The University of Nottingham

Facilitating the Development of Location-Based Experiences

Leif Oppermann, Diplom-Medieninformatiker (FH)

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

April 2009

Abstract

Location-based experiences depend on the availability and reliability of wireless infrastructures such as GPS, Wi-Fi, or mobile phone networks; but these technologies are not universally available everywhere and anytime. Studies of deployed experiences have shown that the characteristics of wireless infrastructures, especially their limited coverage and accuracy, have a major impact on the performance of an experience.

It is in the designers' interest to be aware of technological restrictions to their work. Current state of the art authoring tools for location-based experiences implement one common overarching model: the idea of taking a map of the physical area in which the experience is to take place and then somehow placing virtual trigger zones on top of it. This model leaves no space for technological shortcomings and assumes a perfect registration between the real and the virtual. In order to increase the designers' awareness of the technology, this thesis suggests revealing the wireless infrastructures at authoring time through appropriate tools and workflows. This is thought to aid the designers in better understanding the characteristics of the underlying technology and thereby enable them to deal with potential problems before their work is deployed to the public.

This approach was studied in practice by working with two groups of professional artists who built two commercially commissioned location-based experiences, and evaluated using qualitative research methods. The first experience is a pervasive game for mobile phones called 'Love City' that relies on cellular positioning. The second experience is a pervasive game for cyclists called 'Rider Spoke' that relies on Wi-Fi positioning. The evaluation of these two experiences revealed the importance of an integrated suite of tools that spans indoors and outdoors, and which supports the designers in better understanding the location mechanism that they decided to work with. It was found that designers can successfully create their experiences to deal with patchy, coarse grained, and varying wireless networks as long as they are made aware of the characteristics.

List of Publications

Parts of this work have been previously published at international conferences, workshops, and in several book chapters. The following list provides an account of the dissemination and is, for the core contributions, organised by relevance to this thesis, and afterwards sorted by date.

Core contributions:

L. Oppermann, G. Broll, M. Capra, and S. Benford, *"Extending Authoring Tools for Location-Aware Applications with an Infrastructure Visualization Layer"* (Oppermann et al., 2006), full paper, 8^{th} International Conference on Ubiquitous Computing (Ubicomp), Orange County, CA, 2006 (acceptance rate: 13 %)

L. Oppermann, B. Koleva, S. Benford, R. Jacobs, and M. Watkins, *"Fighting with Jelly: User-Centered Development of a Wireless Infrastructure Visualization Tool for Authoring Location-Aware Experiences"* (Oppermann et al., 2008), full paper, 3^{rd} International Conference on Advances in Computer Entertainment Technology (ACE), Yokohama, Japan, 2008 (acceptance rate: 24 %)

L. Oppermann, *"An Abstract Location-Model for Mobile Games"* (Oppermann, 2009), workshop paper, Mobile Gaming, Lübeck, Germany, 2009

L. Oppermann, R. Jacobs, M. Watkins, R. Shackford, C. v. Tycowicz, M. Wright, M. Capra, C. Greenhalgh, and S. Benford, *"Love City: A Text-Driven, Location-Based Mobile Phone Game Played Between 3 Cities"* (Oppermann et al., 2007), book chapter, Pervasive Gaming Applications - A Reader for Pervasive Gaming Research, Shaker, 2007, ISBN 3832262245

Other:

M. Capra, M. Radenkovic, S. Benford, **L. Oppermann**, A. Drozd, and M. Flintham, *"The Multimedia Challenges Raised by Pervasive Games"* (Capra et al., 2005), full paper, ACM International Conference on Multimedia, Singapore, 2005 (acceptance rate: 16 %)

I. Lindt, J. Ohlenburg, U. Pankoke-Babatz, S. Ghellal, **L. Oppermann**, and M. Adams, *"Designing Cross Media Games"* (Lindt et al., 2005), workshop paper, 2nd International Workshop on Pervasive Gaming Applications, Munich, Germany, 2005

G. Broll, S. Benford, and **L. Oppermann**, *"Exploiting Seams in Mobile Phone Games"* (Broll et al., 2006), workshop paper, 3rd International Workshop on Pervasive Gaming Applications, Dublin, Ireland, 2006

A. Chamberlain and **L. Oppermann**, *"The Digital City: Sex, Cams and Scams"* (Chamberlain and Oppermann, 2006), workshop paper, Exurban Noir, Orange County, CA, 2006

M. Wright, **L. Oppermann**, and M. Capra, *"Visualizing Data Gathered by Mobile Phones"* (Wright et al., 2007), book chapter, Geographic Visualization: Concepts, Tools and Applications, Wiley, 2007, ISBN 0470515112

L. Oppermann, *"Towards a User-Centric Location-Based Service Approach for Mobile Edutainment and Tourism Applications"* (Oppermann, 2007b), position paper, International Conference on Humans and Computers, Düsseldorf, Germany, 2007

V. Paelke, **L. Oppermann**, and C. Reimann, *"Mobile Location-Based Gaming"* (Paelke et al., 2007), book chapter, Map-Based Mobile Services: Design, Interaction and Usability, Springer, 2007, ISBN 3540371095

S. B. Davis, M. Moar, R. Jacobs, M. Watkins, R. Shackford, M. Capra, and **L. Oppermann**, *"Mapping Inside Out"* (Davis et al., 2007), book chapter, Pervasive Gaming Applications – A Reader for Pervasive Gaming Research, Shaker, 2007, ISBN 3832262245

C. Greenhalgh, S. Benford, A. Drozd, M. Flintham, A. Hampshire, **L. Oppermann**, K. Smith, and C. v. Tycowicz, *"Addressing Mobile Phone Diversity in Ubicomp Experience Development"* (Greenhalgh et al., 2007a), full paper, 9th International Conference on Ubiquitous Computing (Ubicomp), Innsbruck, Austria, 2007 (acceptance rate: 19 %)

C. Greenhalgh, S. Benford, A. Drozd, M. Flintham, A. Hampshire, **L. Oppermann**, K. Smith, and C. v. Tycowicz, *"EQUIP2: A Platform for Mobile Phone-Based Game Development"* (Greenhalgh et al., 2007b), book chapter, Concepts and Technologies for Pervasive Games: A Reader for Pervasive Gaming Research, Shaker, 2007, ISBN 3832262237

L. Oppermann, *"On the Choice of Programming Languages for Developing Location-Based Mobile Games"* (Oppermann, 2008), workshop paper, Mobile Gaming, Munich, Germany, 2008

D. Rowland, M. Flintham, **L. Oppermann**, B. Koleva, A. Chamberlain, J. Marshall, S. Benford, and C. Perez, *"Ubikequitous Computing: Designing Interactive Experiences for Cyclists"* (Rowland et al., 2009), full paper, 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI), Bonn, Germany, 2009, (acceptance rate: 24 %)

S. Benford, C. Greenhalgh, P. Tolmie, T. Rodden, M. Flintham, **L. Oppermann,** and S. Reeves, *"Lessons from Touring a Location-Based Experience"* (Benford et al., 2009), full paper, unpublished, 2009

Acknowledgements

Although this thesis represents the culmination of my studies, this work did not happen in isolation and was certainly influenced by the people that accompanied me. This section is a small thank you to those that mattered in this context.

I would like to thank my parents Inge and Uwe for a happy childhood and their love and continued support over the years. Thanks also to my brother Jan for teaching me the first lessons on our home computer and to my sister Birgit for exploring it all together with me in a playful process. As my interest in computers grew, I was lucky enough to find friends that shared my passion and with whom I could evolve. Thanks to Jan-Christoph Sievers, Marcus Kaluschke, and Jochen Oppermann for being my mates for so many years.

I always considered it as a privilege to study Medieninformatik in Wernigerode. Many thanks in this respect to Eberhard Högerle and Helmut Eirund for having created a fascinating curriculum, and to Christian Geiger for awakening my interest for research and supervising my Diplom thesis. For my post-graduate studies, I felt privileged again, as I was able to conduct them at the Mixed Reality Lab at Nottingham University. Many thanks in this respect to Steve Benford, Tom Rodden, and Chris Greenhalgh for providing many thrilling opportunities for research, and especially to Steve and Tom for supervising my PhD thesis.

Thanks to my examiners, Annika Waern and Chris Greenhalgh, and to my colleagues Andy Crabtree and Boriana Koleva for their share in making this a better thesis. Thanks also to the many fellow students and colleagues with whom I discussed and collaborated over the years, especially to my office mates Bartosz Wietrzyk, Joe Marshall, Joel Fischer, Duncan Rowland, and Mauricio Capra. And a very special thank you to the artists from Active Ingredient, Rachel Jacobs, Matt Watkins, Robin Shackford, and Blast Theory, Ju Row Farr, Matt Adams, Nick Tandavanitj, with whom I collaborated for my studies.

Last, but not least, I would like to thank my wife Tanja for having moved to a new country with me and for being supportive and encouraging all the way along. I love you darling, and this thesis is, of course, dedicated to you!

Contents

ABSTRACT I

LIST OF PUBLICATIONS II

ACKNOWLEDGEMENTS V

CONTENTS VI

GLOSSARY XIV

1. INTRODUCTION 1

1.1 Setting the Scope 1

1.2 Goal of the Thesis 2

1.3 Methodology 2
1.3.1 The Reflective Practitioner 3
1.3.2 From Action Research to Prototyping 4
1.3.3 Participatory Design 5
1.3.3.1 First Generation 5
1.3.3.2 Second Generation 6
1.3.3.3 Third Generation 9
1.3.4 Philosophical Backdrop 10
1.3.4.1 Language Games 10
1.3.4.2 Tradition and Transcendece 11
1.3.4.3 Ready-to-hand and Present-at-hand 11
1.3.5 Approach of this Thesis 13
1.3.5.1 Designing for Designers 14
1.3.5.2 Work Organisation 14
1.3.5.3 My Role and Tradition 16

1.4 Summary of Contributions 17

1.5 Guide to the Document 18

2. RELATED WORK 20

2.1 Jargon Buster 20
2.1.1 General Terms 20
2.1.1.1 Position 20
2.1.1.2 Location 21
2.1.1.3 Game 21
2.1.1.4 Experience 21
2.1.2 Computing Terms 22
2.1.2.1 Ubiquitous Computing 22
2.1.2.2 Pervasive Computing 23
2.1.2.3 Context-Aware Computing 23
2.1.2.4 Location-Aware Computing 23
2.1.2.5 Location Service 23
2.1.2.6 Location-Based Application 24

2.1.2.7 Location-Based Service ... 24
2.1.2.8 Augmented Reality ... 24
2.1.2.9 Pervasive Games ... 24
2.1.2.10 Location-Based Experience ... 25

2.2 Examples and Seminal Work ... 25
2.2.1 Cyberguide Project, 1995-1997 ... 25
2.2.2 MARS Project, 1996-1999 ... 28
2.2.3 GUIDE Project, 1997-1999 ... 29
2.2.4 Geocaching, 2000 ... 32
2.2.5 ARQuake, 2000 ... 33
2.2.6 Botfighters, 2001 ... 35
2.2.7 Can You See Me Now?, 2001 ... 37
2.2.8 DefCon 10 WarDriving Contest, 2002 ... 39
2.2.9 Noderunner, 2002 ... 42
2.2.10 Uncle Roy All Around You, 2003 ... 44
2.2.11 Mogi, 2003 ... 47
2.2.12 Riot! 1831, 2004 ... 48
2.2.13 Treasure, 2004 ... 49
2.2.14 MOSES, 2004 ... 50
2.2.15 CAERUS, 2005 ... 52
2.2.16 Epidemic Menace, 2005 ... 53
2.2.17 Hitchers, 2005 ... 54
2.2.18 Insectopia, 2006 ... 58
2.2.19 Momentum, 2006 ... 59
2.2.20 REXplorer, 2007 ... 63
2.2.21 Mercury Platform, 2008 ... 65
2.2.22 Summary of Example Applications ... 68
2.2.22.1 Production Categories ... 68
2.2.22.2 Game-Design ... 69
2.2.22.3 Overview of Technological Choices ... 69
2.2.22.4 Mobile Devices ... 70
2.2.22.5 Positioning Strategies ... 71
2.2.22.6 Communication Strategies ... 72

2.3 Building Location-Based Experiences ... 73
2.3.1 Communication Technology ... 74
2.3.1.1 Voice ... 74
2.3.1.2 Data ... 74
2.3.2 Positioning Technology ... 76
2.3.2.1 Global Positioning System ... 76
2.3.2.2 Cellular Positioning ... 77
2.3.2.2 Wireless Local Area Network ... 79
2.3.2.4 Short-Range Sensors ... 80
2.3.2.5 Image Analysis ... 82
2.3.2.6 Self-Reported ... 83
2.3.2.7 Summary of Positioning Technology ... 83
2.3.3 Spatial Authoring ... 86
2.3.3.1 MARS Authoring Tool ... 87
2.3.3.2 Mediascapes ... 89
2.3.3.3 Colourmaps ... 93

2.3.3.4 CAERUS Authoring Tool ... 94
2.3.3.5 Yamamoto ... 98
2.3.3.6 Gamecreator ... 100
2.3.3.7 Mercury Platform ... 103
2.3.4 Summary of Authoring tools ... 106
2.3.4.1 Background Map ... 106
2.3.4.2 External Content Creation ... 107
2.3.4.3 Content Trigger Zones ... 107
2.3.4.4 Content Trigger Events ... 108
2.4 The Problem of Uncertainty ... 110
2.4.1 Spatial Disparity in CYSMN ... 111
2.4.2 Temporal Disparity in CYSMN ... 112
2.4.3 Manifestation in other Projects ... 113
2.4.3.1 Mediascapes ... 114
2.4.3.2 REXplorer ... 114
2.4.3.3 Treasure ... 115
2.4.3.4 Place Lab ... 115
2.4.3.5 ARQuake ... 116
2.4.3.6 Cyberguide ... 116
2.5 Motivation for this Thesis ... 116
3. CONCEPTUAL FRAMEWORK ... 118
3.1 Extending Authoring Tools ... 119
3.1.1 Current Two Layer Model of Authoring ... 119
3.1.2 Extending the Model to Three Layers ... 121
3.1.2.1 Physical World Layer ... 123
3.1.2.2 Infrastructure Layer ... 123
3.1.2.3 Content Layer ... 124
3.2 Location Mapping ... 124
3.2.1 Location and Position ... 124
3.2.2 Meaningful Locations ... 125
3.2.3 Location in the Context of Space and Place ... 127
3.2.4 Mappings between Layers ... 129
3.2.4.1 Create Mappings between Layers ... 130
3.2.4.2 Relationships between Layers ... 130
3.3 Abstract Location Model ... 131
3.3.1 Reproducible ID-Strings ... 132
3.3.1.1 Locations ... 132
3.3.1.2 Regions (Location Data) ... 132
3.3.2 Flow of Data through the Model ... 132
3.3.3 Defining Location Data ... 133
3.3.3.1 Location Data for the Position Intersecter ... 133
3.3.3.2 Location Data for the Wi-Fi Intersecter ... 134
3.3.3.3 Location Data for the GSM Intersecter ... 135
3.4 Challenges ... 135
3.4.1 Map Data Access ... 136
3.4.2 Infrastructure Data Collection ... 136
3.4.3 Presentation ... 137

3.4.4 Workflow ... 137
3.4.4.1 Desktop ... 138
3.4.4.2 Server ... 138
3.4.4.3 Mobile ... 139
3.5 Plan of Action ... 139

4. FIRST STUDY: LOVE CITY ... 141

4.1 Overview ... 141
4.1.1 Designers' Tradition ... 141
4.1.2 Author's Role ... 142
4.1.3 General Description ... 142
4.1.4 Game Walk-Through ... 144
4.1.4.1 The Public Website ... 145
4.1.4.2 Getting In and Out of the Game ... 147
4.1.4.3 Initial Messages ... 148
4.1.4.4 Messages Sent by Players ... 148
4.1.4.5 Authored Messages ... 149
4.1.4.6 Nearing the End ... 150
4.1.5 Public Staging ... 151
4.2 Development Iteration 1: Initial Exploratory Phase ... 151
4.2.1 A Need for Understanding the Environment ... 151
4.2.2 Network Surveying and Initial Visualisations ... 152
4.2.3 User-Feedback through Design Workshops ... 155
4.2.4 Suggested Functionality ... 158
4.2.4.1 Infrastructure Collection Tools ... 158
4.2.4.2 Infrastructure Visualisation Tools ... 159
4.2.4.3 Summary of Feedback ... 161
4.3 Development Iteration 2: Towards the Prototype ... 161
4.3.1 Technical Overview ... 162
4.3.1.1 Location Mechanism of the Prototype ... 162
4.3.1.2 Content Mapping ... 162
4.3.1.3 Preliminary Distributed System Architecture ... 163
4.3.2 Authoring of Love City Regions ... 164
4.3.2.1 Overview of the New Visualisation and Authoring Tool ... 164
4.3.2.2 Authoring Process ... 167
4.3.2.3 Overview of the New Data Collection Tool ... 169
4.3.3 Feedback ... 171
4.3.3.1 Aid Orientation ... 171
4.3.3.2 Editing of Regions ... 171
4.3.3.3 Testing and Verification In Situ ... 172
4.3.4 User Testing the Prototype ... 173
4.3.4.1 Overview of the Game Prototype ... 173
4.3.4.2 Testing with End-Users ... 173
4.3.4.3 Outcomes from the Test ... 174
4.4 Post-Event Reflections ... 175
4.4.1 Artists' Experience ... 175
4.4.1.1 Physical World Layer ... 176
4.4.1.2 Infrastructure Layer ... 176

4.4.1.3 Content Layer ... 177
4.4.1.4 Final Comments ... 177
4.4.2 Author's Experience ... 178
4.4.2.1 Physical World Layer ... 178
4.4.2.2 Infrastructure Layer ... 178
4.4.2.3 Content Layer ... 180
4.4.2.4 Summary ... 180

5. SECOND STUDY: RIDER SPOKE ... 181

5.1 Overview ... 181
5.1.1 Designers' Tradition ... 181
5.1.2 Author's Role ... 181
5.1.3 General Description ... 182
5.1.4 Game Walk-Through ... 183
5.1.4.1 Recording an Answer ... 183
5.1.4.2 Listening to other Players' Answers ... 186
5.1.5 Public Stagings ... 187

5.2 Technical Background ... 187

5.3 Development Iteration 1: Initial Exploratory Phase ... 190
5.3.1 Deciding on a Content Triggering Mechanism ... 190
5.3.1.1 Informative War-driving ... 191
5.3.1.2 Content Triggering Tests ... 192
5.3.1.3 Storylines ... 195
5.3.1.4 Making a Decision ... 197
5.3.2 Network Survey and Initial Visualisations ... 199
5.3.2.1 User Requirements ... 199
5.3.2.2 First Tests of the Mobile Client and Data Collection ... 199
5.3.2.3 Initial Visualisations ... 203
5.3.2.4 Requested Functionality for Tools ... 206

5.4 Development Iteration 2: Towards the Premiere ... 207
5.4.1 The Rider Spoke Location Mechanism ... 208
5.4.2 A Need for Understanding the Mechanism ... 210
5.4.3 Visualising the Game-Graph ... 211
5.4.3.1 Data Collection ... 212
5.4.3.2 Technical Overview ... 212
5.4.3.3 Functional Overview ... 213
5.4.4 Practical Use by the Artists ... 216
5.4.4.1 Using the Visualisation Tool for Setup ... 216
5.4.4.2 Using the Visualisation Tool for Orchestration ... 219
5.4.5 Post-Event Reflections ... 222

5.5 Development Iteration 3: Post-Premiere ... 225
5.5.1 The "What-If" Simulator ... 226
5.5.1.1 The Server Component ... 227
5.5.1.2 The Analysis Program ... 228
5.5.2 Collecting more GPS Data ... 232
5.5.2.1 Using an Existing Data Collection Tool ... 232
5.5.2.2 Using a Custom Data Collection Tool ... 232
5.5.2.3 Incorporating Wigle Data ... 233

5.5.2.4 Local Database ... 233
5.5.3 Setting Up in Athens ... 234
5.5.3.1 Collecting Data ... 234
5.5.3.2 Choosing the Settings ... 237
5.5.4 Setting Up in Brighton ... 241
5.5.4.1 Collecting Data ... 241
5.5.4.2 Choosing the Settings ... 242

5.6 Post-Event Reflections ... 249
5.6.1 Workflow in London ... 249
5.6.2 Workflow in Athens ... 250
5.6.3 Workflow in Brighton ... 251

6. DISCUSSION ... 253

6.1 The Benefit of the Infrastructure Layer ... 254
6.1.1 For the Designers of Love City ... 254
6.1.2 For the Designers of Rider Spoke ... 255
6.1.3 On a General Level ... 255

6.2 Supporting Design Work ... 256
6.2.1 In the Field ... 256
6.2.1.1 Mobile Authoring Tool ... 256
6.2.1.2 Practical Use of the Tool ... 258
6.2.1.3 Four Approaches to Scale Up ... 259
6.2.2 In the Studio ... 260
6.2.2.1 Visualisation Tools ... 260
6.2.2.2 Practical Use of the Tools ... 262
6.2.2.3 Do not Rely on Visualisations Alone ... 265
6.2.3 Making Sense of Location ... 267
6.2.3.1 Understanding Location in the Design ... 267
6.2.3.2 Mapping Content to Locations ... 270
6.2.3.3 Understanding Places in the Post-Event Analysis ... 273

6.3 The Engineering Side ... 274
6.3.1 The Challenges Revisited ... 274
6.3.1.1 Map Data Access ... 274
6.3.1.2 Infrastructure Data Collection ... 278
6.3.1.3 Presentation ... 278
6.3.1.4 Workflow ... 282
6.3.2 Ideas for System-Level Improvements ... 285
6.3.2.1 Easy Data Synchronisation ... 285
6.3.2.2 Coordinated Visualisations ... 286
6.3.2.3 Scripting Interface ... 287
6.3.2.4 Flash Interface ... 290

6.4 Best Practices ... 290
6.4.1 Opportunity ... 291
6.4.2 Development Process ... 291
6.4.3 The Artistic Umbrella ... 295
6.4.4 Legal Issues ... 297

6.5 Outlook ... 298
6.5.1 Playful Process ... 298

6.5.2 Learn from Others ... 299
6.5.3 Use What is Available ... 300
6.5.4 Build Up Experience ... 300

7. SUMMARY AND CONCLUSIONS ... 301

7.1 Summary ... 301

7.2 Reflection on Key Principles ... 304
7.2.1 Three Layer Model ... 304
7.2.2 Abstract Location Model ... 305
7.2.3 Participatory Design ... 305

7.3 Ideas for Future Work ... 306
7.3.1 Improved GPS Triggering ... 306
7.3.2 Work on Edges ... 307
7.3.3 Tagging ... 308
7.3.4 Transfer Knowledge ... 308
7.3.5 More Layers ... 309
7.3.6 GIS Integration ... 311
7.3.7 Temporal Variation ... 312
7.3.8 Gaps and Conflicts ... 312
7.3.9 Ideas First ... 313

7.4 Conclusions ... 314

APPENDIX 1 – INFORMATION DESIGN ... 316

A1.1 Charts ... 318

A1.2 Maps ... 320

APPENDIX 2 – SOFTWARE DEVELOPMENT ... 326

A2.1 Developing for Mobile Devices ... 326
A2.1.1 Programming Languages ... 327
A2.1.2 Device Capabilities ... 328
A2.1.3 Web-Based Applications ... 329

A2.2 Developing for Servers ... 329

A2.3 Distributed Architecture ... 330
A2.3.1 General Architecture ... 330
A2.3.2 Supporting Libraries ... 332

APPENDIX 3 – METRICS FOR PERVASIVE TECHNOLOGY ... 335

APPENDIX 4 – LOVE CITY DETAILS ... 337

A4.1 Distributed System Architecture for 2 Types of Users ... 337
A4.1.1 Central Web-Application ... 337
A4.1.2 Admin Interfaces ... 339
A4.1.3 Player Interfaces ... 339

A4.2 Final Location Mechanisms ... 341
A4.2.1 Operator Based Positioning ... 341
A4.2.2 Random Positioning ... 342

APPENDIX 5 – RIDER SPOKE DETAILS **343**
A5.1 Distributed System Architecture for 4 Types of Users 343
A5.1.1 Players 345
A5.1.2 Staging Administrator 345
A5.1.3 Content Ranking Artists 346
A5.1.4 Graph Administrator 347
A5.2 Rider Spoke's Fingerprint Location Mechanism 348
A5.2.1 Algorithm Details 348
A5.2.2 Important Aspects 351
A5.2.3 Reflections on the Location Mechanism 352
A5.2.3.1 Stabilising the Location Mechanism 353
A5.2.3.2 Judging the Location Mechanism 357
A5.2.3.3 Optimising the Algorithm 358
A5.2.3.4 Adaptive Algorithm 358
APPENDIX 6 – RIDER SPOKE VISUAL DESIGN DOCUMENT **360**
A6.1 Definition of Terms 360
A6.1.1 Rider Spoke and Graphs 360
A6.1.2 Rider Spoke's Fingerprinting Mechanism 360
A6.2 Visualising the Network / Game Infrastructure 361
A6.3 Tasks to Support the Generation of the Views 365
APPENDIX 7 – AUTHOR'S INVOLVEMENTS **366**
A7.1 IPerG 366
A7.2 INSCAPE 367
A7.3 Other Involvements 367
A7.3.1 City as Theatre 367
A7.3.2 Crossmedia Gaming 368
A7.3.3 Tycoon 369
A7.3.4 Heartlands (a.k.a. 'Ere be Dragons) 370
A7.3.5 Boxed Pervasive Games 372
APPENDIX 8 – LIST OF FIGURES **373**
APPENDIX 9 – LIST OF TABLES **379**
ONLINE RESOURCES **381**
REFERENCES **387**

Glossary

3G	Third Generation of Mobile Phones
AP	Access Point
API	Application Programming Interface
BSSID	Basic Service Set Identifier
GIS	Geographic Information System/Science
GK	Gauss-Krüger
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HMD	Head Mounted Display
HSPA	High Speed Packet Access
HTTP	Hypertext Transfer Protocol
IR	Infrared
J2ME	Java 2 Platform, Micro Edition
MAC address	Media Access Control address
Modem	MOdulator/ DEModulator
NFC	Near Field Communication
NMEA	National Marine Electronics Association
NTP	Network Time Protocol
OSGB36	Ordnance Survey Great Britain 1936
PD	Participatory Design
PDA	Personal Digital Assistant
RFID	Radio-Frequency Identification
SDK	Software Development Kit
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
UTM	Universal Transverse Mercator
WGS84	World Geodetic System 1984
Wi-Fi	Brand for WLAN, used synonymously
WLAN	Wireless Local Area Networks (IEEE 802.11)

“We shall not cease from exploration, and the end of all our exploring will be to arrive where we started and know the place for the first time.”

– Thomas S. Eliot

1. Introduction

When building location-based applications one usually relies on the availability and reliability of wireless infrastructures like Wi-Fi, mobile phone networks, or GPS - but these technologies are not available everywhere and anytime. Due to physical and economic constraints any such technology can only ever have partial coverage in space and time, and this will affect the performance of every application that relies on this technology.

It is in the interest of the people who design location-based applications to be aware of the restrictions of the underlying technology. Knowing about possible technological shortcomings allows the designers to design with these in mind and come up with work-arounds, if needed, rather than running into problems when it might be too late. This technological awareness might also foster the creation of designs which have been previously unthinkable. In any case, it is believed that the quality of a location-based application will be improved if its designers have been aware of potential technological limitations during the authoring phase and not just taken the technology for granted.

1.1 Setting the Scope

This thesis is going to limit its scope of discussion to an artistically influenced sub-set of location-based applications that we call location-based experiences. Like location-based applications, they provide content based on the user's physical location, but their focus is more user-centred, as they are made to deliver some kind of experience to the user. Location-based experiences are usually built to entertain (e.g. pervasive games), to educate (e.g. tour guides) or just for their own sake (e.g. art installations). Additionally, they are also usually bound to some sort of event, which often limits their temporal extent. These limitations separate location-based experiences from location-based applications which would cover any concept that delivers content based on location, e.g. car navigation system or mobile yellow pages.

The Mixed Reality Lab at the University of Nottingham has a history of building location-based experiences in cooperation with professional designers. Having started to work in the lab in 2004, the author became acquainted with building location-based experiences in cooperation with external partners on several occasions. Drawing upon the experiences gained from these cooperations as well as from studying the literature, it became apparent that the utilised wireless network infrastructures, most importantly their limited availability and performance, have a big influence on the final performance of a location-based experience. However, none of the currently available systems for authoring location-based experiences seem to cater for this effect in their authoring workflows.

1.2 Goal of the Thesis

It is the goal of this thesis to make the designers of location-based experiences more aware of the limited availability and varying performance of the wireless networks that they are using. The overarching hypothesis is that by diminishing some of the guesswork and assumptions about the characteristics of the utilised technology which are omnipresent in current authoring workflows, designers will be able to produce designs that better suit the technology that they rely on.

In other words: by revealing the patchy infrastructure of pervasive computing to its designers at design time, it is hoped to make the accompanying problems of authoring location-based experiences more evident and thus more manageable. This is believed to enhance the usability of an application which will in turn benefit its users and thus ultimately increase the economic viability of the application.

1.3 Methodology

The procedure of conducting the work in this thesis, to use a colloquial term for guidance, was that of “learning by doing”. The two studies of this thesis were conducted in collaboration with experience design practitioners who had to deliver working public experiences within a given time-frame. The author participated in the development of these two experiences as a system developer and facilitator, proposed changes to the designers’ development workflows, prototyped supporting tools, and studied the resulting effects. In this light, the term "learning by doing" can be seen in two ways:

1. The experience designers were learning about the technology while working with the technology.

2. The author got involved in the process that he intended to improve.

1.3.1 The Reflective Practitioner

This first meaning of "learning by doing" can be seen as a general approach to generating new insights and knowledge through, as Donald Schön coined it, an epistemology of practice. Schön's point of departure was that *"competent practitioners usually know more than they can say",* as they posses tacit knowledge about the things they do, and that they *"often reveal a capacity for reflection on their intuitive knowing in the midst of action and sometimes use this capacity to cope with the unique, uncertain, and conflicted situations of practice."* (Schön, 1983: viii). His idea of "reflection in action" highlighted the importance of doing, and reflecting about this doing while doing it (and afterwards), for the development of a better appreciation of the situation at hands. Schön thus grounded the development of professional knowledge in the phenomena of practice and the practitioner's response to the experienced situation. In doing so, he gave higher value to knowledge gained by practical means over traditional intellectual knowledge, e.g. as gained by rationalist (observed and proven) or critical rationalist (observed and falsified) methods. Schön also demarcated his "reflective practitioner" from the traditional "expert": both ought to know solutions for a particular problem situation. But the expert must claim to possess this knowledge, even if uncertain, as part of his role, whereas the reflective practitioner does not have to do this, in fact: he should not do this. The reflective practitioner knows that he is *"not the only one in the situation to have relevant and important knowledge."* (Schön, 1983: 300). Although Schön, being an educationalist, was not particularly concerned about design or computers, his view of practical knowledge is also reflected in design methods for computer systems, such as prototyping (Floyd, 1987), or participatory design (Ehn, 1993), which will be described and brought together in the following.

1.3.2 From Action Research to Prototyping

This brings us to the second meaning of "learning by doing" which is that of researchers *getting involved* in the processes that they set out to improve. This call for action lies at the heart of action research, a term which was originally coined in the social sciences by Kurt Lewin (Lewin, 1946). A more recent work from the field of computing summarised that *"in action research, the researcher wants to try out a theory with practitioners in real situations, gain feedback from this experience, modify the theory as a result of this feedback, and try again"* (Avison et al., 1999). The point of action research is thus to test and evolve a theory while making a difference to the real world situation that is being studied (compare Thoresen, 1993: 275). Making a difference is the key point here. Furthermore, action research is also inherently iterative.

In the field of computing, prototyping is an iterative approach to system development which has been advocated by many to be more fit for the purpose of developing complex systems than traditional phase-oriented engineering approaches such as the often cited Waterfall model (Royce, 1970). Buxton and Sniderman provide an early and comprehensive case for prototyping as an iterative approach to designing human-computer interfaces. Their human factors oriented approach regarded *"each iteration of a design as being a prototype whose purpose is to test a critical mass of the overall problem"* (Buxton and Sniderman, 1980). Such a protoype needs to be tested by *"guinea pig"* users, who are monitored while performing certain tasks. The collected information would then be used to assess the performance of the current prototype and plan the next iteration – a process which the authors described as *"keep trying until you get it right"*. In a similar vein, Floyd highlights the importance of seeing prototyping as a process that serves as a learning vehicle about the fitness of a proposed solution for a particular task in a practical situation. She advocates the use of prototyping as a point of departure from more traditional product-oriented views of software-engineering, which largely disregard the relationship between the software-product, the human users, and their environments. In her view the product of a software-development process is not a particular program or a piece of code, but a tool that is applicable to a real work situation (Floyd, 1987). In an earlier work, Floyd provides a structured account of the characteristics of prototypes (e.g. horizontal or vertical functional selection), and argues that the software tools that are used for prototyping (e.g. very high-level programming languages, widget toolkits, and database-systems) need to be integrated so as *"to make effective work possible"* (Floyd, 1984) – a point which was also raised by Buxton and Sniderman. This is all to allow for rapid prototyping, i.e. quickly testing different alternative solutions to a problem, which serves the ultimate purpose of generating feedback from human users in real-world-like situations to find the fittest solution.

1.3.3 Participatory Design

The idea behind prototyping is taken one step further by what is called participatory design (PD). Rather than trying to learn from *simulated* work situations in laboratory settings, researchers and system designers delve into *actual* work situations, which they call "use situations", and see them as the start and end points of the design process (Greenbaum and Kyng, 1991b). This calls for designers to take work practices seriously (Wynn, 1991) and respect humans as actors, not merely as factors, or even idiots (Bannon, 1991). Moreover, the human actors and their actions must not be seen in isolation, but in the context of the tasks at hand, and in the social context of work, which requires communication and cooperation. Consequently, as designers of computer systems need to communicate with practitioners in order to be able to take their practices seriously, this design approach requires cooperation between those who are involved, i.e. system designers, end-users, and other stakeholders. Participatory design is a cooperative design approach that is especially present in the Scandinavian design tradition (Bjerknes et al., 1987, Ehn, 1993, Kuhn and Muller, 1993). It has evolved with different projects over the last four decades and can be broadly categorised into three generations.

1.3.3.1 First Generation

As outlined in the introductory paragraph, participatory design is characterised by taking an understanding of the respective users as key for a successful design. Moreover, it requires that this understanding must originate from an equal participation of the end-users in the design process. Depending on the work environment, this might provoke value conflicts and resentments due to power shifts in the organisation, as workflows and tools would no longer be given from the top, but be designed from the bottom. In fact, this design tradition originates from a political situation in Scandinavia in the 1970s, where changing labour laws and strong trade-unions gave workers a bigger influence on their own working conditions. This democratisation of work was meant to improve health and safety of the workers, but also as a response to the subtle 'deskilling' of the workforce which was brought about through industrialisation (cf. Braverman, 1974).

The Norwegian Iron and Metal Workers Union (NJMF), for example, initiated an early PD project that sought to research the use of computer systems in their workers context through consultation. In its initial inception the project was set up with two workers assisting two researchers in their work, which was basically to produce a list of requirements for working conditions for the NJMF so that they could represent their members more effectively in contract bargaining situations. However, the results of this initial work were found to be meaningless by the unionists, as they could neither apply the researchers' findings as intended nor transfer the results to other work settings. The researchers had effectively missed the target, although they presumably delivered a competent job in their own work tradition. The research strategy for the NJMF project was subsequently radically redesigned so that the local unions would no longer support the researchers, but instead the researchers would support the local unions (Ehn, 1993: 51).

This paradigm change paved the way for what was to become known as the Scandinavian approach to participatory design, a.k.a. the Collective Resources Approach (CRA). CRA is characterised by the intent to increase the workers influence on technology using two instruments: *"1) action oriented and trade union based strategies and 2) cooperative design"* (Kyng, 1994: 93).

1.3.3.2 Second Generation

A good example for a more active participation of the workers in a cooperative design process can be found in the UTOPIA project. This second generation PD project was initiated by the Nordic labour union for graphic workers (NGU) in 1981. It complemented the democratisation in the design process with the idea to produce "*tools and work environments for skilled work and good quality products and services"* (Ehn, 1993: 57). UTOPIA sought to *"contribute to the development of powerful skill enhancing tools for graphic workers, in the light of the emerging graphic workstation technology"* (Bødker et al., 2000). To achieve these goals, the researchers applied a design approach that they called mocking-it-up, where designers could learn about the workers' working practices and the workers could learn about new technological possibilities through the co-construction of (cardboard) mock-ups (Ehn and Kyng, 1991).

The routine of the newspaper workers was traditionally split between the journalists, who wrote the text, and the typographers, who made up the pages of the newspaper. Journalists would send their texts to the typographers; the typographers would then composite the text to page layouts, send so-called "proofs" back to the journalists for final checks, and finally send the approved pages to the press for printing. This process worked well and provided a clear division between the work of the journalists and the typographers' work, which was much to everybody's consent. However, this way of working became obsoleted by technology. The introduction of phototypesetting, which became popular in the 1970s, modified this workflow considerably. As proofing became too expensive and time-consuming on the new machines, the journalists were frequently found in the typographers' proximity to check the page layout. A negative side-effect of this emerging routine was that the journalists were thereby effectively invading the work-space of the typographers and controlling their work, which was not well received. The advent of computer-based layout and page make up at the end of the 1970s worsened the situation, as it effectively took work and responsibility away from the typographers and put it into the hands of the journalists. Moreover, it also resulted in a degradation of typographic quality, as the journalists were not skilled to do the layout work. To remedy the situation for the future, the UTOPIA project started with the following design question: *"Are there technical and organizational design alternatives that support peaceful and creative coexistence between typographers and journalists, where both readability and legibility of the product could be enhanced?"* (Ehn and Kyng, 1991: 171).

The UTOPIA project effectively investigated possible workflows for what was later to become known as desktop publishing. But more importantly, the project team carried out this work without having the technology available. For example, they envisioned the use of laser printers for proofing at a time where laser printers only existed in advanced research labs and therefore seemed unreal to appear at the workplace. The authors noted: *"It was our responsibility as professional designers to be aware of such future possibilities and to suggest them to the users"* (Ehn and Kyng, 1991: 172). Altogether, the researchers thus aimed at gaining real feedback from real users in work-like situations, but without having access to the real technology. Now this might seem like a classic chicken and egg problem. The researchers approached this seemingly futile situation by building prototype workstations from cardboard and testing the workflows that surround these workstations with experienced typographers and journalists in design language games. For example, in one of their settings the aforementioned laser printer was simply a box with the words "desktop laser printer" written on it that had a small stack of paper next to it, which would serve as simulated proof prints. Thus, with a little guidance, the journalists and typographers could built on their experience and fit their traditional workflows to the new environment in a playful way. The idea behind this approach is that by mocking-up future work places and inhabiting them with experienced workers that have a shared understanding of familiar situations (their traditional way of working), participants could interact with each other in a playful way, experience the situation, reflect on their actions, and provide meaningful feedback before an actual system is built for real.

A particularly interesting aspect of this method is how it can uncover flaws in the design quite effectively. As the participants engage in their activity, they exploit their familiarity of the situation and their shared language to gloss over the fact that they are dealing with cardboard prototypes. Ideally, this flow only breaks down when they arrive at a situation that they cannot deal with in their playful imagination – this must then be seen as an anomaly in the prototype that should be investigated. *"Hence, an important aspect of a mock-up is its usefulness for involved activity where the users' awareness is focussed on doing the task, rather than on analyzing objects and relations. Detached reflections on alternatives become part of the process when the fluent use of the typographical design tools –*

their readiness-to-hand – breaks down. These reflections are then grounded in a practical experience, an experience shared by users and designers in a design-by-doing language game" (Ehn and Kyng, 1991: 180).

This might seem as childs-play, and in a way it is. But this is not to say that this method of experiencing prototypes does not warrant professional results. Rather, this creative and imaginative way of dealing with unfinished artifacts exploits the situation of humans engaging in simulated work tasks quite efficiently – much like children making the most out of their play with limited resources where chalk lines on the street or sandboxes can provide endless possibilities for interaction. The designers of UTOPIA argued that their method of using hands-on mock-ups in playful settings can be useful if the mock-ups remind participants of previous experiences in similar situations and thus support their interaction and reflection in action (Ehn and Kyng, 1991: 177).

1.3.3.3 Third Generation

Third generation participatory design projects combine the cooperative design approach with actual system development. After mere consulation in the first generation, such as in the NJMF project, and active participation through mock-ups in the second generation, such as in the UTOPIA project, end-users in the third generation of participatory design projects are not only presented with computer prototypes but also influence their development. Bødker and Grønbæk summarise: *"The cooperative prototyping approach aims to establish a design process where both users and designers are participating actively and creatively, drawing on their different qualifications. To facilitate such a process, the designers must somehow let the users experience a fluent work-like situation with a future computer application; that is, users' current skills must be brought into contact with new technological possibilities"* (Bødker and Grønbæk, 1991: 200).

Bødker, Greenbaum and Kyng describe how the role of a designer and system developer who works with cooperative prototyping shifts from that of project manager to project facilitator: *"The role of designers becomes rather complicated in the cooperative design process. Designers are in charge of the project, they are responsible for getting the work going, and they must be able to act as facilitators* *of workshops and similar events, and, in general, as resources for the groups. And designers are the ones who make sure that all the technical details are in place"* (Bødker et al., 1991: 152).

There is no fixed method of doing participatory design, as pointed out by the editors of one of the leading books on the subject (Greenbaum and Kyng, 1991a: 6), but its authors collective agreed on a few corner stones which have been outlined in this section on participatory design. The focus of cooperative prototyping, being a form of participatory design, is on a process that draws upon the expertise and tacit knowledge of all who are involved to facilitate learning and provide feedback for the design of a computer system that is deeply grounded in practical activities. In this sense, cooperative prototyping embodies Schön's idea of the reflective practitioner (Suchman and Trigg, 1991: 85) and adapts it for the design of computer systems that meet the needs of the people that use them. Cooperative prototyping provides for highly functional designs that are deeply grounded in the essential concerns of current practice, but which also challenge this established practice to allow for new practice to emerge.

1.3.4 Philosophical Backdrop

This section provides an overview of the philosophy behind participatory design. It briefly elaborates on why design should be done this way, and why it works.

1.3.4.1 Language Games

Pelle Ehn, a key figure in the Scandinavian PD tradition, emphasises that participatory design does not only mean that users participate in design, but also, and perhaps more importantly, that designers participate in use and build up their own practical understanding and tradition (Ehn, 1993: 68). By participating in the practice of the end-users, the designers learn their language, and the end-users learn the language of the designers. Thus, this mutual learning provides space for language games in a Wittgensteinian sense of *language as action* (Ehn, 1993: 62), where words are only meaningful when seen in their practical context of use (Ehn and Kyng, 1991: 176), which needs to be understood. Sharing the language, and thus practice, allows for truly grounding the design in the work tradition.

1.3.4.2 Tradition and Transcendece

Ehn further elaborates how any design has to deal with the contradiction of *tradition* and *transcendence*. All work activities are grounded in their own traditions, which participatory design regards as key. Yet, any particular tradition that is inspected for redesign is also implicitly regarded as no longer sufficient, as otherwise it would not need to be redesigned – this departure from the tradition is the transcendence. Design is thus always concerned with both, tradition and transcendence, and needs to find a balance between the two. Ehn sees this tension between tradition and transcendence as the *"dialectical foundation of design"* (Ehn, 1993: 70). By bringing the language games of designers and end-users together, participatory design can be firmly grounded in the work tradition, and based on these grounds support envisioning the future by focussing on the transcendence.

1.3.4.3 Ready-to-hand and Present-at-hand

An alternative explanation of the effectiveness of using prototypes in participatory design can be based on the philosophy of Martin Heidegger, who distinguished between the terms *ready-to-hand* and *present-at-hand.*

Heidegger disapproved of the rationalistic tradition of science that separated the mind from the body (Heidegger, 1927: 95). This so-called Cartesian dualism is based on the thinking of René Descartes in the 17th century, who postulated: "cogito ergo sum" – I think, therefore I am. Dourish summarised that this dictum "*had reflected a doctrine that we 'occupy' two different and separate worlds, the physical reality and the world of mental experience.… Most of the philosophers since Descartes had taken the position that the mind is the seat of reasoning and meaning. The mind observes the world, gives it meaning by relating it to abstract understandings of an idealized reality and, on the basis of that meaning, formulates a plan of action."* (Dourish, 2001: 107). Heidegger reversed this chain of reasoning because he questioned the possibility of an objective and complete mental viewpoint. Although he did not disapprove of abstract reasoning altogether, he introduced his idea of *being-in-the-world* as our basic state of existence and as central to our understanding of the world (Heidegger, 1927: 53).

Winograd and Flores summarise Heidegger's perspective: *"There is no neutral viewpoint from which we can see our beliefs as things, since we always operate within the framework they provide"* (Winograd and Flores, 1986: 32) – therefore our understanding of the world must be based on our real world experiences. Heidegger's phenomenological perspective is more concerned with the *how* than with the *what* of the objects of investigation (Heidegger, 1927: 27). This reminds us of the importance of *doing*, as expressed by the thoughts of Schön, Lewin, Floyd, Ehn et al. which were presented earlier in this methodological section.

In the context of using equipment for carrying out tasks (i.e. doing), Heidegger introduced the term *ready-to-hand* as a kind of property that equipment possesses in use. This property of the equipment can not be discovered by merely looking at an object, as it only becomes apparent in use (Heidegger, 1927: 69). In the original text, Heidegger presents the use of a hammer to illustrate the meaning of these two terms, but Dourish's more recent example of the use of a computer mouse seems more appropriate in the context of this thesis. Dourish writes: *"consider the mouse connected to my computer. Much of the time, I act* ***through*** *the mouse; the mouse is an extension of my hand as I select objects, operate menus, and so forth. The mouse is, in Heidegger's terms,* ***ready-to-hand****. Sometimes, however, such as when I reach the edge of the mousepad and cannot move the mouse further, my orientation towards the mouse changes. Now, I become conscious of the mouse mediating my action, precisely because of the fact that it has been interrupted. The mouse becomes the object of my attention as I pick it up and move it back to the center of the mousepad. When I act on the mouse in this way, being mindful of it* ***as*** *an object of my activiy, the mouse is* ***present-at-hand****."* (Dourish, 2001: 109, bold emphasis originally in italics).

When equipment is *ready-to-hand*, it disappears from our conscious attention and disappears into our mental background. One could say that the equipment "just works", which therefore allows us to use it in a natural way to *do the work* without thinking about the equipment. Heidegger emphasises that humans are usually not interested in equipment for its own sake, but rather to use it as a tool that allows them to do their work: *"That with which our everyday dealings proximally dwell is not the tools themselves. On the contrary, that with which we* *concern ourselves primarily is the work"* (Heidegger, 1927: 70). It is only when the workflow breaks down that a tool requires conscious attention again and it becomes *present-at-hand*.

Heidegger's notion of *ready-to-hand* and *present-at-hand* is of interest to this thesis in two ways.

1. *Ready-to-hand* lies at the heart of Mark Weisers' vision of ubiquitous computing, which is the field of research that this thesis contributes to. Weiser argued that *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it"* (Weiser, 1991).

2. Prototypes and mock-ups can "just work", or be *ready-to-hand*, when they remind participants of familiar use situations where they know what to do. And the same prototypes can most effectively reveal problems in the design when they break down, i.e. participants no longer know how to use them. In Heidegger's terms this happens when human perception of equipment in use switches from *ready-to-hand* to *present-at-hand*. We often take background equipment for granted and only appreciate its existence in the case of a breakdown (Winograd and Flores, 1986: 37).

These two views combined chart the course for the work in this thesis. While for the artistic productions that were exposed to the public we ultimately strived to achieve smooth running ubicomp experiences as in the first sense, it is the idea of this thesis to reveal the patchy infrastructures on which these experiences rely to their designers. The intent is to force the invisible wireless background to crack open and become recognisable so that it can be better designed for, which is more in the second sense. This, in turn, ultimately allows the wireless infrastructure to become a tool in the hand of designers that is *ready-to-hand* quite authentically.

1.3.5 Approach of this Thesis

For the two studies in this thesis a participatory design approach was choosen to design and evaluate prototype tools and workflows that bring out the character of the invisible wireless infrastructure layer at design time. The participating end-users were professional designers that had previous experience in designing location-based experiences and thus possessed the necessary competence and tacit practical knowledge that enabled them to provide meaningful feedback on the prototypes. Although we did not work with trade unions and had no political background, our approach to designing was very much in the tradition of Scandinavian participatory design: we also engaged our end-users in practical activities that would allow them to experience the future by using mock-ups and early prototype tools and reflect on their actions. And we cooperated with our end-users at eye level.

1.3.5.1 Designing for Designers

The word "designer" needs to be clarified in the context of this thesis as it will otherwise cause ambiguity. In the terminology of the Scandinavian design tradition designers plan, facilitate and study the participatory design process in cooperation with end-users and other stakeholders. In this sense, I – the author of this thesis – am the designer. But my end-users just happen to be designers by profession! In order to not constantly use the word "designer" to speak of both, my role in the process *and* the profession of my end-users, I will henceforth refer to my own role as that of a researcher or author and refer to my end-users as designers or artists.

1.3.5.2 Work Organisation

This thesis evaluates the experiences gained from working with two artist groups on the production of two location-based experiences that implemented their location mechanisms in ways other than just relying on GPS. For these two case studies, the author organised workshops, participated in the design discussions, built and evaluated tool prototypes, suggested changes to the artists' workflows and analysed their effects. These changes to the development processes were thought to increase the designers' awareness of the utilised wireless infrastructure at authoring time, which is the goal of this thesis. Consequently, parts of the thesis work had to be conducted outside, where location-based experiences are staged. Overall, in both studies, the work was organised in several iterations that went from technical design to experiencing a prototype and back again. Several activities, such as outside data collection or mock-up authoring on paper, have been conducted with the end-users to stimulate their technological fantasy and advance the design. As our designers were dealing with *invisible* infrastructures, which are naturally hard to sense, we had to make use of sensing technology and visualisation techniques to devise new workflows for the designers. Thus, we were not merely working with mock-ups and language games, but building computer system prototypes, which means that our work was in the tradition of cooperative prototyping, i.e. third generation participatory design.

Our end-users' feedback directly influenced the development of subsequent prototypes. For time-management reasons, and to keep the balance between *must-have* and *nice-to-have* features, we have asked our end-users to prioritise future prototype features based on their perceived usefulness for their current and forthcoming tasks. This functional selection often happened informally, but we also made it a topic at workshops that were organised to decide on future developments. In the case of the Rider Spoke study, we additionally agreed on functionality for specific use situations through a design document.

Although the artists were shaping the overall public experiences, including the game designs, all media assets, and the staging process, they were not directly changing the game-applications or the tools that supported their creations; this work was carried out by teams of researchers and system developers that worked in close cooperation with the artists. Additionally, each of the artist groups also hired an individual free-lance Flash programmer who worked on the public front-end under their direction. Our artists were primarily interested in delivering the result of their work to *their* end-users, which was the public audience. Although this thesis is not immediately concerned with the experience of our end-users' end-users (!), i.e. how the artists' final designs were presented to the public, this aspect might be less familiar to the reader than, say, the presentation of journalistic text in a newspaper[1]. Therefore, in order to provide a holistic view on the design and enable the reader to have a better appreciation of the overall use

[1] To draw an analogy to the UTOPIA project which mocked-up tools for making newspapers.

situation, each study chapter will start with a brief overview of the final experience and with details of the artists' traditions.

1.3.5.3 My Role and Tradition

My role as the author of this thesis was specifically concerned with anticipating practical use situations, and providing appropriate prototype tools for the artists (compare Bødker and Grønbæk, 1991: 200) that helped them to better understand the wireless infrastructure that they were using. I did not write the game-engines for the two experiences that are discussed in this thesis. Nevertheless, I was heavily involved in all technical issues to the extent that I inducted a development team of four programmers to enable them to write the Love City game-engine. I was the facilitator for workshops, coordinator of outside activities, and acted as a general point of call for my colleagues throughout the development phase.

In order to gain the relevant practical knowledge to do this, I got engaged in the development of two location-based experiences before starting to work on the two thesis studies. The first "training" experience was called *Tycoon* and it was a cell ID-based multi-player game for mobile phones that was developed by a visiting student under the supervision of my supervisor. Tycoon needed an authoring tool to define game regions and I had the chance to build a very first prototype of a tool that later evolved in the Love City study. This work was the seed for this thesis and a paper about it got presented at Ubicomp 2006 (Oppermann et al., 2006). The second "training" experience was called *'Ere be Dragons* (now Heartlands) and it was a GPS and heart-rate based multi-player game. I took a leading role in transferring this game from a single-player into a multi-player game, wrote the communication protocol and database back-end, programmed a spectator interface, and also proposed and sourced different mobile technologies to run the experience, which included getting custom built pre-production Bluetooth heart-rate sensors from a foreign research institute. 'Ere be Dragons was commissioned by Hewlett Packard to be presented at the Game Developers Conference 2007; it also won the Nokia Ubimedia MindTrek Award in 2007. Further details of these two training experiences, as well as of other relevant practical involvements, are summarised in appendix 7 (page 366).

I am trained as an interdisciplinary multi-media specialist. I have a degree in Medieninformatik (computer science for the arts and media) and have previous practical experience of developing realtime augmented reality and virtual reality applications. I am an experienced programmer with expertise ranging from very low-level to very high-level languages. I have practical experience from working in the newspaper industry, in online-marketing, in video-production, and in research and development. Apart from my studies, I am used to working with artists in collaborative computer projects in the so-called demoscene since 1994.

1.4 Summary of Contributions

The focus of this thesis is on facilitating the development of authoring location-based experiences. Specifically, it proposes, implementes and evaluates a new approach to authoring this type of applications that more deeply implicates the characteristics of the wireless computing infrastructures into the authoring workflow. This thesis makes the following contributions.

The methodology section provides a condensed overview of participatory design that leads from Schön's "reflective practitioner" over action research and prototyping to the Scandinavian design tradition. It also presents some of the philosophy behind this approach, including Wittgenstein's "language games", Ehn's "dialectical foundation of design" and Heidegger's "ready-to-hand / present-at-hand".

The comprehensive literature review covers examples of location-based experiences and related seminal work, as well as how to build these applications in practice. It also raises awareness of the problem of uncertainty in the wireless computing infrastructure, which is prevalent in these kinds of applications.

The conceptual framework presents a novel approach to authoring location-based experiences that provides a better awareness of the wireless infrastructure. This is achieved by working with 3 layers of information that cover the physical world, the infrastructure (novel contribution), and the content. The framework chapter also presents the design for a general abstract location model that can be used for triggering location events, and outlines foreseeable challenges.

Two user-centred participatory design studies put the proposed framework into practice on two occasions of commercially commissioned location-based experiences that were built in collaboration with two groups of professional experience designers.

The discussion reflects on the feasibility of the conceptual framework after it has been put into practice and studied on two occasions. It highlights the need for a distributed workflow that seamlessly links work in the field and work in the studio. It presents desirable features for the tools that support this work and reflects on how to make sense of location. A reflection on the engineering side returns to the challenges that were set out earlier and provides ideas for system-level improvements. The discussion is closed with reflections on the wider principles for bringing about successful location-based experiences.

The final chapter provides a summary of the thesis document, presents ideas for future work, and draws a conclusion.

1.5 Guide to the Document

Although this document has been designed to be readable from cover to cover, it is expected that few people will actually do so. This section provides a guide to the document so that readers may find their way through the text more quickly.

References are cited in two styles: Harvard and numbered. Harvard style referencing (name, year: optional page) is used for references with clear authorship information, also for online resources. Numbered style [number] is used for the remaining online references. The respective full details are listed at the end of the document in sections "References" and "Online Resources".

Chapter 1 is mostly interesting as a general introduction and for its condensed methodology section about cooperative prototyping. This approach is described in the context of Scandinavian participatory design and some philosophy (page 2).

Novices in the field of location-based experiences might want to start reading about the related work in chapter 2 (page 20). This chapter first provides a jargon buster and presents example applications, before leading over to the technical details. It also presents tools and techniques for spatial authoring (page 86), and closes with the description of a prevalent problem that needs attention (page 110), which leads the reader to the motivation for this thesis (page 116).

Chapter 3 (page 118) provides the conceptual framework for this thesis and presents the core innovation that underpins both studies.

Chapters 4 (page 141) and 5 (page 181) present the core material of this thesis. They provide detailed accounts of two cooperative design studies which have been conducted in collaboration with external partners. Both studies describe development processes that span from inception to delivery of the respective project to a public audience, and highlight the tools and workflows that enabled the designers to accomplish their tasks.

Chapter 6 (page 253) reflects on the two studies, refers back to the conceptual framework, and provides a broader discussion. It highlights best practices and gives recommendations for bringing about successful location based experiences (page 290).

Chapter 7 (page 301) presents a summary and final reflection of the document and gives ideas for future work. It also draws a conclusion and provides a closing thought.

2. Related Work

Location-aware computing has been studied for the past two decades across a variety of disciplines. Depending on the respective discipline, different terms are used to basically describe the same thing, but with slightly different connotations. As a result, terms like location service, location-based service or location-based application are often mixed up, or used synonymously. Küpper identified this lack of a common terminology and called it a dilemma (Küpper, 2005). He suspects this to be caused by the historical growth and use of the terms in separate fields, most notably in telecommunications and ubiquitous computing.

This thesis focuses on the creation of location-based experiences, which are a subset of location-aware computing. To minimise confusion and to assure a common terminology, this chapter starts with a very brief definition of the terms that are being used throughout this document and then reviews the seminal work. Furthermore this chapter summarises how location-based experiences are built, highlights an immanent problem that needs to be addressed and concludes with the motivation for this thesis.

2.1 Jargon Buster

The following provides a brief definition of terms as they are used in this document. It is not intended to provide universally valid text-book definitions, but should rather illustrate how certain terms are used in the context of this thesis.

2.1.1 General Terms

2.1.1.1 Position

A position specifies a point in a coordinate system and can be defined either in absolute or relative coordinates. (52.953412° latitude, -1.187508° longitude) is an example for an absolute position that is defined in the World Geodetic Coordinate System (WGS84). WGS84 is the standard coordinate system for the Global Positioning System (GPS). A relative position is defined in relation to another coordinate, e.g. 20 m north of the above.

2.1.1.2 Location

A location is a position or area with an associated semantic. Locations are more humane in the sense that they are easier to grasp and talk about, but they might be ambiguous. The previously defined absolute coordinate of (52.953412° latitude, -1.187508° longitude) could refer to a location called "Computer Science building" and 20 m north of that could refer to "the bicycle stand".

2.1.1.3 Game

A game is an engaging and rewarding leisure activity that is usually governed by rules and could have an element of competition as well as a winning condition. Webster's provides numerous definitions for the word game, ranging from a brief and general *"an amusement or pastime"* to the more specific *"a physical or mental competition conducted according to rules in which the participants play in direct opposition to each other, each side striving to win and to keep the other side from doing so"* (Webster, 1961). Everybody intuitively knows what a game is, yet it is very hard to define. This is reflected in the existence of a full body of literature on the subject. For example, Huizinga provides a study of the play element in culture and argues that man is because he plays ("Homo Ludens"), but that the game precedes human culture, because animals also play (Huizinga, 1939). Instead of trying to summarise the whole literature in a single paragraph, the interested reader is referred to Wikipedia's extensive overview article (1) which lists many notable definitions, including those of Wittgenstein, Caillois and Crawford. However, for the sake of the argument in this thesis, the first sentence of this section is elaborate enough. Examples of games include soccer, board-games and catch.

2.1.1.4 Experience

Experience is a human-centred, Anglo-American term that can be used as a verb and as a noun. The term is especially hard to differentiate for speakers of the German language, due to different connotations in the English and German language, as pointed out by the *Experience the Experience!* art project (2). In the context of this thesis, an experience often has an artistic background, a limited duration and might be used to transport a certain message. As such it is akin to the performing arts, but might integrate other media. An early example for an experience in this sense is *Poème électronique* by Le Corbusier (direction), Xenakis (architecture), Varèse (music) and Petit (projections), which was staged at the Philips Pavilion at the Expo 1958 in Brussels. It was the first *"electronic-spatial environment to combine architecture, film, light and music to a total experience made to functions in time and space"* (3). The pavilion had no other than this experiential function and was destructed at the end of the exposition; it has been described as a *"Gesamtkunstwerk, or total work of art"* (Cohen, 2006). Experiences are similar to the broad definition of games, in that they are engaging and rewarding leisure activities, and a game can be an experience, but not all experiences are games.

2.1.2 Computing Terms

2.1.2.1 Ubiquitous Computing

Ubiquitous computing (Ubicomp) is a post-desktop computing metaphor that is attributed to Mark Weiser. He argued that *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."* (Weiser, 1991). Ubicomp can be seen as a logical step in the evolution of computing metaphors, which started with mainframes, where one computer was shared by many users, passed through the desktop metaphor, which provided a personal computer for everyone, and now reached the stage where one user controls many computers of different complexities. Ubicomp research is necessarily tethered with a human-centred approach to computing, in order to ensure that people stay in control of the many machines that surround them (Shneiderman, 2002). Weiser argues that the disappearance of computing into the mental background is a fundamental consequence of human psychology. Humans cease to be aware of something once they have learned it sufficiently well and are henceforth freed to use it without thinking so that they can focus on new goals beyond it. He draws several analogies, including literacy, which was once reserved for a few, but is now a common instrument for many.

2.1.2.2 Pervasive Computing

Pervasive computing is another name for the concept of ubiquitous computing, which means exactly the same thing: computers pervade the everyday life, or, as Weiser put it: *"they weave themselves into the fabric of everyday life"* (see above). Other notable names for the same concept include Ambient Intelligence and Everyware (Greenfield, 2006).

2.1.2.3 Context-Aware Computing

Context-aware computing is another post-desktop concept. It seeks to sense and process the human context in order to provide the most appropriate services to the user in his current context. Such a system comprises of a range of networked mobile and stationary devices that span a *"multitude of situations and locations covering the office, meeting room, home, airport, hotel, classroom, market, bus, etc."* (Schilit et al., 1994). The seminal work in the field was done at the Palo Alto Research Centre (PARC), which built and evaluated the ParcTab (Schilit et al., 1993) prototype. Its authors highlighted the important aspects of context to be: *"where you are, who you are with, and what resources are nearby"*. They further elaborated that context-aware computing should go far beyond a user's location and also sense other information, such as *"lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social situation, e.g. whether you are with your manager or with a co-worker."* (Schilit et al., 1994).

2.1.2.4 Location-Aware Computing

Location-aware computing is a subset of context-aware computing and a wide field on its own. As the name suggests, work in this field is mainly concerned about sensing and processing location.

2.1.2.5 Location Service

A location service is a tool or device that is used in location-aware computing. It provides location information about objects or persons of interest. An example of a location service is a GPS device, which provides an application or individual with position information in geographical coordinates.

2.1.2.6 Location-Based Application

A location-based application is program or product or usage scenario in the field of location-based computing that integrates one or more location services, but does not necessarily comprise a network component. Generally speaking, it is something that adds value based on location information. An in-car navigational system is a typical example of a location-based application.

2.1.2.7 Location-Based Service

Location-based service is a term that is very similar to location-based application and sometimes used synonymously (Küpper, 2005). The use of this term usually suggests the existence of a network component and a service provider, thus making location-based service the preferred business term.

2.1.2.8 Augmented Reality

Augmented Reality (AR) is the plausible combination of digital information with the real environment. The term is usually used to describe a sophisticated system which adds computer generated imagery to a user's view of the world. According to Azuma's definition, an augmented reality system bears the following three characteristics (Azuma, 1997): it combines the real and the virtual, it is interactive in real time, and it correctly registers the virtual with the real environment (in all three visual dimensions). Augmented reality research is usually concerned with the engineering problem of improving tracking algorithms in order to minimise the visual registration errors. An overview of different applications of AR can be found in (Azuma et al., 2001).

2.1.2.9 Pervasive Games

Pervasive Games are games that make use of ubiquitous computing technology in order to *"deliver a gaming experience that changes according to where users are, what they are doing"* and potentially also how they are feeling (Waern et al., 2004). On first sight, pervasive games have a particular focus on unchaining players from their desktop and making use of location information, but this is not necessarily the most important aspect, as they also challenge the boundaries of games per se. Montola summarises that pervasive games seek to expand the boundaries of play in three dimensions: space, time and social (Montola, 2005). More on the theory and design of pervasive games can be found in a recent book (Montola et al., 2009a).

2.1.2.10 Location-Based Experience

Location-based experiences are akin to pervasive games; but, as previously discussed on page 21, experiences are not necessarily games. Moreover, pervasive games are not solely concerned with spatial expansion, as outlined above. This thesis is concerned with making location information more manageable for designers that seek to deliver some kind of location-based experience (Benford, 2005) to the user. This term also better reflects the human-centric approach that we take in the development phase, and it will therefore be used in this thesis.

2.2 Examples and Seminal Work

Location-Based Experiences allow the retrieval of geo-coded content based on the user's location. Over the past years these applications have become increasingly popular and hence well studied in a variety of fields such as tourism, games or art. This section provides an overview of archetypical work in the field, presented in a chronological order. The selection process was guided by the principle of presenting work that exhibits key features of location-based experiences and utilises technology in much the same way as it would be used today and in the near future.

2.2.1 Cyberguide Project, 1995-1997

Between July 1995 and August 1997, the Cyberguide project at Georgia Institute of Technology investigated possible applications for mobile location-aware systems. Rather than inventing new hardware, the project focussed on utilizing what was readily available at the time or foreseeable to come, and also tried to keep everything low-cost. Cyberguide envisioned a number of usage scenarios for the then emerging technologies such as mobile computers, wireless positioning technology, and wireless communication. The project built a series of prototypes of mobile, location-aware tour-guides that would assist visitors to the Georgia Tech GVU Lab during the monthly open houses. Their idea was to collapse the typical paper-based map and information pack into a system which would know where visitors were, what they were looking at, and that would also answer typical visitor questions (Long et al., 1996a).

For practical reasons the Cyberguide project built separate indoor and outdoor systems. A modular system architecture – inspired by Schilit (Schilit, 1995) – was applied to encapsulate the overall system functionality into four pluggable components: the map, the information base, the positioning system, and the communication system. Each of these components could be modified or replaced without affecting the remaining components. One of their indoor systems used an array of infrared (IR) remote controls hanging from the ceiling, each repeatedly beaming a unique pattern. Each mobile unit consisted of an Apple Newton (a.k.a. Message Pad) which was linked to an IR receiver via a custom programmed microcontroller that was attached to the Newton's serial port. So whenever a visitor stood under one of the strategically place remote controls, their mobile unit would recognise its beamed pattern, switch the location context of the application and present the relevant information to the user. The Cyberguide indoor system is essentially a near-field cellular location system where the user is placed in one spatial cell until he is reported to be in another one. Subsequent versions of the indoor system utilised the Newton's built-in IR port to get rid of the external IR receiver and microcontroller. Usability issues were reported which were said to be caused by the limited range of the communication channel (1 m) which resulted in too small locations. Similar to the indoor system, the Cyberguide outdoor system also incorporated an Apple Newton. But instead of IR it utilised GPS for positioning via an external GPS mouse. The hardware setup is shown in figure 2.1 (right); the Trimble GPS unit is connected via a serial connection. Figure 2.1 (left) shows a screenshot of the outdoor Cyberguide application. The user's position on the map is updated according to the reported GPS position and users can also freely zoom into and out of the map (the buttons at the bottom of the image belong to the Newton OS).

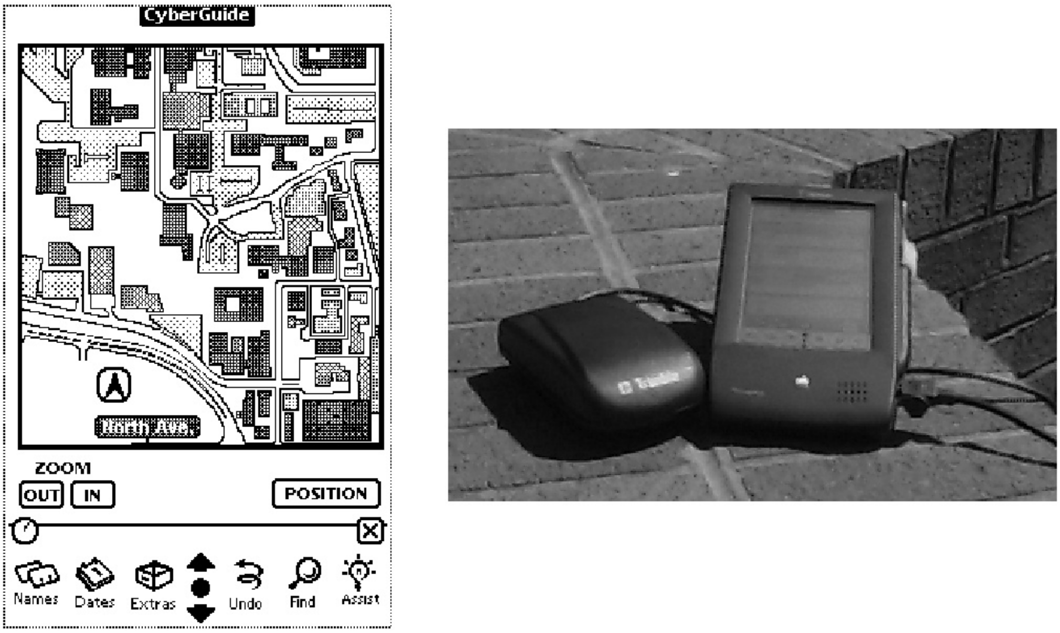

Figure 2.1: The outdoor Cyberguide application (left), hardware (right)
(Source: (Abowd et al., 1997))

The outdoor application as reported in the main Cyberguide literature (Long et al., 1996a, Long et al., 1996b, Abowd et al., 1997) has only been developed to a proof-of-concept stage (Abowd, 1999). A later development called "CyBARguide" sought to put the technology to use by providing a location-based index of local bars, which could also be browsed based on personal preferences and extended with users' annotations. Notes on the CyBARguide webpage (Long et al.) suggest that the accuracy of the GPS-based outdoor positioning were inaccurate by about a city-block. This might have been caused by the so-called "Selective Availability" of the GPS signal (Graas and Braasch, 1994), which was in effect until 02.05.2000 and purposefully caused inaccuracies of about 100 m to non-US-military users. The authors highlighted three lessons that they learned from the overall project (adapted from (Long et al., 1996a)):

1. Cost-effective mobile location-aware applications can be built with readily available hardware (in 1995-1997 – ed.)
2. Absolute positioning information throughout an entire space is not so important. It is far more useful to know what someone is looking at than to know someone's exact physical position and orientation.
3. Positioning and communications should be separated as their coverage might differ over space.

2.2.2 MARS Project, 1996-1999

The MARS project at Columbia University researched into the construction of Mobile Augmented Reality Systems (MARS). It aimed at exploring the synergy of augmented reality and mobile computing and built a couple of prototypical applications. According to the project website (Feiner et al.), the project focused on *"identifying generic tasks a mobile user would want to carry out using a context-aware computing system, defining a comprehensive set of reusable user interface components for mobile augmented reality applications and making combined use of different display technologies, ranging from head-worn, to hand-held, to palm-top to best support mobile users."* The project combined available off-the-shelf hardware into a wearable system, which allowed researching into experimental mobile user interfaces. As the available hardware at that time was still quite bulky, the developers had to make a compromise and traded a lightweight and comfortable wearable system in favour of a system which allowed rapid prototyping but weighted 20 kg. Their first system consisted of a backpack computer (133 MHz Pentium PC with 64 MB RAM, 2GB HD and a 3D accelerated graphics card running Windows NT), a handheld computer with stylus (75 MHz 486 DX4 with 16 MB RAM, 340 MB HD running Windows 95), an optical see-through head-worn display with orientation tracker (Virtual I/O i-glasses), a GPS receiver (Trimble DSM with subscribed differential correction allowing 1 m positioning accuracy), Wi-Fi cards in both computers and a battery belt. The system was used within a campus-wide wireless network and could thus connect to the Internet.

Their first prototype, called the Touring Machine (Feiner et al., 1997), implemented a campus information system. The user could freely roam the campus and retrieve information about buildings. The availability of information was marked through virtual labels which were attached to the buildings in the combined augmented reality view (figure 2.2, top). The user could select the desired label and the system would then open a link in a web browser on the handheld computer, which could branch to other pages (figure 2.2, bottom). Due to accuracy constraints, the virtual text labels were only loosely attached to the buildings and merely floated on the buildings, which did not allow for highlighting specific features. This was a conscious design decision when building the system. The authors summarised that this kind of approximate tracking could be useful for many applications, while many others would require a higher accuracy.

Their second prototype extended the system with a higher resolution HMD (Sony LDI-100B), a separate orientation tracker (Intersense IS-300Pro), a faster handheld computer (233 MHz Pentium MMX) and a real-time kinematic differential GPS (Ashtech GG24, allowing cm-level accuracy) (Höllerer et al., 1999).

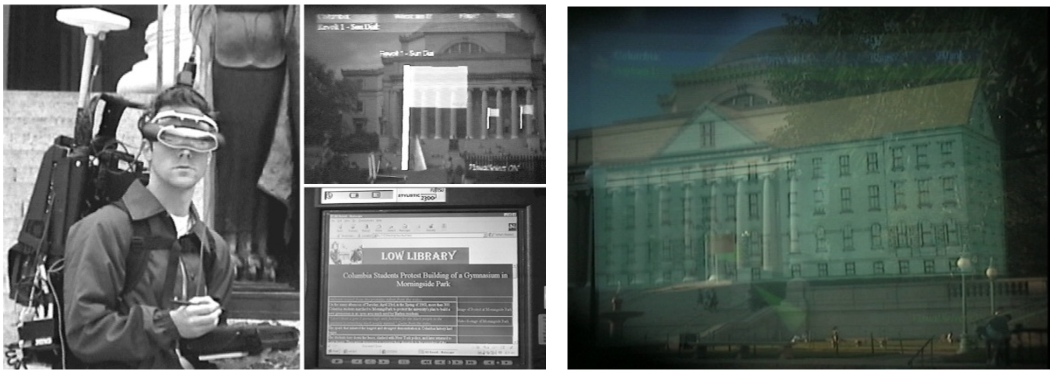

Figure 2.2: User with full kit (left), augmented view (top middle and right), browser view (bottom middle) (Source: (Höllerer et al., 1999))

The system had a similar form-factor to the first iteration and still weighted about 20 kg (see figure 2.2, left). It allowed for visually more complex augmentation as can be seen in figure 2.2 (right), which shows a 3D model of a historic building superimposed onto the user's current view of the world. The model was correctly registered with the real environment and updated at interactive frame rates.

2.2.3 GUIDE Project, 1997-1999

Similar to Cyberguide, the Lancaster GUIDE project (4) produced a location-aware tourist guide. But it considerably extended the scope of view by taking into account real user requirements, sourced from semi-structured interviews with domain experts from the Lancaster Tourist Information Centre (Cheverest et al., 1998), and actually deploying the application in a practical real-world environment. This was followed by an evaluation of the visitor experience (Cheverest et al., 2000). The overall idea of the GUIDE project was to allow users to flexibly configure their journeys according to personal and environmental contexts, i.e. interest, time budget and opening times. The domain experts specifically highlighted the need for a dynamic system which would allow the database to be changed and the system to rearrange a visitor's scheduled tour while he was out and about, as for example Lancaster Castle also hosts the court and is only open for visitors when the court is not in session.

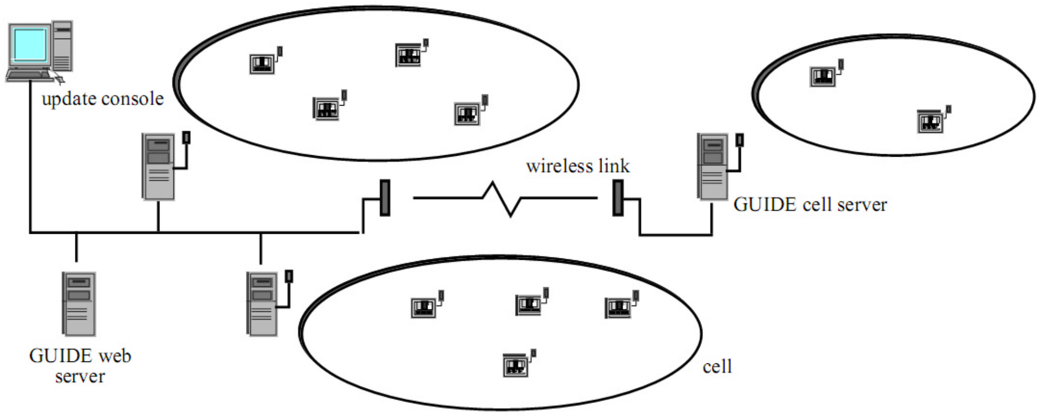

Figure 2.3: Lancaster GUIDE distributed architecture (Source: (Davies et al., 1999))

The distributed GUIDE system relied on a purpose-built high-bandwidth cellular wireless infrastructure (Wi-Fi), which also provided location. Relevant locations in the city were grouped into cells, each of which was provided with Wi-Fi coverage with the help of a designated cell server (see figure 2.3). The cell servers were normal PCs with Wi-Fi adapters that were connected to a central backbone web server via cable or over the air. The system's content could be centrally managed via an update console. When a mobile unit entered a location it received the relevant local content via a custom cell broadcast protocol, i.e. content was periodically sent to all devices in the cell. Of course, because the system was dynamic, what is regarded as "relevant" could differ from user to user and many permutations of the content were possible. Therefore the system also allowed the retrieval of personalised content from the web server over the cell server. The system was designed to have a very responsive user interface to avoid user frustration, yet it should scale to support potentially thousands of users. The available network bandwidth was limited to 2 Mbps and could have proven to be a bottleneck when too many mobile units constantly used the shared bandwidth. To mitigate this problem the GUIDE system used several heuristics to ensure that the most relevant content was likely to be already available on the device. It also implemented several proxy-caches on the server side as well as on the clients.

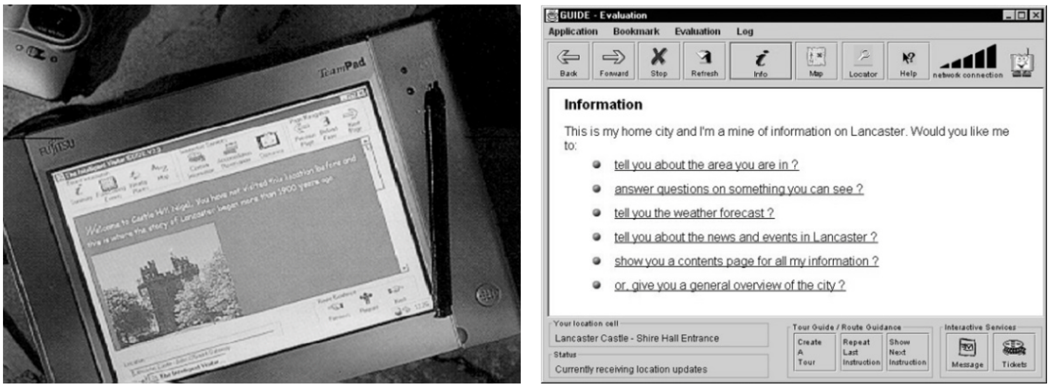

Figure 2.4: Lancaster GUIDE hardware (left), application (right) (Source: (Cheverest et al., 2000))

The mobile units consisted of a pen-based mobile PC, a Fujitsu TeamPad 7600 (A5 sized, 850g), which ran a purpose built browser software (see figure 2.4). The browser's main purpose was to display the supplied HTML pages for the respective location, as received from the cell server. These pages were built with a template mechanism (the authors called it "tags") which created custom HTML pages by combining the HTML template with the user's contextual information.

An interesting feature is the browser's revelation of the current state of the network connection. As the GUIDE project was conducted before the availability of city-wide Wi-Fi coverage, their designers had to cope with intermittent connections. The mobile units were only connected to the network when they were physically close to a network cell and would be disconnected when the user meandered through the city. Some features of the interactive application were not available in disconnected state. In order to avoid frustrating users with seemingly not working devices, the designers implemented a visualisation of the state of the connection via the bar metaphor, which they borrowed from mobile phone user interfaces (see top right of the image). The system also informed the user about the time of the last location fix and where the user is currently located. This was done in textual form at the bottom left of the window. Together, this was reported to aid the user in building a mental model of the availability of services rather than being confronted with imperceivable "failures" in the system. This is one example for seamful design in the sense of Chalmers and Galani that makes seams in systems perceivable rather than hiding them (Chalmers and Galani, 2004).

Another interesting aspect of the GUIDE project is the construction of locations for the experience. Six cells have been strategically placed in Lancaster, each of which was made up by a cell server with Wi-Fi card and antenna. The authors reported how they designed cells to be non-overlapping in order to avoid cell handover and maximise bandwidth usage. They have been physically tweaking the placement of the antennas in order to reach the desired coverage on the ground and reported of one occasion where they managed to configure two adjoining cells to be situated only metres apart without overlapping. But this was an exception and the general consensus was that cells had very blurry boundaries which were affected by environmental factors and thus a bit unpredictable. The granularity of locations in the GUIDE project was rated as sufficient and the authors summarised that designers of similar location-based experiences *"must not be over zealous when deciding to constrain the information or functionality provided by the system based on the current context"*.

2.2.4 Geocaching, 2000

Geocaching is a simple treasure hunt game that uses a minimal set of rules and is played all over the world. It has minimal hardware requirements as it can be played with any GPS device. Its inception is directly linked with the discontinuation of the artificial degradation of GPS signals – called Selective Availability (Graas and Braasch, 1994) – as announced by US president Clinton on 1 May 2000 (Clinton, 2000) and put into effect on 2 May 2000. This means that overnight the accuracy of civil GPS increased by an order of magnitude, which made it more useful for civilians. On the same day, Dave Ulmer posted his proposal for "The Great Stash Game !!" to the "sci.geo.satellite-nav" newsgroup (5). He proposed that with the new GPS accuracy *"[it] should be easy to find someone's stash from waypoint information. Waypoints of secret stashes could be shared on the Internet, people could navigate to the stashes and get some stuff. The only rule for stashes is: Get some Stuff, Leave some Stuff!!"* He further elaborated that closed plastic containers could be used for the stashes. Each stash should contain a logbook and a pencil so that people could record their find with: date, time, what they got and what they put in. People should create own stashes and post their coordinates on the Internet. Ulmer created the first stash, posted its coordinates and by doing so invented a hobby game that is still played like this today (Cameron and Ulmer, 2004). So far hundreds of thousands of caches have been created world-wide. Several websites are used for circulating and discussing the caches; most of them are free of charge whereas others commercialised Ulmer's idea. The name "geocaching" itself was proposed on 30 May 2000 by Matt Stum to avoid the negative connotations of the word "stash" (6).

Geocaching is *the* archetypical example of a GPS-based game. Moreover it is an example of how a location service such as GPS can be turned into a location-based game or experience by having a good idea, structuring it with a few compelling rules and putting it out into the wild.

2.2.5 ARQuake, 2000

ARQuake was a prototypical Augmented Reality (AR) game developed at the University of South Australia. It presented an immersive mixed-reality environment in which the player could roam the game area on the University campus in order to hunt virtual monsters (see figure 2.5). ARQuake (Piekarski and Thomas, 2002) was a research effort which combined the engine of the computer game Quake (7) with the Tinmith Augmented Reality system (Piekarski).

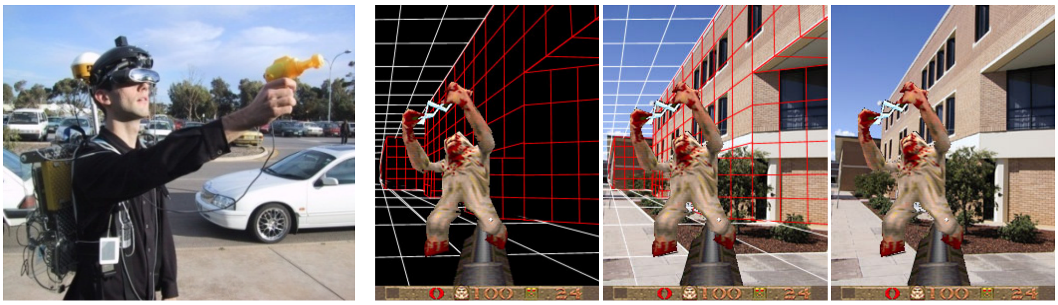

Figure 2.5: ARQuake player (left), virtual monster in real environment (right)
(Source: (Piekarski))

The original Quake was developed and published in 1996 by id Software. It was a genre-defining game: the first truly immersive first-person 3D shooter that allowed total freedom for the player and the designer (Curry, 2007) and one of the first games to be playable in multi-player mode over the Internet. In 1999, id Software made the source code of the Quake engine available under the GPL license and thus allowed hobbyists and researchers to build their work on it.

The Tinmith Augmented Reality system is a combination of hardware in the form of a wearable computer and is originally intended to support interactive 3D modelling in an outdoor AR environment (Piekarski, 2004). The version used for ARQuake comprises a laptop computer, a headmounted display, GPS, a digital compass, a camera and a trigger device (see figure 2.5, left). The developers of ARQuake modified the freely available Quake engine so that it accepted user input from the Tinmith system instead of the conventional keyboard and mouse input channels. They also designed a virtual world that matched their campus so that the game's monsters would be appropriately placed in the real environment and would not walk through walls. This virtual campus model was only used to correctly align the virtual and the real world and was not displayed to the user (see figure 2.5, right).

ARQuake used a hybrid positioning system which combined sensor data (GPS and digital compass) with computer vision. The developers anticipated that the quality of the position and orientation values from the sensors alone would not be sufficient for believable AR, especially when the user would be closer to a building, where GPS is known to be less precise due to signal noise caused by a multipath effect[2]. They also wanted the system to work within buildings, where GPS was not available, and thus also used marker based tracking through ARToolkit (Kato and Billinghurst, 1999). The visual markers would be printed out in a large format, attached to building walls and corridors, and aligned with the corresponding place in the virtual world. Whenever the Tinmith system recognised such a marker through the camera, it would calculate the marker's orientation and relative distance from the user's current real world. This information would then be used to deduce the required virtual camera position and shift the virtual view to (hopefully) match the user's real view. If no markers

[2] GPS signals are reflected off building walls and confuse the receiver by arriving multiple times, causing it to calculate an erroneous position

could be found through the camera, the system would revert to its GPS and digital compass sub-systems and calculate the user's position and orientation from those sources.

Due to its technical focus, ARQuake must be seen as a technological demonstrator, rather than a user-centred experience. As a game it could not possibly hope to improve the most successful original. Its players reported some level of enjoyment, but also noted that misalignment of the real and the virtual caused irritations (Thomas et al., 2002). But as a research prototype, ARQuake was very successful. Despite its technical achievement, ARQuake basically demonstrated that even the combination of the most advanced technology available at a time does not ensure an enjoyable user experience.

2.2.6 Botfighters, 2001

Botfighters was one of the first commercial location-based games. It was a massively multiplayer online role-playing game that utilised the location-based aspect offered by mobile phones (Dreher, 2008). Players used the game website to design their robot game avatars and subsequently went outside on the streets with their mobile phones to battle against other players' robot avatars. The game constantly scanned the virtual environment based on the player's physical position and signalled players if their robot was near another robot (Struppek and Willis, 2007). Players then received a description of where the other robot could be found, e.g. "MegaBot is at Times Square, 700 m NW" (see top inset in figure 2.6, left), and had to physically move towards that location in order to get into fighting distance. Fights were carried out via SMS and players exchanged bullet SMSs in order to attack each others. A fight lasted until one robot won or its player surrendered (Woznicki, 2002). Players could be contacted by the game or attacked by other players at any time, even at night. They had the option to park their robots, but this did not protect from them from being attacked by other players, it only suppressed the delivery of SMS messages at inconvenient times (Bjerver, 2006). On the game website (see bottom inset in figure 2.6, left), players could view highscores, buy weapons, receive missions, monitor their progress and participate in an online-community – but the actual fighting took place outside.

Botfighters was developed by Swedish company It's Alive and originally announced in November 2000. The first version used SMS messages to communicate with mobile players. This technological decision (SMS is the least common denominator for non-speech data communication in mobile phones) made it an attractive and immediately marketable game. The game has been licensed to network operators in different countries and has been played in Sweden, Finland, Ireland, Russia and China (Dreher, 2008). Botfighters was launched in April 2001 and had 40000 registered players. At its peak time, about 3 month after its launch, the game processed 4 Million SMS / month (8). The game received a distinction at the Ars Electronica 2002 (9). In their statement, the jury highlighted the pervasive aspect of the game: *"[..] the player is a part of a continuously ongoing role play adventure, taking place in a virtual world draped over the real physical world. Through the mobile phone and the web the player may interact and build relationships with other players out on the streets. As the player's physical location will influence how the game evolves, it's not always easy to tell reality from fiction."*

A second version of the game was designed to employ a graphical J2ME user interface, but it remains unclear whether this version was actually ever deployed to the public, as it could not be tracked down beyond some initial concept art and marketing presentations, and was reportedly not online by 2006 (Bjerver, 2006).

Figure 2.6: Botfighters artwork with inset web- and mobile interface (left), Battletech artwork (right) (Sources: (Söderlund, 2005), (10))

As can be seen in figure 2.6, Botfighters' theme (left) seems to be inspired by the much older table-top role-playing games from the fictional BattleTech universe (Weisman, 1998) (right), as originally devised in 1984. The contribution of the designers of the Botfighters game is thus not the original game idea but rather that they transferred the original pen and pencil based game design and turned it into a commercial computer game that made use of the Internet and location-based technology.

2.2.7 Can You See Me Now?, 2001

Can You See Me Now? (CYSMN) is an artistic location-based game of catch that mixes play in a virtual world with play on the streets. Up to 20 online players at a time are chased trough a 3D virtual model of a city by up to 4 runners – street players who have to traverse the real city in order to catch the online players (Benford et al., 2006). While online players are members of the public, the street players are professional performers and members of the artists' team (see figure 2.7, left).

Figure 2.7: A CYSMN runner checking his PDA for nearby online players (left). An online player (white silhouette) in proximity to a runner in the 3D virtual world (right)
(Source: (Benford et al., 2006))

The virtual world (see figure 2.7, right) is implemented in Adobe Director and makes use of Shockwave 3D. This means that the game can be accessed from any web browser that has the Shockwave plug-in installed. The players' view of the virtual world is in a third person perspective, like in many other 3D computer games, i.e. the players see themselves as avatars. The controls are also similar, as players use the keyboard to navigate in the 3D world. Players can freely roam the game area, which usually has a side length of a few hundred metres, but they cannot enter buildings or cross virtual fences that limit the game area. The players' objective is to stay in the game as long as possible, i.e. they have to avoid being caught by the runners. The game has a chat function which enables players to send public text-messages to everyone, including the runners.

Runners are equipped with GPS-enabled PDAs which are wirelessly connected to the central game server. Each runner's PDA shows a digital map of the area with the online players' positions and names super-imposed. The runners' GPS positions and the online players' positions in the 3D virtual world are mapped into the same virtual coordinate system to create a shared virtual environment between players and runners. The important notion here is that the virtual city model is geo-referenced with the real city so that coordinates of players in the online virtual city can be exactly transformed into real world GPS-coordinates, and vice-versa. Runners use walkie-talkies to coordinate their activities and exchange crucial information such as traffic conditions. This communication channel is broadcast to the players so that they are aware of the physical labour involved in chasing them and to build up the atmosphere of the game. As the runners are professional performers, they also personally address those players that they are after in order to further increase this effect. A player is caught when a runner comes within 5 m of their vicinity. Their game is then over and the time that they lasted constitutes their final score. As a final action the runner takes a picture of the location where the online player was caught and then turns towards hunting the next prey. After the game, these pictures are revealed on the project website (11) together with game statistics and a spatial overview of where in the virtual world players have been caught.

Can You See Me Now? has been developed by the artist group Blast Theory in cooperation with the Mixed Reality Lab at Nottingham University and premiered in Sheffield in 2001. Since then it has toured internationally, visiting cities such as Rotterdam, Oldenburg, Cologne, Barcelona, Tokyo and Chicago (Benford, 2007). The game has been nominated for an Interactive Arts BAFTA Award in 2002 and won the Golden Nica for Interactive Art at Prix Ars Electronica in 2003 (12). In their statement, the jury highlighted the interweaving of the real and the virtual world and how the online players' understanding *"that the runners are tired, cold, struggling with the environment on the KopVanZuid (a part of Rotterdam – ed.) can become a powerful emotion. A player from Seattle wrote: I had a definite heart stopping moment when my concerns suddenly switched from desperately trying to escape, to desperately hoping that the runner chasing me had not been run over by a reversing truck (that's what it sounded like had happened)."*

2.2.8 DefCon 10 WarDriving Contest, 2002

WarDriving is originally a hobbyist activity carried out by computer security experts who were interested in statistics about the uptake of wireless network technology, esp. their security settings and spatial spread. The name of the activity is derived from the popular 1983 hacker-movie "WarGames", in which a computer whizz kid finds a route into a secret military computer by means of an automated modem scan of telephone numbers from his home computer and accidently almost starts a war (13). This activity of scanning the wired telephone network for connectivity to remote computer systems was known as WarDialling. With the advent of wireless networks in the late 1990s, this activity was gradually transferred from wired home computers to wireless laptops and people started to roam the public space in order to discover external wireless networks that would broadcast their identity into the streets. The automation of this survey process is credited to Peter Shipley of Berkeley, who wrote a dedicated piece of software that would augment the wireless scan results with GPS positions to allow spatial analysis and map making. He installed the kit (laptop, GPS, antennas, logging-software) in his car, conducted a survey of the wireless networks in the San Francisco Bay Area in 1999/2000 (Poulsen, 2002), and presented his results at the DefCon 9 hacker conference in Las Vegas in 2001 (Hurley et al., 2004: 3). Shipley's primary interest was in network security. For ethical reasons he did not reveal the locations of the networks that he discovered during his survey as a significant number of them were unsecured and could have been easily abused if their location was made public.

Others were less concerned about this and picked up on Shipley's idea, which led to the creation of community websites such as Wigle.net (14), where participants share their survey data and collaboratively produce maps of the wireless network environment. The Wigle website, for example, is online since 2001 and has so far (at the time of writing in December 2008) mapped over 16 Million access points with a community of over 81000 users. Figure 2.8 (left) shows a map of the world according to Wigle. The overlaid heatmap highlights areas of Wi-Fi coverage that have been surveyed by the Wigle community. It can be seen that the Wigle WarDriving community is predominantly active in North-America and Europe.

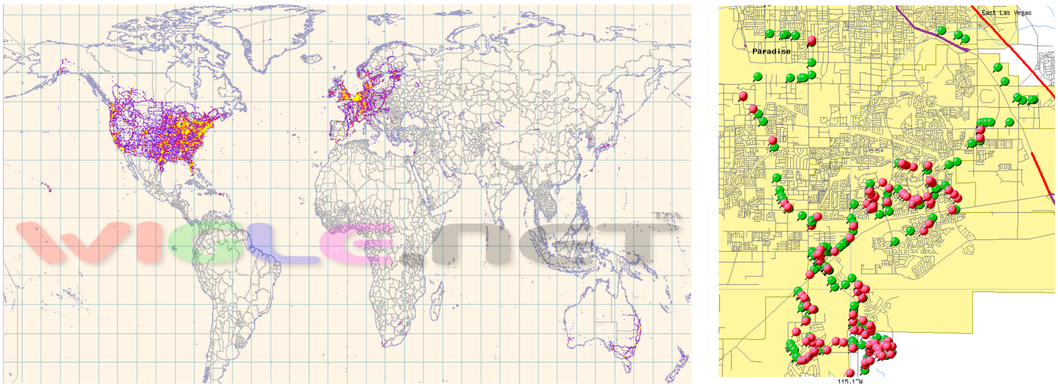

Figure 2.8: Heatmap of 16 Million geo-referenced wireless access points from Wigle (left), trail of one team from DefCon 10 WarDriving Contest (right) (Sources: (14), (15))

WarDriving has been turned into a contest for the DefCon 10 conference which took place in Las Vegas from 2nd – 4th August 2002. The organisers highlighted the social aspect of carrying out WarDriving and presented a set of simple, yet compelling rules that emphasised the idea of mapping out a city without breaking into other people's wireless networks. At DefCon 10, the game was played by teams of four for the duration of two hours. Participants roamed the city of Las Vegas with their own equipment in search for wireless networks. When players returned, a program would analyse their collected log-files and award points based on the following schema:

- 1 point for every discovered access point
- 2 points extra if it still had a default Service Set Identifier (SSID) and did not have Wired Equivalent Protocol (WEP) encryption enabled
- 5 points extra if no other team found this access point

The DefCon 10 WarDriving Contest was played by about 80 players in 21 teams. In their post-event reflection, the organisers highlighted the need for a legal disclaimer when organising such event-based games. Even though (or because) they presented their game at the world's largest annual gathering of hackers, they deemed it necessary to waive the conference hosts from any liabilities that might occur from the players actions on the streets (Hurley et al., 2004) (pp. 259-263). Although this put off some potential participants – 20 people refused to sign the disclaimer – it was seen as a crucial measure. A second crucial measure was that the organisers had to spontaneously relax the game rules in order to make the game safer for its participants. The organisers did not expect to attract too much interest with their event and anticipated maybe 20 participants. The original game design allocated a fixed two hours time-slot and basically expected participants to perform a Le Mans start into the game. Having received four times the number of people for their game, the organisers were concerned about road safety when that many people simultaneously tried to leave a hotel car park in Las Vegas and spill out into the streets. To counteract this potential risk they devised the game rules which then stated that each team's game would start when they turn on their logging equipment and run for two hours from that moment. The attitude of DefCon attendants (read: hackers) caused a big problem for the organisers. They got hacked in multiple ways soon after they announced the presence of an FTP server on the DefCon wireless network where participants' should upload their log-files. In addition to distress for the organisers, this also imposed an additional burden upon participants who just wanted to complete the game by uploading their log-files. Due to continuous attacks, about one third of the participants finally had to hand in their data on physical media such as CDs or USB sticks. This experience led the organisers to reconsider the importance of security and access control for future revisions of the contest where they would use a wired connection to the central server to better shield the system from outside attacks. Another problem for the organisers was the diversity of data-formats that had been submitted. It took them half a day to transcode and manually fix all log-files so that they could be rated by the rating script which relied on a specific data-format. This task was originally perceived to be much smaller and caused serious delays to the overall process. Nevertheless the organisers managed to rank participants and also produce maps from their submitted data (see figure 2.8, right). Subsequent version of the contest required that participants submit their data in a specific format.

The success of the first DefCon WarDriving Contest made it a recurring event at subsequent DefCons, sometimes with slightly modified rules that led to different game experiences. It also led to four WorldWide WarDrive events that took place between late 2002 and mid 2004 in which participants from all over the world mapped the wireless networks in their area. The DefCon 10 WarDriving Contest is an early example of a location-based experience that employs both GPS and Wi-Fi. By defining a few simple rules, the organisers turned a niche hobbyist activity into an engaging competition that led participants to probe the availability of urban Wi-Fi networks in order to win the game. The organisers encountered and acted upon many of the obstacles that would occupy organisers of location-based experiences for years to come. Examples of these include legal issues, safety matters, inappropriate workflows, and general provisions when staging such events, as well as security concerns and ethical reasoning about acceptable use of the infrastructure provided by others.

2.2.9 Noderunner, 2002

Noderunner is a race game that exploits the ubiquitous urban wireless infrastructure and incorporates it into the game mechanics. It is much similar to the WarDriving Contest described above in that two teams of players are sent out into the city and race against time to find as many open wireless access points as possible. They are provided with taxi money, so that they could traverse the city quickly. Players use laptop computers to scan for open wireless networks. If they find one, they must use a digital camera to provide a photographic proof (figure 2.9, left) of the sighting in the exact spot where they found it and submit that photo with annotations about the location and the access point's name (SSID) to a central weblog; the game rules require that this must be done by using the discovered access point only. Each game session lasts for about 1-2 hours. During this time the photos from the street players are projected onto a map of the area which provided a spectator interface at the central game headquarter (figure 2.9, right).

The weblog documents the process and can be used to revisit a game's locations and have an overview of public spaces with free wireless connectivity (figure 2.10, left).

Figure 2.9: Players found an access point (left), spectator interface (right) (Sources: (16), (17))

Noderunner was created for an exhibition called "We Love NY: Mapping Manhattan with Artists and Activists" in August 2002 (8 - 24) and was awarded a Golden Nica at the Prix Ars Electronica for Net Vision / Net Excellence in 2003. In their statement the jury highlighted the ubiquity of wireless networks in urban environments and how these got incorporated into the Noderunner game: *"Noderunner's playing field is the available WiFi spillover in a densely populated area. The density of this spillover is so great that it can be used as a legitimate wireless network. For example, in New York City it is now easier to find an open and free 802.11 hotspot than it is to find a public restroom."*

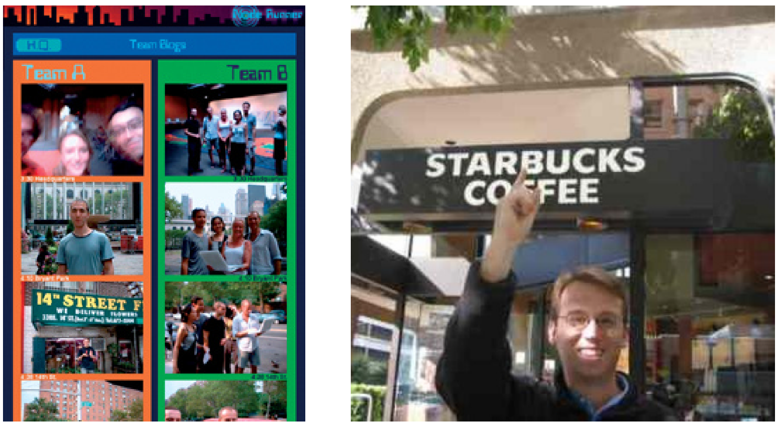

Figure 2.10: Weblog (left), player piggybacking on a Starbucks hotspot (right)
(Sources: (16), (17))

2.2.10 Uncle Roy All Around You, 2003

Uncle Roy all around you (URAAY) (18) was a follow-up game to Can You See Me Now?. It was also developed by Blast Theory in collaboration with the Mixed Reality Lab at Nottingham University and premiered in June 2003 at the Institute of Contemporary Arts near St. James's Park in London. URAAY was a much slower paced experience where participants collaborated in order to solve clues and finally find an elusive character called Uncle Roy. The game altered some of the key aspects of CYSMN while retaining others. It still mixed online play in a virtual world with play on the streets but this time the technology was put into people's hands, as both the online players and the street players were members of the public. Performers were also involved and they were crucial to the experience, but they were actors in a real-world adventure game that took place on the streets of London rather than relentless hunters in a fast virtual action game. The game mixed elements of street theatre and computer games in a real world setting. Its authors summarised that *"Uncle Roy All Around You was primarily focussed on delivering an experience to the street player in which online players could also engage"* (Flintham et al., 2003a).

Street players arrived at the hosting venue and were asked to hand over all their personal belongings, including wallets and mobile phones, in exchange for a handheld computer. They were then introduced to their mission as well as taught how to use the game interface on the handheld computer. The interface showed a digital map of the game area (see figure 2.11. left). Street players would self-declare their positions in the game by using the PDA's stylus to drag the "ME" icon on their screen to the desired location and then pressing the "I AM HERE" button (Benford et al., 2004b). They could also rotate the map according to their preference (Seager and Fraser, 2007) and zoom in to get a more detailed view of the area with annotated street names. The street players' first task was to find a red marker on the map, move to the physical location and then declare their position. Once that task was completed, they were led through the game area by several clues from the elusive character of Uncle Roy which appeared on their screen whenever they declared their position with the PDA. These clues appeared to be sent by a human being, but were actually triggered based on a player's reported position. Not all of the clues were helpful and some were even misleading. Street players had to cooperate with online players in order to solve all clues and finally find the office of the elusive character. For this reason they could record short audio messages and send them to the game server which made them available in the online virtual world.

Figure 2.11: Mobile user interface w. map overview (left), 3D virtual world (right)
(Source: (Benford et al., 2004a))

Online players navigated their own (white) avatars through a 3D virtual model of the game area. They could follow the progress of street players, which they usually saw represented by red auras (see figure 2.11. right). They could also browse a profile page for each street player which contained their name, gender, a short description, and a picture of the player which has been taken at the beginning of the game. Online players were made aware of the clues that street players received from the game when they declared their positions. Whenever this happened, a dramatic sound was played and the red aura of a street player changed its shape in an animation to visually emphasise the event. Online players knew more than street players. They could explore the game area and find visual artefacts – one of them revealed the location of Uncle Roy's office, which was what the street players were looking for. It was the online players' decision if they wanted to help players on the street or mislead them. In either case they communicated with them via text messages that appeared on the screen of the street players PDA just like the location-based clues of Uncle Roy did. The game exploited issues of trust. When street players arrived at the office within the allotted game time they entered the final phase of the game. They rang the door bell, the door slid open and their device instructed them to step inside and have a look around. The office was deserted but showed signs of recent habitation: the lights were on and the radio was playing. They found a postcard which contained the question: "when can you begin to trust a stranger?". They noted their answer on the postcard (figure 2.12, left), looked into a CCTV camera imagining a stranger and were finally asked to leave the office, taking the postcard with them.

Figure 2.12: Player in the office (left), actor in limousine (right)
(Sources: (Benford et al., 2004a) (19))

Players were told to wait in a telephone box. When the telephone rang, they answered it and were instructed to enter a waiting limousine. When they did, an actor climbed in as well (figure 2.12, right) and the limousine left the spot, taking them back to the starting point. On their way, the actor asked them a series of questions related to trust which culminated in the optional commitment to be available for their online counterparts for the period of a year if they needed someone to attend to. When they agreed, the car pulled over by a post-box and the players were asked to post their prepared card to manifest their commitment (Benford et al., 2004a). At the same time online players were asked the same series of questions. If an online- and a street-player mutually agreed, their addresses were exchanged to remind them of their promise.

The game deliberately avoided using GPS for positioning and instead relied on self-reported positioning. One of the artists commented: *"We dumped GPS because it's too flaky, and we knew it would mess with players' enjoyment."* (Adams, 2003). Uncle Roy All Around You has been nominated for several awards, including two BAFTA Awards for Interactive Arts and Technical & Social Innovation in 2004, a Webby Net Art Award in 2004, and received an Innovation Award from the Arts & Humanities Research Board in 2003 (20).

2.2.11 Mogi, 2003

Mogi is a location-based scavenger hunt or collection game. Players move around with their mobile phones to hunt down, collect and subsequently trade virtual items which are placed at locations all over Japan (see figure 2.13). Similar to the example of Uncle Roy All Around You it links play on the street with web-based play and encourages communication and collaboration amongst players to the extent that a web-based player might send a message to a street player in order to guide them. But instead of requiring live performers for the real world setting, this game builds on its community whose players can either play online or on the streets as they see fit (Joffe, 2005).

Figure 2.13: Mobile interface: map, radar, collecting items, trading items (f.l.t.r.) (Source: (21))

Due to its lowered requirement of human resources, Mogi could be deployed country-wide. The game is available on the Japanese KDDI network and utilises the networks "eznavigation" service, a mixture of A-GPS and CDMA cellular technology that builds on Qualcomm's gpsOne technology (22), to provide seamless game play whenever it is required. The game's website (21) mentions that the game uses two positioning methods, cellular and GPS, *"as they both have advantages"* and elaborates that A-GPS is precise but more complicated to use, due to the time it needs to obtain a satellite fix, and more expensive, due to the occurring network costs. Cellular positioning on the other hand is described as cheap and fast but less precise. Apparently the game has been designed to accommodate the lack of precision in cellular positioning as the website recommends that players used cellular positioning most of the time and only switched to GPS when they needed the added precision to hunt down an item in the streets. Mogi's game concept does not require the presence of a specific technology and merely needs some x/y coordinates in order to work (Joffe, 2004). At the time of writing Mogi is still available on KDDI network but seems to have failed economically. In a late article (Joffe, 2007), one of the authors of the game summarised that "*the lack of marketing and the prohibitive data-pricing prevented the spread of Mogi.*"

2.2.12 Riot! 1831, 2004

Riot! 1831 is a location-based audio guide that was staged at the Queen Square in Bristol for a period of three weeks in spring 2004. It took place on the same square where the Bristol riots of 1831 occurred (see figure 2.14, left). The riots were one of the most dramatic episodes in the history of the United Kingdom and caused great damage to the city, injuring or killing many of its inhabitants (23). Participants of Riot! 1831 experienced a location-based audio play which let them immerse into the past and listen to fictional anecdotes that are based on the real events. The audio-content is triggered by the GPS position of the participant and delivered by a PDA through a set of headphones (see figure 2.14, right).

Figure 2.14: Aerial view of Queen Square (left), participant with headphones (right)
(Source: (Reid et al., 2005a))

The creators of Riot focussed on a lightweight mobile platform that would transport their content. As such the piece is similar to the GUIDE project, but the hardware is even more lightweight and the assets are professionally produced. The installation also does not require any kind of on screen interaction as it is audio-only and basically presents a location-based "sea of voices" which allowed its participants to immerse into the stories of a historic event of the past (Reid et al., 2005b).

2.2.13 Treasure, 2004

Treasure (a.k.a. Bill) is a location-based multi-player game developed at the University of Glasgow. It combines GPS positioning and Wi-Fi communication in a way that forces its players to reason about the gaps, or seams as proposed by Weiser (Weiser, 1994), in the wireless infrastructure while they are playing (Barkhuus et al., 2005).

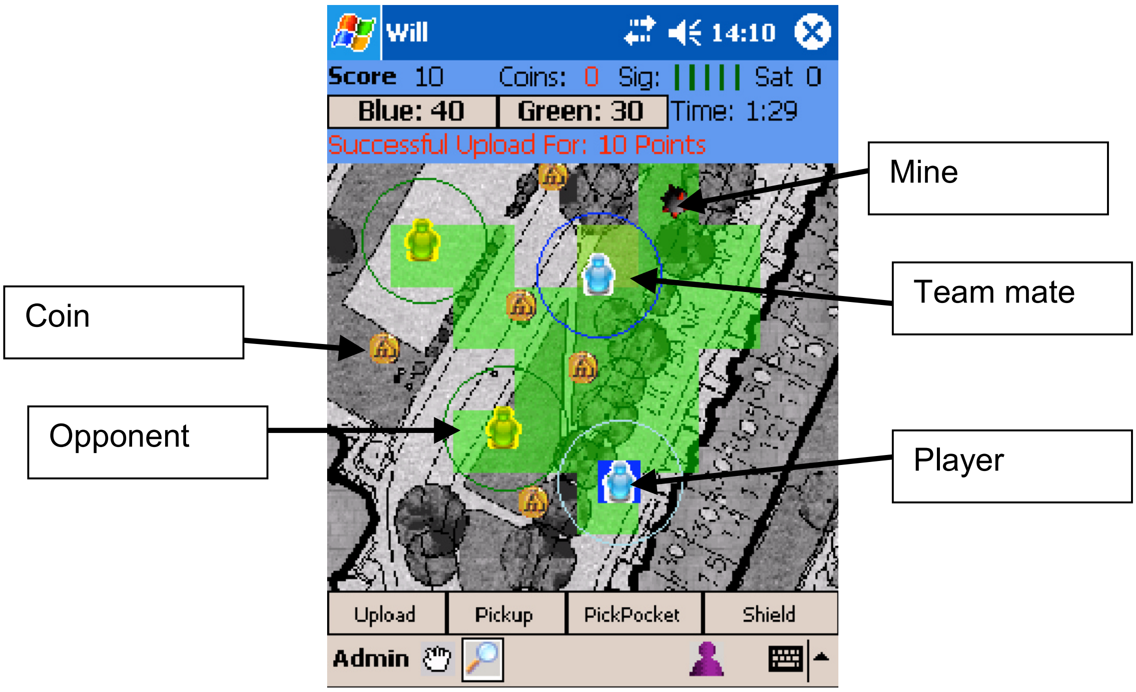

Figure 2.15: Treasure interface, showing the position of players, coins and network availability (Source: (Chalmers et al., 2005a))

The goal of the game is to collect coins, which are only available in areas of poor Wi-Fi coverage, and then bring these coins back into areas of good coverage and "upload" them to gain points. The players' positions are constantly tracked by GPS and this information is also used to determine if a player picked up a coin. Position data is combined with the Wi-Fi coverage data so that over time the participants of the game map out the Wi-Fi coverage of the gaming area. This information is represented in the graphical user interface (see figure 2.15) as coloured squares, where green represents a good signal quality and yellow a weaker one. The game is played with two teams, each consisting of two players. Extra points can be gained through a combined upload of team-mates. Players can also steal coins from opponents by using the "PickPocket" command in their vicinity; this attack can be blocked by using the “Shield”. As a further obstacle, mines are randomly placed in the area. When players walk over them, they loose all their coins and are disconnected from the game for 60 seconds.

The game is similar to Noderunner in that it requires participants to probe the availability of Wi-Fi in order to advance in the game (Chalmers et al., 2005b, Chalmers et al., 2005a). Treasure requires participants to move in and out of Wi-Fi networks in order to find and claim virtual coins that are scattered across the gaming area but which may lie outside of Wi-Fi coverage, thus forcing the players to reason about the availability of the game infrastructure. As such it makes the patchy structure of Wi-Fi coverage an integral part of the game and provides an interesting example of how a survey exercise can be incorporated into the game design.

2.2.14 MOSES, 2004

MOSES is a location-based tour guide which was developed at the University of Applied Sciences Trier for the State Garden Show 2004 in Trier. The funders required that the system should work on off-the-shelf mobile devices without requiring any additional location tracking hardware such as GPS (Schneider and Greving, 2005). The resulting system runs on Windows Mobile PDAs and locates the visitor’s location via infrared, as infrared interfaces were built into many PDAs of the time.

Figure 2.16: State Garden Show site (left), position update via infrared (right) (Source: (24))

The State Garden Show in Trier was open to the public for 6 months from 22nd April – 24th October 2004. It hosted about 3000 events and attracted over 700000 visitors; its site encompassed 44 hectare. The MOSES system was made available for public hire for 119 days (6 days a week) and attracted over 700 visitors. In addition, the system was also available for tech-savvy visitors to self-install on their own devices from a smart-card (15 requests) and as a download (132 downloads).

The MOSES system features a complete site map (zoomable in 3 steps), information to all exhibits, the complete event calendar and the menu of the catering. The information was prepared for two different display modes: location-tracking mode and browse-mode. To support the location-tracking mode, the map has been subdivided into 15 thematic zones (figure 2.16, left). An infrared beacon in a weather protected shell was placed at a central location in each of the zones. When received by a PDA on location-tracking mode, the transmitted location ID caused the display of the PDA to switch to a local area map and highlight the current position (figure 2.16, right). From here, the user could tap on exhibits of interest in her vicinity. When set to browsing mode, MOSES resembled a hypertext system. The available information was grouped into 6 categories (overview, map, children, catering, events, and service) and presented in a hierarchical fashion.

The evaluation of the system reported an average usage-time of over 3 hours. The authors reported a high acceptance rate amongst their participants and considered PDA-based tour-guides as mature enough for commercial deployment. A reoccurring request of the users was for a path-finding function which would guide visitors between two exhibits of interest on the large site. The authors concluded that press coverage was essential for their system to be noticed. They initially had very few visitors asking for a tour, but this changed significantly after repeated TV and newspaper coverage.

MOSES is an example for how even a very coarse grained positioning technology (15 beacons across 44 hectare equal 1 beacon per 30000 m^2) can be incorporated into a working system that can be deployed to the public.

2.2.15 CAERUS, 2005

CAERUS is an integrated platform for location-based learning which has been developed at the University of Birmingham. The system has been designed for outdoor tourist sites and educational centres and has been trialled with a tour through the University's own botanic garden in May 2005 (Naismith et al., 2005). CAERUS comprises of a desktop content management system for curators and a handheld content delivery system for visitors.

Figure 2.17: Borrowing a PDA (left), roaming the botanic garden (right) (Source: (25))

Visitors to the botanic garden could borrow a GPS-enabled mobile device (a Windows Mobile PDA) from the reception where they would also be instructed on how to use it (figures 2.17, left). On their PDA, they could choose from a list of theme-based tours that they could follow. The mobile application shows an overview map of the garden with relevant content items highlighted on top of it. Navigation through the content is done by roaming the area (figure 2.17, right), with the visitor's position constantly being tracked via GPS.

Figure 2.18: Listening to audio content (left), browsing for additional content (right) (Source: (25))

The mobile CAERUS system was deliberately designed to not require too much of the visitor's attention so that they could enjoy the beautiful botanic garden (figure 2.18, left). When the visitor enters one of the predefined content areas, the corresponding audio clip is played back to her. Additional information is available on demand through a touchscreen interface with clearly labelled (read, listen, watch, more) big buttons at the top of the screen (figure 2.18, right).

CAERUS was neither the first nor the biggest mobile tour guide that is mentioned in the literature. But its combination of a desktop-based authoring tool for curators and a mobile content delivery system for visitors make it complete platform that is directly relevant to this thesis. As such, the authoring tool will be reviewed further down in this literature review (see page 94).

2.2.16 Epidemic Menace, 2005

Epidemic Menace is a cross-media game which means that it is played using several gaming interfaces across different devices (Lindt et al., 2005, Ohlenburg et al., 2006). It was developed as part of the IPerG project and first played on the Fraunhofer campus in Sankt Augustin in Germany on August 24th and 25th of 2005. Epidemic builds upon a story of a mad scientist contaminating a university campus with a life-threatening virus. The players arrive on site as teams of "medical experts" whose mission is to locate, capture, analyze and destroy the virus using the different interfaces. After an initial briefing, players are divided into teams and assigned to the different indoor and outdoor interfaces, which include an outdoor augmented reality system (figure 2.19, left), a mobile Geiger counter-like audio interface (middle), an indoor communication headquarter (right) and some other gadgets such as a Sony Aibo robotic dog and a big touchscreen display showing the game board with all players and viruses on it. The spread of the virus across the game area is influenced by the wind through the use of a local weather station that is connected to the game server. Actors were cast for some key roles and high quality video snippets featuring the same actors were used to frame the story line from the initial briefing to the final resolution. A second version of the game additionally provided a live video stream as a spectator interface for remote audiences (Lindt et al., 2006, Fischer et al., 2007).

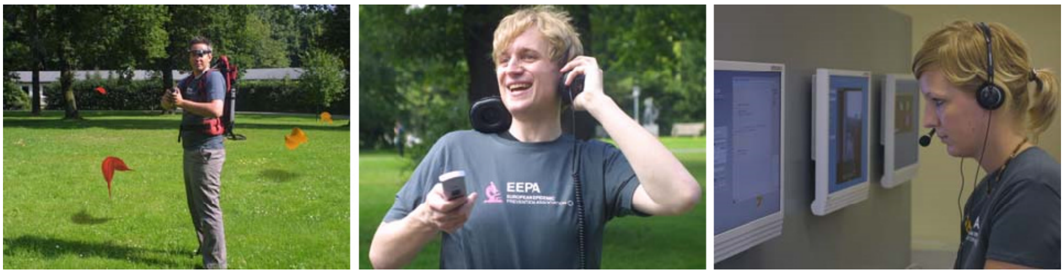

Figure 2.19: An AR player surrounded by virtual viruses (left), an audio-player captured a virus (middle), a stationary indoor player guiding the mobile players (right)
(Source: (Paelke et al., 2007))

The gaming interfaces are dependant on the respective devices and their associated roles. The stationary indoor interface gives an overview of the game-state including all mobile players and viruses. Indoor players in the game's headquarter must use this information to assist their outdoor team-mates in their tasks. Outdoor players can only stay outside for a limited period of time to avoid being contaminated by the virus. The mobile audio-interface allows sensing the different virus types and to capture them eventually when the mobile player is in their proximity. Captured viruses can be analysed when back in the headquarters in order to find an anti-virus. Once an anti-virus is found, the mobile augmented reality system can be used to exterminate the virus on the ground. Finally, players have to solve a puzzle to identify the villain and conclude the game.

Epidemic Menace is an interesting example because of its integration of a diverse set of devices. Each device on its own has been previously integrated into location-based experiences, e.g. the outdoor augmented reality system is technically very similar to the MARS system (page 28) or the Tinmith system as used in ARQuake (page 33). But the integration of *all* these devices into a coherent whole which also makes use of live actors, role playing, and is marketable via Internet television is unprecedented. As such Epidemic Menace is an early example of a location-based experience which makes use of ubiquitous computing technology in its more original sense.

2.2.17 Hitchers, 2005

Hitchers is a location-based multi-player game for mobile phones which has been developed at the University of Nottingham and first trialled in Summer 2005. Players create virtual hitchhikers by giving them a name, a destination and a question to ask if they are picked up by another player. They are then released into the wild so that they can be found, picked up and given a lift by other players (Drozd et al., 2006). The game can be played be any number of players who can create any number of hitchers, but each player can only pick up and interact with one hitcher at a time. Location in the game is determined by network cell ID.

Figure 2.20: Searching for, picking up and answering a hitcher (Source: (Drozd et al., 2006))

A player can scan his network neighbourhood and look for nearby hitchers (figure 2.20, left). Once picked up, a hitcher presents its destination and its question (figure 2.20, middle). A player would answer this question (figure 2.20, right) and carry the hitcher for some time before he drops it again. When a hitcher gets dropped, it asks a second question ("Where are we now?"), which encourages players to enter a textual description of the current location. The player's response to this question is associated with the current cell ID. The mobile game-client for Nokia S60 phones continuously logs the ID of the currently serving mobile phone cell, building up a list of adjacent cell IDs over time. When a hitcher is dropped, this information, together with the description of the last location as entered by the player, is transmitted to the central game server and combined with information from other players.

Hitchers can be quickly deployed in any mobile phone network without requiring any pre-game survey activity. The game is usually played for several days. In addition to the mobile game client it also comprises a website where players can check the history of hitchers that they have discovered and the answers that other players had given to their questions. As the game is played, it gradually builds up a common graph of cell IDs (with associated textual descriptions) and connections between them.

This complex data-set can be inspected with a custom visualisation tool that is depicted in figure 2.21. The main visualization window shows a graph where the nodes are distinct cell IDs and the edges are the encountered connections between them. Some of the nodes are annotated with location tags as given by the players when they drop a hitcher. The view of the main visualisation window can be freely modified, i.e. the user can change the zoom level and pan the canvas in order to focus on areas of interest. Using the controls to the right of the application window, the user can filter the data-set by country, network and location tags, and also modify the appearance of selected nodes through changing their colours and sizes. The two small windows at the bottom show the selected sub-graph both in graphical and textual form.

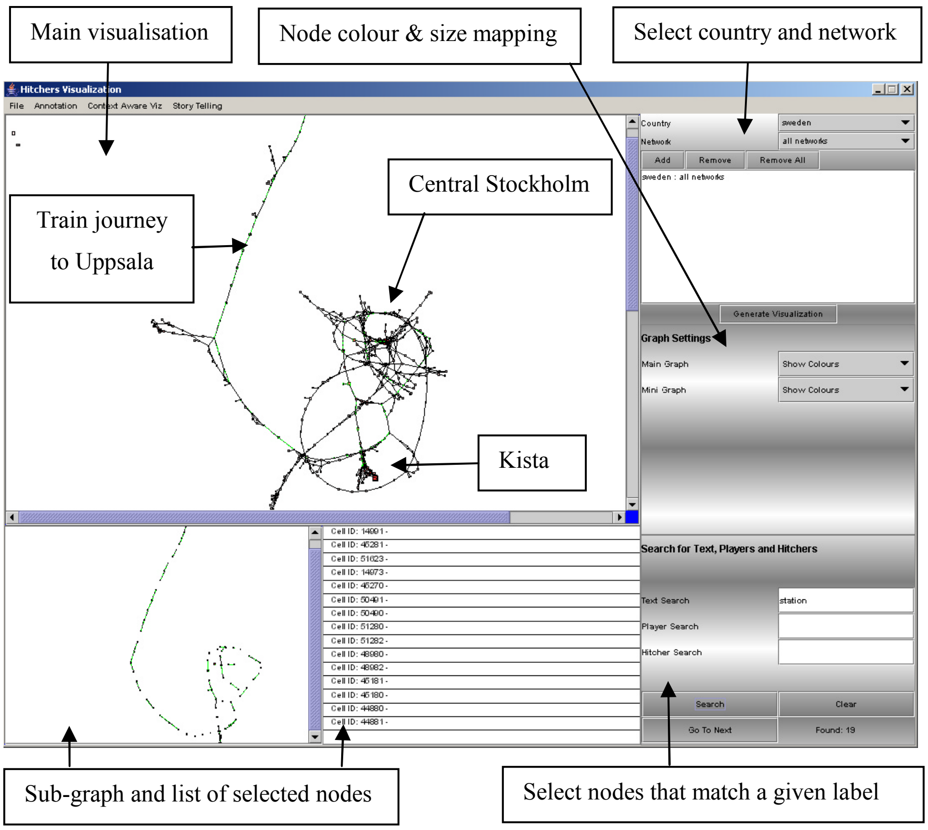

Figure 2.21: The Hitchers cell ID graph visualisation tool (Source: (Wright et al., 2007))

Figure 2.21 shows data from a Hitchers trial in Sweden which took place in September 2005. Participants of an IPerG plenary meeting played the game over the course of several days. The figure reveals some distinct clusters as well as long thin connection routes between them[3]. Inspection of the player given cell labels revealed the physical locations of some of the key clusters, such as the cities of Stockholm and Kista. Furthermore, the long trail on the left could be related to a train journey to the city of Uppsala.

Figure 2.22: Highlighting cells that players have labelled 'Kista' (left), 'SICS' (middle) and 'ICE' (right) by being shaded. Is ICE a place within SICS which in turn, is within Kista? (Source: (Drozd et al., 2006))

A closer investigation of the Kista cluster of cells is depicted in figure 2.22. It shows a zoomed-in view of the previous figure. The three different images in this figure highlight those cells that players have labeled "Kista" (left), "SICS" (Swedish Institute of Computer Science – ed., middle) and "ICE" (Interactive Computing Environments Lab – ed., right). It can be seen how these three location tags have been applied to a different range of cells, but also that the two cells that are highlighted as "ICE" in the rightmost image are a subset of the other two selections. The authors suggested that ICE is a place within SICS, which is a place in Kista.

Hitchers is probably more interesting as a general data collection framework and research vehicle than it is as a game. The authors themselves noted that the game could be much more elaborate, but that the current game design was enough to conduct the intended study of the use of cellular positioning for location-based play. The authors' later work "MobiMissions" (26) revisited the Hitchers

[3] Please note that the figure is not geo-referenced and therefore orientation in the figure does not represent actual orientation (i.e. up is not north).

framework and built a more elaborate experience on top of it (Grant et al., 2007a, Grant et al., 2007b).

2.2.18 Insectopia, 2006

Insectopia is a proximity-based single- and multi-player game for mobile phones which has been developed by the Interactive Institute in 2006. It is one of four game prototypes from the IPerG socially adaptable games showcase. Insectopia requires a Java and Bluetooth enabled handset.

Players of Insectopia use their mobile phones to catch virtual insects wherever they go. Each virtual insect in the game is generated from a Bluetooth device ID that was discoverable in the player's vicinity at the moment of interaction with the game. Insectopia was designed as a long-running, slow-paced game that fits participants' lifes, and its designers deliberately avoided the need for intense participation to support this style of interaction. Players can play the game at any time and do so either in single-player or in multi-player mode. In single-player mode, players can search for virtual insects by scanning the Bluetooth environment up to every three minutes. If such a scan resulted in multiple insects being found, the single player can still only catch a single insect per search. This limitation is lifted for the multi-player mode, where multiple players need to be co-located and can then catch all insects in their vicinity. Each insect is assigned a score, based on its rarity, and by collecting an insect a player advances his score. Scores are tracked on a global scale and it is the game's goal to advance on this global high-score list and become the best collector. Players thus need to find their balance between using the increased search power of the multi-player mode and advancing their personal score independently.

Players can trade their collected insects with each others. They also need to refresh their finds within eight days of the initial discovery to keep them in their collection. In summary, Insectopia has four modes of play: searching for new insects, refresh previously found insects, browsing the collection, and multi-player search (Peitz, 2006). Figure 2.23 shows several screenshots of the game.

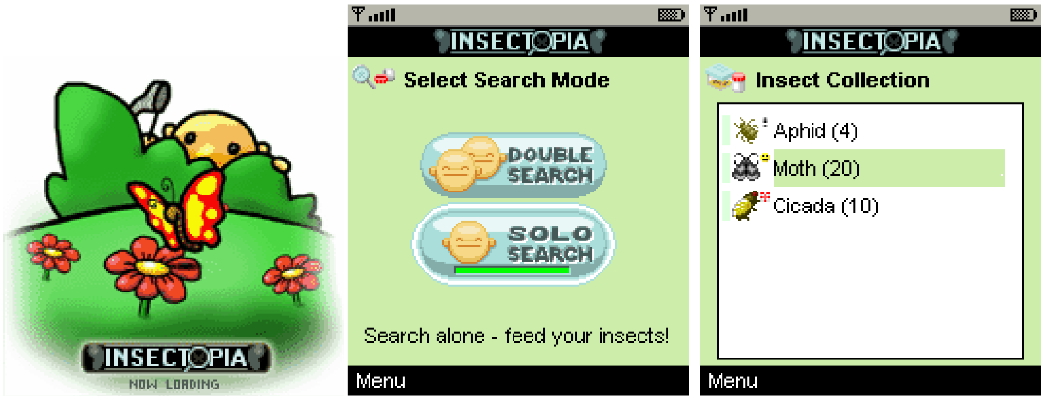

Figure 2.23: Screenshots of Insectopia (Source: (Peitz, 2006) (27))

Much like Hitchers, Insectopia is a mobile game that piggy-backs its location mechanism on an existing layer of location beacons that are present in the urban world that surrounds us, and which can be measured with a mobile phone. While Hitchers built on the ubiquitous, and largely static commercial mobile telephone networks themselves, Insectopia instead utilises the presence of discoverable Bluetooth devices. Insectopia thus makes use of the presence of Bluetooth enabled cell phones, printers, laptops, heatsets, or photo kiosks in the world around us. Most of these devices are inherently mobile and can usually not be expected to always be found at the same place. In order to be successful in the game, players of Insectopia need to build their own mental model of which of their insects belongs to which Bluetooth device in their vicinity. This is part of the fun of the game, as certain Bluetooth devices might only cross the player's radar at certain moments in time, e.g. on a specific bus at a specific time.

Locations in Insectopia are not fixed to a certain real world area; instead they move around with the Bluetooth devices that are carried by other people in the players' proximity. The term proximity-based is thus more appropriate to describe this type of location-based game.

2.2.19 Momentum, 2006

Momentum was a technology enhanced live action role-playing game (LARP) that was staged in and around Stockholm between 6 October and 5 November 2006 (Hansson et al., 2007). It sought to expand the boundaries of pervasive game-play in all three dimensions: spatial, temporal, and social.

Live action role-players, a.k.a. larpers, use the world as a game board in which they impersonate their game characters. This is usually done *"within a strictly confined area or in the middle of nowhere, in wilderness"*, but a LARP can also be staged in a city environment, so as to implicate unsuspecting bystanders (Montola et al., 2009b). Non-players might be unaware of the game rules, or even the existence of the game altogether, but can still be part of the game from the players' perspective – this is the social extension. Social extension serves the purpose of blurring the boundaries between play and reality for the players so that they can never be sure who or what is part of the game. The authors of Momentum called the process of fabricating a layer of game-content on top of reality to support social extension as "reality hacking" (Waern and Stenros, 2007).

Momentum's story line was themed around the possibility of contacting dead people through the use of bespoke technology and techno-occult rituals. Momentum was organised to mix high-intensity, co-located weekend play with low-intensity, remote play time during the week. The game employed custom game interfaces, which were built from off-the-shelf technology, such as GPS devices or RFID readers. But the technology was hidden inside game artefacts that were plausible within the theme of the game. To provide an overall framing, Momentum made use of mystical headquarters which were located at an abandoned nuclear reactor 30 meters below Stockholm (see figure 2.24, left). Players could visit the headquarters at anytime throughout the game and make use of the game artefacts that were permanently installed at that site, e.g. a radio-like device to communicate with the dead. This *Electronic Voice Phenomenon* (EVP) device was mounted to a steel bed, to which players had to strap themselves when they wanted to use the device for communication (see figure 2.24, right). They could then use four knobs to try to tune into a radio-channel with "the other side". When they found the right setting, they would be presented with a piece of pre-fabricated audio content that advanced their story-line. In addition to simply receiving pre-fabricated content, players could also speak into a microphone which was said to serve as a link to the dead. As players were free to enact their characters in any way they liked, Momentum, like any LARP, sometimes required sudden intervention to *"react to player improvisation*" (Jonsson and Waern, 2008). It was thus constantly monitored by a crew of game-masters who could intervene at anytime to ensure a smooth running game. For example, if a player's interaction with the EVP device required intervention, the game-masters could quickly record a fitting piece of audio in their secret room, feed it into the EVP device, and present it to the player while staying within the narrative of the game.

Figure 2.24: An abandoned nuclear reactor served as the headquarter for the game (left), The EVP device mounted to a bed and hosted in a room at the site (top-right), Player in action (bottom-right) (Source: (Stenros et al., 2007))

In addition to this stationary mode of interaction with a game artefact at the headquarters, Momentum also provided technology-enhanced mobile play with a variety of devices off-site. Much like the EVP device at the bed, these devices were fabricated into artefacts that integrated with the overall athmosphere of the game. Two devices that are of interest in the context of this thesis are the *Steele* and the *Thumin glove*, both of which were used to locate players, but with different levels of positioning accuracy. Both devices were used to play a sub-game of Momentum, where players searched for "magical nodes" in the wilderness. The Steele was used to find an area that warranted further investigation, or paranormal activity in in-game terms. Once players discovered such an area, they used the Thumin glove to investigate further and determine the exact location of the paranormal activity within centimetre level accuracy.

Technically, the Steele (figure 2.25, left) contained a consumer-grade Bluetooth GPS device which could determine the position of players up to a few metres accuracy. The Thumin glove (figure 2.25, right) contained a Bluetooth RFID reader which could sense RFID tags that were hidden in the environment.

Figure 2.25: The Steele device with built-in Bluetooth GPS (left), the Thumin glove with built-in Bluetooth RFID reader in use by a player (right) (Source: (Hansson et al., 2007))

To make these devices work together in a sensible way, they were communicating with a central game-server via mobile phones. The phones were carried by the players and periodically scanned for known Bluetooth devices in their vicinity, e.g. the Steele or the Thumin. When close to such devices, the phones automatically connected, read the sensor data, and forwarded it to a central game-server via GPRS. In addition to logging data and generally providing an overview of the game for the game-masters, the server also intersected the reported GPS positions with predefined game-regions. If a group of players and their Steele device were located within a previously defined area, the mobile phones of the corresponding players were triggered by the server to play an audio alert. If players heard this sound, they knew that they were nearby a "magical energy-source" which they could precisely locate with the Thumin glove, and thus switched to using the RFID device. The glove vibrated when upon discovery of a "magical node", i.e. when the RFID reader in the glove sensed a hidden tag.

Figure 2.26 provides an overview of the utilised positioning and communication technologies in the node game.

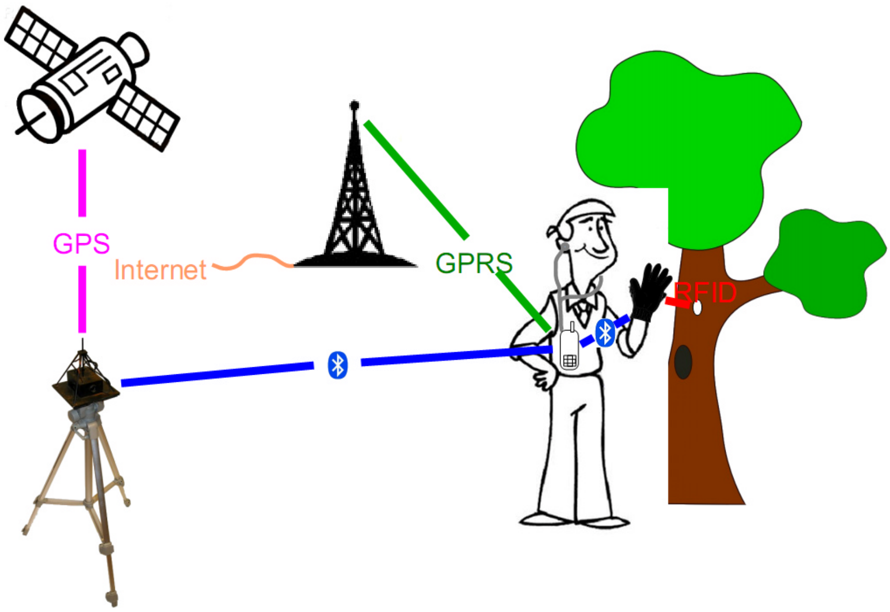

Figure 2.26: Technical overview of the node game in Momentum
(Source: (Hansson et al., 2007), slightly modified)

Momentum is generally interesting as an example of how to design and stage a high-profile pervasive game that extends the boundaries of play on all three dimensions (space, time, social): the game was long-running, could be played almost everywere, and integrated non-players into the game. The team behind Momentum made big efforts to integrate technical artefacts into the game narrative without exposing the players to too much technical details, e.g. players haven't even been told that the Steele contained a GPS device. This was all done to keep players in the flow of the game, rather than having them struggle with the technology, which can be seen as an analogy to the Scandinavian design tradition of mocking-it-up (page 6). In the context of this thesis, Momentum serves as an example of how wide-area GPS positioning can be combined with near field RFID sensing to provide in-game positioning across different levels of positioning granularity.

2.2.20 REXplorer, 2007

REXplorer was a location-based experience for tourists. It was designed to be a combination of a game and a tourist guide and was staged as part of the

"Regensburg Experience" (28) for a period of three month – July to September – during the summer of 2007. The game was intended to make learning history fun and be especially attractive to younger audiences. But it was not designed as a tour guide replacement (Ballagas et al., 2007) and therefore worked slightly different from those. REXplorer adapted a magic wand metaphor through the use of bespoke technology. Participants would pay a fee to borrow a device from the hosting venue, get introduced to how the game works and watch an introductory video. Equipped with their borrowed device and a paper leaflet – showing an overview map and an instructions summary – they would leave the tourist office and enter the old town of Regensburg to start their experience, which would last for about 30 minutes.

Figure 2.27: Locations of REXplorer (left), interaction through gestures (right)
(Sources: (29) (Walz et al., 2006))

REXplorer mystical theme allowed participants to interact with virtual game characters that were attached to 29 landmark locations throughout the city (figure 2.27, left). These characters appeared as spirits from the past, e.g. traders or famous figures such as the mathematician Johannes Kepler. Each of them would tell a short story that relates to the history of the city and ask players to help them on a small quest. If players accepted a quest, they would be sent to another location in town where they would perform a gesture with their wand-like device (figure 2.27, right). The resolution of this characters quest would be presented to the players and they would be free to look for another character and their quest, if they so wished. Players could also use the device to take geo-coded pictures, which would appear on a personalised weblog after the end of the game.

The literature about REXplorer does not cover much technical detail. GPS was originally intended to be used as a positioning technology, but in a footnote (Ballagas et al., 2008: section 2.1) the authors stated that the deployed version of the game actually used manual location selection for triggering the game content and only used GPS for the weblog data and user-initiated map-based location checks. The wand-like device consists of a Nokia S60 camera phone (model N70) and an external Bluetooth GPS unit, both combined in a protective shell. A central game-server communicates with the mobile devices via a 3G network.

REXplorer was designed to get younger audiences interested in the city's history. Although the game implemented a scoring system to allow for competition, its ultimate goal was to route participants through the city and offer them some interesting sights. The Regensburg Experience was planned as a permanent installation in the course of Regensburg's application as the European Cultural Capital 2010; unfortunately the application was not successful. As a consequence the Regensburg Experience fell short of support from authorities, especially premises, and had thus been abandoned (Weidmann, 2009) in September 2007 [4].

2.2.21 Mercury Platform, 2008

The Mercury Platform is an integrated platform comprising of content authoring, management and delivery components for location-aware information systems. The system was developed over a period of three years (2004-2007) and was planned to be subsequently installed at fifteen main museum and archaeological sites of Greece, including Acropolis, Olympia, Delphi, Knossos, Mycenae and Crete (Savidis et al., 2008). The system was designed for everyday use and needed to meet specific requirements from the Greek Ministry of Culture: support massive inquiries, e.g. incoming groups of 50-100 visitors, and the renting procedure must not take longer than twice the amount of time it would take to issue a ticket receipt. This was achieved through an optimised, semi-automated barcode-based rental system which allows handing out the device to the visitor in less than 15 seconds. At this point the application would already be up and

[4] In a 2008 paper the authors still spoke about it as a permanent installation

running in the visitor’s preferred language (English, Greek, French or German) and could be used immediately.

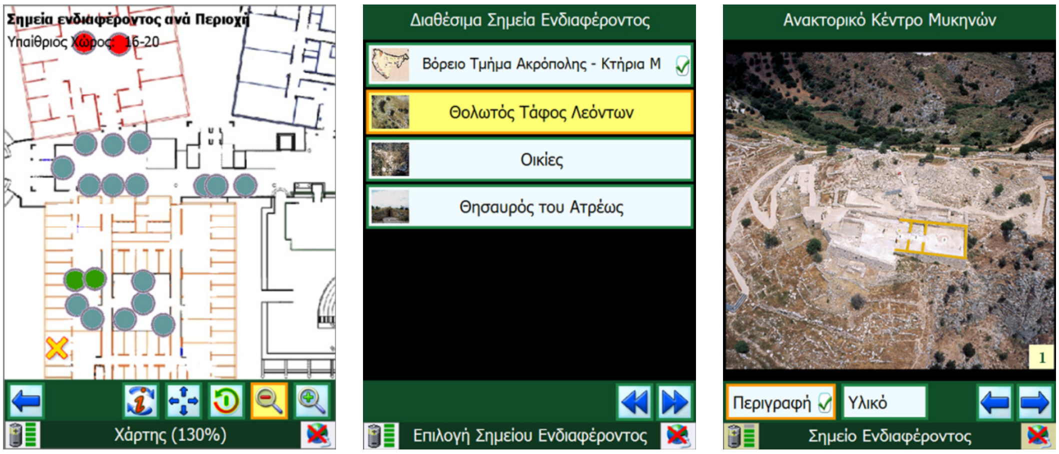

Figure 2.28: Overview map (left), content selection menu (middle), content presentation (right) (Source: (Savidis et al., 2008))

The system offers location-triggered information presentation as well as user-initiated content browsing and works indoors as well as outdoors. Visitors are free to explore the prepared multimedia content either in a sequential, tour-like fashion across the site, or to deviate from a tour at any time to satisfy their immediate interests. The mobile client supports the visitor’s overall orientation by showing his current position marked on a map (yellow X in figure 2.28). Content areas are marked as filled circles on the same map, with the colour indicating if an area was already visited (green) or would still need visiting as part of the current tour (red). Additional information, such as the ordering of areas or their labels, can be cycled through by clicking on the info icon at the bottom of the screen. The other icons can be used to freely zoom, pan and rotate the map view. When a visitor enters an area, its available content is automatically presented. If there is more than one item assigned to an area, a selection menu is presented (figure 2.28, middle), otherwise this menu is skipped and the content is directly presented (figure 2.28, right). Apart from this automatically triggered content presentation, the visitor is also free to browse through the available content areas without physically moving. This feature was implemented after interviews with experienced tour-guides revealed how museum visitors frequently use books or leaflets to revisit already seen exhibits or plan ahead and search for exhibits that they would like to see.

On the technical side, the Mercury Platform runs on Windows Mobile 5 PDAs (Fujitsu Siemens Pocket Loox N560 with integrated GPS, Wi-Fi, Bluetooth and Infrared). It provides flexibility through a generic location interface which can be instantiated by the programmer in two different ways:

1. as a point-based interface, which tracks the user's position (e.g. GPS reporting absolute coordinates)
2. an area-based interface, which tracks the user's contextual state (e.g. infrared reporting a user to be in room A near exhibit B)

Both ways were found necessary in order to support the range of current and possible future technologies that might be used for location-sensing. The location-triggering mechanism was implemented as a statically prioritised stack of location-sensing technologies. And each of these technologies was implemented through a generic wrapper interface, thus allowing new technologies to be integrated into the system when they become available, without rewriting the complete system. For the pilot system, a pragmatic choice of technologies was made and ordered according to the expected accuracy and reliability of the delivered position estimate:

1. Tags, mostly infrared (highest priority, regarded as nearly perfect)
2. GPS
3. Wi-Fi
4. Bluetooth (not in current system, but planned for special cases)

At runtime, the system would try to obtain a user location by querying the available sensing technology one after another. This means that if the infrared sensor already yielded a location, no other sensor would be queried; otherwise GPS and Wi-Fi would be queried subsequently. If none of the sensors yielded a location, the user's last known location would be retained.

The Mercury Platform had a reported project budget of 11 million Euros and is said to employ 5000 PDAs across 15 different sites. To the best of the author's knowledge this would constitute that it is to date the largest permanently deployed location-aware information system in the world. However, it seems that the project is still not completed to its ambitious plans, as personal communication with one of the authors revealed that the system is currently only installed on one site at the Natural History Museum in Crete.

2.2.22 Summary of Example Applications

Based on the reviewed examples, this section provides a brief summary of some of the key aspects of location-based experiences.

2.2.22.1 Production Categories

From a very general perspective, the presented location-based experiences can be divided into three production categories: "research", "event-based" and "commercial". Commercial location-based experiences, such as the Mogi game, operate on available consumer technology and can be directly deployed to the market. This is, yet, a somehow small group, as most of the given examples belong to the other two categories, but it is growing over time as the market leans towards commercialisation. Most of the presented examples were event-based, i.e. they were staged for a limited amount of time, in a limited area and to a limited audience. They were usually linked to an external event or some other kind of opportunity, including funding, that provided a frame for development. Irrespective of the financial and marketing possibilities provided by such a setting, it also allows incorporating less standard and more expensive hardware into the experiences that people would not normally own and that would therefore be prohibitive in a mass market context. The limited duration of an event also allows hiring support staff that manages the smooth flow of the overall experience through human intervention, be it as part of the experience (acting, game-mastering (Jonsson and Waern, 2008)) or behind the scenes (orchestrating (Crabtree et al., 2004), 'Wizard of Oz'-style simulation (Dow et al., 2005)). An event's limited time-frame also makes it feasible to study the experience. In fact most of the presented example applications derived from a research background and were built to be studied. While some of them took the form of an event, others were just prototyped and evaluated as part of a research project.

2.2.22.2 Game-Design

Some of the reviewed location-based experiences were pervasive games. It is noteworthy that most of these games utilised existing game designs and simply transplanted them into a location-based environment: Can You See Me Now? is a game of catch, Mogi a scavenger hunt and Botfighters resembles Battletech. These transformations alone led to new experiences that were exiting in their own right, and it does not seem to be strictly necessary to come up with completely new ideas in order to design compelling location-based experiences. This argument is also supported by the Geogames research, which focuses on designing location-based games from classic board games (Schlieder et al., 2006). Overall it can be stated that when building a new location-based game, it would be beneficial to have a good understanding of traditional games and their underlying game design patterns (Salen and Zimmerman, 2003, Björk and Holopainen, 2005), as this will ease the design task and leave some time for the knotty technological choices that are to be made. More on the theory and design of pervasive games can be found in a recent book (Montola et al., 2009a).

2.2.22.3 Overview of Technological Choices

From a pragmatic technological point of view, location-based experiences utilise mobile devices and usually combine positioning- and communication-technology. The following table provides an overview of the different technological choices made in the presented example projects.

Project	Mobile Devices	Positioning Technology	Wireless Communication Technology (Data)
Cyberguide	PDA (Newton OS)	Indoors: IR Outdoors: GPS	-
MARS	Backpack PC (Windows NT), Tablet PC (Windows 95)	D-GPS, orientation tracker	Wi-Fi
GUIDE	Tablet PC (Windows 95)	Wi-Fi	Wi-Fi
ARQuake	Laptop (Linux)	D-GPS, orientation tracker (for HMD connected to laptop), computer vision	Wi-Fi

Botfighters	Mobile Phone (any)	Operator-based	SMS
CYSMN	PDA (Windows Mobile)	GPS	Wi-Fi (later: GPRS), Walkie-Talkie (voice)
WarDriving	Any	GPS	Wi-Fi
Noderunner	Any	Photo, location description	Wi-Fi
URAAY	PDA (Windows Mobile)	Self-reported on map	GPRS
Mogi	Mobile phone (from au by KDDI network)	A-GPS, Cell ID	3G (CDMA EV-DO) (30)
Riot!	PDA (Windows Mobile)	GPS	-
Treasure	PDA (Windows Mobile)	GPS	Wi-Fi
MOSES	PDA (Windows Mobile)	IR	-
CAERUS	PDA (Windows Mobile)	GPS	-
Epidemic Menace	Mobile Phones (J2ME), PDA (Windows Mobile), Laptop (Windows XP)	GPS, orientation tracker (for HMD connected to laptop)	Wi-Fi, Phone calls
Hitchers	Mobile Phone (S60)	Cell ID	GPRS
Insectopia	Mobile Phones (J2ME)	Bluetooth	GPRS
Momentum	Mobile Phones (J2ME)	RFID, GPS	GPRS (with server), Bluetooth (with devices)
REXplorer	Mobile Phone (S60)	Self-reported, GPS (only logs)	-
Mercury Platform	PDA (Windows Mobile)	IR, GPS, Wi-Fi, Bluetooth	-

Table 2.1: Chosen mobile devices, positioning- and communication technologies

2.2.22.4 Mobile Devices

When looking at the reviewed examples, it becomes apparent that the chosen hardware was usually off-the-shelf. This is even the case for the earliest cited example, the Cyberguide project, which already relied on readily available hardware. As for the actual mobile devices, PDAs and mobile phones have been the most popular choices, but some early projects also used laptops or even full

PCs. Some results in the table are slightly skewed towards older technology, as the selection of reviewed examples focused on seminal work rather than on contemporary work. For example Windows Mobile PDAs are overrepresented as they were the first choice for building these location-based applications several years ago. Depending on the application scenario, this is not necessarily still the case today. It is a recent trend of the mobile phone market (since about 2007 in Europe, a bit earlier in the Asian-Pacific region) that increasingly more devices feature a GPS receiver and a multitude of communication interfaces, thus making mobile phones an ideal choice for the development of location-based experiences in the near future (Varshavsky et al., 2006) – within reason (Patel et al., 2006).

2.2.22.5 Positioning Strategies

Various positioning technology has been used with different expectations regarding the achievable positioning accuracy. On the one end of the spectrum, projects such as ARQuake or Epidemic Menace required an optimum precision as they sought to create an Augmented Reality experience, where a virtual 3D model is precisely registered with the real world and visually laid on top of it in the user's field of view. Under such premises, the smallest misalignment between the real and the virtual naturally has a big impact on the user's perception. And albeit utilising sophisticated and expensive technology, the authors of ARQuake finally had to report that misalignment confused their users (Thomas et al., 2002). On the other end of the spectrum, several other projects were less concerned about the positional accuracy. The Cyberguide research suggested that *"absolute positioning information throughout an entire space is not so important"* and that *"it is far more useful to know what someone is looking at than to know someone's exact physical position and orientation"* (Long et al., 1996a). This view is supported by the GUIDE project, which stated that designers of similar systems "*should be careful not to be over zealous when deciding to constrain the information or functionality provided by the system based on the current context"* (Cheverest et al., 2000). Finally, the more conventional use of positioning in projects such as Riot!, Treasure, or CAERUS, relies on somehow accurate absolute positions from consumer GPS receivers with errors in the range of tens of metres under good conditions. As this level of accuracy can currently only be achieved with a clear and broad view of the sky – to establish a direct line of sight between the GPS receiver and the broadcasting satellites – these kinds of projects are usually staged outside. Finally, several short-range radio or vision-based positioning approaches lean towards detecting proximity to meaningful locations or objects, rather than absolute position. These can be viable approaches for many applications where precise geographical position is less important, and they also work particularly well indoors.

2.2.22.6 Communication Strategies

Communication is not a strict requirement for location-based experiences. This can be seen in the right column of table 2.1 which shows that a large number of projects worked without any wireless communication at runtime. While many experiences, especially games, benefit from multi-user communication, others work perfectly in a single-user mode, where all the required content is pre-loaded onto the mobile devices. This seems to be especially the case for tour guide applications, as most of the reviewed examples (Cyberguide, Riot!, MOSES, CAERUS, REXplorer, Mercury) worked without any runtime communication components.

Those experiences that utilise runtime communication implemented different strategies to ensure connected operation. The first strategy is to erect a custom wireless network and/or limit the experience space to an existing network coverage area. This has been done for projects like MARS (used an existing campus network), GUIDE, CYSMN or Epidemic Menace (all custom networks). While this is a feasible strategy in many cases, it is also a costly one. And it might replicate work which has already been carried out by network operators who strive to cover populated areas with their services for commercial reasons. Thus, a second strategy would be to use an existing wireless network such as GPRS, 3G or commercial Wi-Fi. A third strategy is to design for disconnected operation, as pointed out in (Benford et al., 2003). While in wired settings, a relatively cheap and stable network connection can be assumed, the various technical and monetary constraints imposed on the project by the chosen wireless networking technology must be considered. This means that disconnection of a communication channel in a distributed system should be seen as a reoccurring natural state rather than an error. This view would then be reflected in the design of the experience, so that it would not break for the user if a communication channel fails. Example applications of this third strategy include the walkie-talkie audio-channel in CYSMN and the overall design of the Treasure game which exploits the idea of disconnection by making it a central game mechanic.

With this variety of hardware and infrastructure readily available, the technological choices to be made for developing a location-based experience nowadays resemble cherry picking within budget rather than covering basic technical needs. Consequently the attention is now more in the software-side, with an increasing focus on the authoring of content and the shaping of the overall user experience. This causes the production process to be slowly handed over from engineers to artists. While this is a natural and welcome evolution, it also introduces space for a potential gap between what the artists want and what the technology can provide.

2.3 Building Location-Based Experiences

The basic requirement for building location-based experiences is to know the available technologies and tools that can be used for this task. This section provides a high-level overview of the available communication- and positioning technologies that form the basis of many location-based experiences. It then presents the tools and techniques than can be used for spatial content authoring, i.e. the configuration of media assets in a spatial context. As this thesis is more concerned with the spatial nature of location-based experiences, rather than the communication or social side, this section puts more emphasis on positioning technology and spatial authoring.

Location-based experiences are usually built in cooperation between designers and technicians, and the material presented in this section is relevant to both groups. In addition, appendix 2 (page 326) provides an overview of the software development side of the process. While this side is often deemed uninteresting by designers, it is a fundamental part of the process and deserves some attention in order to provide a holistic view of the design.

2.3.1 Communication Technology

Location-based experiences that are not intended to be solitary experiences often include one or more communication channels, which are either voice or data.

2.3.1.1 Voice

Experience designers often use elaborate, scripted language to make face-to-face communication with participants as effective as possible, e.g. at the reception of an experience. Transmitted voice can also be used as the driving force behind experiences, as it is the case in the following two example experiences. *Single Story Building* by Blast Theory was an interactive audio experience that used call centre software to allow participants to navigate through a tree-like structure of pre-recorded audio-snippets by answering a series of either/or questions, effectively enabling them to build their own story when they called the central game number (31). *I Love Bees* by 42 Entertainment was another audio experience which took an opposite approach. Instead of having a central number that participants would call, the game presented pairs of GPS-coordinates and time-stamps on a central website that pointed to public payphones which would ring at the given time (Stewart, 2007) (32), requiring participants to move to those locations in order to follow the storyline.

Because voice communication technology is quite mature, it can also serve as a safeguard, e.g. by using a walkie-talkie audio channel as an almost fail-safe alternative to more error-prone components (as in CYSMN, page 37), or as a general hotline or call-back mechanism in case of an emergency.

2.3.1.2 Data

Data communication is a common requirement when building location-based experiences. In this context, its aim is to provide a communication channel between mobile devices and other system components. Options for mobile data communication are closely tethered to the utilised mobile devices and deciding on a particular option usually requires a trade-off between functionality, spatial coverage, adoption rate and cost.

In the domain of mobile phones, the lowest common denominator for data communication is the Short Message Service (SMS), which is available on virtually every mobile phone. It allows sending and receiving short text messages of usually 160 characters length. An extended version called the Multimedia Messaging Service (MMS) allows sending and receiving multimedia messages which can contain images, audio and video in addition to text. More advanced – and correctly configured – mobile phones provide access to the World Wide Web via the universally available Hypertext Transfer Protocol (HTTP). Such phones can display web-pages much like other devices, such as PDAs or laptops, which allows for an easy integration with a server-side component.

On a lower level, data communication between mobile and stationary components can be accomplished over a variety of networking technologies. Inherent to many second generation GSM mobile phones (2G) is the General Packet Radio Service (GPRS), which is widely available and allows moderate transfer speeds, similar to those of analogue modems of the 1990s. The more recent third generation of mobile phones (3G) provides higher speeds through the High Speed Packet Access (HSPA) family of protocols.

A recent addition to mobile phones, and an inherent component of PDAs and laptops, is the Wireless Local Area Network (WLAN, a.k.a. Wi-Fi) adaptor. WLAN provides high transfer speeds, but is currently not as widely available as 3G or GPRS technology. A notable difference between 3G/GPRS and WLAN is that the former operate in licensed frequency bands and are thus only provided by network operators, whereas the latter operate on license-free bands and are thus mostly installed for individual or institutional use. As a consequence, WLAN coverage tends to be more closely clustered around residential and urban areas than 3G/GPRS, whose operators have a contractual obligation to cover a certain percentage of a country's population with their wireless networks.

In addition to the above mentioned wide- and local area networks, more ad-hoc Personal Area Network (PAN) technology is also frequently used in location-based experiences. This includes wireless technologies such as Bluetooth, infrared (IrDA, Infrared Data Association), or even wired technology such as the Universal Serial Bus (USB). The availability of PAN technology is often limited in time and space and is thus more likely to be used for data synchronisation than for live communication during the experience.

2.3.2 Positioning Technology

Positioning technology (a.k.a. location services) is the technological enabler for location-based experiences. Many applications rely on GPS, but GPS is neither the only solution to positioning, nor always the best one. This section provides a brief overview of relevant technology that has been used to build the reviewed examples.

2.3.2.1 Global Positioning System

The Global Positioning System (GPS), or originally NAVSTAR GPS, is a satellite-based global navigation system that was developed by the United States Department of Defense (Parkinson, 1994). It consists of 24+ satellites that orbit around the Earth, constantly transmitting signals which can be received on the ground. GPS receivers use these signals to independently calculate their current position, i.e. the receivers are not known to the satellites. Location-based applications became a big research topic after the full operational capability of GPS was announced on 17.07.1995 (33) and precise civil use became possible on 02.05.2000 (Clinton, 2000) when the so called "selective availability" (Graas and Braasch, 1994), which artificially degraded the precision of the signal for most non-military users, has been turned off. Since then the most successful civil location-based applications for end-consumers are arguably GPS-based car navigation systems. They became widespread since they work in an environment where GPS works best: outside, with good view to the sky and in a fixed mode of transport within an object that doesn't significantly change speed and direction all the time. The precision of a GPS-coordinate directly depends on the quality (and price) of the utilised GPS receiver. While expensive GPS receivers can achieve up to centimetre accuracy, the more commonly used consumer-grade GPS receivers usually provide accuracies in the range of a few metres, under good conditions, up to hundreds of metres under bad conditions. GPS has been designed to work outside and cannot be expected to provide a solution to current indoor positioning needs. Although work in this direction is already undertaken (Eissfeller et al., 2005), such systems have not, yet, arrived on the mass market.

GPS receivers that are tightly integrated into mobile devices are often read out via vendor-specific APIs. Otherwise they can usually be read out via a serial connection (wired or wireless, often Bluetooth), using a protocol defined by the National Marine Electronics Association (NMEA) (34).

2.3.2.2 Cellular Positioning

Cellular positioning uses the mobile phone networks' cellular structure to generate position information that is less precise than GPS, but could be used anywhere the phone works. Mobile phone networks such as GSM are built from base station towers which provide the link between the mobile end-users and the operator's backbone network. Such towers are equipped with one or more designated antennas and their attached base transceiver stations which together provide network coverage for a certain area around the cell tower. A frequently found layout is that three antennas are attached to a cell tower in an angular fashion around its mast, so that each antenna covers an area of 120 degrees around the tower. Depending on the power of the base station, the reach of a network cell might range from a few hundred metres to several kilometres, but this distance is also influenced by the transmission power of the individual phone. Each cell in a GSM mobile phone network can be uniquely identified by its cell ID. There are two approaches to employ cellular positioning in practice: client-based and operator-based; both work indoors and outdoors.

Client-Based Cell ID Positioning

Mobile phones in a GSM network need to be connected to a cell at all times in order to function. A GSM mobile phone constantly observes its surrounding cells and automatically connects to the strongest cell when the signal of its currently serving cell drops below a certain quality threshold value. Client-based cellular positioning makes use of the phone's knowledge about the currently serving cell and uses the cell ID as a position reading. This cell ID data can then be used to infer a user's personally important locations without knowing exactly where they are in geo-spatial coordinates (Laasonen et al., 2004, Smith et al., 2005), or to geo-code them with the help of previously recorded training data and build a location service that provides absolute position readings (LaMarca et al., 2005, Varshavsky et al., 2006). Client-based cellular positioning requires a mobile phone handset that provides an API for reading out the cell ID information. This functionality is only supported by a subset of phones, currently most notably on Series 60, Windows Mobile, and Android, and is often not available from J2ME.

Operator-Based Positioning

The limited applicability of client-based cellular positioning can be overcome by using operator-based positioning where the mobile phone operator measures the proximity of a given mobile phone to the network's antennas. These proximities are then triangulated to result in an absolute position reading of better accuracy. The advantage of operator-based positioning is that it works with every mobile phone on a network. The disadvantage of operator-based positioning is that each position reading has to be paid for which makes it currently too expensive for extensive use. It also requires formal overhead to setup the service, get each individual's consent to being tracked, and provide an easy way to opt-out. Operator-based positioning is not available on every network.

Cell ID Details

This paragraph briefly elaborates on the structure of a cell ID reading as it might be obtained from a suitable mobile phone or GSM modem. A full cell ID in a GSM network consists of four integer values: the mobile country code (MCC), the mobile network code (MNC), the location area code (LAC) and the cell identity (CI). The MCC and MNC numbers are assigned by the International Telecommunication Union (ITU). The MCC explicitly defines the country. A country can have several MCCs but an MCC is always unique to its country, e.g. 262 for Germany and 234 or 235 for the United Kingdom. The MNC represents the operator of the mobile phone network. Note that this number has to be used in combination with the MCC to clearly identify an operator. This combination of MCC and MNC is called the Home Network Identity (HNI) and is required as the MNC numbers differ between countries even for very big multi-national operators such as Vodafone or T-Mobile (Vodafone UK is 234_15 and Vodafone Germany is 262_02). Very large countries like the United States have several MCCs (310-316) assigned, but a mobile's home network identity is limited to a single MCC.

The LAC and CI numbers represent the actual network structure. Every operator's network is divided into so called location areas (LA) which in turn are further subdivided into the actual network cells. These numbers are non-geospatial which means that they bear no reference to the physical world, i.e. they can not be put on a map without additional geo-coding work.

2.3.2.2 Wireless Local Area Network

Wireless Local Area Network (WLAN, a.k.a. Wi-Fi) technology is nowadays widespread in office environments, private dwellings, and in public urban areas. Bahl and Padmanabhan demonstrated an early positioning system that piggy-backed on Wi-Fi technology and algorithmically processed the raw access point measurements to improve the positioning granularity. Their system called "RADAR" (Bahl and Padmanabhan, 2000) continuously measured which Wi-Fi access points were visible from a mobile device and how strong their respective signals were received. By knowing the precise position of each (immovable) access point in their indoor setup, they could combine this environmental knowledge and apply triangulation and signal propagation modelling to get a position reading. Ekahau (35) commercialised this idea and nowadays ships a series of products related to Wi-Fi network monitoring and precise real-time indoor positioning. Going beyond the office walls, Intel Research Seattle evaluated the feasibility of building a metropolitan-scale Wi-Fi positioning system with different algorithms, including centroid, fingerprinting, and particle filters (Cheng et al., 2005). Similarly, Sony CSL studied unobtrusive long-term location logging with custom built key rings for their PlaceEngine (Rekimoto et al., 2007). They logged snapshots of the Wi-Fi environment only every few minutes and thus gained a battery life of 4-5 days. In a separate post-processing step, these log-files could then be converted into geographical coordinates with the help of a web-service (36). In 2008, the Fraunhofer Institute for Integrated Circuits (IIS) estimated that the city of Nuremberg in Germany featured about 2000 Wi-Fi access points per square kilometre (37). It was summarised that it would therefore be feasible to build a commercial city-wide Wi-Fi based positioning system on top of this infrastructure.

In the non-commercial domain, mapping of Wi-Fi networks, an activity known as wardriving (Hurley et al., 2004), has been done for many years and some of the resulting geo-coded AP data is shared through a big online-database called Wigle (14). Wigle presents its data in an aggregated fashion, forbidding access to the raw-data that was originally collected. Although this might be prohibitive for certain applications and algorithms, Wigle still holds a wealth of data (in September 2009, there were 18 million geo-coded access points in its database).

2.3.2.4 Short-Range Sensors

A different approach to automatic identification of locations is to use relatively short range mobile communication technologies. Instead of building a location-service on top of a more or less static network layout, as is the case with cellular and Wi-Fi positioning, this approach emphasises on proximity. Proximity-based systems are based on the assumption that an object or individual must be near a known location if its associated tag can be sensed. Locations in this sense don't have to be static, or rather: opportunities for meaningful interaction are not bound to a specific physical space, as they are associated to objects in that space, which might be mobile. In their spatial model of interaction, Benford and Fahlén called such opportunity for interaction an "aura" and defined it as *"a sub-space which effectively bounds the presence of an object within a given medium and which acts as an enabler of potential interaction"* (Benford and Fahlén, 1993).

There is a variety of short-range sensing technology that might be used for the purpose of proximity detection, e.g. Bluetooth, infrared, Radio-Frequency Identification (RFID), or Near Field Communication (NFC).

Using Bluetooth for client-side proximity detection in a slow-paced pervasive collection game was demonstrated by Insectopia (page 58). Other implementations of this approach can be found in the action game *Lex Ferrum* (38), a Nokia N-Gage launch title from 2003, or *Visby Under*, an adventure game prototype from around the same time (39) (Ericsson, 2003). An obvious advantage of using Bluetooth for proximity detection is its almost ubiquitous availability, as many people carry this technology around with their mobile phones. But its biggest disadvantage is the slow scanning process, which can easily reach tens of seconds in busy environments. As this is usually too slow for an enjoyable user interaction, client-side Bluetooth scanning is seldomly used for proximity detection. But similar to cellular positioning, it is also possible to reverse the direction of sensing so that mobile clients are actively sensed by the environment and notified about state changes via some other communication channel. This ambient sensing approach requires preparation of the environment with Bluetooth scanners, which might be a feasible approach for certain projects. It was for example part of the second REXplorer location mechanism (page 63), although it was not used in the final experience, as the authors could not place their Bluetooth scanners at the desired locations.

Many PDAs, laptops, and mobile phones from about the last ten years came with a built-in infrared (IR) port, which stipulated its use for proximity detection. Infrared proximity sensing is far more responsive than Bluetooth scanning, and can be very reliable, if it can be used for the intended application (compare Mercury, page 65). Infrared was also used in the Cyberguide (page 25), where remote-controls were mounted to the ceiling, and in Moses (page 50), where weather-proof IR beacons were distributed throughout a park. However, infrared started to slowly disappear from the mass-market with many devices like the Apple iPhone, Google Android phones, or various netbooks no longer supporting this standard.

Other short range radio technologies such as Radio-Frequency Identification (RFID) are currently on the uptake. RFID is already widely used in logistics (Want, 2006) and e-passports (Carluccio et al., 2006), but has also been used for proximity detection in pervasive games such as Momentum (page 59) or Pac-Lan (RASHID et al., 2006). Near Field Communication (NFC), which is technically very similar to RFID, presents an interesting alternative for proximity detection and physical interaction (Broll et al., 2009), as manufacturers started to built it into their smart phones, e.g. Nokia 6131 NFC, 6212, or 6216 (40).

2.3.2.5 Image Analysis

As an alternative to using radio-based technology, it is also possible to employ sophisticated computer vision algorithms for the purpose of location or proximity detection. This usually implies a pattern-matching approach which would search for pre-defined visual patterns in a live camera image. Pattern detection is often helped by visual markers that are easy to identify by a computer program.

ARToolkit (Kato and Billinghurst, 1999) is an example of a marker-based system that searches for rigid black and white visual markers (figure 2.29, left) in a live camera image and calculates their positions relative to the camera in high precision with six degrees of freedom (position and orientation). More sophisticated image analysis systems can also work without dedicated markers and track features of real 3D objects, or on deformable surfaces (Lepetit, 2009). The OSGART system unites the aforementioned approaches with the OpenSceneGraph API and facilitates their use for rapid prototyping (Looser et al., 2006). Another valuable resource for image analysis is the OpenCV computer vision library, which provides high-performance implementations of many algorithms for free (41) (Bradski and Kaehler, 2008).

Figure 2.29: ARToolkit marker (left), barcode (middle), semacode (right)

Image analysis does not necessarily have to perform six degrees of freedom tracking. The analysis of a sequence of images can be used to detect motion, which can in turn be used to support human-computer interaction (Paelke et al., 2004). It is also possible to simply detect the presence of a marker and use this proximity information as an event trigger, without performing computationally expensive analysis. Finally, some markers carry encoded information in their visual representation, which can be used as parameters to the triggered event.

Examples of this latter category are barcodes and semacodes (figure 2.29). Image analysis has been used in various ways in a number of location-based applications that are relevant in the wider context of this thesis. Six degree of freedom marker-based tracking with ARToolkit (and other technologies) has been used in ARQuake (page 33), semacodes have been used in the urban treasure-hunt game *Conqwest* (Lantz, 2007), and barcodes have been used in the *Barcode Battler*, which was a Japanese gaming console from the early 1990s (Pias, 2007).

2.3.2.6 Self-Reported

Self-reported positioning is a low-tech alternative to all the technology-driven solutions that have been presented so far. Instead of trying to automatically determine a location, this technique simply asks the user to actively declare it. (Benford et al., 2004b) evaluated how a self-reported positioning system, where a user declared their position by clicking on a digital map on a PDA, has been put into practice and how it performed in the context of an urban multi-user experience (Uncle Roy All Around You, page 44). Self-reported positioning has also been used in the final version of REXplorer (page 63). Despite its simplicity, this technique is more than a last resort when "all else failed", as it provides a great deal of flexibility when dealing with locations, which was found necessary by seminal research such as the GUIDE project (page 71).

2.3.2.7 Summary of Positioning Technology

Choosing a positioning technology for a project requires making a trade-off between coverage and accuracy of the technology, the resulting interaction style, the general availability in the desired target market, and the associated costs. It is also a decision that defines whether one builds a location-based or a proximity-based application. Positioning technology provides readings in absolute and/or relative coordinates. In the context of this thesis I regard absolute position readings as those which are defined in a global frame of reference, such as latitude/longitude GPS readings in the World Geodetic System 1984 coordinate system (WGS84) (see page 20). In geodesy, relative coordinates are usually defined in relation to a fixed point of reference, i.e. an absolute position, which makes them transformable into absolute coordinates by mathematical means. But with some of the technology that was used to build the presented example applications, such reference points are often mobile (RFID, Bluetooth, etc.) and lack an associated absolute position so that they can usually not be transformed back into absolute coordinates. Thus, using such reference points or tags often merely allows sensing proximity to those tags (and potentially distance and orientation), which results in a proximity-based interaction style. In order to still quantify those relative points of reference in absolute terms (i.e. on a map) and make them usable on a larger scale, it is possible to avoid mathematical transformations and simply log the occurrences of the desired data in conjunction with associated absolute cordinates that were obtained from an appropriate source, e.g. a GPS receiver. This procedure is known as geo-coding.

This whole section provided an overview of the most commonly used positioning technologies from the previsouly presented example applications. Albeit some of the example applications used additional technology to enhance their position context with orientation data – e.g. a digital compass in ARQuake (page 33), or a six degree of freedom orientation tracker in Epidemic Menace (page 53) – these are not covered in this discussion in order to keep the focus on positioning technology only. The following table 2.2 provides an overview of key characteristics of the covered technologies. The subjective accuracy ratings in the table can roughly be translated as: centimetres (very high), metres (high), hundreds of metres (medium), and kilometres (low).

Technology	Absolute	Relative (Proximity)	Accuracy	Coverage
GPS	X	-	High (consumer grade) to Very High (professional grade)	Global, currently only outdoors w. clear view to sky
Cellular	(X) [5]	X [6]	Low to Medium	Inhabited areas, indoors and outdoors
Wi-Fi	(X) [7]	X	Medium to High	Inhabited areas, indoors and outdoors

[5] After geo-coding by operator, self or others

[6] Pure cell ID without geo-coding

[7] After geo-coding by self or others

Short Range	-	X	Medium (Bluetooth) to Very High (NFC)	Prepared areas, indoors and outdoors
Image Analysis	(X) [8]	X	Very High	Prepared areas, indoors and outdoors
Self Reported	X	X	Low to Medium	Depending on reporting mechanism

Table 2.2: Key characteristics of selected positioning technologies and techniques

It can be seen in the table that only GPS provides absolute coordinates without further ado, whereas the other technologies mostly sense proximity to a location. The major drawback of GPS is that it can only be used outside with a clear view to the sky, whereas most other reviewed technologies can also be applied indoors. Self-reported positioning is a special case as it not a technology but a technique whose reporting mechanism can be tailored to the application's needs to provide absolute as well as relative coordinates. Cellular and Wi-Fi can easily be geo-coded and thereby transferred from relative to absolute coordinates. In theory, this is also possible with other technologies, such as short range sensors or image analysis, but it makes less sense in those cases as the sensed artefacts are inherently mobile and can thus change their position at any time. Certainly this also applies to cellular and Wi-Fi networks to some extent, but these infrastructures are generally far more stable in their spatial layout and thus make sense to geo-code. They are also usually "given" in a sense that they are already an invisible part of the urban landscape that can be exploited for positioning, whereas short range or image analysis tags need to be deliberately placed in the environment, which could be a hindrance, as it does not scale well. Furthermore, it has to be considered that image analysis is potentially ambiguous, as some tags, especially bar codes on products, will be deployed more than once in practice.

A typical approach to mitigate the weaknesses of one particular technology is to combine it with another technology with different characteristics and build a hybrid positioning system. This is also called sensor fusion. The Place Lab project from Intel Research Seattle (42) is one example of such a hybrid system. It provides the building blocks for a location-service that is built on top of Wi-Fi, GSM cell ID, GPS, and Bluetooth (LaMarca et al., 2005). Hybrid systems can

[8] Absolute coordinates usually in relation to tag, not in global frame of reference

help to provide GPS-like services that work under conditions where GPS can not. However, none of these technologies, techniques, or hybrids is going to replace GPS – and it would be foolish to neglect GPS in cases where it can be applied.

2.3.3 Spatial Authoring

With a wide range of building blocks for location-based experiences readily available (hardware, software, positioning and communication technology), the attention increasingly turns to the question of how best to author and configure them. Media assets such as images, sounds, text or videos need to be attached to locations in the physical world to be triggered by participants who enter, leave or dwell in them.

A simple solution to this problem would be to hard-code the required values in the source-code or a separate configuration-file. While this might be a feasible approach for early tests of a system, it certainly does not make much effort to be usable by anyone else than the programmer or for anything else than simple tests. A slightly more advanced approach is to make use of the freely available web-based mapping tools, such as Google Maps, to define locations in relation to maps of the area. While this still leaves out dedicated support for the creation of location-based experiences, it is easily available, free to use, and therefore rapidly gaining popularity. Projects such as WikiMapia or the Brussels Guide use this feature to geo-reference their content to locations (figure 2.30).

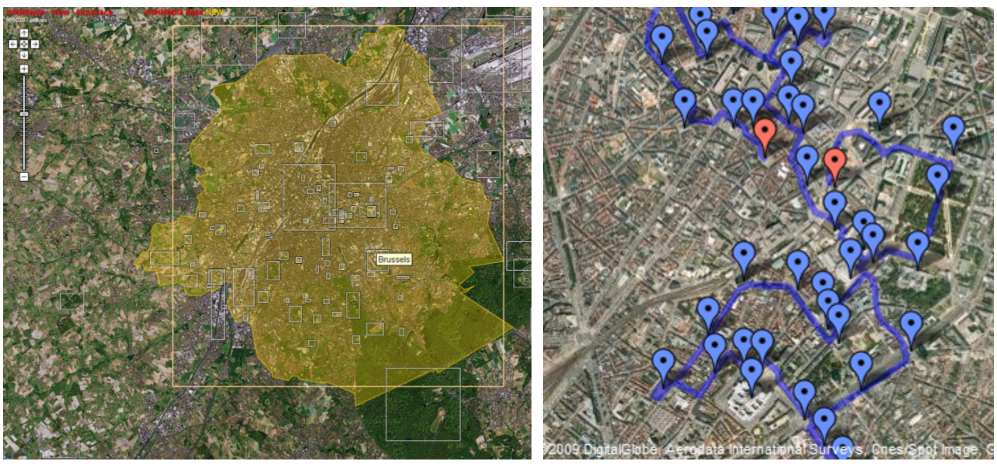

Figure 2.30: Two examples of using Google Maps for defining locations (Sources: (43), (44))

To satisfy the more sophisticated needs of content experts, dedicated tools and techniques for authoring location-based experiences have begun to emerge. All of the examples produced to date share a common overarching approach:

1. Spatial authoring is done using a map of the physical area in which the experience is to take place and then somehow placing or drawing a series of regions, locales or hotspots on top of this that are triggered according to a set of basic events such as participants entering and leaving them.

2. Actual media assets are left to be produced with whatever tools the artists prefer and integrated into the experience based on a variety of supported standard file-formats such as MP3 for audio, JPG for images, etc.

This section provides an overview over a range of different authoring tools and techniques for building location-based experiences and concludes with a summary of their key concepts.

2.3.3.1 MARS Authoring Tool

The MARS Authoring Tool emerged from a late iteration of the MARS research, which was already presented on page 28. The succession of publications about MARS reveals the evolution of this work and can be seen as archetypical for similar projects:

- Systems paper from 1996, describing the underlying software architecture that will make things easier (MacIntyre and Feiner, 1996)
- Application paper from 1997, describing an initial prototype application, built on the new architecture, light on the content side, "only just works" (Feiner et al., 1997)
- Application paper from 1999, with content authored by domain experts, still built with engineers who programmed according to the authors input (Höllerer et al., 1999)
- Application paper from 2003, describing an authoring tool that liberates the authors from the engineers (Güven and Feiner, 2003)

The MARS authors described their location-based experiences as "Situated Documentaries". Due to the journalistic focus of their application vision, they strived to empower future journalists from a university class to work with their system. The resulting authoring tool focused on authors who are not programmers and allowed them to spatially arrange pre-authored media assets in a 3D model of the actual area (figure 2.31). Assets can be linked with each others to produce a hypermedia structure that can be browsed by the user.

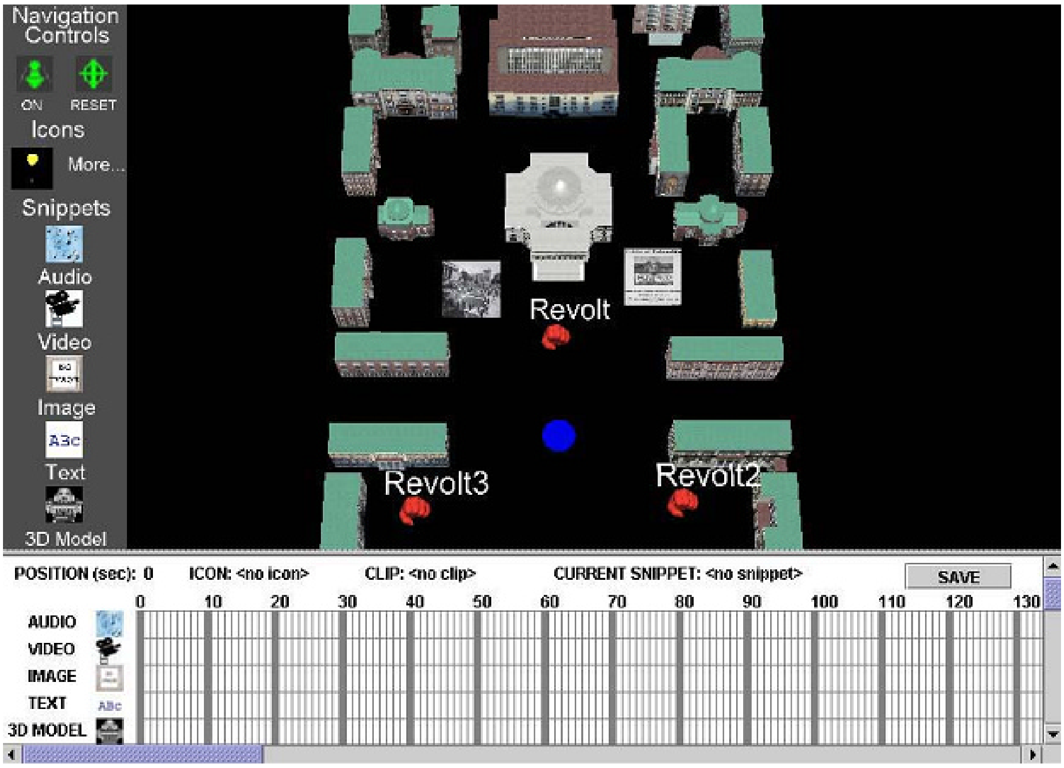

Figure 2.31: MARS Authoring Tool with 3D model of Columbia Campus, three defined locations (red), current user position (blue) and two image snippets (squares)
(Source: (Güven and Feiner, 2003))

The MARS Authoring Tool consists of an authoring component and a presentation component, which can be used to preview the experience in a desktop VR setting without having to go outside with the full kit. An author starts by choosing an icon to mark a location of interest and placing it at the desired position in the 3D model. Locations are named (as done in figure 2.31 with Revolt, Revolt2, Revolt3). The author then decides which media assets should be associated with locations and imports them from the file-system (supported assets include audio, image, text, video and 3D).

Assets[9] can be arranged on a timeline (see bottom of figure 2.31) that resembles those found in desktop animation of video editing programs. Timelines in this system are per location so that an image asset arranged at time 0 and set to endure for 5 seconds would be shown for 5 seconds starting at when a user enters that location. Finally, the author can decide if an asset is to be screen-stabilised or world-stabilised. This distinction is made as MARS utilised sophisticated AR technology that supports relatively precise measurements of a user's current position and orientation. A screen-stabilised image asset would appear in the user's head-mounted display as he enters a location and would stay at the same position on the screen for the defined period of time, no matter how the user moves his head, given that he stays within that location. A world-stabilised image, on the contrary, would seem to be attached to its location and would move across the screen according to the user's position and orientation.

No information is given as to how the 3D model of the Columbia Campus has made it into the editor. It has to be assumed that the model has been included and geo-referenced prior to providing the software to the authors.

2.3.3.2 Mediascapes

Mscape Suite, formerly known as the Mobile Bristol Application Development Framework (Hull et al., 2004) is a commercial authoring tool for location-based experiences that the authors call mediascapes. It has been used to author several experiences, including Riot! 1831 (see page 48) and Scape the Hood (45) in which users explored an area of San Francisco, triggering local stories as they entered different city blocks. Mediascapes are played back on GPS-enabled Windows Mobile devices; the required player software is available free of charge. The authoring software is also available free of charge and runs on Windows XP and Vista. Figure 2.32 shows the overall production workflow for creating a mediascape in the desktop authoring tool and deploying it to a mobile device. Mediascapes are internally represented in an XML-based language called Mobile Bristol Markup Language (MBML), a snippet of which is shown at the centre of

---

[9] Imported media assets are called "snippets" in the MARS system

the figure[10]. It was one of the main principles of the project to put the authoring tool into the hands of domain experts that are not necessarily programmers. Consequently, authors are not required to program MBML by hand and would instead use the visual authoring tool which provides the most common features and saves the authoring state in the required MBML file-format. In addition to the visual interface, the authoring tool also provides a programmer's editor where advanced users can script additional code-snippets that will subsequently be available for use in the visual editor.

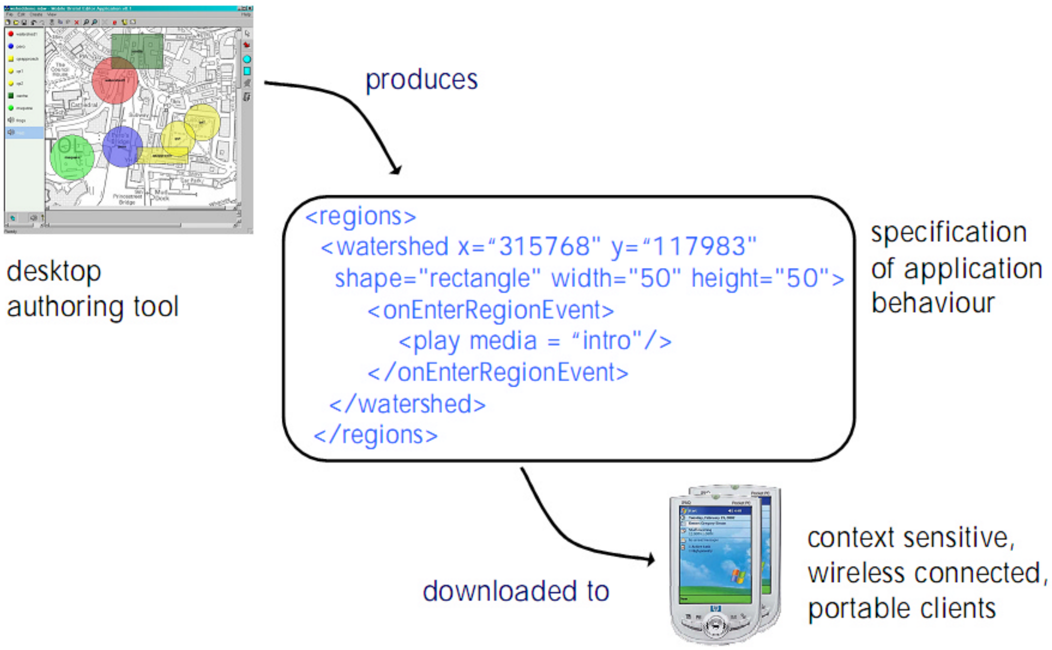

Figure 2.32: Mscape production workflow (source: (Hull et al., 2004))

The desktop authoring tool as seen in figure 2.33 provides a workflow which essentially consists of 4 main tasks:

1. Define *what* digital content is to be used, i.e. the media assets
2. Define *where* the content is to be encountered, i.e. spatial arrangement
3. Define *how* the content is triggered, i.e. trough movement, time, button presses on a graphical user interface
4. Define *how* the content is presented to the user, i.e. as audio, HTML, Flash

[10] The snippet defines a rectangular region called "watershed" of a certain size and associates this region with a media file called "intro" which will be played when a user enters the region.

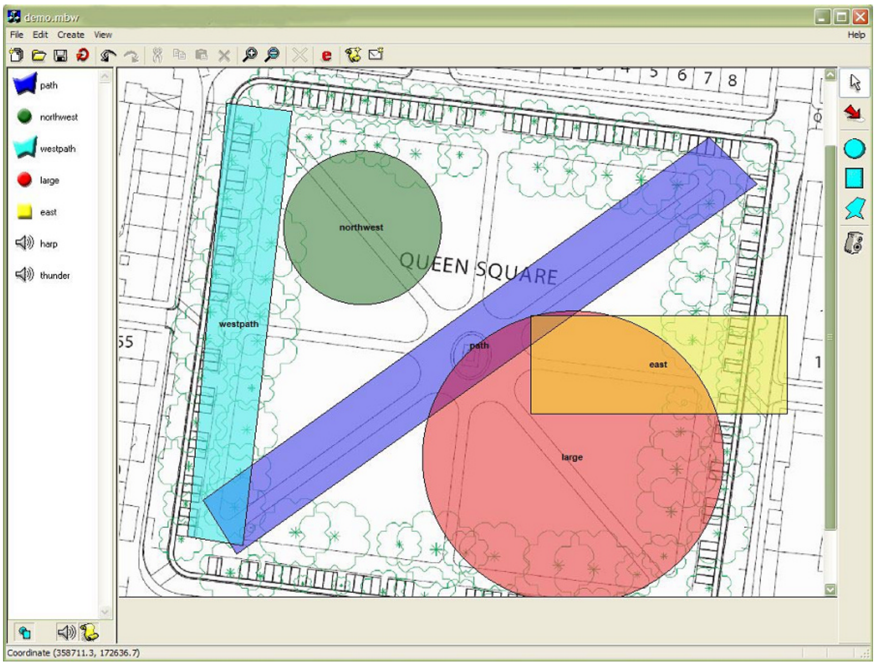

Figure 2.33: Spatial layout of triggering regions (Source: (Hull et al., 2004))

Task 1 is supported through a media manager that resembles those found in other media production programs, e.g. video editing suites. Task 2 is supported through a graphical layout editor which provides a map background that represents the target area on which the designer can place spatial triggering regions such as circles, rectangles or polygons. Task 3, the definition of content triggering rules for regions, is supported through an action editor which can be brought up by double-clicking on a region (figure 2.34). Using that editor, handlers for specific events can be defined in a point and click fashion by selecting a function from the tree control to the left of the window and specifying the desired function-parameters such as the media name, duration, etc.

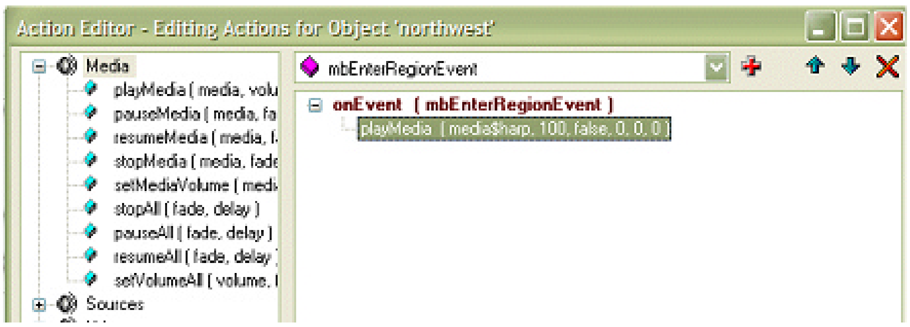

Figure 2.34: Action editor (Source: (Hull et al., 2004))

The example in figure 2.34 defines that a media called "harp" should be played when the user enters a region called "northwest" (as of the window title bar). Upon closer inspection of the regions defined in figure 2.33, it becomes apparent that the green circular region on Queen Square is labelled "northwest"; thus the playMedia-action would be triggered when a user enters that part of the square. Apart from this kind of location-based triggers, events can also be triggered based on time and custom programming logic. To accomplish task 4 (*how* to present the information), the author can decide whether to communicate with the user audibly or visibly, or both. Visible communication and user-interaction is supported through established web technologies, namely HTML pages and Flash movies.

Mscape suite provides a test-environment which can be used to check the authoring state without having to leave the desk. In this test-environment, the user's GPS position is emulated by clicking on a map. Once satisfied with the work, the author can package and deploy the mediascape to a hosting website by using an integrated publishing tool. Basically, the overall concept of the Mscape suite was inspired by the success of HTML and its authoring workflow with tools such as Dreamweaver (46), which provide high-level graphical authoring as well as low-level source-code access. The authors stressed the point that the democratization of web-publishing enabled everyone to build and deploy a website. Results would differ, as would the expertise of the users, but the important point for the authors was that the low entry-barriers enabled everyone to participate in the search for the possibilities offered by the new technology and *"the emergence of new application genres with real value to their users"* (Hull et al., 2004: 126). Mediascape sought to achieve the same for the authoring of location-based experiences and appropriated many of the concepts found in web-publishing to the creation of location-based experiences. Following this mantra, the Mscape suite has been rolled out to the public. This was supported by community building measures such as a website (47) with forum, blog and a gallery of user-generated mediascapes, as well as user gatherings. More recently, the application of Mediascapes to education and learning has also been studied (Loveless et al., 2008) (48).

2.3.3.3 Colourmaps

Colourmaps are a concept rather than a tool. With colourmaps, designers can use their favourite drawing package to paint the desired trigger regions on maps, using a different colour for each new region (Flintham, 2005). The resulting image would then be saved to a lossless bitmap format and used as a lookup table within the game-engine. An example of a colourmap that has been used to trigger location-based clues in *Uncle Roy All Around You* (page 44) is shown in figure 2.35, left. Each of the coloured areas defined in this image was a different piece of content that would be presented to the player if his current position would fall within that region. A second example use of colourmaps is given in figure 2.35, right, where possible start positions for online players of *Can You See Me Now?* are defined in blue colour. As it can be seen, colourmaps do not require a continuous definition of regions. Most graphic-design packages, including the popular Photoshop and Freehand, provide a layering functionality, where the final image is composed of several layers of graphics that can be individually modified and disabled when not needed. This is useful as it allows a map of the area to be loaded into the background to provide guidance while painting. That background layer can then be turned off when exporting to obtain the desired pure colourmap image, free of any other information. This can be seen in figure 2.35, where the left image shows only the colourmap as it would be used in a game-engine and the right image shows a colourmap with a background map and annotations that have been added for illustration purposes.

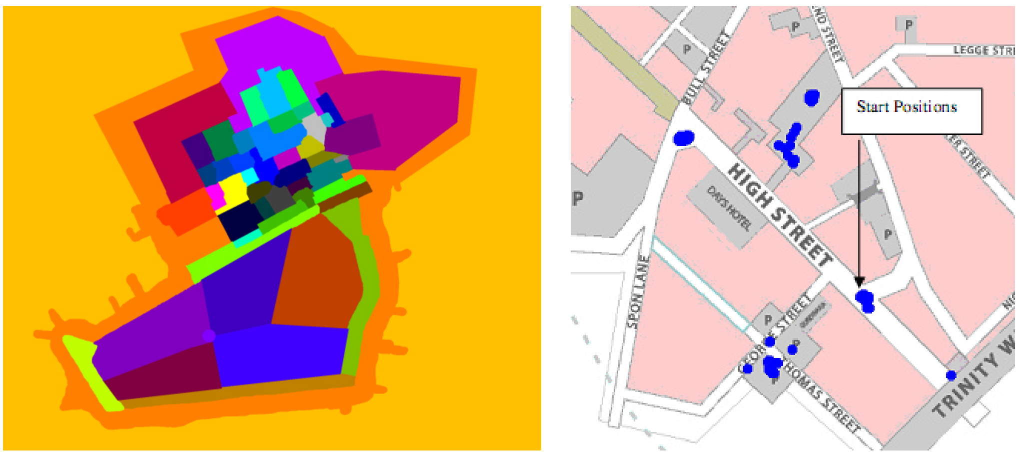

Figure 2.35: Colourmaps from Uncle Roy All Around You (left) and Can You See Me Now? (right) (Source: (Flintham, 2005))

Colourmaps use a raster-based representation of spatial content, in contrary to the vector-based approach taken by tools such as Mediascapes. To enable their use in a location-based experience colourmaps have to be geo-referenced. That is, the corner coordinates of the resulting image-file have to be known in geo-coordinates as well. At runtime, the geo-referenced colourmap would reside in memory on the mobile device. Each position update would be translated into image space and the corresponding pixel's colour value would be read out. This value would then be used to look up the associated content. This mechanism implies that there is no space for tolerance as even the slightest difference in the colour value would cause a mismatch between colour and content. This has been reported by the authors whose artists unintentionally tried to use the colourmaps-system with compressed images (which have compression artefacts) and soft-brushes (which mix colours to produce a softer image).

The concept of colourmaps allows designers to spatially arrange location-based content in a straightforward manner without having to leave their familiar tools behind. It has been successfully applied to the creation of artist-led location-based experiences such as the aforementioned *Can You See Me Now?* and *Uncle Roy All Around You* as well as *Savannah* (Benford et al., 2005) and *I Like Frank* (Flintham et al., 2005), all of which have been produced with involvement of the Mixed Reality Lab at Nottingham University. One limitation of this approach is the rather high storage requirement for the colourmap-files, as raster data takes up much more space than vector data. Another limitation is the external definition of content. The presented systems used XML configuration files that the artists had to manually edit, which was reportedly difficult. This limitation could be mitigated by integrating colourmaps into a complete solution that combines spatial layout with content association.

2.3.3.4 CAERUS Authoring Tool

CAERUS, the Context Aware Educational Resource System, has already been generally introduced on page 52. This section is more specifically about the authoring tool which is the central part of the system. The CAERUS desktop authoring tool is a Windows application which provides a graphical user-interface for creating location-based experiences and deploying them to mobile devices that run Windows Mobile. Synchronisation between desktop and mobile devices is done via Active Sync, Microsoft's standard protocol for this task.

The CAERUS authoring workflow is divided into three main categories of tasks: map tasks, region tasks and tour tasks. The task panel to the left of the application window allows switching between these categories. Figure 2.36 shows the application window with a map of the University of Birmingham's botanic garden loaded into the background and the task panel set to Map Tasks.

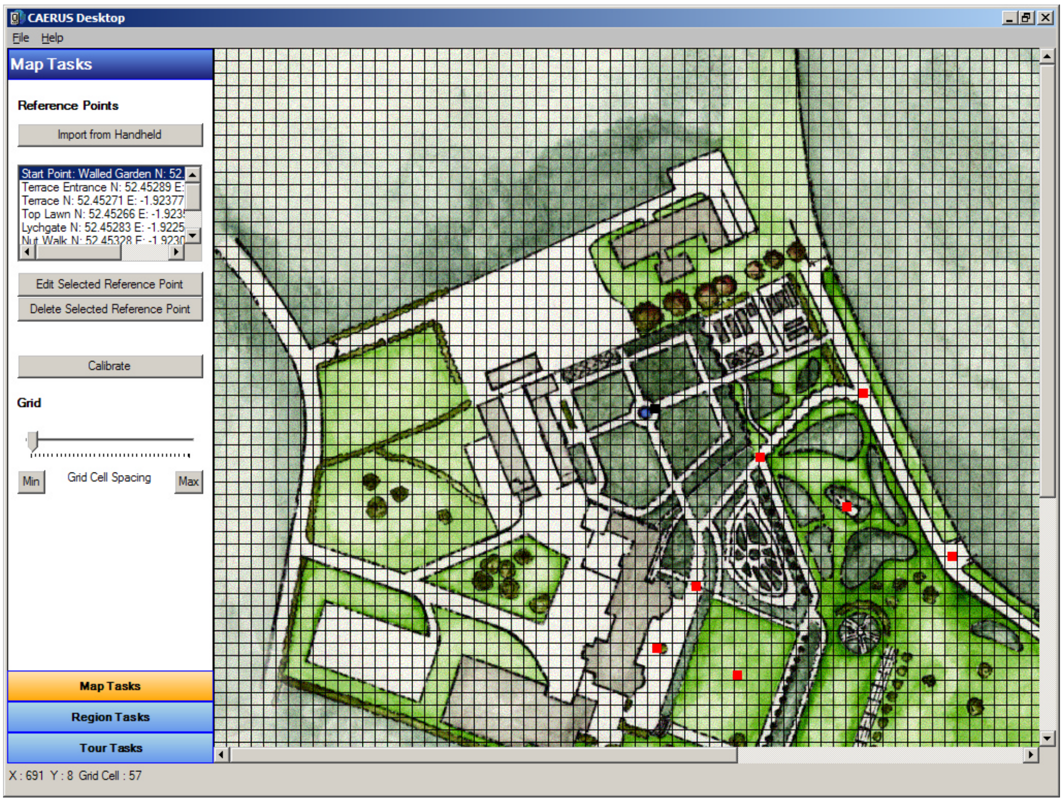

Figure 2.36: CAERUS Desktop Authoring Tool, showing "Map Tasks"

Map Tasks

The CAERUS authoring tool provides a map-based view of the area of interest. The map is a mandatory component for the system to work and has to be supplied by the author in a standard bitmap format when creating a new project. The map needs to be calibrated before actual authoring can begin. To do so, the author supplies geo-coordinates of at least three reference points in the area and relates them to their corresponding positions on the map (the red dots in figure 2.36 mark the reference points given to calibrate the map in this example). The system then calibrates itself and overlays the map with a grid image, whose cells can be used for content placement. The spacing of the grid can be adjusted by the author to reflect the author's anticipated accuracy of the GPS positioning and the required granularity of locations in the area. The task of registering a map with the physical world is aided by the mobile client, which allows defining locations on site that can afterwards be imported back into the authoring tool. The calibration step must be completed thoroughly before starting to author, as the calibration information is the foundation for all associated definitions and any change to it afterwards will cause the authoring tool to loose all defined content regions!

Region Tasks

Content triggering regions are defined via direct manipulation of the grid overlay. The author first creates a symbolic region by giving it a name and potentially already relating it to premade media assets and texts (figure 2.37, left) and then draws its spatial spread directly on the map (figure 2.37, right).

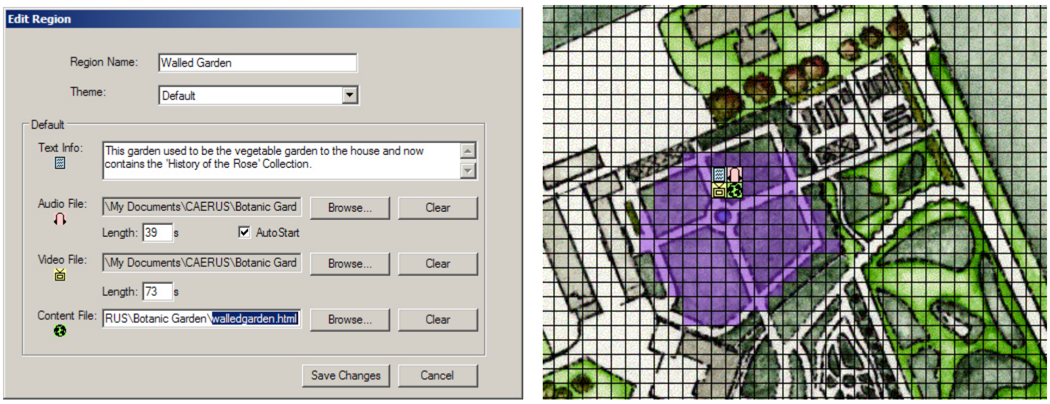

Figure 2.37: Definition of a symbolic region with associated content for the "Walled Garden" (left) and defining its spatial spread on the map (right)

Tour Tasks

Regions of interest can be grouped into tours that allow visitors to experience the prepared content and environment in a sequential way. To do so, the author either uses the already defined regions or sets up simple waypoints (without content) that a user must pass through in order to advance the tour. However, the system evaluation revealed the visitors' frustration with the tour mode and the authors concluded that it needed to be reworked. Some visitors had general navigational problems when trying to follow the suggested tour in a linear fashion, others were frustrated when they reached a waypoint and received no content from the system.

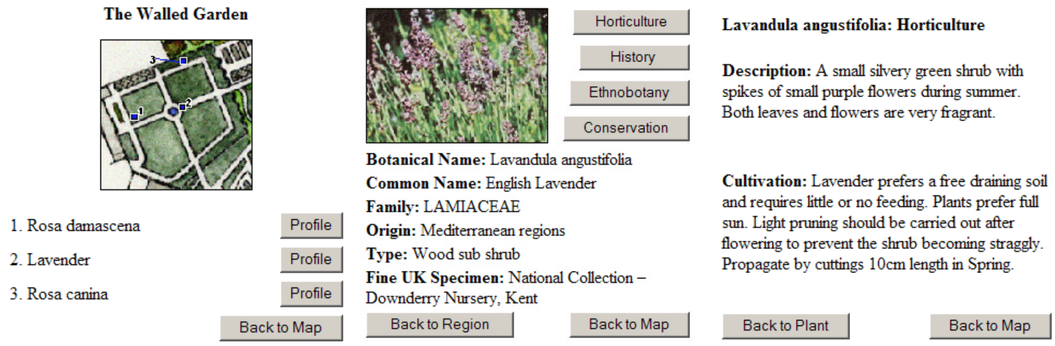

Figure 2.38: Three screens of additional HTML pages provided for the “Walled garden”

CAERUS’ content triggering mechanism provides hooks to directly associate audio, video and text to spatially defined regions. Audio files are automatically played when a visitor enters a region and text and videos can be started by pressing big buttons on the touch screen interface. Additional content can be associated with a region and explored by the user upon demand. This additional content can be encoded in any file-format that is understood by the underlying Windows Mobile operating system and registered with an application that is installed on the PDA. In the example of the botanic garden tour this feature was used with standard HTML files that would open up in the device’s web browser. A visitor could freely browse through the provided information of plants in the current region and get back to the map interface once satisfied (figure 2.38).

Summary

The CAERUS desktop authoring tool provides a complete solution that spans from map calibration, over content creation to deployment. Authors can use the system to build location-based experiences without any programming knowledge, as all required work can be done using a standard graphical user interface. A limitation of CAERUS is that it does not provide a test-environment, such as Mediascapes, hence experiences can only be tested outside on location. The map calibration task is nicely supported in the desktop software as well as on the mobile devices. However, there is a severe problem in the calibration process, as recalibration of the map at a later stage of authoring would erase already defined regions. This can only be seen as a design flaw and should by no means be adapted for future systems!

2.3.3.5 Yamamoto

Yamamoto is a Windows application which provides sophisticated tools for map-based polygonal modelling of instrumented environments[11] (Haupert, 2005). The application was developed at the University of Saarbrücken as part of a larger research project. One of the project goals was to go beyond car navigation and research into pedestrian navigation, especially on how to best guide a user to his final indoor destination. Yamamoto provides a hierarchical model that supports nesting of high detail models inside low detail models, e.g. a detailed building model with internal features can be nested (linked) in a city model that only provides the outside appearance of buildings and the transport network. Authoring begins with loading a reference image, e.g. an aerial map or a floor plan, into the background of the authoring tool and then geo-referencing that image.

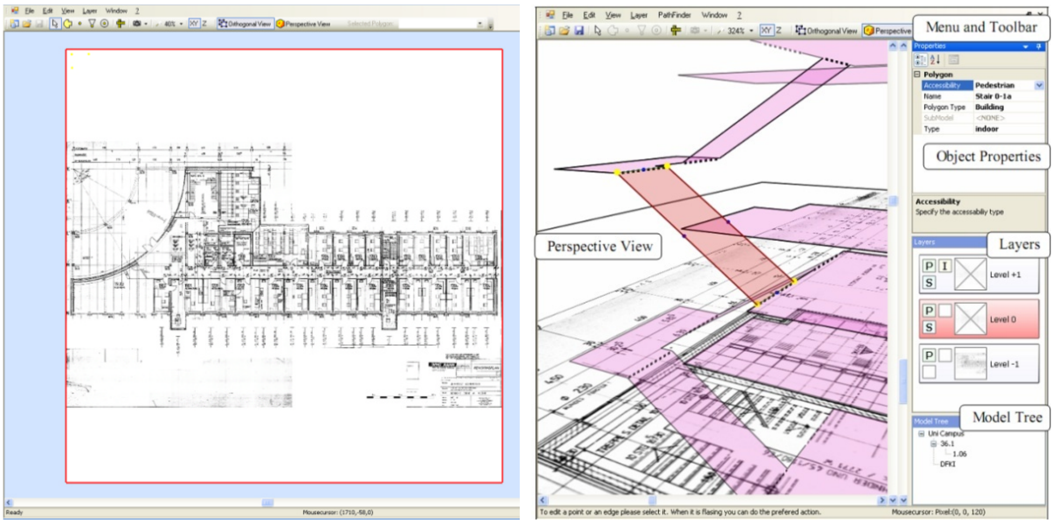

Figure 2.39: Yamamoto with background floor plan in orthogonal view (left) and with multi-floor model in perspective view (right) (Sources (Haupert, 2005, Stahl and Haupert, 2006))

Figure 2.39 (left) shows the editor with a floor plan of a building loaded into the background. Yamamoto allows modelling of polygonal surfaces in three dimensions. Edges between adjacent polygons can be marked as passable (e.g.

[11] Project term for ambient intelligence / intelligent environments: a mobile user is supported by technology that is embedded in the environment (often indoors)

corridors, doors) or non-passable (e.g. walls, fences) to support the route-finding algorithm that was developed as part of the project. Different floors in a building can be connected for the same reason.Due to its support for multi-floor buildings, Yamamoto provides two projection modes: orthogonal and perspective. Orthogonal projection is useful when working on a single floor while perspective project is useful when working with several floors. Figure 2.39 (right) shows a perspective view of a multi-floor model that is connected via stairs. Note that the stair polygon is currently selected and its object properties are revealed on the right side of the application window, which also provides means to organise the current view in layers and within a larger model tree. Several building models can be organised in a larger area or city model.

Figure 2.40: Campus overview with several buildings (Source: (Haupert, 2007))

Figure 2.40 shows a campus model of the University of Saarbrücken with several low-detail building models overlaid on top of an aerial image of the area. The building towards the right of the figure is activated and reveals a high-detail multi-floor model. Yamamoto utilises web-standards to mitigate potential storage problems which would arise when the system would be used on a larger, ideally planetary scale. Its hierarchical structure allows splitting larger area models into separate XML files that could be loaded on demand.

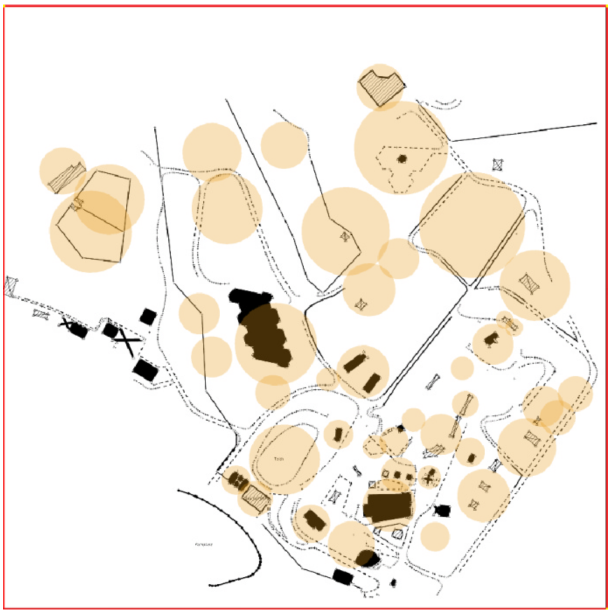

Figure 2.41: Locations in the Zoo of Saarbrücken (Source: (Haupert, 2005))

While Yamamoto is predominantly used to allow detailed modelling of intelligent environments for pedestrian navigation, it has also been used to help building a location-based experience that was staged in the Zoo of Saarbrücken. Figure 2.41 shows the locations that have been marked within that zoo, each of which was associated with an audio file that was played when the user entered that location. Yamamoto does not provide a runtime for a mobile client, which thus had to be programmed, but supports the location modelling task with sophisticated tools. The resulting model is stored in XML and thus easy to use externally.

2.3.3.6 Gamecreator

Gamecreator is a web-based authoring system that was developed in the IPerG project. Gamecreator is oriented towards the amateur developer who has some proficiency with Web 2.0 services but is not necessarily a programmer (Heliö et al., 2008). The idea behind Gamecreator is to separate input and output technologies from the internal game-logic and provide a general authoring solution just for the logic (Waern, 2008a). This approach was chosen to allow supporting different target architectures and also facilitate quick adaption of new technologies as needed. As a consequence, Gamecreator does not provide its own mobile runtime component. But as part of the Boxed Pervasive Games work package of IPerG it has been trialled with different mobile hard- and software setups to stage several games that focussed on location-based play (through GPS and RFID, in a game called Crash (Waern et al., 2008)), augmented reality play (through computer vision, in a game called Alchemist (Wetzel et al., 2008)) or both (in a game called Interference (Bichard and Waern, 2008)).

Gamecreator basically allows linking locations, items and people into a coherent data-structure. It clearly separates real entities from game entities and requires those that belong together to be explicitly linked, e.g. real world locations need to be explicitly linked to in-game locations. Gamecreator's core concepts are assets, game objects and game events (Waern, 2008b):

- *Assets* comprise all digital media as well as real world places and their corresponding triggers
- *Game Objects* comprise all logical objects, including items, places and people (used to define roles in the game), all objects have properties
- *Game Events* translate real world triggers into in-game events, e.g. a player enters a real location which creates a game event for entering an in-game location

Gamecreator provides a tagging mechanism to allow custom bundling of objects, events and assets. Game events are handled through scripts, or rules, that can be constructed from predefined snippets of actions. Gamecreator provides online interfaces to create and manage the complex relationships between those assets, objects and events. It features a programmer's API that provides access to the data-structure. Gamecreator also allows inspecting and modifying any object at runtime to allow game mastering (Jonsson and Waern, 2008), which sometimes requires human intervention based on a history of events.

Gamecreator supports triggering locations based on GPS, RFID and ARToolkit markers (Kato and Billinghurst, 1999). RFID and ARToolkit markers are identified by their corresponding ID, which is assigned to an object or location. GPS locations are identified by their corresponding areas, circles or polygons, which get defined via an integrated map-editing component that is based on Google Maps (49).

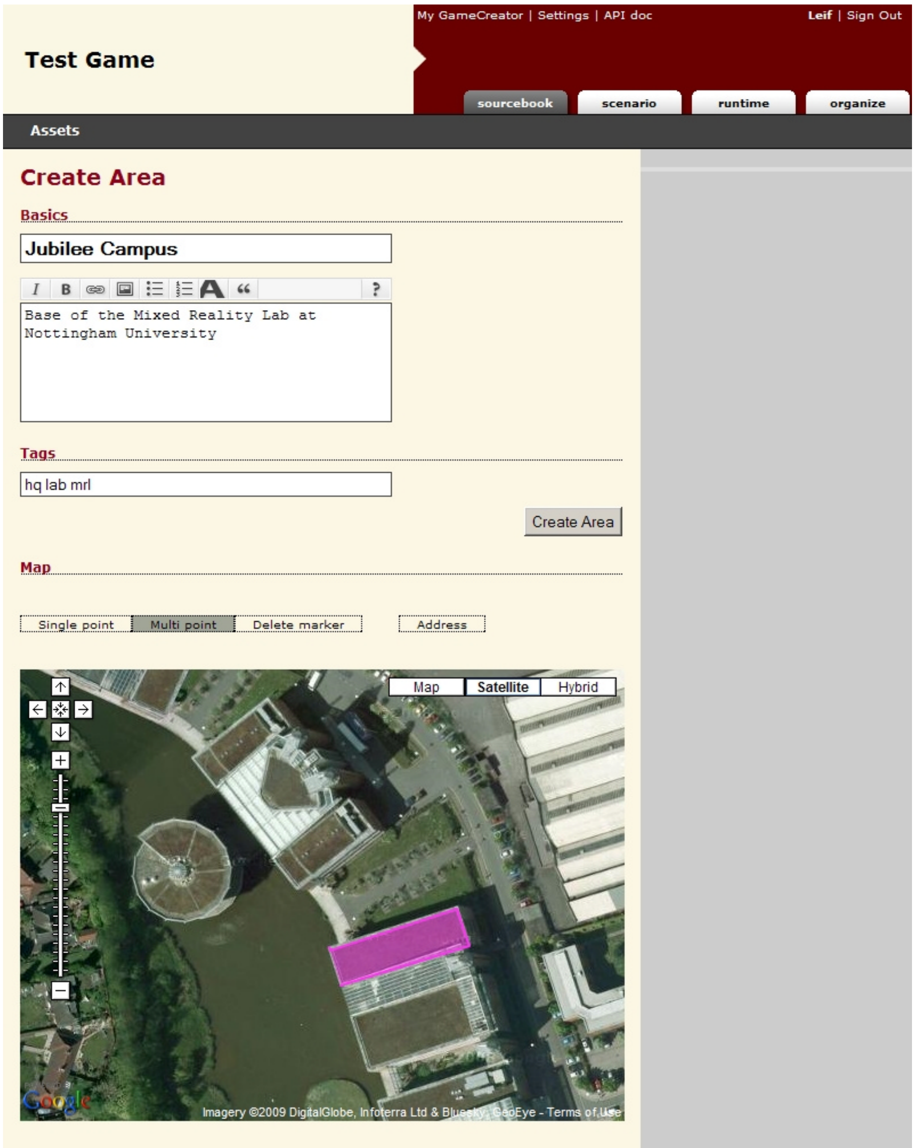

Figure 2.42: Created a rectangular area in Gamecreator

Figure 2.42 shows a screenshot of Gamecreator as it has been used to define a GPS area on the Jubilee campus of the University of Nottingham. This area can then be used to cause as a trigger for in-game events, such as a participant entering or leaving it. Figure 2.43 shows a view of the Gamecreator as it has been used to associate a media-asset to the virtual "Jubilee Campus Location" which will be activated when a participant enters the area that was defined in figure 2.42.

Gamecreator was still in a prototype stage when the IPerG project ended. The applicability of its general concept of encapsulating the game logic separate from the technology had been verified in several research games. One of the games was even restaged in another city through appropriation of assets, mostly redefinition of real world triggers. But Gamecreator was also reported to be in need of another iteration of development, as the system was reportedly hard to use. Making the games actually work still required a lot of custom programming on the mobile side, which was supported by the Gamecreator API. The authors concluded that Gamecreator, in combination with a middleware such as PART/PIMP (see page 332), would provide a good platform for rapid prototyping of pervasive games.

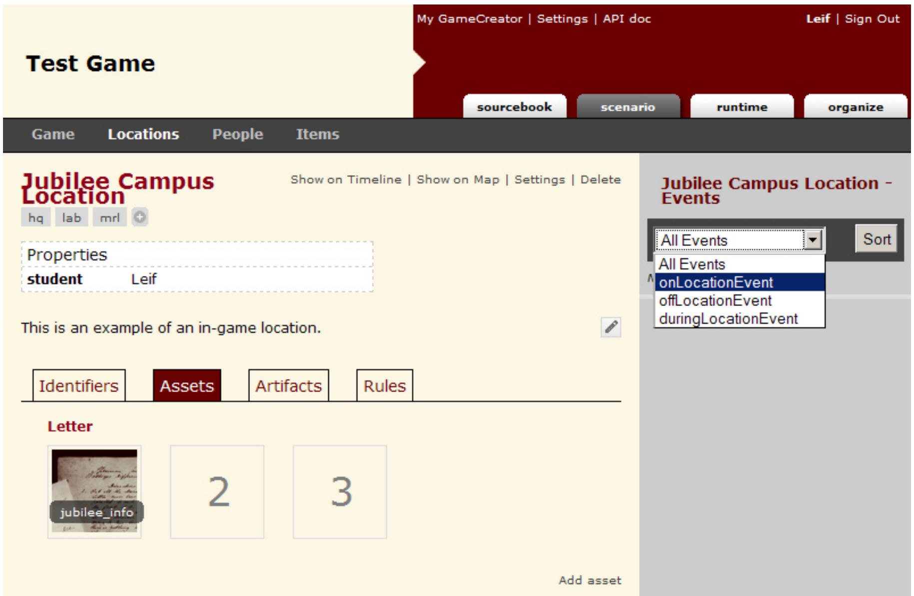

Figure 2.43: Associated asset to in-game location

2.3.3.7 Mercury Platform

Mercury Platform, the integrated location-aware information system that is deployed at several Greek tourist sites, was already generally described on page 65. This section is specifically about the authoring components of the system. As a reminder, the system was built for Windows Mobile PDAs and integrated a range of client-initiated location-sensing technologies (IR, GPS, Wi-Fi) into a portable information system that could locate visitors indoors as well as outdoors. Authoring for this system is done via two components: a desktop authoring component and a mobile authoring component.

The desktop authoring component is used for content management and spatial editing. Authors start by defining trigger zones in geo-spatial coordinates via direct manipulation on a map-background (figure 2.44). One or more information entities, represented by blue circles in the figure, can be associated to each trigger region. This was found to be necessary as for many locations it would not be possible to clearly distinguish the visitor's location from another one nearby. The project's solution to this technological problem was to support grouping of several exhibits into one location and present the user with a selection menu when they enter that location. The task panel to the right of the application window provides access to the main content management functions, such as the linking of the available digital media to different locations on the site.

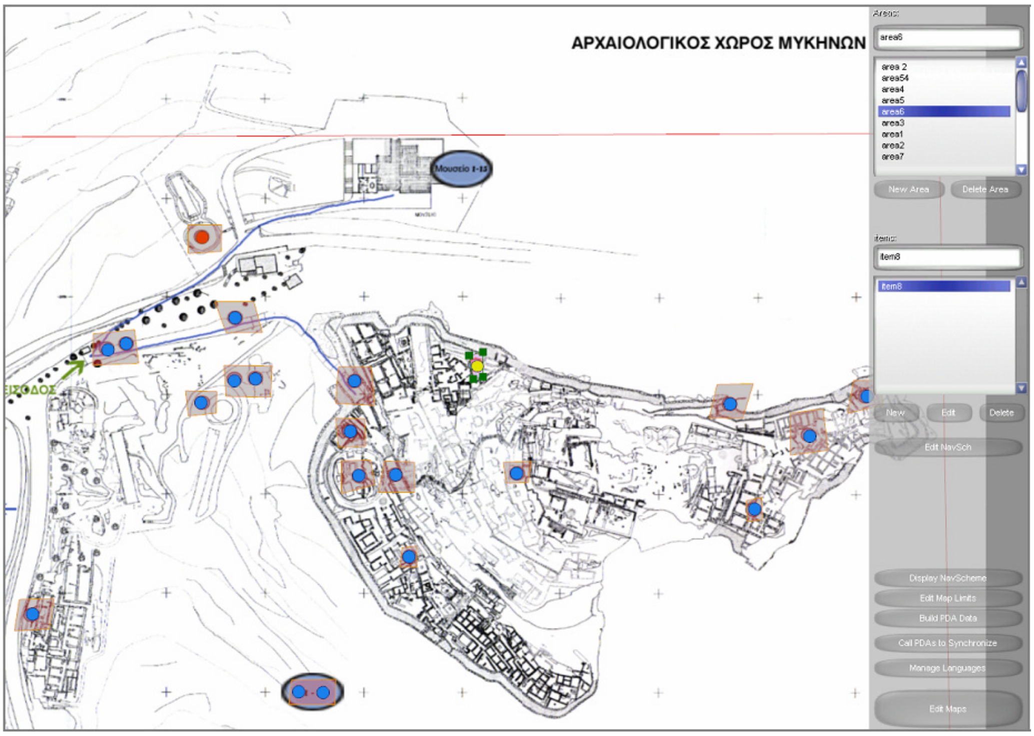

Figure 2.44: Map-Based spatial content editing in Mercury Platform
(Source: (Savidis et al., 2008))

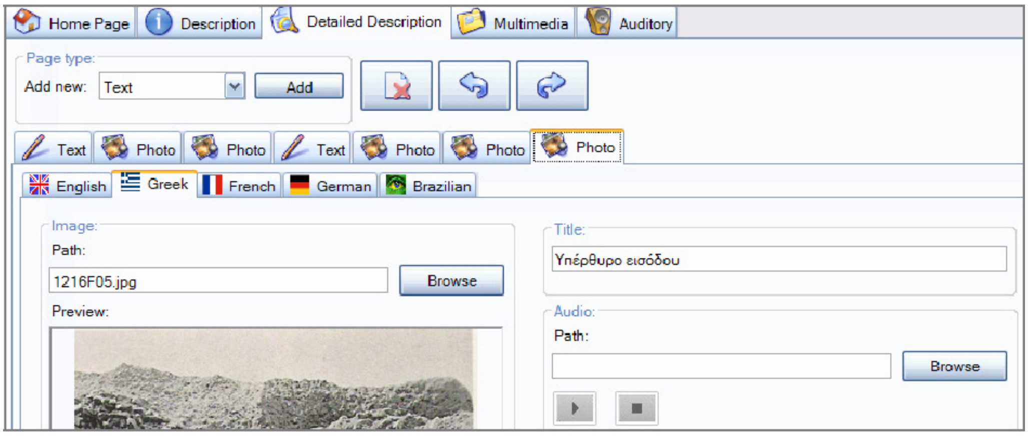

Figure 2.45: Content editing opened from the spatial administration pane
(Source: (Savidis et al., 2008))

A separate window is opened when the author clicks on the edit-button when a location is selected (figure 2.45). In this window, the available multi-media content can be associated with the selected location. A typical workflow would include creating a new content page of a pre-defined type, e.g. text, photo, audio, video, etc., and then importing the desired assets into the content database. A notable difference to similar systems is the multi-language supports, which allows grouping the content into different language schemas. This was necessary as the Mercury Platform targets international visitors.

In a second step, the authoring commences on site. This is supported by a mobile authoring tool that runs on the actual hardware which is also later used by the visitors. The mobile tool was devised as the authors were aware of the limiting factors of the technology that they employed, especially their varying accuracy, and wanted to support in-situ authoring of locations to tackle potential problems. The mobile tool (figure 2.46) provides a map-based overview of the current area that is overlaid with content trigger zones. Additional trigger zones can be created on the spot, using the same point-and-click mechanism as in the desktop tool.

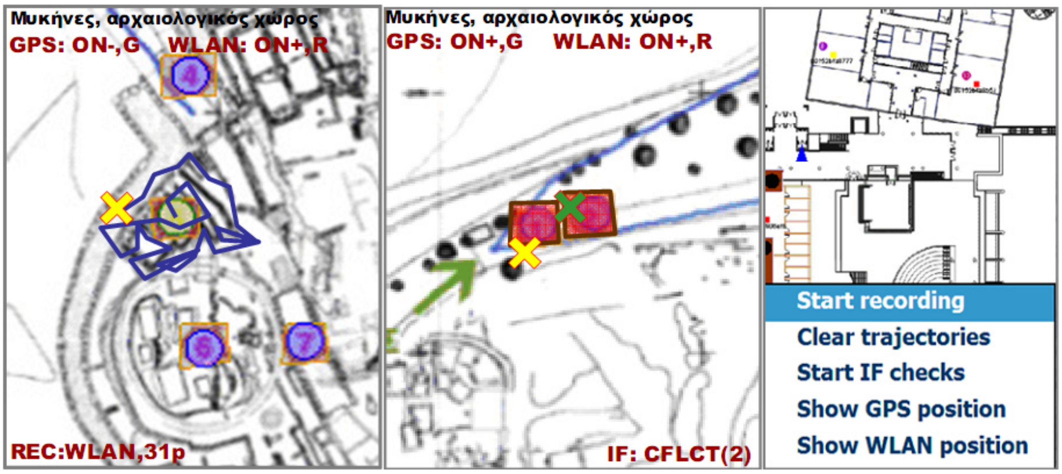

Figure 2.46: Mobile authoring tool (Source: (Savidis et al., 2008))

The mobile system can also be used to measure the performance of the utilised location technologies through analysis of the printed on-screen debug messages that are written in red. The left and the middle images of figure 2.46 show examples of such messages. Looking at the top of the images, it can be seen that both GPS and WLAN (Wi-Fi) were active ("ON") when the screenshots were taken. The suffixed "+" and "-" signs indicate if the respective subsystem actually returned a usable position value. As such, it can be seen that in the left image, GPS was actually not getting a position update, but WLAN did. In the middle image, both GPS and WLAN have been reported to work, but the bottom line ("IF: CFLCT(2)") reported a problem in the infrared sub-system where two infrared markers were visible from the current position and thus causing a location conflict. The location areas associated with those two infrared markers are highlighted to aid the mobile author in fixing the location ambiguity. Finally, trajectories of position information can be recorded over time and visualised on screen. The right image in figure 2.46 shows the menu from which this option can be invoked. A resulting visualisation is shown as a connection of lines in the left image, where the user's current position is marked by a yellow X. The displayed trail corresponds to the WLAN subsystem, and the additional debug message at the bottom of the image ("REC:WLAN, 31p") reveals how many different positions (31) are contained in the current trajectory.

2.3.4 Summary of Authoring tools

Based on the presented authoring tools for location-based applications, this section briefly summarises the key concepts that underpin their designs.

2.3.4.1 Background Map

All authoring tools provided a background map of the relevant areas with the intention to aid the orientation of the author during the task of spatially arranging the content. The background map is always registered with real world coordinates. The registration step is handled differently between tools, but usually in one or more of the following three ways:

1. The tool provides a pre-registered background map which cannot be changed by the author.
2. The tool provides a way to self-register an image or model by defining several reference points in both image-space and geo-space.
3. The tool is connected to a map-service that provides registered map data.

With the advent of freely available map-data on the Internet, it can be expected that more and more tools will revert to option 3. For example, the latest

Mediascapes version includes a feature which allows searching and importing maps from an online service.

2.3.4.2 External Content Creation

None of the reviewed authoring tools provided functionality for asset production beyond simple text-entry. Creation of digital content, such as images, audio, video or HTML pages, is assumed to be done with available external programs. Media assets are usually imported from the file-system and associated to virtual locations.

2.3.4.3 Content Trigger Zones

The virtual locations are merely bundles of associated media assets that should be presented to the user of a system when he enters the corresponding real world location. This is achieved by defining content trigger zones that allow making containment tests with the position readings that are obtained from the respective positioning technology. For many of the presented examples, GPS was used as the positioning technology and GPS provides an absolute position reading in geo-spatial coordinates. In these cases, trigger zones are usually simple geometric primitives such as circles, spheres or polygons. These primitives are often clamped to the ground and used to delimit an area of unlimited height that starts on the surface of the earth. This is often seen as a feasible trade-off between flexibility in modelling a location and ease of use, as a user with a GPS-enabled device would usually be outside and roaming the area on the ground rather than changing his altitude. An exception to this rule is found in the Yamamoto tool, which provides more sophisticated polygon-based multi-floor modelling capabilities due to its orientation towards indoor location modelling. An alternative to vector-based primitives is the raster-based approach as represented by colourmaps and CAERUS. The raster-based approach registers a rectangular grid within the geo-spatial coordinate-system. The cells of this grid overlay are then somehow associated with different virtual locations, e.g. by colouring an associated image or by selecting a series of cells in an interactive editor. At runtime this process would be reversed: incoming position readings would be transformed from geo-space into overlay space. In case of an intersection between the transformed position and the overlaid grid, the corresponding value at the intersection point would be read out and used to look up the current location from a table.

Some of the presented examples did not, or not only, support absolute positioning technology, but also provided means to utilise relative positioning technologies. Relative position is usually used as an indicator of nearness to a real-world object and thus acts as a proximity trigger. Examples include RFID tags, as supported by Gamecreator, and infrared and Bluetooth beacons, as supported by the Mercury platform. Proximity triggers are usually handled by associating the unique identifiers of the chosen tags or beacons with the desired virtual locations. This technique can also be used to tag moveable objects and players, as demonstrated by some of the games created with Gamecreator.

2.3.4.4 Content Trigger Events

With locations modelled in one way or another, the next step for any location-based experience is to somehow programmatically react upon this changing variable. Due to their inherent dependency upon the user's location these applications are non-linear in their structure. This non-linearity has to be taken into account when designing the program control flow and the common solution to this in the context of location-based applications is to employ an event-driven model. In its simplest form this would incorporate a central event dispatcher and a number of event handlers that are known to the dispatcher and will be called upon pre-defined calling conditions. The dispatcher receives incoming location readings and uses these to create a number of events that are passed on to the associated event handlers. The event handlers would then react accordingly and usually update the user interface so that the change of locations would become apparent to the user.

The event dispatcher can generally be built in two different ways: to either work on a single location reading or work on a history of location readings. The first approach would typically generate events upon a user entering or leaving a location. This resembles the way that established multi-media authoring tools such as Adobe Flash or Director work, where authors can place objects on a timeline that get called when the current runtime of the application enters ("onEnter") or leaves ("onExit") the timeframe of the object. The second approach can go beyond atomic states and generate events based on inter-location relationship. This is commonly done by taking the user's current and last locations into account, deducting transitional states from this information, and then act upon these transitions rather than on locations alone. Whatever way is chosen to build the event dispatcher, the underlying event model eventually leads to the use of state machines, which can be used to model the behaviour of software, based on a number of states (here: locations), transitions and actions (Wagner et al., 2006). A simple example of a state machine comprising of two locations (home and office) and two transitions (going to work and going home) of an imaginary office worker is shown in the state diagram in figure 2.47. Different actions might be attached to its discrete locations and transitions, e.g. the person following the depicted routine might want to make himself a cup of coffee upon entering the office, but would make tea instead when coming home. Similarly he might buy a packed lunch on the way to work, but would buy the evening newspaper when going home. State machines are part of the theoretical side of computer science. They provide an unambiguous way of describing *what* needs to be done in a computer system (Eirund et al., 2000). Of the reviewed examples, so far only a beta-version of Mediascapes (50) explicitly supports a state machine.

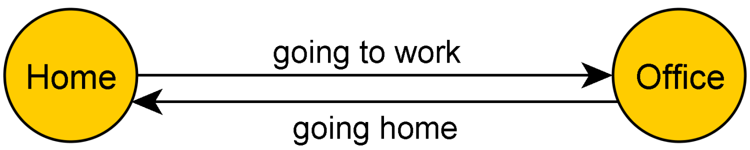

Figure 2.47: State diagram with two states and two transitions

Most of the presented authoring tools provide at least an onEnter handler and often also the onExit handler to model the flow of the experience. This is especially true for those tools that are specialised towards a specific application, such as the tour guide authoring tools MARS, CAERUS and Mercury Platform. Another frequently supported handler is based on the time that a user resides in a location. In addition to these spatial and temporal containment tests, more sophisticated authoring approaches allow the definition of conditional logic; this can be found in Gamecreator and Mediascapes. Both of these tools utilise a snippet-based approach where pre-defined actions can be selected and arranged into a more complex set of rules. This is thought to enable non-programmers to construct conditional logic in a visual manner. In addition, Mediascapes also provides a scripting interface that allows the more advanced user to formulate solutions to problems that are not already covered by the pre-defined snippets and add them to the snippet-library so that they will be available to others. Finally, Yamamoto and colourmaps do not provide any event handling mechanisms, as their focus is slightly different to the other approaches: they only intend to support the actual location modelling task, leaving runtime implementation details to be dealt with separately in the individual project.

2.4 The Problem of Uncertainty

Regardless of all that has been said so far about the authoring of location-based experiences, published studies suggest that there is more to designing them than simply attaching media assets to a map and devising an event triggering model. Rather – in strong contrast to traditional wired applications – designers need to be aware of the inherent uncertainties in the ubiquitous infrastructure, most notably limited coverage of wireless communication and limited coverage and accuracy of positioning systems.

The analysis of the game *Can You See Me Now?* (CYSMN, see page 37) revealed the inherent patchiness and limited coverage of wireless systems such as Wi-Fi and GPS, how it affected the game experience, and how the supporting staff dealt with these issues (Benford et al., 2006, Crabtree et al., 2004, Flintham et al., 2003b). For the first two outings in Sheffield and Rotterdam, CYSMN used a custom built Wi-Fi infrastructure for communication. The Sheffield setup consisted of a single access point which was made accessible via an eight metre mast on a rooftop supplemented by two omni antennae. The Rotterdam outing had a more elaborate setup that consisted of seven access points, four of which were on buildings, one on a lamppost, one in a van, and one on a ship. Despite the effort that went into building these custom networks, street players were repeatedly disconnected from the game server due to network coverage problems. In an attempt to mitigate connection problems, later versions of the game abandoned custom Wi-Fi communication in favour of GPRS (Benford, 2007), as this is almost ubiquitously available on mobile phone networks. Similarly, the accuracy of GPS positioning affected the performance of the game, as it varied significantly over space. While some parts of the game area provided a clear view to the sky and therefore good GPS conditions, others were built up and provided only a narrow view to the sky. This is known to lead to problems with GPS reception – the so called multipath effect – where the radio signals from the GPS satellites reflect off surrounding buildings and terrain and arrive at the GPS receiver several times and with slight delays. This degrades the quality of the signal, confuses the GPS receiver and causes inaccuracies in the calculated position.

Performers, game managers and designers had to understand and account for the local characteristics of the ubiquitous computing infrastructure at each new city that they visited, building up a common stock of knowledge as to coverage and performance blackspots. It also became apparent that the witnessed effects did not only vary in space, but also over time.

2.4.1 Spatial Disparity in CYSMN

The extent of the spatial disparity problem can be seen in figure 2.48 which was produced in the aftermath of *Can You See Me Now?* (CYSMN). It visualises the characteristics of GPS and Wi-Fi over the course of a two hour performance on a peninsula in Rotterdam. The brightly coloured areas correspond to locations where performers acquired a GPS fix and their PDAs were able to transmit the received position data back to the game server over Wi-Fi, i.e. both GPS and Wi-Fi were available at that moment. Light green shows areas of high estimated GPS accuracy and light red denotes areas of lower accuracy. The dark areas represent areas that were either not accessible to performers (e.g. water, buildings, depicted in black) or where some combination of GPS and Wi-Fi was not available (depicted in grey).

Given that the performers ventured widely during the course of the game, there were clearly many areas around the peninsula that were not playable as intended. This is especially true for the narrow thoroughfares between the buildings at the centre of the peninsula, where performers reportedly went, but experienced disconnection as either GPS or Wi-Fi failed to function.

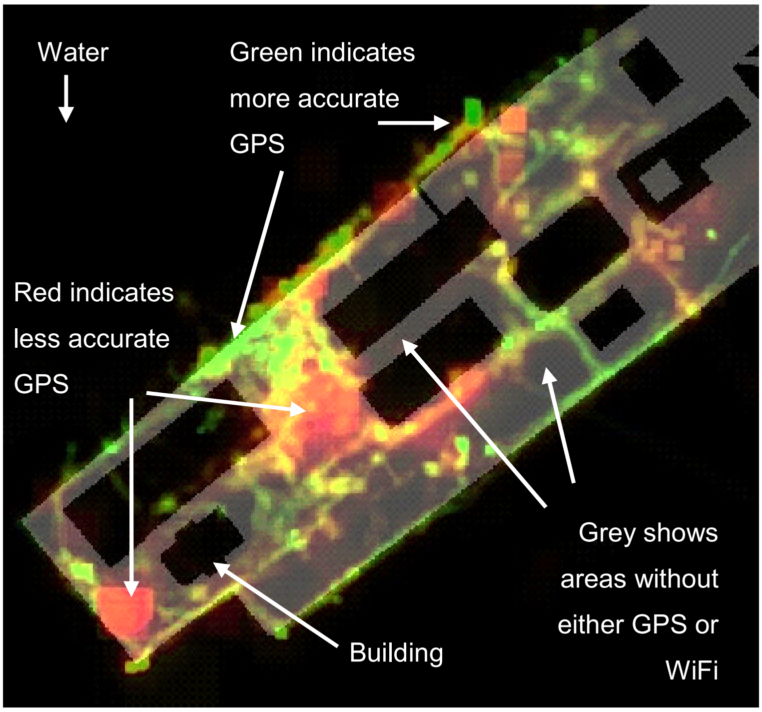

Figure 2.48: GPS and Wi-Fi coverage and accuracy from CYSMN Rotterdam
(Source: (Benford et al., 2006), recoloured)

It is also noteworthy that even reportedly good GPS readings did not accurately match the underlying model of the area. This can be seen on the upper side of figure 2.48, where a series of GPS readings are plotted over the black area that denotes the water. None of the performers ever went out onto the water, so there was clearly some kind of registration error between reality and virtual overlay that summated to several metres of misalignment. This error might have been caused by a general calibration error in the geo-referencing, but given that figure 2.48 only visualises two hours worth of data, it could also have been due to temporal drifts of the GPS signal.

2.4.2 Temporal Disparity in CYSMN

Further to the spatial disparity, the availability and performance of wireless infrastructure such as GPS and Wi-Fi also varies over time, as objects might move into and out of the environment and GPS satellites move over the sky. Consequently, static visualisations of historical trends, such as the one depicted in figure 2.48, often do not provide a rich enough picture of the dynamic situation to offer guidance for future coverage predictions.

Another visualisation produced in the aftermath of CYSMN sought to reveal the temporal variations in GPS coverage. It consisted of a geo-reference 3D model of a city which was overlaid with coverage shadows that were calculated from the model in combination with time and orbital information of the GPS satellites in the sky. The visualisation takes the form of an animation, a screenshot of which can be seen in figure 2.49. It effectively reveals the anticipated areas of good and bad GPS availability throughout the model over time, as the GPS coverage shadows move according to the number of satellites (3 or more for "good") that are in direct line of sight from the respective point on the ground at the given time.

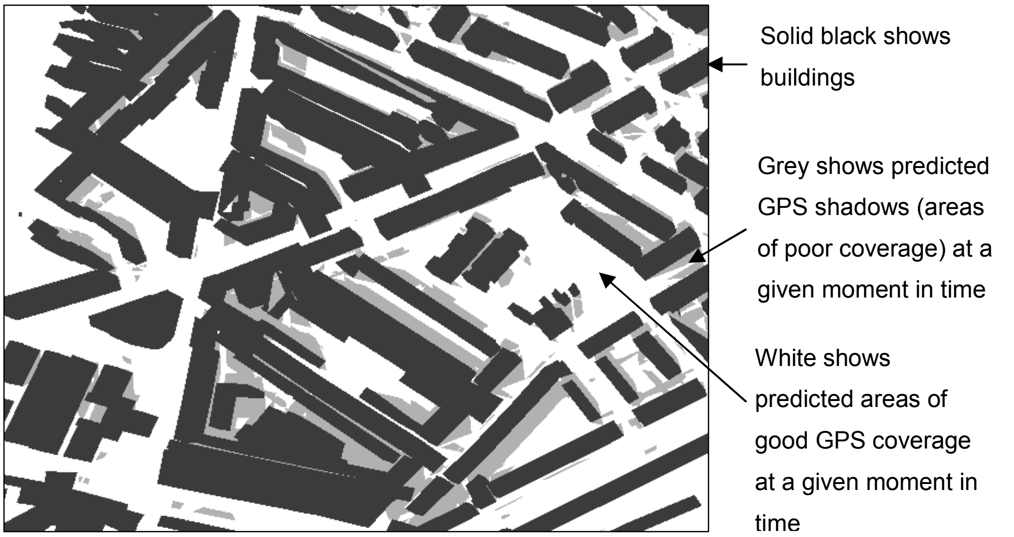

Figure 2.49: Visualisation of predicted GPS availability (Source: (Benford et al., 2006))

2.4.3 Manifestation in other Projects

Spatial and temporal disparities in the accuracy and coverage of wireless technologies did not solely appear in the evaluation of CYSMN. They are much rather inherent uncertainties of any wireless positioning and communication system and undermine the behaviour of all location-based experiences that are built from these components. The following provides a brief account of related effects that have been reported in the evaluations of some of the previously presented examples.

2.4.3.1 Mediascapes

The Mediascapes desktop authoring tool provides an emulated mobile component, which provides the same full multi-media output as it would eventually be experienced outside on a PDA. But the emulator does not use real GPS data and instead utilises a self-reported positioning technique, where an author declares the position of interest on a map. The emulator can be used by authors to quickly test their current state of the work and proved to be a valuable tool. However, the evaluation of the system (Hull et al., 2004) revealed that the authors were then quite surprised when they tested their work in the wild and it did not behave and feel like they anticipated. Different kinds of noise have been discussed in the paper. In addition to the audible and visible noises, such as traffic noise and bright sunlight (hard to read the screen), this also included the accuracy of the GPS positioning. Their authors experienced problems as they tried to trigger some of the smaller regions that they had no problem with triggering in the emulator; but due to the inaccuracies of the GPS positioning, they found it too hard to trigger them using a real GPS receiver on the ground. To compensate for the effect of GPS jitter, the authors increased the sizes of those trigger zones that were too small. The Mediascape evaluation concluded that noise, and especially GPS jitter, should be emulated as well, to give authors a more realistic impression of the situation on the ground.

2.4.3.2 REXplorer

REXplorer (see page 58) was originally devised to trigger its content based on the user's GPS position (Walz et al., 2006). This was later found to be impractical as the narrow streets and built up environment of the old town of Regensburg would frequently prohibit a usable GPS fix. So the authors devised a hybrid mechanism that would additionally include Bluetooth beacons to define locations, as well as a manual selection mechanism in case the location detection fails (Ballagas and Walz, 2007). The evaluation of REXplorer revealed that automatic location detection had after all been completely removed from the final system and users would merely select their current locations from a menu (Ballagas et al., 2008). Not using the Bluetooth mechanism was attributed to a lack of support from local shop owners, who were reportedly unwilling to provide space for the required Bluetooth beacons in their shops. The failure of the automatic content triggering mechanism in REXplorer illustrates what can happen to a project if the uncertainties of the ubiquitous computing infrastructure are not taken into account from the beginning of the design process and instead one relies on assumptions. It also shows that even when these uncertainties are dealt with, external factors such as lack of support might still endanger a project.

2.4.3.3 Treasure

Treasure's overall design (page 49) is geared towards exploiting the uncertainties of wireless positioning and communication technologies by making them an integral part of its game-design. Players are constantly crossing the boundaries of Wi-Fi coverage in order to collect coins, which are located outside of coverage, and then subsequently claiming points for the coins when they are back in Wi-Fi range and can connect to a central server. This game-mechanic is supported through Wi-Fi coverage visualisations on the mobile user-interface, which provides players with visual hints that allow them to develop tactics and strategies about how to best cope with the problems of uncertainty (Barkhuus et al., 2005).

2.4.3.4 Place Lab

The Place Lab project (42) from Intel Research Seattle and the University of Washington implemented an hybrid positioning system which allowed locating a user based on GPS, Wi-Fi and the GSM mobile telephone network. The project conducted an experimental study (LaMarca et al., 2005) where they tested the availability of these positioning technologies throughout their participants' daily lives (see table 2.3). This positioning technology / time coverage experiment for humans concluded that GPS would only produce a useable fix for a tiny fraction of a user's daily time (4.5% in their experimental setup), with Wi-Fi covering more than 90% and GSM close to 100% of the user's daily time. This experiment is highly interesting as it reveals that GPS is obviously not the ideal solution to every positioning problem, especially not when it comes to ensuring the availability of a service at "anytime, anywhere".

Test Subject	GPS		GSM		Wi-Fi	
	coverage	avg. gap	coverage	avg. gap	coverage	avg. gap
Immunologist	12.8%	68 min	100%	-	87.7%	1.6 min
Home maker	0.6%	78 min	98.7%	2 min	95.8%	1 min
Retail clerk	0%	171 min	100%	-	100%	-
Average	**4.5%**	**105 min**	**99.6%**	**1 min**	**94.5%**	**1 min**

Table 2.3: Place Lab time-coverage study for 3 wireless technologies
(Source: (LaMarca et al., 2005), average times rounded)

2.4.3.5 ARQuake

The evaluation of ARQuake (see page 33) revealed that the inaccuracy of the GPS tracking affected the playability of the smaller levels that were tested with users (Thomas et al., 2002). The GPS inaccuracy was less noticeable to users in larger levels; this emphasises the relationship between the precision of the positioning system and the overall size of the area, which is the granularity of locations in the system.

2.4.3.6 Cyberguide

The Cyberguide project (see page 25) reported that in their experience absolute positioning was less important than knowing the users' semantical location (*"to know what someone is looking at"* (Long et al., 1996a: 294).

2.5 Motivation for this Thesis

Building location-based applications using off-the-shelf hardware has at least been possible since the late 1990s, which is when the seminal work in this field has been produced. This is now ten years ago. Over the past ten years, a lot of research and development work went into the continuous improvement of the hard- and software components and wireless infrastructures that drive these distributed systems. Keeping Moore's law in mind – which expresses how electronics get cheaper and integrated circuits double their density every couple of years (Moore, 2005) – it becomes obvious that the available hardware can not be the limiting factor when building such systems today. The same goes for the software, as a multitude of supporting libraries are available to support the developers. We are now in a state of transition where the technology gets handed over from developers to experience designers and so the attention increasingly turns towards the authoring of content and the shaping of the overall end-user experience.

While developers might be aware of technological pitfalls – which is not necessarily the case – this awareness can definitely not be expected from the non-technical designers who are nowadays in charge of the production process. The problem of uncertainty in the wireless infrastructure is a hard one, especially because its effects are invisible and its causes are manifold. Fragments of advice on how to best design location-based experiences and cope with the effects of wireless technology can be found in the literature. Such advice usually stems from evaluations of previous experiences, which, in other words, means that the respective authors gained some knowledge from their own mistakes and wanted to share it with others through their papers. But it appears that these messages do not always get through to those that devise another new system; which might be for a number of reasons. For one, a message might simply not reach its destination, because a particular paper is not read by the person who would benefit from it. A second reason might be that the academic approach of advancing knowledge through the exchange of written papers does not necessarily fit into the workflow of creative professionals. A third reason might be that humans tend to prefer learning from their own mistakes, or "learning by doing" (Baden-Powell, 1908), rather than accepting knowledge gained by others as applicable to their current problem.

Finally, and most importantly, current tools and workflows for building location-based experiences provide no way of telling what is going on with the wireless computing infrastructure. It must be concluded that the available information about the behaviour of wireless communication- and, especially, positioning-components in location-based experiences is currently not presented in appropriate forms to support the designers in their work.

3. Conceptual Framework

The literature review concluded that current authoring concepts, tools and workflows for building location-based experiences do not sufficiently support the mobile experience designer in thinking about the wireless computing infrastructure, which causes a gap between what experience designers expect and what wireless technology can provide.

This raises a fundamental question for location-based authoring tools: what is the point of placing assets at physical locations where there is insufficient network coverage or positioning accuracy to reliably access them? Or in other words: how can we enable designers to take into account wireless computing infrastructure as well as the nature of the physical environment while authoring an experience? And how can designers learn to master the usually invisible wireless computing infrastructure to an extent that enables them to realise their envisioned experience *through* the technology rather than being continuously hampered *by* the technology? These are the main challenges addressed by this thesis.

This chapter sets out the conceptual framework for this thesis. The basic idea is actually very simple: to extend the conceptual model of current location-based authoring tools so that they can reveal to the designer the complexities of the wireless computing infrastructures on which they rely.

The idea of revealing complexity originates from the field of information design, which deals with the question of how information can be prepared to be consumed by humans with efficiency and effectiveness. Information design is of interest to this thesis. As this field might be new to the reader, an introduction to information design, including examples of excellence in information design for maps and charts, has been compiled and is presented in appendix 1 (page 316). Section 3.1 (page 119) outlines the core idea of this thesis: it presents a general plan for how principles of information design could be reflected in the design of location-based authoring tools and proposes an extension of their conceptual model, including visualisations of the wireless computing infrastructure, so that experience designers can make informed decisions about where to place content. Section 3.2 (page 124) elaborates on my definition of location and how they are mapped onto the physical world. Section 3.3 (page 131) presents an abstract location model that facilitates the handling of locations. Implications for the overall development workflow and supporting tools are listed in section 3.4 (page 135), which outlines several key challenges to be explored in the remainder of the thesis. Section 3.5 (page 139) presents a plan of action to guide the practical work of this thesis.

3.1 Extending Authoring Tools

Location-based authoring tools allow experience designers to spatially arrange digital content, and to define how and when this content is going to be presented to a user of the resulting system. The review of the literature about deployed location-based experiences revealed that the performance of the utilised wireless positioning and communication infrastructures had an impact on the end-user experience. Yet, current location-based authoring tools do not sufficiently support experience designers in reasoning about this effect and are rather built on the assumption of a perfect registration between the real and the digital by means of a precise location-service. Consequently, this means that as soon as the utilised infrastructure starts to behave in ways different to how the designer expected, the location-based experience authored with such a tool is going to behave strangely, run out of synchronisation or even stop functioning altogether. This is because the designer had no chance to reason about potential technological shortcomings when he authored the experience.

To remedy this situation, and to break false assumptions, it is necessary to increase the designers' awareness of the uncertainties of the technologies that they work with. This thesis seeks to facilitate this task by revealing the characteristics of the utilised wireless infrastructure to the author of an experience at authoring time. This requires the general approach to authoring location-based experiences to be extended from a two layer model into a three layer model.

3.1.1 Current Two Layer Model of Authoring

The current state of the art authoring tools for location-based experiences share one common overarching approach: the idea of taking a map of the physical area in which the experience is to take place and then somehow placing or drawing a series of regions or hotspots on top of it that define the locations where specific pieces of content are going to be triggered. This approach can be thought of as a two layer model as depicted in figure 3.1.

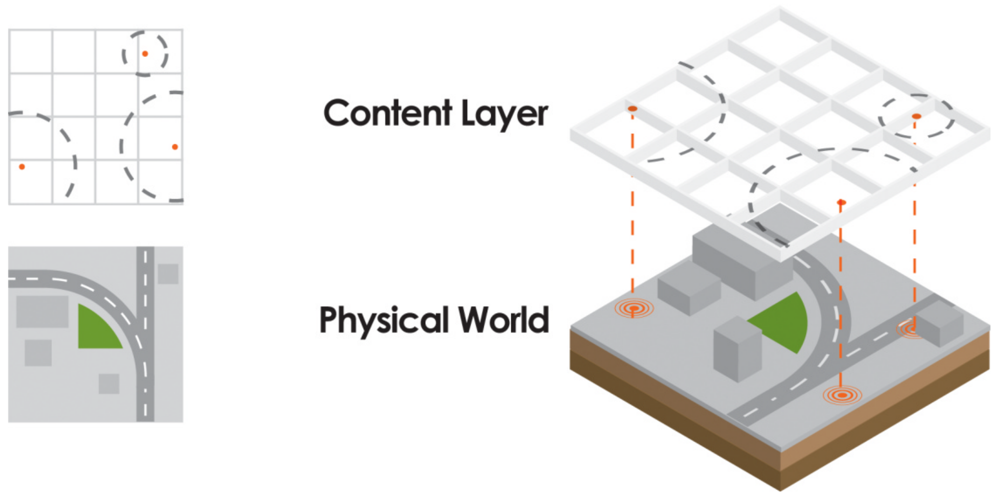

Figure 3.1: Conceptual model of current location-based authoring tools[12]

In the current model, the "Physical World" layer represents the target physical environment and usually consists of a map or an aerial photograph. The "Content Layer" represents the experience designer's authored content and usually consists of a combination of digital assets and their associated trigger zones and events. The presentation of the digital assets is scheduled based on the participants' location in the physical world intersecting with predefined content zones and triggered according to a set of basic events such as participants entering, leaving or dwelling in them[13].

In this model, the physical world and its digital content overlay are aligned in the same coordinate system and stitched together via a location service that measures the position of the user in the physical world. Building location-based experiences according to this model makes them rely heavily on the precision of the location service in use. While this authoring approach works well for many cases, it is still too simplistic, as it neglects the inherent uncertainties of the wireless computing infrastructure – in fact it does not represent them at all!

---

[12] Figures 3.1 & 3.2 drawn by Markus Hammonds

[13] see page 106 for a more detailed summary of the key concepts of current authoring tools

3.1.2 Extending the Model to Three Layers

In the new extended conceptual model, the designer gets supported with knowledge about the physical world and the wireless computing infrastructure at the same time. The authoring process now involves overlaying information characterising communications and sensing systems on meaningful backgrounds that facilitate the orientation of the experience designer, e.g. for geo-coded data this could be a map of the area or for time-stamped data this might be a timeline.

Figure 3.2 depicts this extended conceptual model of authoring location-based experiences, which is the core innovation of this thesis. It enables the designer to work with three layers of information: the *physical world layer*, with representations of the target physical environment; the *infrastructure layer*, with representations of the wireless computing infrastructure across this environment, and the *content layer*, with representations of digital media. In practice, each layer may consist of a series of sub-layers that reveal different types of information: maps, aerial photographs and data from geographic information systems (GIS) for the *physical world layer*; recorded and predicted information about the coverage and accuracy of communication and location services for the *infrastructure layer*; and assets, content trigger zones and events for the *content layer*.

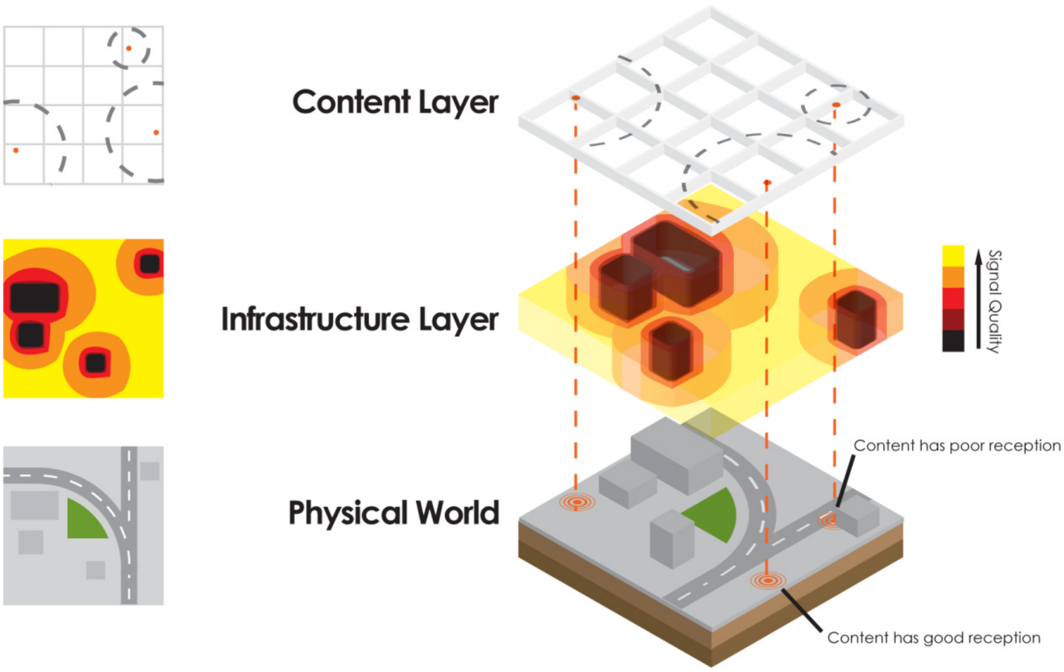

Figure 3.2: Conceptual model of extended location-based authoring tools with infrastructure layer

The key innovation here in relation to previous work is the insertion of the infrastructure layer. This layer may show previously measured values, simulated data, or both. It helps the experience designer to better understand the spatial and temporal spread of the otherwise invisible wireless computing infrastructure (e.g. mobile phone networks, Wi-Fi, GPS) at authoring time. The advantage of this new model is that the author of a location-based experience can now make informed decisions about the placement of digital content based on the revealed characteristics of the wireless infrastructure, whereas without this layer, he would have had to make assumptions or even disregard these characteristics completely.

The advantage of the new model is best illustrated by an example. Figures 3.1 and 3.2 illustrate the same authoring scenario in both conceptual models. Imagine an experience designer intends to place three different audio-samples in the depicted area, which contains some buildings and roads. He defines three circular content trigger zones (the dotted circles on the content layer) and associates each of them with an event that plays the desired audio-clip when a participant enters the respective zone. GPS is used as the location service, so that a zone is programmatically entered when the participant's reported GPS coordinate falls within the range of one of the circles. While this example appears to be straightforward and unproblematic, it actually already contains a catch. GPS is known to work best with a clear view to the sky and with some distance to buildings. But the small circular region is located too close to a building, which might affect its retrievability, and is thus arguably a bad location for content.

Such knowledge could be represented by the infrastructure layer, which, in this example, shows a visualisation of anticipated good (yellow) and bad (dark) areas for precise GPS coverage. It can be seen that one of the defined content regions falls within an area of anticipated bad coverage (dark, close to a building). Only authoring tools that provide such information about the wireless computing infrastructure to designers empower them to make well-founded decisions regarding the placement of digital content and to avoid defining content regions that could not be reliably triggered in the final experience. Representations on the infrastructure layer are of course not limited to what is shown in this example. They could be applied to different technologies and could focus on different aspects depending on what is needed for a particular project. The following provides a brief summary of the intended use of the different layers in this three-layer model; anticipated challenges are outlined in section 3.4 (page 135).

3.1.2.1 Physical World Layer

This layer represents the world around us and provides the user with meaningful backgrounds in regard to the physical target environment. Due to the spatial nature of location-based experiences, such backgrounds will often consist of maps. Although maps are not strictly required when authoring location-based experiences they help to give an understanding of the physical area in which an experience is going to take place and should therefore be used to the fullest. Other possibly meaningful backgrounds include 3D models and time-lines.

3.1.2.2 Infrastructure Layer

The infrastructure layer is the core innovation of this thesis. It provides an experience designer with information about the infrastructure that is part of the design, e.g. availability, granularity and reliability metrics. It contains historical samples and predictions of infrastructure information that are projected in the same way as the meaningful backgrounds in the physical world layer and which are overlaid on top of them. This overlay puts the sampled data into the right perspective and enables the user to see it in context in order to better reason about it. In many cases this context will be geospatial, as the infrastructure layer is aligned to the physical world layer, which in turn will often consist of maps. But dependant on the chosen technology and the requirements of the project, the presentation on the infrastructure layer could also be non-geospatial, e.g. temporal or topological, and it might even be non-visual.

Data for this layer stems from mobile survey activities which need to fit into an overall development workflow. The infrastructure layer is all about revealing the uncertainties of the utilised wireless infrastructure (pages 110, 137).

3.1.2.3 Content Layer

Drawing upon the information contained in the physical world and infrastructure layers, the actual location-based experience itself is going to be defined on the content layer. Here, the author can make informed decisions about where the experience should take place, *where* and *how* to place assets and *when* and *how* to trigger them.

The literature review summarised that existing authoring tools for location-based experiences rely on external programs for digital content creation, i.e. assets like audio, video, images, 3D models or interactive animations would still have to be created in domain specific programs like Steinberg Cubase, Avid Media Composer, Adobe Photoshop, Autodesk Maya or Adobe Flash. This is for a good reason – because these programs are very good tools – and this work does not want to change this model. In the context of this thesis, everything on the content layer is about scheduling the delivery of content to the user based on her location.

3.2 Location Mapping

Location is the key concept of location based experiences and this section clarifies how the term is understood and used in this thesis, and how locations can be mapped onto the physical world.

3.2.1 Location and Position

Location was already briefly defined in the jargon buster (page 21) as a position or area with an associated semantic, such as "Computer Science building". This attached semantic context sets a location apart from a position, which is just a point that is discretely defined by a set of coordinates. For example, the tuple (52.953412° latitude, -1.187508° longitude) defines a geographical position that falls within the boundaries of the above mentioned building and could thus be associated to it – but it is not the same. Sometimes the term location is even used synonymously for defining a geographical position, but the author of this thesis disapproves of this usage in the context of this thesis to avoid confusion.

Position or location should not be used interchangeably, as a location is a disembodied concept with attached semantics, whereas a position is a more tangible definition that is directly linked to its underlying metrics, such as geographical coordinates, GSM cell IDs, or Wi-Fi fingerprints.

3.2.2 Meaningful Locations

With location and position being separated like that, it is noteworthy that a similar distinction has originally been introduced to the computer supported cooperative work (CSCW) community by Harrison and Dourish, who distinguished between space and place (Harrison and Dourish, 1996, Dourish, 2006). They argued that people *"are located in 'space', but act in 'place'"* and that *"a place is generally a space with something added"*. By this they mean that the world around us is more than just the space it provides. The humans that inhabit this world fill their spaces with personal meaning and thereby eventually transform their spaces into places that are meaningful to them. This idea is based on Martin Heidegger's *being-in-the-world,* i.e. how our perception of the world is the basis for our understanding of the world (page 11).

While Harrison and Dourish could use space in its everyday sense, the introduction of pervasive technology for sensing a user's spatial context in location-based experiences complicates the matter a bit. In addition to measurements of space in geographical coordinates through GPS, other technologies might also be used for the task of location sensing (page 76). Each technology potentially introduces its own units of space, most notably the many different ID numbers (cell ID, tag ID, MAC address, fingerprint ID, etc.). Nevertheless, Harrison and Dourish's separation can be appropriated for the design of location-based experiences. Now the location is the notion that matters in the design and the location-based content is the "something" that gets added. This situated content allows people to interact with the system and thus turns the disembodied concept of location into a place that (hopefully) matters to them. This may seem unsurprising at first sight, as *location* is already part of the phrase *location*-based. However, it appears on second sight that many of today's location-based experiences are actually much more concerned about position (i.e.

space, as in latitude/longitude GPS coordinates) than they are about location in this symbolic sense.

Previous research in the field of location-based applications suggested a similar view. Hightower argued that many contextual applications require such a symbolic notation as they *"want to reason about 'place' instead of or in addition to coordinates"* (Hightower, 2003). The outlook of his PhD thesis (Hightower, 2004), which presented a framework for sensor fusion that was adopted by Intel's Place Lab project, concluded that further work should be undertaken in this direction. Although the Place Lab project (LaMarca et al., 2005) mainly continued along the route of sensor fusion and related engineering tasks, they also devised a study about automatically learning and recognizing places that people visit (Hightower et al., 2005), based on traces of Wi-Fi and GSM cell ID data from mobile devices. Similarly, Ashbrook and Starner presented a system that clustered traces of GPS data to automatically find *"meaningful locations"* from participants' movements in a post-event analysis (Ashbrook and Starner, 2002). Laasonen et al. provided another similar system that sought to reveal meaningful locations, which they called "bases", from GSM cell ID data in an automated fashion by running dedicated software on mobile phones (Laasonen et al., 2004). Finally, Nurmi and Koolwaaij discussed several algorithms that could be used to identify meaningful locations, which they called "places", and presented their own algorithm that provides unsupervised learning from traces of GSM cell transition data that is enriched with GPS data (Nurmi and Koolwaaij, 2006).

The multitude of meanings that surround the term location alone is already confusing. This section hopefully did not add to that confusion as it tried to put the term into the context that it should be seen in. In short, a location is an empty hook which can be associated with meaning (content for user interaction) and spatial representations (regions of IDs or coordinates), but that is independent of any metric in itself. In theory it is thus independent from any spatial representation[14]. In practice it is, of course, dependent on some form of

[14] 3D computer graphics artists will find this abstraction similar to the concept of "null objects" in Softimage or "locators" in Maya

embodiment to provide an opportunity for interaction (compare Dourish, 2001), and must thus be associated with both content and regions to frame the action for the location-based experience.

3.2.3 Location in the Context of Space and Place

Previous work (Hightower, 2003, Nurmi and Koolwaaij, 2006) already used the term *place* to describe meaningful locations. Nurmi and Koolwaaij additionally employed social identity theory to motivate that places could be defined based on the roles that people enact in them, i.e. worker, spouse, music lover, etc. But both papers ultimately used *place* as something that could be algorithmically expressed and automatically determined. The author of this thesis does not agree with that underlying assumption and will instead base himself on Dourish's notion of place, which is that humans turn spaces into places by accepting them and that places cannot be designed – only designed *for* (Dourish, 2001: 91).

Dourish provided a much needed separation between space and place from a sociological perspective, as he argued that humans appropriate their spaces to eventually turn them into places. The typical example in this context is the difference between a house and a home, and what these terms mean to an individual: a home is cherished place that is also a house, but not every house is a home. Dourish proposes that the evolution from space to place is something that can only be designed for, as it is the humans that have to trigger this transition by accepting a place. As a design principle that facilitates this transition, Dourish advocates supporting human appropriation of space, i.e. to be flexible about its structure. When using this principle as a lens to analyse the concepts of current location-based authoring tools, we see that most of them are not very flexible about the structure of space and rely solely on a single coordinate system, i.e. GPS coordinates[15]. This inflexibility and unnecessary constraint is arguably caused by unquestioned technological decisions, and it makes it very hard for the experience designers to appropriate the space at hand, if the pre-defined structure does not fit for what they have in mind.

---

[15] There are of course exceptions to this rule, like Mercury (page 103) or Gamecreator (page 100), which both provide an abstraction from positioning technology.

To illustrate the point of a flexible space becoming a place by human practice, Dourish provided the colloquial example of a meeting room whose chairs and tables can be re-arranged to provide for different settings such as a circle for discussion and a different layout for presentations. Compared to our domain, the GPS-only location-based authoring tool would be the equivalent of a meeting room whose chairs and tables are bolted to the floor, because it does not allow for structural appropriation.

In order to provide for some of the much needed flexibility, we need to allow for a more flexible definition of space when using pervasive location-sensing technology. It is therefore proposed to decouple the notion of space from any particular reference coordinate system, to allow it to be whatever is required. This can be achieved by subdividing *space* into a metric level and an abstract level, i.e. *position* and *location* (see table 3.1).

	Metric Level	Abstract Level	Interactional Level
Term	**Position** (Space)	**Location** (Space)	**Place**
Example	GPS coordinates, GSM cell ID, Wi-Fi fingerprint	"House" An empty shell; needs to be filled with meaning, i.e. content and regions in our case.	"Home" A location with associated semantics; content and regions in our case.
Usage	Define regions in space according to a particular metric	Links the metric level with the interactional level	Point of interaction with end user

Table 3.1: Position, Location, Place

In this model, the location provides an abstract, empty shell to which any number of attributes can be attached. Attributes of locations could be definitions of regions in space (e.g. geospatial or other coordinates), content (i.e. digital media assets), or meta-information such as name, notes or tags. The advantage of this model is that the notion of location provides an abstraction of the interactional level – that is the anticipated user-experience – from any particular sensor technology that is tied to a particular metric of space. Of course, in practice, this model still requires a connection to a sensor system, but this connection is now more flexible and allows for an easy reconfiguration or extension with other technologies. At runtime, incoming position data gets processed on the metric level and subsequently triggers different locations on the abstract level. These location events can then be used to control the flow on the interactional level.

This combination of a positioning technology (and the regions that are defined in its metric) with the concept of location is a location mechanism. Basically, such location mechanisms span the metric level and the abstract level, i.e. Harrison and Dourish's notion of space. It is in the designers' interest to understand this interplay and the characteristics of the technology that is used to facilitate it, because it forms the structure on which their interaction with the end-user is based when they design an interactive system. It is quite important to understand that employing a flexible location mechanism does not warrant a good interaction design or game design in itself. The resulting experience might still be dull and fail to impress the end user at all. Consequently the locations in such a boring experience would not be regarded as places by the people that use the system. Discussions about good and bad designs fill books, some of which are proposed for further reading later on (page 299). Beyond that, a discussion about interaction design at an end user level falls out of the scope of this thesis.

3.2.4 Mappings between Layers

A location mechanism that is based on the proposed separation of position and location is applicable to all kinds of location-based experiences. This generally includes proximity-based interaction, where the opportunity for interaction is not limited to a certain spatial area (however loosely defined that might be) but rather depends on the proximities of subjects and objects (wherever that might be). Proximity-based interaction is often supported by short-range sensors (page 80) and represents an important branch of interaction for mobile applications. However, the idea of visualising the infrastructure in the proposed three layer model was based on an overlay of the sampled data on top of the physical world layer. This calls for geocoding and appropriate projections. Although tracking and geocoding the movements of short-range sensors on objects or subjects in the physical world might be possible with a lot of effort, this would certainly contradict the idea behind (relative) proximity-based interaction, which is to not be location-based in an absolute sense. Therefore this thesis limits the scope of its further discussion to the more stationary wireless positioning technologies and disregards proximity-based interaction from now on.

3.2.4.1 Create Mappings between Layers

Designers want to provide engaging location-based experiences to their end users. They cannot simply map their content directly onto reality as children do in their games, or as artists do in their street performances. Instead, they have to use technology to map their content to reality, i.e. create a mapping between the physical world layer and the content layer *through* the infrastructure layer.

Designers effectively want to make the infrastructure work for them and consequently make it disappear from their consciousness. They want to use the infrastructure as a tool that is *ready-to-hand* in a Heideggerian sense (page 11). In Heidegger's terms, they don't want to be concerned about the hammer, but instead want to be concerned with hammering. This means that they want to use the infrastructure layer to create meaningful locations in their designs, i.e. on the content layer. Therefore, the relationship between layers is of interest to designers.

3.2.4.2 Relationships between Layers

In a child's game, chalk lines on the street can become quite accurate boundaries between being in and out. This is not so with pervasive technology, as was previously outlined in the section about uncertainty (page 110). Designers of location-based experiences have to somehow map their digital content to the physical world with the help of technology (infrastructure). One straightforward way of doing so is to use GPS and allow for some imprecision in the design. Using relative positioning technologies for location mapping (e.g. cellular positioning or Wi-Fi) introduces an extra layer of complexity. This is because they first need to be surveyed or geocoded before they can be visualised and used in a meaningful way. This survey step is usually done with the help of GPS. Thus the imprecision in that system will be carried forward through the data and needs to be considered in the resulting system (where it might be insignificant). GPS is not the only tool for surveying. It is not even the best tool for this task, as civil engineers with high precision requirements would still rather use a builder's level or a theodolite.

Generally speaking, one technology has to be used to geocode another. In theory it would thus be possible to map Wi-Fi technology with the help of GSM technology, or vice versa – but the resulting data set might not be very useful. For practical reasons, any geocoding work that supports location mapping throughout this thesis will be done with consumer grade GPS receivers. Location mapping will be done from the infrastructure layer to the content layer.

3.3 Abstract Location Model

This section proposes a design for a general abstract location model that can be used for triggering location events. Most existing location-based experiences, including the two studies presented in this thesis, can be mapped onto this model. A very similar model is also found in the Gamecreator authoring tool (page 100), which stems from the author of this thesis collaborating with the author of Gamecreator during its inception phase as part of the IPerG project.

A key task in authoring location-based experiences is defining its locations. The way of defining locations depends mostly on the positioning technology that is going to be used in the experience. A flexible and extensible abstract location model is needed to cater for all the different technological choices. The model must support defining locations across a wide range of current technologies, such as GPS, Wi-Fi, GSM cell ID, and it should be extensible to satisfy future requirements.

The proposed model works by processing raw sensor data and finding out which location(s) they might resolve to and cause to trigger. In this model, multiple sensor readings of one technology can be grouped into regions, and regions across technology boundaries can be attached to abstract locations. For example, a location called "home" could be represented by a combination of its geographical position, a region of cell IDs that cover the area, and a region of Wi-Fi access point MAC addresses that are discoverable in the neighbourhood. Any of those regions definitions could then be used to trigger the location.

3.3.1 Reproducible ID-Strings

Unique and reproducible identifiers for locations and regions are the basis for the proposed model. They allow for a flexible mapping of digital content to locations (page 270) that can be defined using various metrics. Regardless of the chosen technology, any measured sensor data needs to be referenced by unique ID-strings to allow for database storage and unambiguous mapping. Additionally, these identifiers must also be quickly reproducible on the target mobile device from the incoming sensor data to allow for triggering of content based on the current location-context.

3.3.1.1 Locations

In this model, each location is identified by a unique string of text (which acts as the database primary key), and also carries a human readable name (optional).

3.3.1.2 Regions (Location Data)

A location can be associated with any number of region definitions that can be defined using any metric that is suitable for the underlying technology. The current design supports the definition of location data for GPS, Wi-Fi and GSM networks. If needed, the model is easily extended for new technologies through an abstract region interface. Like locations, regions are also uniquely identified by an ID-string and have a name.

3.3.2 Flow of Data through the Model

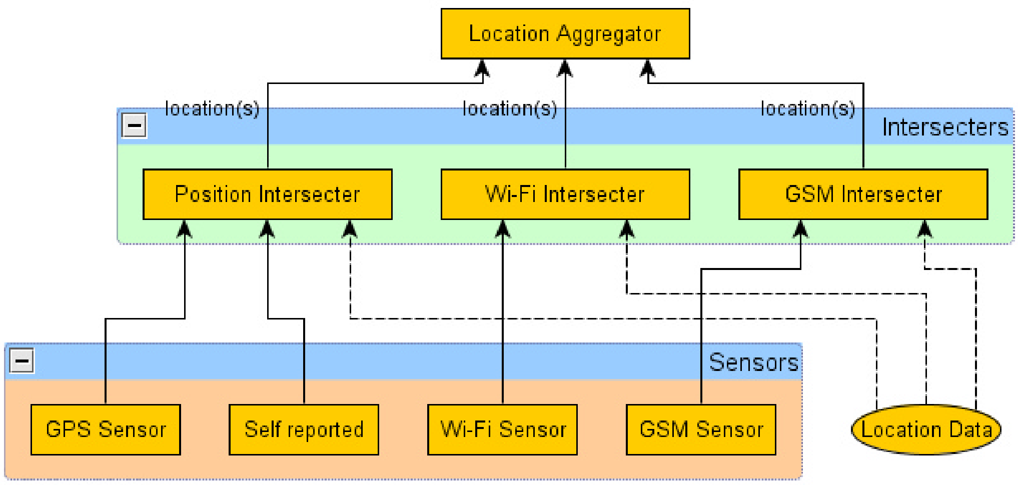

Figure 3.3: Abstract Location Model

The flow of data from the device sensors, through the model to the location-based application is depicted in figure 3.3. The model abstracts the notion of location from the underlying sensors by introducing a layer of (what I call) intersecters. Intersecters are methods that turn sensor-readings into location readings by intersecting the incoming sensor data and the predefined regions of location data. All triggered locations from the different intersecters are collected by the "Location Aggregator", which is the interface for the location-based experience.

Support for GPS positioning is indispensable for a general location model and the "GPS Sensor" in combination with the "Position Intersecter" provides just that. Similarly, "Self reported" positioning, where a user declares her position explicitly by setting it on an interactive map interface (c.f. page 44), could also be used to trigger a location in combination with the "Position Intersecter". The "Wi-Fi Sensor" periodically scans the network environment and outputs a list of nearby access points and possible other metrics like their corresponding signal strengths. This data is then analysed by the "Wi-Fi Intersecter" which triggers a location if its corresponding trigger condition is met. Similarly, "GSM Sensor" and "GSM Intersecter" would allow checking for locations that are defined based on GSM metrics like the currently serving cell ID.

3.3.3 Defining Location Data

Whatever combination of sensor and intersecter is going to be used in a location-based experience, the location data will have to be encoded and stored in some way. This section proposes location definitions for a selection of currently common positioning technologies and has been written with database-storage and realtime performance on mobile devices in mind. Appendix 3 (page 335) provides a non-exhaustive list of metrics for pervasive technology that can be useful as a reference for the purpose of defining location-data, but the most relevant metrics are also briefly convered in the following text.

3.3.3.1 Location Data for the Position Intersecter

This type of location data supports vector- and raster-based location definitions in absolute real world coordinates (WGS84). The vector-based location definition uses a few basic bounding shapes: 2D polygon, box and sphere. The polygon defines a closed area on the surface of the earth in latitude and longitude coordinates. Please note that altitude is disregarded for this shape as it is assumed that participants will usually dwell on the earth's surface. The box or sphere shapes might be used for those cases where elevation data is meaningful, e.g. when designing for skyscrapers, TV towers, or anything else that is located above the ground. In either case a vector-based location will be triggered if the tested geospatial coordinate is contained within the associated vector shape.

The raster-based location definition resembles Flintham's work on "Colour Maps" (Flintham, 2005) and is basically a two-dimensional look-up table. It allows defining locations by colouring geo-coded raster images. Each location has an associated colour-value which can be freely used in the image. The following steps are then needed to trigger a raster-based location:

1. Test if the coordinate intersects the area of the geo-coded raster image
2. Find out which pixel in the image it intersects and get its colour value
3. Use the colour value to look up which location it belongs to

3.3.3.2 Location Data for the Wi-Fi Intersecter

This type of location data is non-geospatial. Locations in Wi-Fi network space have to be defined using the measurements that are available from "Wi-Fi Sensor", namely:

- BSSID (the unique MAC address of the access point)
- SSID (the non-unique name of the access point)
- RSSI (the received signal strength indication)
- encryption information (off, WEP, WPA, etc.)
- mode of the discovered Wi-Fi device (infrastructure, ad-hoc).

A very basic location definition would apply a 1:1 mapping between measured characteristics and location, e.g. a location would be triggered whenever a specific BSSID is discovered. However, due to the fluctuating nature of the Wi-Fi environment, where access points might disappear at any time, this might not be a very stable definition. Location data for the "Wi-Fi Intersecter" should therefore utilise a composite of measurements, e.g. a list of BSSIDs per location or something slightly more elaborate, like the Rider Spoke Wi-Fi fingerprint location mechanism (page 208).

3.3.3.3 Location Data for the GSM Intersecter

This type of location data is non-geospatial. Similar to Wi-Fi, locations in GSM network space have to be defined using the measurements that are available from "GSM Sensor":

- MCC (the Mobile Country Code)
- MNC (the Mobile Network Code)
- LAC (the Location Area Code)
- CI (the Cell Identity)

These four measurements need to be combined in order to form a unique cell ID. It would then be possible to assign locations to a composite of one or more cell IDs. It has to be noted that implementing even just a simple GSM Sensor, which just reads out the ID of the currently serving network cell, is only possible on a few smart-phones like Nokia S60, Google Android, or Windows Mobile. Support for additionally reading out a list of surrounding cell IDs is available on even fewer devices, but can be achieved on some Windows Mobile phones, or by using specialised external hardware like GSM modems.

3.4 Challenges

Authoring location-based experiences is a relatively young, but not completely new topic. Several authoring tools are already available; they have been summarised in the literature review (page 86).

This thesis proposes extending the common approach to authoring by additionally revealing information about the infrastructure at authoring time. This core innovation is deceptively simple. In fact there are many deep & complex challenges involved in realising this and it is them that the main part of this thesis explores in depth.

A number of issues need to be addressed if we want to study and generalise the proposed extended authoring approach beyond lab-conditions. The issues that are immediately foreseeable can be grouped into the following categories: map data access, infrastructure data collection, presentation and workflow.

3.4.1 Map Data Access

Maps form the background of many location-based authoring views. Map data is copyrighted in most parts of the world and so access to that data is always going to be limited in some way. There are two general data-sources for this layer of information: printed maps and digital maps. Existing printed maps and aerial images can be used if their coordinates are known or can be measured. Such data can be digitised, geo-referenced and would then be ready to be used as a backdrop in an authoring tool. Digital maps or map-data can be accessed as allowed by the respective data provider. Digital map-data has the advantage that it is usually precisely measured, so that an overlay of infrastructure data on top of such maps will be as precise as possible. There are currently a number of online data providers emerging and it appears to be necessary to allow for some flexibility and to stay independent from any particular service, as the way to access such data is going to change over time and a provider might withdraw or change his service for different reasons at any time or might even cease to exist altogether.

3.4.2 Infrastructure Data Collection

Data about the wireless infrastructure needs to be collected before it can be processed and presented to an experience designer. This requirement will almost certainly lead to a survey activity that needs to be supported by the overall workflow. Surveys are time-consuming and it can be expected that any such activity will eventually face scalability issues, where the amount of work required to complete a survey is too big to be mastered by the available manpower. It is therefore interesting to see how the data collection process could be interwoven with other activities, so that the desired data is obtained as a by-product. Several of the reviewed applications, e.g. the Defcon WarDriving Contest (page 39), Treasure (page 49) or Hitchers (page 54), demonstrated how such interweaving could be accomplished. Moreover, it could be interesting to interoperate with existing communities that collect similar data and share it online.

3.4.3 Presentation

The presentation of information about the wireless infrastructure is anticipated to be predominantly visual; but other forms of presentation, e.g. audible, might be effective as well and should also be investigated. Choosing an appropriate visualisation style for the data is the key question for any information visualisation application. Knowing that the wireless infrastructure is inherently variable, its visualisation in a location-based authoring tool should be dynamic, rather than static. The amount of raw infrastructure data to be visualised is potentially huge, which could lead to information overload. Such a situation would contradict the claim of supporting experience designers in their work and so a usable abstraction of the data needs to be provided. This thinking is influenced by Shneiderman's visual information seeking mantra: *"overview first, zoom and filter, then details-on-demand"* (Shneiderman, 1996). It is anticipated that any single visualisation style alone is not enough. Many views on the same data are possible and it should be the user's choice to select the most appropriate for a given task.

On a general level, there are two main challenges that need to be addressed: to cope with uncertain data and to cope with incomplete data. Any collected data will be subject to uncertainties in the measurements, as for example the underpinning GPS data is subject to errors. Furthermore, the available data is never going to provide a complete picture of a situation, so that some degree of interpolation will be unavoidable. Finally, temporal variations in the data provide a further challenge, as visualisations may need to account for the ways in which coverage and accuracy vary over time as well as over space.

3.4.4 Workflow

In order to effectively support experience designers in their authoring process, every tool needs to fit into a coherent workflow. This workflow will include data collection in situ as well as working at the desktop. It will include a lot of testing and tweaking – quite possibly up to the last minute before the deadline – and will integrate a range of different devices with varying capabilities.

Building a location-based experience will also almost certainly involve some degree of custom programming for the different components. This necessary task can be eased by providing an environment that makes the exchange of data between components as simple as possible without oversimplifying it. This calls for a distributed computing architecture that links mobile and stationary work as well as data collection, content creation, testing and delivery. Such a distributed architecture will consist of a number of components that are loosely coupled over the Internet and can be grouped into three categories:

1. Desktop, for visualisation and authoring
2. Server, for data exchange, persistent storage and orchestration
3. Mobile, for data collection, testing and runtime

3.4.4.1 Desktop

The desktop tools visualise the invisible network infrastructure and implement the extended conceptual model as described in section 3.1.2 (page 121). They allow the designer to freely choose his view on the data, e.g. specify the map-area to look at and enable the layers of interest. Desktop authoring tools could be used to define locations in an interactive fashion and sync them with the server. Desktop visualisation tools could highlight certain aspects of the data and might even be incorporated into the actual experience to provide runtime visualisations.

3.4.4.2 Server

Servers act as global data-hubs for the whole distributed system. They provide flexible communication possibilities and host two main data services: flat file-storage, potentially with versioning, and a database with remote access. The file storage is used to store compacted incoming file-archives from the mobile devices which are the ground-truth of all other data and therefore need to be kept in a safe place. The data in these files gets processed and is then added to the database from which it can be accessed remotely. The web-service is deliberately designed as this two-step process for the following reasons:

1. The collected ground truth data does not change, but the post-processing applications will improve over time, so that the database will need to be regenerated from raw data.

2. This two-step model has a built in security gate which allows to control which data goes live and which does not.

3. Current network traffic on mobile devices is slow and expensive; therefore sending compressed file-archives is both faster and cheaper.

4. The burden of processing and filtering the collected data is taken off the mobile devices and transferred to more powerful computers.

In addition to these essential data related tasks, web-servers are of course ideal to host interactive web-applications that could be incorporated into the location-based experience as a front-end interface to the participant and as a back-end interface for supporting staff.

3.4.4.3 Mobile

The mobile tools allow collecting the ground truth data for the whole development process. Initial data collection is done by roaming the area of interest with a GPS receiver and a mobile device running a data collection software. The software time-stamps and logs the GPS information, i.e. the position, as well as the relevant network information, e.g. Wi-Fi access points or GSM cell IDs. Such software can be potentially run in parallel on a large number of mobile devices, which can speed up the data collection process. The collected data can be uploaded from the mobiles to a server from where it will be picked up and post-processed for inclusion in the database. The mobile devices will also be used to test the validity of the location triggers and the overall experience, and might be used to run the actual experience once it is finished.

3.5 Plan of Action

There is a gap between what experience designers expect and what wireless technology can provide. This gap needs to be filled with a better understanding of wireless technology and it seems timely to investigate this problem now. The following plan of action has been devised to guide the practical work of this thesis:

- Visualise the invisible wireless infrastructure at authoring time
- Study real users, i.e. professional designers of location-based experiences
- Get the designers more engaged with the technology that they are using
- Substitute assumptions with knowledge (or at least educated guesses)
- Do this in a way that fits the playful creative process
- Act on the principles of information design
- Facilitate the reasoning about locations
- Provide tools where necessary
- Support all of this from the beginning of the process rather than at the end

This work should not be understood as an improvement to technology alone, but rather as a tribute to the needs of those humans that will design the next generation of location-based experiences. Their needs will have to be analysed to inform the design of new tool prototypes, and these tools then need to be tested for their usability from early on in an iterative cooperative prototyping process (Bødker and Grønbæk, 1991: 200). The concept of user-centred design in general has been studied by many over the past four decades. Shneiderman provides a thoughtful and enjoyable high-level overview of this concept, which can be recommended for further reading (Shneiderman, 2002: 53).

This document now continues with the description of the practical thesis work. It is expected that the proposed extended conceptual model and the distributed workflow will facilitate the development of location-based experiences by supporting designers to better understand the ubiquitous wireless infrastructures they are relying on. Evidence for this is going to be sought in two large independent research trials which are described in the following two chapters.

The conceptual framework, as it was presented in this chapter, is later revisited for the discussion in chapter 6, starting on page 253.

4. First Study: Love City

This fourth chapter is about the *Love City* trial which has been designed and developed in collaboration between the performance art group Active Ingredient and the Mixed Reality Lab at the University of Nottingham as part of the INSCAPE project. The author's work on Love City was all about helping the artists in understanding a location mechanism which was based on the 4 UK GSM networks.

4.1 Overview

This chapter starts with an overview of the designers' tradition, the author's role in the process, and the general game idea. This is followed by a detailed description of the iterative development process that took the project from initial workshops to a fully functional graphical smartphone prototype. The development process can be broadly seen as having progressed through three phases: exploration, prototype development and final development. This chapter covers the critical period up to the smartphone prototype. Love City subsequently went through another iteration of work that transformed it from a graphical smartphone application to a server-based SMS application which was finally deployed to the public. Details of the final version can be found in appendix 4 (page 337).

The description of the Love City development process is based on the author's active participation throughout all phases of the project, and on the minutes, plans, reports, notes and emails that have been written during that time. It is also based on videos which have been recorded at a workshop and during a post-event semi-structured interview with the artists. Parts of this material have been previously disseminated in a book-chapter (Oppermann et al., 2007) and in a conference paper (Oppermann et al., 2008).

4.1.1 Designers' Tradition

Active Ingredient (51) is an arts company that specialised in interactive media and was founded in 1996. Based in Nottingham, the group comes from a performing arts background and has experience in interactive web and video productions. More recently, they traversed into the mobile location-based domain and started cooperating with the Mixed Reality Lab at Nottingham University in 2006. Their work *Heartlands* (page 370) makes use of GPS and heart-rate sensors to build a location-based multi-player experience that is controlled through one's heart-beat. Active Ingredient have been commissioned by Hewlett Packard to present Heartlands at the Game Developers Conference 2007 and also won the Nokia Ubimedia MindTrek Award in the same year.

4.1.2 Author's Role

Throughout the development process the artists have been supported by the author who led the technical development and provided them with mobile survey and desktop authoring tools. In addition to designing and programming these custom tools, the author acted as a general point of call for the development team throughout the development process, i.e. he was a facilitator. He inducted a team of four developers on how to develop with the Java-based Equip2 framework (page 332), so that they could write the game application. He also documented the participatory design process, especially the artists' experience with the tools.

4.1.3 General Description

Love City is a text-driven and location-based multi-player game for mobile phones which connects three English cities via one imaginary virtual "Love City". The artists envisioned the game as an "urban ménage a trois" that connects strangers over three cities during a month long "whirlwind romance" and thus brings these three cities closer together; diminishing former rivalries. Love City is played anonymously and uses text-messages as the catalyst of game play. It subdivides each participating city into a series of zones that players are put into based on their physical location.

Love City was conceived to be an imaginary place, combined and overlaid on the three cities taking part. It was initially developed to fit the brief from the Create and Connect Fund (52) to create a game which would involve participants from each of the major East Midlands cities in the UK: Derby, Leicester and Nottingham. The fictional Love City is comprised of a series of zones that relate to real world landmarks or regions common to these three real cities such as parks, train stations, universities or city centres. To progress in the game and to score points it is vital that players send messages to other players from other cities. They can only do this when they share a common space in the virtual world of Love City. While freely moving around their city and carrying out their daily routines, players can interact with the game at any time in order to see if they share their current space with players from other cities. In the moment of interaction, players are tracked by the game-engine and then placed accordingly in the virtual game space, where they might meet players from other cities. If this is the case they might exchange messages of love in an attempt to bond with each other. A bond takes place when a receiving player accepts a sent message. If now a player bonds with another player from the third city, meaning that bonds between players from all three cities have been made, this generates a "ménage a trios", which in turn creates a virtual offspring – a new inhabitant of Love City. Points are scored for making bonds and for making the offspring that populate Love City. Love City is a multi-player game that utilises a central server. Players participate with their own mobile phones, which they use to send and receive text messages. Constrained by this technological choice, the game's core mechanism is essentially SMS but adds to this the use of location. The artists wanted players to have some sense of their real location in relation to the other people playing. But this did not have to be exact, just enough to carry across the idea that they are sharing a similar space.

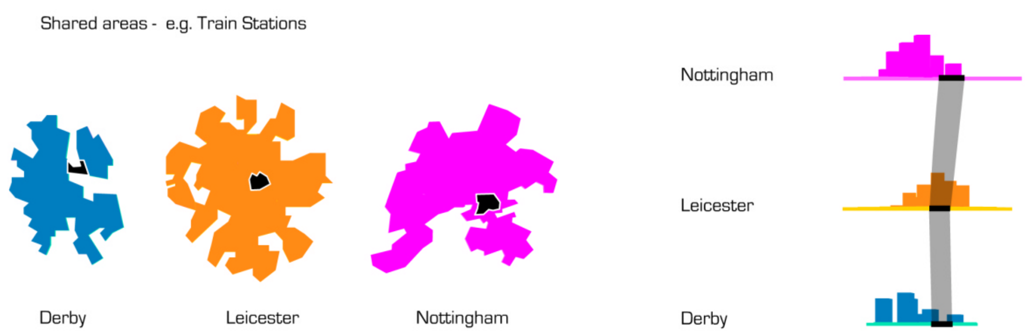

Figure 4.1: Defining shared areas in all 3 cities and linking them to a virtual space

Using GSM-based location services, individual players' phones could be tracked and positioned in loosely defined areas of the city they are resident in. Areas common to all cities, e.g. the train station, would be mapped on each network and in every city and then virtually overlaid to form a common space (see figure 4.1).

Love City is intended to serve as an imaginative link between real locations with common characteristics in the three cities allowing players to effectively share one space even when they are physically remote from each other. A player at home in the suburbs would know that they share that space with other users who are also in the suburbs, but not necessarily in their city. Similarly, a user who is physically in the Shopping Centre in Derby could also be present in the Shopping Centre in Love City, named in the game world as "Shopping's Soft Centre". In the same way a player who is in a shopping centre in Nottingham also shares this virtual space with the player from Derby. This mapping of virtual locations to real world areas by means of the GSM networks demands for a good understanding of the technical infrastructure in order to work; section 4.2.1 (page 151) elaborates this thought further, while section 4.3.1 gives a technical overview (page 162).

4.1.4 Game Walk-Through

While this chapter generally describes the crucial design phase of Love City that led the project from the initial idea to a fully functional smartphone prototype, the final SMS-based game emerged after another iteration of work. This section describes that final version from an end-user perspective.

The game starts when a player registers their details on the public Love City website. Interest for this has been generated through a marketing campaign which included several interviews on the local radio stations, interactive public screens in the city centres and through advertised launch events in the local arts venue of each participating city as well as flyers and stickers.

Figure 4.2: Large Screen in Nottingham Market Square (left), Registering players in Derby (right)

The interactive screens were prominently located and challenged passer-bys to send text messages to the big screen, which also showed the website URL (figure 4.2, left). The launch events were advertised in advance by the respective venues and provided interested parties the possibility to easily get into the game. Members of the public would arrive at the event, listen to a talk by the artists and then sign up for the game with the help of an assistant (figure 4.2, right). This allowed addressing the potentially off-putting elements of the game, answering questions about cost, privacy and making sure players got their vital details correct, especially their mobile phone number. The events also created a social meeting space where people could have a drink, listen to music and chat to each other in person.

4.1.4.1 The Public Website

Once people got attracted by the marketing campaign, the Love City website was the most visible side of the game and gave players and members of the public an insight into what was happening in Love City. It also provided a detailed information section where interested parties could read an introduction to the game as well as the “frequently asked questions” and “terms and conditions”. The website provided details about the city, the players, and the story of Love City. Through a visualisation it also provided a way in which the game could be observed and play could be enhanced.

The central component of the website was an interactive heart-shaped map which showed the areas of Love City as well as the players and their virtual offspring inhabiting them (figure 4.3). The artists’ impression of Love City was inspired by heart-shaped pictorial maps of the mid 20th century from artists like Frederic Holland, or Jo Lowrey; examples of their work can be found in (Harmon, 2004: 52-55). Visitors to the website could pan and zoom the map using their mouse. Visitors could also highlight the different game locations by selecting them from the drop-down menu in the upper right hand corner. Each location in the game was represented by an iconic photograph. Figure 4.5 shows the photographs for two of the locations which have already been mentioned: the shopping centre and the station. These images have been taken by local photographers as part of a wider campaign of activities that took part to complement the game.

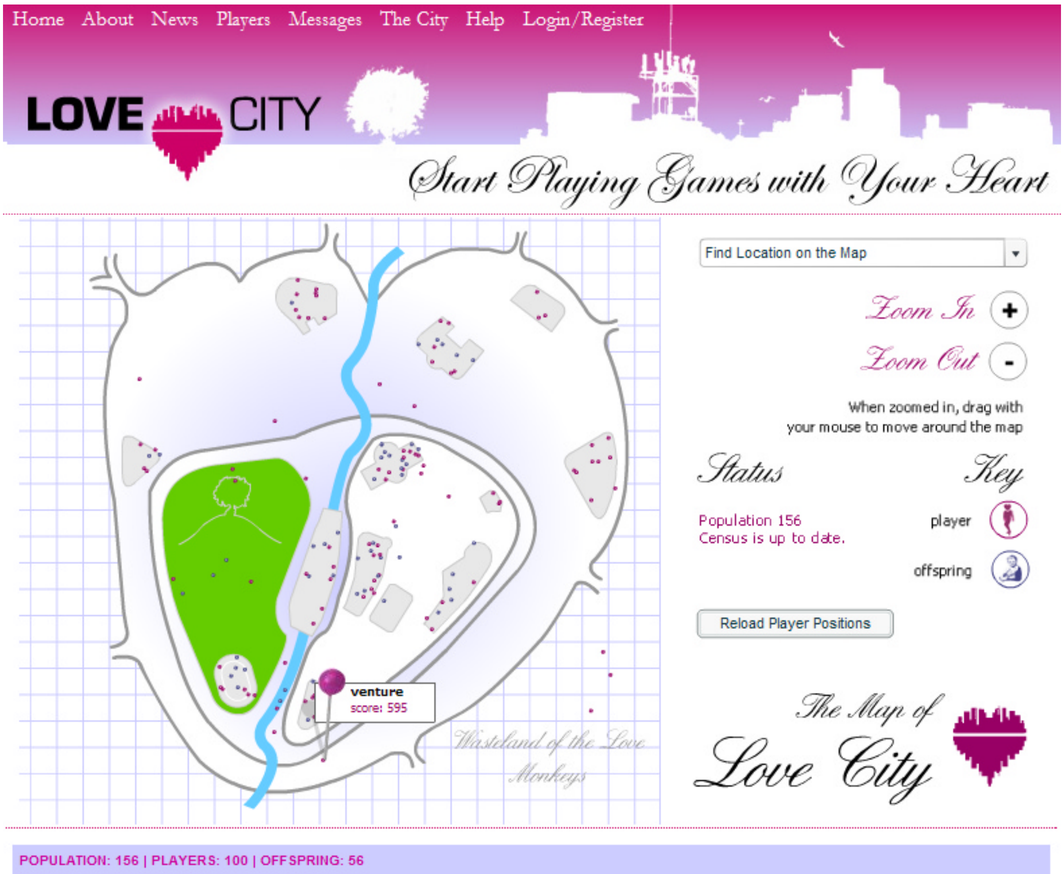

Figure 4.3: Love City website, pin-pointing a player on the map

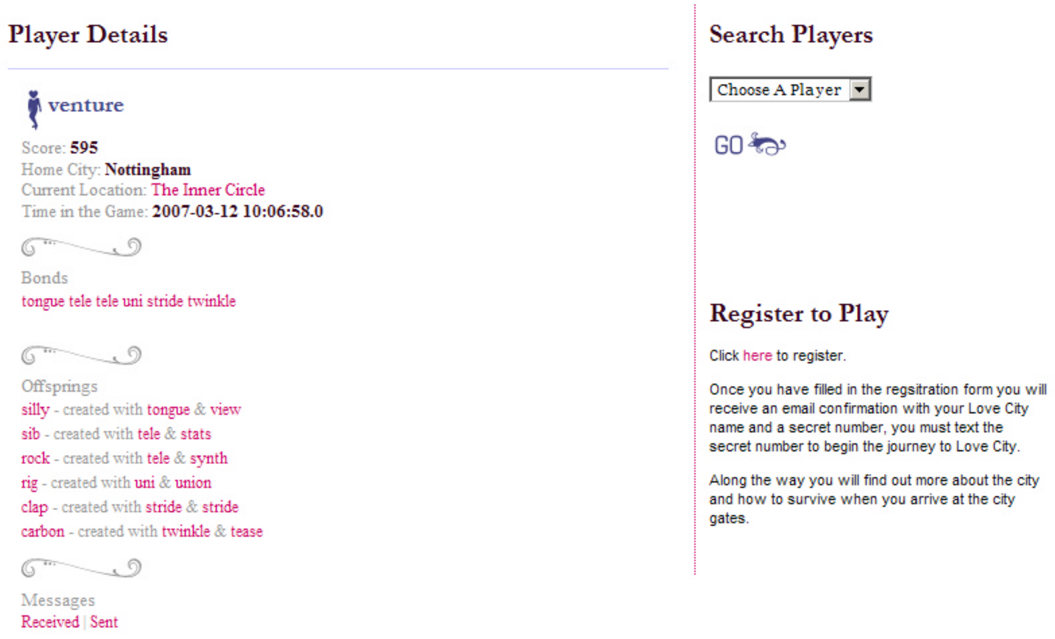

Figure 4.4: Love City website, showing player details

Figure 4.5: Photos for Locations “Shopping’s Soft Centre” (left) and “Tender Station” (right)

Each of the dots on the map in figure 4.3 represents either a real or a virtual player, i.e. an offspring that was generated as result of a successful bond between players from three cities. Hovering over the dots revealed their name and current score. By clicking on the dots in the map one was taken to the player details page, which revealed their score, home city and current location, all their bonds (accepted links with other players), offspring (awarded virtual characters) as well as links to all messages sent and received (figure 4.4).

4.1.4.2 Getting In and Out of the Game

The Love City website was the entrance into the game. Players first registered their details (mobile phone number, email, name, etc.) on their own or with the help of an assistant at one of the launch events. The system then sent them a confirmation email with further instructions and their game name, which they were told to use instead of their real name to protect their privacy. In order to complete the registration process a player then had to use their registered mobile phone and text the “JOIN” command to the game’s short code number. This double opt-in process across different communication channels has been established to ensure that the registered details were correct and that players were ready to play on their mobiles once they had completed the process. It also prevented spam, i.e. people accidentally or purposefully being signed up for the game by third parties.

The tracking of an individual mobile phone was only enabled after an explicit consent (the “JOIN” statement of a registered individual from their phone) – this was required by law. It was also required to make the opt-out as simple as possible and thus a player could stop being tracked at anytime by sending the "STOP" command to the game. Alternatively they could phone a hotline or send an email and be immediately taken out of the game.

4.1.4.3 Initial Messages

After sending the "JOIN" command the player would start receiving introductory messages to their phone. These messages aimed to make the player comfortable with the available interfaces and get them into the game. A secondary purpose of the initial messages was to bridge the time for those that signed up first as the game only really started once the start-events in all three cities were over to ensure that there are players in all participating cities. Examples of these initial messages can be found in table 4.1.

Your journey has begun. You are now crossing the turbulent sea of serenity that surrounds Love City. We'll be in touch soon. Good Luck.
Your journey to Love City is not over. You have now entered the wasteland of the love monkeys, be on your guard and await further instructions. www.lovecity.tv
To enter Love City's gates text WHERE in reply to this message and then WHO to find other players. Visit www.lovecity.tv to see where you are in the city.

Table 4.1: First messages received on the phone

4.1.4.4 Messages Sent by Players

By sending the WHERE command, the player would receive a description of their current place (table 4.2), similarly the WHO command would list the nearby players.

Shopping's Soft Centre: All products have personality; they chat up shoppers so they pop them in their trolleys. No one gets left on the shelf in Love City.

Table 4.2: Place description, received after sending the WHERE command

The game was intended to encourage users to move around their cities, find new places and interact with new people. After finding a nearby character, the player could send them a message. Table 4.3 lists some typical player messages.

From train: love. You must give a little to receive a lot. Will you give me a little of yours?	22 February 2007	18:57:57
From tele: The icy wind blows warm greetings into your heart.	1 March 2007	19:18:04
From twinkle: maybe a miserable sunday in the east midlands but all is love and blue skies in love city! here's sending you some of both!	4 March 2007	14:53:17
From treat: it's so wet and grey today...you fancy a hot chocolate by the fire?	5 March 2007	11:05:44

Table 4.3: Player generated content

4.1.4.5 Authored Messages

In addition to the user generated content, players could also receive messages authored by the artists in several ways:

- Players could be individually addressed by the artists
- The “Love City News” was sent out daily (table 4.4)

Love City News: The walls of the city have been destroyed in the riots. Love Monkeys have raided the city. Top Scorer is TELE.	3 March 2007	13:49:58
Love City News: The tree on Lovers Hill has been surrounded in flowers and messages for the people who lost their love in the street riots. Top Scorer is TELE.	4 March 2007	10:33:28

Love City News: Now that the city walls have come down, the decay and rubble is being cleared, it is time to believe in utopia, it is time to heal the city.	5 March 2007	10:10:10

Table 4.4: Examples of the daily "Love City News"

- The artists could send location specific messages (table 4.5)

Lost Souls Bar *- From Charlie: I am searching for my friends. I live in the underworld, I track the lost souls destroying themselves throughout the night. Can you help?*
Dreaming Suburbs *- From Ramona: I steal things to leave on the doorsteps of the broken hearted, the jilted and unrequited lovers. So keep an eye where you tread.*
Infirmary of the Beloved *- From Ramona: I heal hearts by day and break them at night. Let me give you hope, because true love really does exist.*
Lovers Hill *- From Billy: I have been waiting for you. I live on Lovers Hill in a van behind the trees, I want to tell you about Love City.*

Table 4.5: Authored content sent to players at specific locations

4.1.4.6 Nearing the End

Later on in the game, players started to receive messages from a fake player called Ramona which was authored by the artists and who started asking questions about two other fake players: Billy and Charlie. Over a number of days these players sent messages back and forth; revealing details and information about the new world of Love City. These messages were usually sent as local messages, as in table 4.5. The game climaxed when this narrative ended with the revelation of a blog which invited the players to a final party in a real bar. At this party, the players had an opportunity to meet each other in the flesh and exchange their experiences. The most active players from each city were finally announced as winners at the party.

4.1.5 Public Staging

The public outing of Love City was launched with an event party at the Broadway Cinema & Media Centre in Nottingham on 14th February 2007 (Valentines Day), followed by similar events in Leicester and Derby on the following days. The game lasted for a month and attracted more than 100 active participants from all three cities.

4.2 Development Iteration 1: Initial Exploratory Phase

Having described the final experience from a user point of view, this document now provides a detailed decription at the iterative development process that led to this experience.

The main focus at the start of the project was to verify the feasibility of the artists' game design and come up with a strategy to implement the game. The artists have been provided with a prototype of a mobile phone network survey tool and inducted on how to use it. Their feedback after having used the tool served as input for its further development. The artists have also been introduced to a prototype of a stationary tool which could visualise the collected data back in the studio. The artists provided feedback on how it could be turned into an authoring tool that they could see themselves using to author the Love City locations.

This early phase of the project was organised as an interleaved series of workshops and tests between the artists and the developers. An internal project manager and an internal technical manager (the author) were appointed to ensure timely progression through the schedule throughout this phase.

4.2.1 A Need for Understanding the Environment

The artists were aware that mobile phone network cells can be used as a way of locating where people are in the city and wanted to explore the use of this location mechanism for Love City. In particular, they wished to explore the ability of some phones being able to look-up their own cell IDs as a potential way of locating players. This meant however that to realise the game, the artists needed to have a good understanding of the cellular networks available in the three cities where it would be played. From the author's point of view, the implementation of the game provided an ideal opportunity to iteratively develop an authoring tool for location aware applications that visualises the wireless infrastructure in a way that meets the needs of creative professionals.

We started the development process by getting the artists engaged with the technology and giving them something to work with. They were set up to conduct a survey of the GSM networks' cellular infrastructures. Based on the survey results they refined the design of their game by identifying the key zones for Love City and explored how these could be mapped onto the underlying infrastructure in each city and on every network.

4.2.2 Network Surveying and Initial Visualisations

As the artists wanted to enable as wide participation in the game as possible, play needed to be supported over the major GSM networks in the UK: Vodafone, T-Mobile, Orange, and O2. For these wireless infrastructures to be visualised in an authoring tool, data about them must first be obtained. In some cases it is possible to get access to data that has been collected by other parties, e.g. through Wigle for geocoded Wi-Fi data (page 79). Other projects such as Place Lab (42) or MobiLife (53) started sharing GSM data through online services but unfortunately never gained enough momentum to attract such an active user base as Wigle did. So in our case relying on the available data was not an option as there is a lot less wardriving data available for the GSM network than there is for Wi-Fi. This meant that we had to define a workflow for collecting our own infrastructure data.

The first step was an infrastructure survey exercise carried out by Active Ingredient, with the intention of working out where in the physical world different cell IDs can be found, i.e. establishing the relationship between cell IDs and physical locations. The artists were provided with a mobile prototype that collected time-stamped and geo-coded GSM network data at a rate of about 1 Hz. Four Nokia S60 2nd Edition mobile phones were used by the artists, each capturing information about one of the GSM networks with one of the phones also logging the GPS position data that was coming in from an external Leadtek

Bluetooth GPS receiver (with a SiRFstar II chip). The complete mobile equipment is shown in figure 4.6, left.

Figure 4.6: Survey equipment (left), Surveying the streets of Nottingham in a car (right)

The GPS data has afterwards been applied to all other phones' log-file data by using a custom post-processing tool which correlated the samples based on their time-stamp. The survey software has been implemented in Nokia Python for S60 and the post-processing tool was written in C++. One city – Nottingham – was surveyed first to explore the feasibility of tracking players' location through cell IDs and to gain an understanding of the granularity of the network cell infrastructure. The artists first planned their routes on paper, then took the equipment with them in a car (figure 4.6, right) and drove through Nottingham in ever decreasing circles. In one day in 5 hours they managed to map most of the inner city area of Nottingham. On another day one of the artists took the kit out on his bicycle, allowing him access to some parks and the city centre.

The collected data was given to the author to create some preliminary visualisations of the infrastructure. The initial version of the authoring tool, running on a desktop PC, was used to create visualisations of the network cells. These images were sent back to the artists for them to reflect on the infrastructures of the different networks and to think about how the city could be subdivided into virtual zones based on this data.

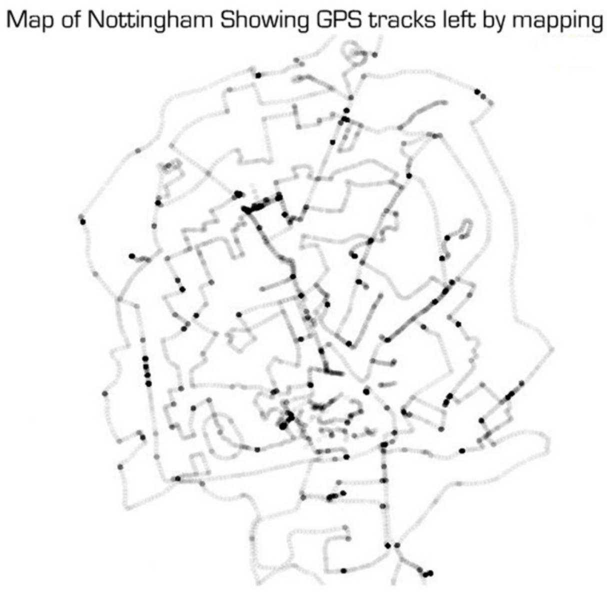

Figure 4.7: GPS samples from the survey (6 x 6 km)

Figure 4.7 shows the GPS positions where cell ID samples were recorded, i.e. the trail of the survey. The shading of the trail shows the speed of movement at which the data has been collected, where dark means slow or stand still (and many dark areas in the figure show traffic junctions) and bright denotes a continuous movement.

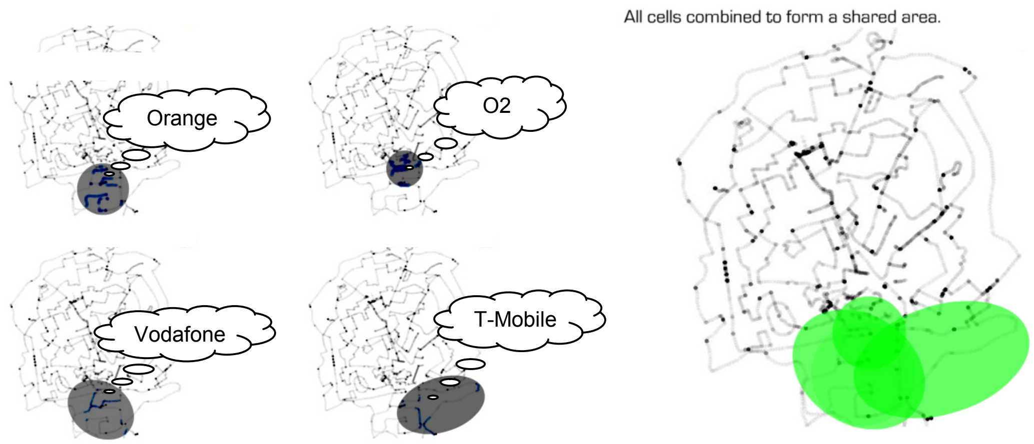

Figure 4.8: Visualisation of cell samples of the closest cell to Nottingham train station on all 4 GSM networks (left), manually combined view to estimate the spread of the shared area (right)

Figure 4.8 (left) shows the nearest cell ID to Nottingham train station on each of the four GSM networks respectively. The highlighted ellipses mark the areas that one could possibly expect to be connected to the corresponding cell IDs. The area covered by all maps in figure 4.7 and 4.8 is 6 x 6 km so the approximated shared area around the marked area is about 3 x 2 km. This visualisation helped the artists in better understanding the level of granularity they were going to face in their project. At this stage it was already clear that the granularity of the networks was much lower than the artists originally envisioned and that an area "around the train station" could potentially be as large as 3 x 2 km.

4.2.3 User-Feedback through Design Workshops

A first workshop was held to discuss the gathered data and the initial visualisations. The survey had verified that cell IDs can only distinguish between quite large areas in the cities. From the artists' point of view a key benefit of the visualisations at this stage was that they helped them to understand the different cell sizes and enabled them to think about their effect on the game. The survey also prompted them to consider the stability of cellular locations which led the discussion to the phenomenon of cell breathing (where cell coverage varies through the day depending on the numbers of users currently in it and neighbouring cells, and atmospheric conditions). Ideas were proposed about how the game could be designed to cope with unstable cell boundaries including using past tense in the game narrative (for example "you have been seen at the park" instead of "you are at the park"[16]) and using history of cells to inform location as previously demonstrated in the Place Lab Reno project (Smith et al., 2005).

Following this workshop the artists surveyed the remaining two cities using the same process as for Nottingham. When GSM cell data had been collected for all three cities a second workshop was held with a focus on identifying the conceptual places in Love City and mapping these to the underlying infrastructure. The process of physically exploring the cities in order to survey the infrastructure had helped the artists to not only gain an understanding of the cellular networks but also to enhance their design ideas in relation to the city environments. They were able to get a better feel for the actual physical nature of the cities and identify popular places and landmarks, thus helping them to refine the set of shared regions that would make up the imaginary Love City.

---

[16] more on this in section *"6.2.3.1 Understanding Location in the Design"* on page 267

To practically explore the process of how these regions can be mapped onto the underlying infrastructure it was decided to first use paper maps. The artists spent part of the workshop marking conceptual places including train stations, shopping centres, parks, football grounds and car parks on paper maps. Figure 4.9 shows an annotated map for Derby. The highlighted regions represent places that were deemed to be meaningful for Love City.

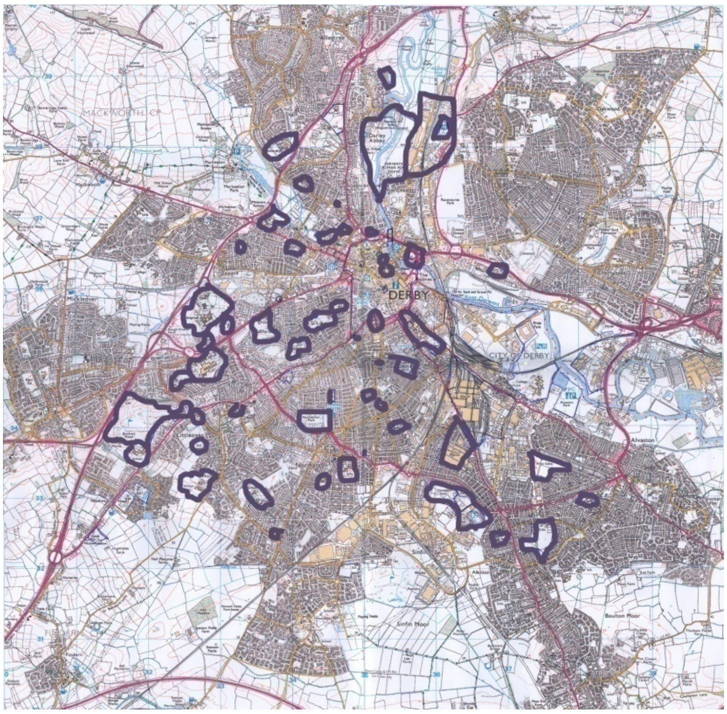

Figure 4.9: Paper map of one of the three cities (Derby) annotated with regions

This process led to further refinements of the mechanics of the game. It was decided that it would be useful to be able to distinguish between places that are far away from the three cities and those that are just slightly outside of the inner city regions that are part of the game. The latter were termed 'wasteland' and became incorporated into the game narrative. Discussions also continued on how best to design the game to handle the imprecision of the cell based positioning. It was decided that there should be some ambiguity in place names and that different containment strategies should be adopted for deciding if a player is in a particular region depending on its type. More specifically, players should be considered to be in smaller specific places, such as a shopping centre, only when the probability for them being there is high (containment should be exclusive). Other less specific places like a neighbourhood or parkland should be treated in the opposite way (inclusive containment). This means players could be reported to be in those places even if the probability of them being there, e.g. due to imprecise positioning, was rather low. The reasoning behind this was that definitions like "neighbourhood" are so blurry in their own right that it would not matter whether a player was in or near one. Whereas the more specific definitions like "cinema" would make it more obvious if there was a location mismatch.

The mapping activity and its resulting visualisation also prompted the artists to suggest that it would be very useful if the authoring tool was able to show not just raw sample data but also visualise the estimated edges of cells, i.e. depict the block region covered by a cell in some way. The author mentioned that there was no such thing as an accurate cell edge, as phones in GSM networks are dynamically allocated to available cells in their vicinity, but the artists insisted on their idea. The author finally suggested that he could visualise the inner workings of the mouse selection mechanism of the visualisation tool, which utilises the geometric Voronoi algorithm (Okabe et al., 2000) to decompose the two-dimensional screen space into polygonal regions for each cell sample, and it was decided to try this out as an additional visualisation in the tool.

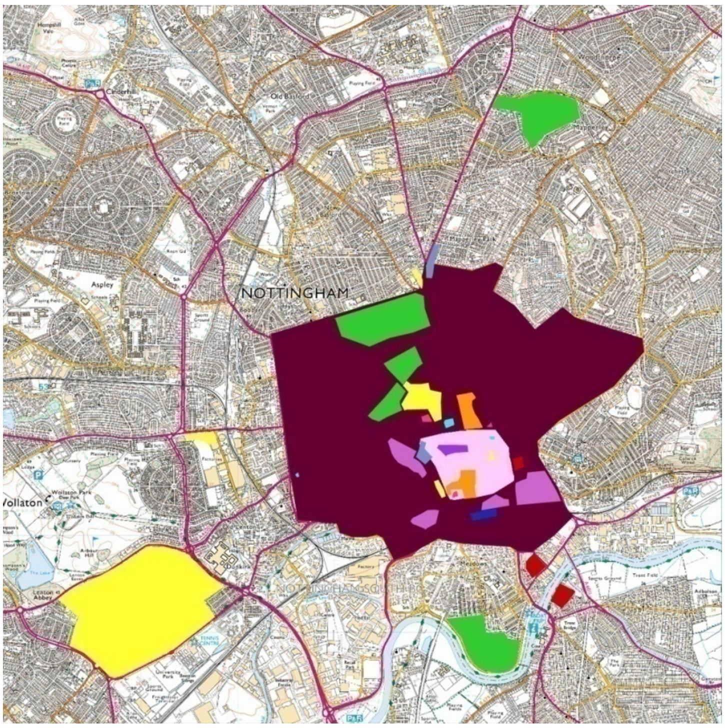

Figure 4.10: Regions of different scale marked on a digital map of Nottingham in Photoshop

After the second workshop the artists continued with the content prototyping process. For their next draft they used digitised maps in Photoshop to map the zones of Love City with different colours. Photoshop's layering capabilities allowed grouping the areas in sets. The high resolution background map and Photoshop's zoom-feature allowed working at different scales – from the most zoomed in view allowing details in the city centre to be annotated, to a zoomed out overview that displayed the whole of the city. The resulting content map for Nottingham is depicted in figure 4.10. It can be seen that the artists marked places at different zoom levels, i.e. some places are very big and some places are very small and detailed. This reflects the artists' earlier thinking of containment strategies and granularity of place. The figure also shows the artists' use of colour for tagging locations, e.g. green is used for parks and yellow for universities.

A third workshop was then hosted where these electronic maps were discussed and suggestions were made about how the infrastructure data collection and visualisation tools should be further developed to support the authoring of the Love City game.

4.2.4 Suggested Functionality

The artists' feedback at the third workshop concluded the initial project phase. This section draws together the observations this far and summarises the functionality that would be desirable in tools that visualise the wireless infrastructure for developing location-aware applications.

4.2.4.1 Infrastructure Collection Tools

The tool for automated logging of network cell IDs with their GPS positions was found to be very useful by the artists. It allowed them to capture infrastructure data by simply moving through a city – driving, cycling or walking. Furthermore, this exploration process enabled them to become more intimately familiar with the three cities where the game would be played and to refine its design – looking for landmarks and choosing the most appropriate shared regions for Love City.

To help with the next stages of the game development, the artists suggested that the network data collection tool should be extended to ***support verification of***

regions that have already been defined. That is, after initial authoring has been done, the data collection tool should – besides logging more data – show the current region that the system thinks the user is in (e.g. “park”, “train station” etc.) based on the current state of the authoring. This would allow the artist to verify their regions in situ, while at the same time collecting more data for a better visualisation. Both tasks, gathering data and checking current assumptions, are required in order to improve the authored regions. Carrying them out in parallel seemed to be an obvious time-saver as well as a productivity enhancer as the data collection task on its own was already described as rather dull and boring. It was suggested that ***annotation*** of some form should be supported to allow the person on the ground to take note of any errors or inconsistencies at game design / content level (e.g. reported location is in the park when actually in the shopping centre). It was also suggested that there should be a mechanism to ***manually indicate position or tag location*** for places where there is no GPS coverage. For example, indoor shopping centres often have dedicated GSM cells on the inside to allow shoppers to use their mobile phones while shopping. But as GPS is not available inside buildings, the discovered IDs of those indoor cells cannot be automatically tagged with a position reading. This makes it harder to utilise these cell IDs with the current authoring workflow, although they would be very relevant to the game design due to the number of people passing through them and the inherent constraints of the place they are in. Support for manually tagging locations would allow such spaces to be included in the game as well.

4.2.4.2 Infrastructure Visualisation Tools

The first iteration of the stationary visualisation tool was capable of loading a file of logged data for a particular cellular network and producing an interactive visualisation of the raw sample data. The infrastructure data could be overlaid on a map (figure 4.11, left) or the map-layer could be turned off (figure 4.11, right). The software continuously tracks the position of the mouse pointer, searches for the nearest cell ID sample and then highlights all samples with the same cell ID, effectively enabling the user to view the recorded coverage of different cells as they move their mouse cross the screen. The user can select one or more cell IDs to highlight them at the same time (figure 4.11 shows a selection of 18 cells on the Orange UK network in Derby).

Figure 4.11: Raw visualisation of samples from 18 cell IDs in Derby (left), without map (right)

This visualisation was seen as a good starting point by the artists, as it enabled them to appreciate the layouts of different networks and thus refine their ideas about how the game could be structured. However, the tool needed to be extended in a number of ways in order to allow the artists to use it themselves to author Love City. It was suggested that a number of ***different views per layer*** would be useful.

For the physical layer it was proposed that authors should be able to ***load custom images***, such as an annotated map, into the background. For the infrastructure layer, in addition to the raw sample view, ***an averaged visualisation of the network coverage of different cells*** should be provided as the existing raw visualisation was received as visually overwhelming. This would assist authors in making judgments about which cells should be used to define a particular conceptual region such as the train station. For this task it was also thought to be helpful to be able to ***visualise the infrastructure data for all networks*** (Vodafone, T-Mobile, etc.) ***simultaneously*** in an overlaid view. Being able to use these multiple layers of information was seen as essential to the authoring process, however, the artists also wanted to be able to ***turn layers on and off*** as appropriate to the current task, so that they could focus on the required details. Finally, based on their experience of defining regions on paper maps and on digital maps in Photoshop, the artists requested that the interface should support ***viewing of the data at different scales*** and that they should be able to ***give*** their ***own names to regions***, possibly also distinguish them with different ***colour codes***.

On a more general note the artists also suggested that the synchronisation of data between mobile devices, the server and the visualisation tool should be made as easy as pressing a single button.

4.2.4.3 Summary of Feedback

The following table summarises the artists' suggestions for improvement at the end of the first iteration.

Collection Tool	• Verification while collecting • Annotation • Manually indicate position / tag location

Visualisation Tool	• Different views per layer
	• Averaged view
	• Improved Layer handling:
	    o Toggle layers on/off
	    o Simultaneous overlay
	• Different scales
	• Support custom images
	• Colour codes
	• Naming

General	• Easier data synchronisation

Table 4.6: Summary of suggested functionality for authoring tools

4.3 Development Iteration 2: Towards the Prototype

New versions of the infrastructure collection and visualisation tools were developed based on the artists' feedback. The main focus of this second stage was then on allowing the artists to author the content regions of Love City for themselves and to engage them in a process of testing and verification. At the same time the technical team started the development of a first mobile prototype of the game.

4.3.1 Technical Overview

This section provides a brief account of the location mechanism and the technical infrastructure which allowed building the first prototype. The distributed system architecture as used in the second prototype and in the final SMS version is described on page 337.

4.3.1.1 Location Mechanism of the Prototype

The location mechanism of the prototype works by relating one or more full cell IDs (as described on page 78) to the pre-defined regions of Love City. For example, the following cell IDs[17] all relate to the Love City football stadium:

- 234_33_655_25408 (on Orange in Derby)
- 234_33_1243_34037 (on Orange in Leicester)
- 234_33_37_29061, 234_33_37_29070 (on Orange in Nottingham)

It can be seen that a game region can consist of several cell IDs, as is the case with the Nottingham example. But to avoid ambiguity each cell ID can only belong to a single game region.

4.3.1.2 Content Mapping

The process of assigning cell IDs to game regions is what we call content mapping, and it has to be done for each game region on every network and in every city. The resulting data-file is transferred to the graphical mobile prototype where it is subsequently used to continuously look-up the player's location by comparing the current network cell against the list of pre-mapped cells. For example, if the player's mobile phone would be logged into cell 234_33_655_25408 (as in the above example), the prototype would locate that player in the football stadium. Figure 4.12 summaries the content mapping process. It shows a map of Derby with the football stadium of Derby located on the right side of the map in the open space (left).

[17] Format: MCC_MNC_LAC_CI

Figure 4.12: Map of Derby (left), overlaid cell centres (middle), highlighted cell (right)

The time-stamped GPS/cell ID data from the network surveys is used for geo-referencing all sampled cell IDs. For one of the visualisations, all position data per cell ID is accumulated and averaged, resulting in an approximated position of the encountered cell centre. This map is overlaid with geo-referenced network cell ID data, in this case the averaged data for the Orange UK network which is visible as a set of point (middle). The point set is used as input data for the Voronoi algorithm which decomposes the two-dimensional space into a set of polygons that are used for finding the sample that is closest to the mouse pointer. Each polygon in the output set has the property that it entails all positions around its seeding point that are closer to this than to any other point in the input set. The resulting polygon-mesh is shown in figure 4.16 (right). The highlighted area shows the polygon that was created around the averaged position sample of cell ID 234_33_655_25408.

4.3.1.3 Preliminary Distributed System Architecture

The Love City prototype utilised a preliminary client-server architecture which was divided into three main tiers: phones, server and desktop (figure 4.13).

Phones

We have built two smart-phone clients using Python for S60. The first client was used for surveying the networks during the development phase and uploaded its log-data to the server. It also downloaded the latest authored content regions from the server and provided the artists with a way to test their content regions in situ, while collecting more data. The second client was used for rapid prototyping of the mobile user interface and user testing.

Server

At this stage the server component acted as a central file-store for the collected survey logs and authored content regions. It simply consisted of a dedicated directory on a server which was accessed via the file transfer protocol (FTP) both from the desktop and the phone clients. The mobile phones uploaded their collected log-data in ZIP-archives and downloaded the latest content as text-files. To avoid filename conflicts that could arise from storing multiple log-files from several phones, a naming scheme had been applied which included the phone's IMEI number and a timestamp into the filename of the archive.

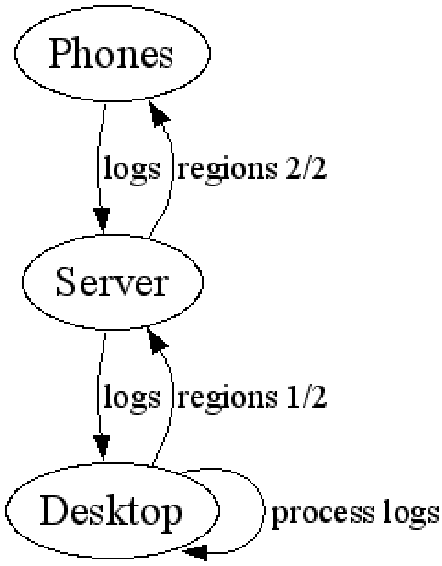

Figure 4.13:
Data flow between components
(source: (Oppermann, 2007a))

Desktop

The desktop component downloaded the archived log-files from the server, processed them and then presented a graphical user-interface which allowed the artist to visually define the content mapping as described in the previous section. The resulting regions were uploaded back to the server to be used by the mobile phones.

4.3.2 Authoring of Love City Regions

This section begins by introducing the new desktop authoring tool, followed by a description of the actual authoring process and the artists' feedback during this phase.

4.3.2.1 Overview of the New Visualisation and Authoring Tool

After starting the authoring tool it is initially set to a 40 x 40 km overview of the game area of Love City (figure 4.14). The initial view shows a splash-screen background that features the project logo instead of a real map but its overlaid sample data is otherwise fully geo-referenced. It shows the averaged sample data for the Orange UK network in the cities of Derby (top-left cluster), Nottingham (top-right) and Leicester (bottom). A predefined set of different views is provided and mapped to the function keys for quick and easy access. These predefined views include the Love City overview (as shown in figure 4.18), an overview for each of the three cities (between 6 x 6 km and 10 x 10 km), a close up view for each of the cities (between 3 x 3 km and 6 x 6 km).

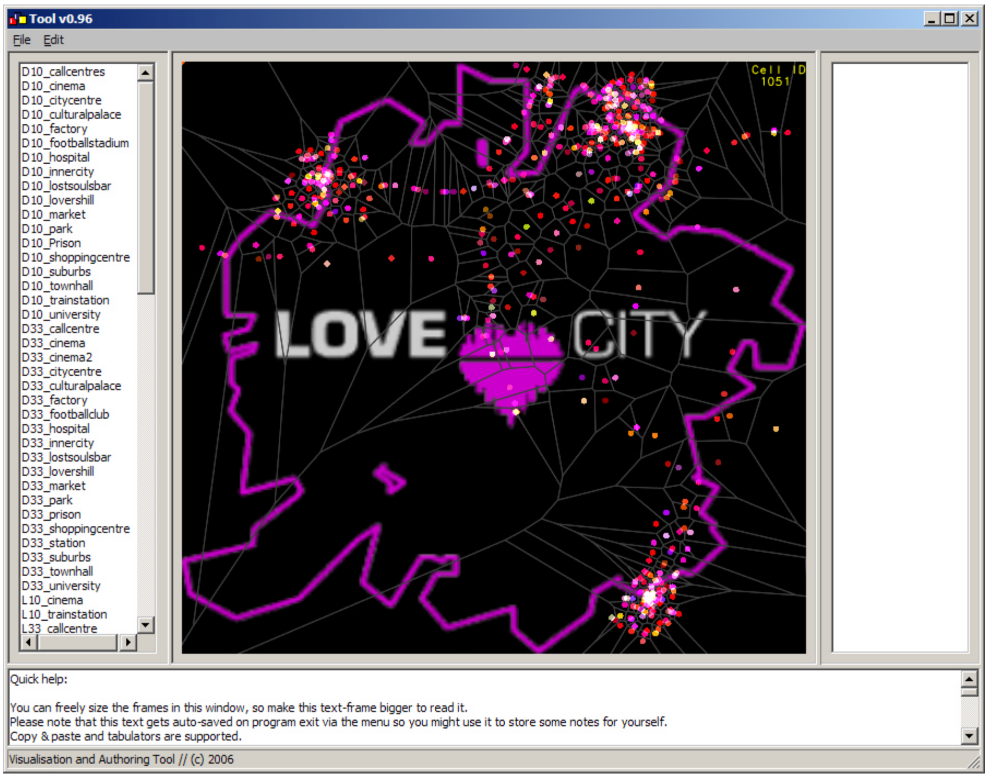

Figure 4.14: Overview of Love City game area at 40 x 40 km

Further keyboard shortcuts allow controlling the visibility of the map background, the visibility of an optionally overlaid vector mesh (that approximates cell outlines), and the transparency of that vector overlay. By pressing one of the map shortcuts, the view changes to the predefined positions, showing an area depending on the cities' scales, e.g. figure 4.15 shows Nottingham at 6 x 6 km.

Users can select to view the raw point samples of the infrastructure by loading the appropriate network-data file through the file menu. In the same way the cell data for the other networks can be loaded and visualised. The overlaid points show the averaged observed cell positions of the mobile phone network. This means that each dot in the visualisation represents one cell ID and its position is the averaged position of all encountered GPS readings for which the telephone was reporting to be connected to that cell ID. The overlaid vector mesh visualises a rough and strictly geometric approximation of the possible extents of cells. It is generated by applying the aforementioned Voronoi algorithm to the set of points of averaged observed cell ID positions. In this view it is desirable to have small polygons as large polygons usually indicate a badly sampled area rather than very big cells.

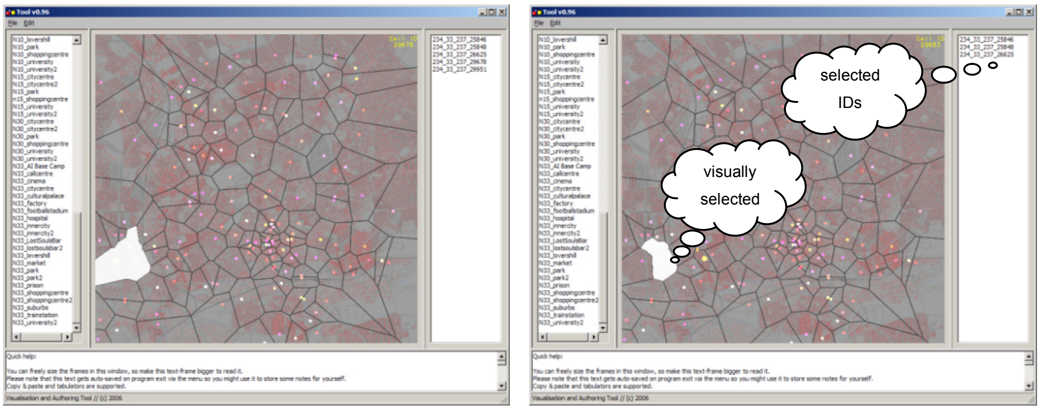

Figure 4.15: Initial result of region selection (top) and revised version with fewer cells (bottom)

Users can directly interact with the visualisation to define regions. An initial selection is made by pressing and holding the mouse button and then hovering over the area (figure 4.15, left). The selected cells are highlighted in the visualisation and also appear as ID-strings in the panel on the right side of the window. If the initial selection is not satisfactory the user can change or discard it, otherwise the selection can be stored and will subsequently appear in the listview to the left of the application window. Table 4.7 summarises the available commands and functions of the authoring tool.

Command	Function
F1 / F2	View Derby / Derby close
F3 / F4	View Leicester / Leicester close
F5 / F6	View Nottingham / Nottingham close
M	Background map on/off

V	Vector overlay on/off
1..0	Set opaqueness of vector overlay between 10% and 100%
Ctrl-O (Menu: File)	Open file of processed survey data
Click and hold mouse-button, move mouse	Select one or more IDs while moving the mouse (clears current selection and starts a new one)
Shift-Click, move mouse	To add to current selection
Alt-Click, move mouse	To remove from current selection
Alt-R (Menu: Edit)	To store current selection as a region (will ask for a unique name), region will appear in listview and is saved to disk
Ctrl-C (Menu: Edit)	Copy current selection of IDs to clipboard
Double-click on region names in listview	Delete the region (will ask for confirmation)

Table 4.7: Commands and functions of the Love City authoring tool

A new selection can be started (by just clicking, holding and hovering again) or the existing selection can be modified by holding the Alt key for taking cells out or holding the Shift key for adding cells. A revised selection is shown in figure 4.15 (right). Once the selection is satisfactory it can be named and saved as a region (a selection of one or more cells). Saved regions will appear in the alphabetically sorted list of regions in the panel to the left of the window and are automatically saved to the file system.

4.3.2.2 Authoring Process

A workshop was then held where this new version of the authoring tool was used by the artists to visualise the cellular network infrastructure and finally author the Love City game regions, i.e. transfer them from sketches into a format that is used by the game-engine. There were 18 key conceptual places of varying sizes already identified by the artists. These were chosen as they are available in all three cities and provide a balance between the need to be sufficiently precise for engaging game play, yet vague enough to cope with imprecision of the underlying infrastructure (e.g. inner city, city centre, shopping centre, or park).

The authoring software was installed on a PC which was connected to a projector to share the view of the interface. One of the artists directly interacted with the tool to implement the game regions. He first defined large regions before going into detail. The preferred infrastructure visualisation view at this stage was the one showing averaged cell coverage and all regions were initially defined using that view (figure 4.16, left). The raw point sample view was only used to check details and confirm selections (figure 4.16, right).

Figure 4.16: Authoring Love City using abstract (left) and raw sample view (right)

A specific naming scheme was followed when the regions were defined with all names consisting of city identifier, network identifier and common region name, e.g. “N33_Park” for the park region in (N)ottingham on the Orange UK network (network code is 33) or “D10_Park” for the equivalent region on the O2 network in Derby. The artists aimed at authoring regions that would contain multiple cells, but this was not always possible, depending on how the network operators planned their networks. Some operators such as Vodafone or O2 seemingly tend to prefer more powerful macro cells which would cover a larger area than the less powerful (and cheaper) micro cells which seem to be preferred by some other operators such as Orange. Consequently a network containing more micro cells does consist of more cell IDs.

This can be seen in table 4.8 which presents a selection of three Love City game regions (Park, City Centre and University) and how these have been mapped to the underlying GSM networks of one city (Nottingham). The table shows the number of cell IDs per region per operator. It can be seen that in the defined game content regions on the Orange network tend to consist of more cell IDs than those on the other operators' networks.

Nottingham:	O2	Vodafone	T-Mobile	Orange
Park	1	1	1	9
City Centre	5	3	5	12
University	2	2	2	5

Table 4.8: Cell ID count of selected regions for one city (Nottingham) across all GSM networks

Table 4.9 shows how the same virtual regions as in the previous table (Park, City Centre, and University) have been mapped to the three cities across one network (Orange). It can be seen that on average the regions defined on the Orange network generally tend to consist of more cells, also in the other cities.

Orange UK:	Derby	Leicester	Nottingham
Park	4	5	9
City Centre	5	26	12
University	2	3	5

Table 4.9: Cell ID count of selected regions for one network (Orange UK) across all participating cities

We have to be careful not to draw too many conclusions about network layouts here, as the numbers in tables 4.8 & 4.9 are based on the subjective mapping of an individual artist working with the author's visualisation and authoring tool. But it can be stated that our artist preferred working on the Orange network as its finer granularity was reported to allow an easier authoring of places for the game.

4.3.2.3 Overview of the New Data Collection Tool

Once authored, the game regions had to be tested. The new mobile infrastructure collection tool supported the verification of already authored content while collecting more data to improve the dataset for the visualisation and authoring tool at the same time. The latest content could be downloaded from the server to the mobile phone by selecting a single menu option (figure 4.17, left). The application continuously updates the cell ID information on screen, together with any potentially mapped regions. Figure 4.17 (middle) shows a screenshot of the tool without the overlaid menu. The full GSM cell ID is displayed at bottom part of the image (here: “234/33/1900/26625”). The line below reads “no known place for cell!”, meaning that this cell has not been assigned to any game region according to the current content mapping. The colour of the background screens has been used as the primary indicator to signify if the mobile would be inside one of the defined regions (light green) or in an undefined area (black, as shown). This allowed quickly getting this vital information at a glance, without interacting with the mobile in any way, then getting further details if necessary.

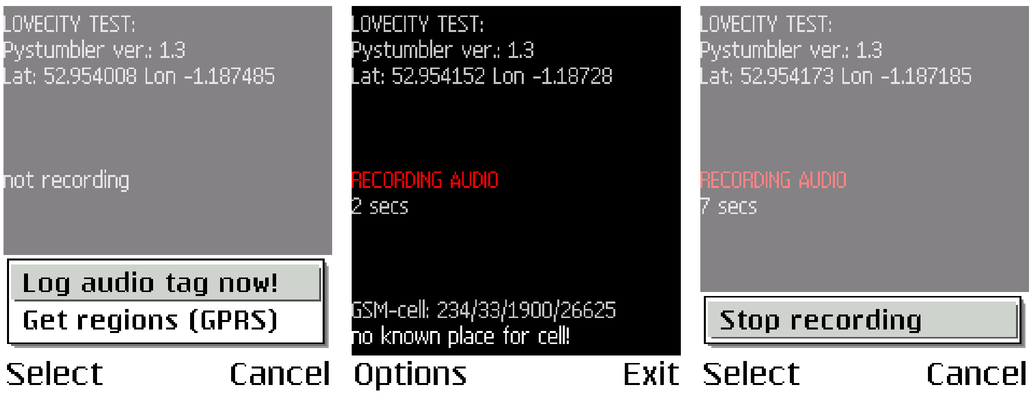

Figure 4.17: Data Collection tool for Nokia S60, recording an audio tag

The requested annotation feature was implemented as a voice recorder that saved geo-referenced audio samples which could later be visualised on the map in Google Earth. The voice recorder was triggered via the phone’s left soft-key menu where it was the default option and thus easy to select without searching for it. Audio samples were saved directly to the memory card in the phone’s native AMR format (adaptive multi-rate), which takes up about 1 kb per second. The maximum length of the recording is not pre-defined and only limited by the size of the memory card. The recording can be stopped by pressing the menu button again (figure 4.17, right). Other separate screens in the application would show more technical in-depth information such as GPS related information.

4.3.3 Feedback

The artist's feedback after the authoring and verification process was generally geared towards achieving a more fluent workflow outside and in the studio. Suggestions were mainly about staying in control and facilitating the editing of regions but also included a feature request for a paper export option.

4.3.3.1 Aid Orientation

The infrastructure viewer allowed editing on 4 networks in 3 cities and changing the networks and cities at run-time. Its programmer (the author) chose to accelerate the access to the city-views by putting them on hot-keys and have the network data to be loaded from files which needed to be selected with a file-requester. The artist mentioned that he would rather have the networks on hot-keys because he would do all networks in one city first before switching to the next city and so the switching of networks would occur much more often because he would continuously switch forth and back between the network layers. To aid orientation between the network layers he also suggested printing the currently active network name to the screen as he sometimes didn't know which network he was currently looking at.

4.3.3.2 Editing of Regions

Similar to the feedback on the orientation task the artist also pointed out that the viewer should automatically switch to the correct network layer if the user chose to get back to an already authored region, which might be on a different network and thus required switching the network layer. He also pointed out that regions did get changed quite often during the authoring process and that it would be an important addition to allow modifying the existing regions more easily. Furthermore he suggested to overlay all of the networks at once, potentially with variable degrees of transparency and different colours, as he frequently sought to compare the networks' granularities with respect to each other and had missed such a functionality in the tool when doing so (compare figure 4.8 which was made by the artist to aid him in reasoning about the definition of places in the game). Another welcome feature would be to get assistance in recognising overlap-conflicts in the editor. Love City used 1:N mapping of regions to cell IDs where each cell ID could only be assigned to one region. This application specific requirement was not supported by the authoring tool. As a result the artist had to keep track of which cell had already been taken to avoid selecting it again in another region – a task which would be much better carried out by a computer. The artist also raised a feature request for a paper export option. He would want to use the visualisations of the networks and the authored regions for other purposes including websites and printed material. To facilitate the export options beyond the existing screen grab and text to clip-board possibilities he suggested implementing a high-resolution "Save Image" function for the current view and being able to modify the colours being used in the visualisation, mostly the black background colour, to better support subtractive colour models like CMYK as they are used in print.

4.3.3.3 Testing and Verification In Situ

Being able to do to the required content-verification in parallel to the rather dull data collection exercise was seen as a boost in productivity by the artists and helped them in gaining an understanding of the behaviour of their piece in the wild while collecting more data for the visualisations. It is worth mentioning that the GPS device never caused any problems. But if it did it could have easily invalidated a lot of wardriving data. To tackle this potential risk the artists suggested that the tool should be able to make the user aware of the state of the GPS even if they are not currently paying attention to the screen. This alert mechanism could also be used to inform the user about other interesting state changes. Their feedback after having used this version of the data collection tool suggested using audio-signals for the state changes that were most important to them:

- Changing regions (relates to their content)
- Changing cell IDs (relates to the underlying network's granularity)
- Discovering new cells
- GPS problems

4.3.4 User Testing the Prototype

After this, a feature-complete graphical mobile prototype has been developed to facilitate early user tests. This second prototype, like the data collection tool, was based on Python for S60 and implemented the same cell ID location mechanism. Client-Server communication was realised with encrypted and compressed RPC (Remote Procedure Calls) over HTTP. The application employed a graphical user interface with a custom design as shown in figure 4.18.

Figure 4.18: Main menu and location screens of the graphical prototype

4.3.4.1 Overview of the Game Prototype

The different menu options as depicted in figure 4.18 (left) can be selected by using the up and down keys of the phone's joystick; pressing its centre button executes the highlighted option. By moving the joystick horizontally, the user can toggle between the main menu and a location screen, which shows the current location of the player within the virtual city, including a representative image and a location description (figure 4.18, right). The phone’s soft-keys provide access to the options menu, i.e. for rerunning the registration process, and to a hide function, which sends the application process to the background without quitting. The different game functions use either custom or native interface elements, depending on the function, the degree of user interaction, and the requirement for native user interface technologies, e.g. for text entry with predictive technology.

4.3.4.2 Testing with End-Users

Students from multi-media classes in Leicester and Nottingham were invited to test this prototype for a period of two days. Each student was given a Nokia Series

60 phone, a set of instructions, and a map of their city. They were instructed to regularly check their location in Love City, locate other players in the same zone, send messages to other players, accept or reject received messages, and note any issues or experiences that they had related to the game play. The students took notes, filled out a form, and were filmed by the artists while giving their feedback. Notes written directly on the map were the most effective feedback tool for the artists as this method gave in situ responses to the game and its functionality, and allowed for the feedback to be related back to the spatial content mappings in the authoring tool. This gave our artists an opportunity to find out how players inhabited the fictional world of Love City while they roamed the space of their actual real cities.

4.3.4.3 Outcomes from the Test

An interesting outcome of the testing was the discovery that when locations did not respond to the players' understanding of real world location, players often used their imagination to engage in the experience. For example, one player was at home in the "suburbs" when the game told them they were in the "park" and they said that they felt pleased that they were "transported to a park" by the game.

The testing also brought up issues of slow response rate of the interface, a high rate of no cell ID feedback from the existing mapping, and inconsistent responses from the game locations in the same real world locations. An important outcome of the testing was that it revealed the priorities of potential players. These were highlighted as:

1. Accessibility, as most students did not own Nokia Series 60 phones
2. The need for quick and regular responses from the game / other players
3. The game needed to mimic SMS messaging to meet player's expectations

Based on these findings, Love City was rewritten to employ an SMS user interface instead of the graphical smart-phone version. Details of this final development are presented in appendix 4 (page 337). But since the content mapping of the game has not been changed for the final version, this chapter now continues with the post-event reflections of the Love City study.

4.4 Post-Event Reflections

A feedback session was held after the game finished. Two artists from Active Ingredient were interviewed by two members of our team. The session lasted for about an hour and a half and took the form of an unstructured interview. The prompting questions were concerned with capturing information about the artists' experience of using the three layer model for authoring a location-based experience and their suggestions for improvements to the tools. The session was video recorded to allow transcribing and analysing the feedback in more detail. The rest of this section summarises the feedback that was gathered from this analysis.

The artists stated that the tools were very useful for visualising and making sense of a large area. They found that the process of taking samples of mobile network coverage in the three cities and then looking at the maps and visualisations of coverage in the infrastructure visualisation tool helped them to better relate their ideas to the maps and understand the cities as well as the wireless infrastructure. In terms of effect on the overall game, the artist's view was that the process of working with the tools did not change the original game concept but rather facilitated its realisation. In fact the artists were not sure how they would have built the Love City experience otherwise. They commented that authoring Love City was particularly complex as it involved a very large area of 3 cities and an infrastructure of 4 mobile networks in its public outing. Furthermore, they needed not just to see the data but also to be able to converge it to produce the merged virtual area of Love City. The infrastructure visualisation and authoring tool was seen as indispensable for this process.

4.4.1 Artists' Experience

The artists felt that the infrastructure visualisation tool needs to be layered as the authoring process of a location-based experience like Love City involves working with different versions of the same data simultaneously. This section splits the artists' feedback and suggestions for extensions into the three conceptual layers of the proposed authoring model.

4.4.1.1 Physical World Layer

The artists suggested that it will be useful to be able to choose between or overlay different types of maps of the physical area. For example, street maps are useful for reasoning about navigation whereas aerial or satellite imagery support more organic authoring and recognition of features such as buildings and parks.

4.4.1.2 Infrastructure Layer

The artists said that when they began working on the project they only knew that the ID of cells in mobile phone networks could be used to pinpoint location. They found it fascinating to learn about the spread and sizes of cells and the different granularities of cells between mobile network operators.

It is essential to be aware of the capabilities of the supporting infrastructure when authoring location-based experiences. Differences in precision or blind spots are not particularly problematic if the artists can be made aware of them and produce a narrative to accommodate them. In fact they said that gaps in the infrastructure (i.e. areas of no coverage) can provide interesting opportunities that could be exploited for storytelling and gaming. It was suggested to provide another visualisation of the infrastructure which would focus on highlighting the gaps in infrastructure coverage.

Indeed how to appropriately visualise the infrastructure was one of the main points of discussion in this part of the interview. The first version of the infrastructure visualisation tool that was presented to the artists only provided a visualisation of the infrastructure as a series of dots representing sample points (as shown on figure 4.11). The artists explained that while they found this view to be aesthetically beautiful it was visually overwhelming and too confusing to work from alone. They commented that the addition of the visualisation of approximate cell location and shapes (e.g. shown in figure 4.12) was liked a lot and simplified the authoring process. The artists preferred working with the abstract overview visualisation as much as possible to prevent themselves from getting overwhelmed by and caught up in detail. This observation is in line with Shneiderman's visual information seeking mantra "overview first, zoom and filter, then details on-demand" (Shneiderman, 1996). The artists mentioned that other visual representations of the infrastructure such as heat maps could be meaningful to them. It was also suggested that it would be useful to make it even easier to upload sample coverage data into the tool. For example, at the press of a button on the mobile device used for surveying, the sample data would be sent to the editor and dynamically integrated into the view.

4.4.1.3 Content Layer

The current support for content creation is very basic; essentially it consists of defining and naming different trigger regions. The artists commented that it would be useful to extend this layer. It was suggested that in the future one might be able to see the unfolding of a live experience on the content layer so that the location of participants could be dynamically represented and updated on the layered visualisation and one could see how participants navigate through the physical area and move between the regions.

4.4.1.4 Final Comments

The artists pointed out how the current approach of the editor leads the authoring process from the data. They proposed that it would be useful to explore how this can be changed so that instead it would be 'ideas first'. For example, the user could begin by drawing desired regions on a map and the editor would reveal how well this could be supported by the underlying infrastructure. Based on this feedback the user would then refine their design to accommodate what is possible with the communication and positioning infrastructure.

In concluding the session, the final comments of the artists were that the data collection and authoring tools worked well and achieved what they needed to do for the task at hands. They felt that these tools were a good starting point for artists to begin experimenting with and building up an own understanding of how a chosen wireless network location mechanism performs.

4.4.2 Author's Experience

This section outlines some additional thoughts and observations that evolved during this collaboration. It complements the artists' experience as described above and shows up possibilities for future work.

4.4.2.1 Physical World Layer

The physical world layer is thought to represent the target physical environment. To do this, the desktop infrastructure visualisation tool utilised digital map data ("Land-Line.Plus") that was acquired from Edina Digimap, a subscription-based service which allows UK students and academics free access to Ordnance Survey maps and other data. The available map-data is of higher quality than those currently available through other channels such as Google Maps/Earth or Open Street Map. The big limiting factor, however, is that the data only covers the UK, which prevents the use of this service for experiences that are not to be staged within the borders of this country. It would therefore be interesting to investigate the use of alternative data sources for the physical world layer.

A repeated request of our artists was concerned with a zoom function in the user interface. The different map views in the infrastructure visualization tool were pre-defined and assigned to the function keys. Although this enabled access to maps of all three cities at different scales, it seems unnecessary in retrospect (as mentioned in the artist's feedback on p. 171). If the user's orientation on the map view could be aided by an easier interaction metaphor then the function keys would be freed up to accelerate other less intuitive tasks like switching views on the infrastructure layer. It would therefore be interesting to investigate the use of zoomable user interfaces, e.g. as proposed for a 2D context by Perlin and Fox (Perlin and Fox, 1993) and also immanent to current 3D modelling packages as well as geo browsers such as NASA World Wind (54) and Google Earth (55) (cf. Scharl and Tochtermann, 2007).

4.4.2.2 Infrastructure Layer

Positioning technologies like GPS or – as in the case of this project – cellular positioning are the foundation of any location-based experience and it is in the interest of the experience designer to thoroughly understand the one that they rely on. Although this seems to be a commonplace, it is surprising how little is often done about it and how much people rely on a technology to just work as they think it would work when they are planning their systems. Similarly, positioning technologies other than GPS are often compared to GPS and expected to work in a similar fashion and with a similar precision – which is of course nonsense. It is a core interest of this thesis to reveal the inner workings of a chosen positioning technology to the designers to allow them to better reason about how its performance affects their work. By diminishing the designers' dependency on their own expectations it is hoped that a lot of guess-work can be replaced by, at least, educated guesses.

In this study we allocated a survey exercise to the designers early on in the development process. We have hoped to get the artists engaged with the technology and thereby become more intimately familiar with its characteristics and limitations. When coming back from their surveys the artists were always keen to look at visualisations of the data and find out about the granularity of possible locations in the area that they just surveyed. This eager interest must be seen as the expression of a learning process by which the artists gained experience about the invisible infrastructures that they had to rely on with their work. A key moment of understanding happened at the authoring workshop (see page 167) where data for all networks in all three cities was available and the artists sought to map their conceptual regions to the different networks. One of the artists was operating the authoring tool and started mapping. When he switched the authoring view to work on the O2 network he suddenly chuckled in disbelief. Looking at a visualisation of his self-collected survey data he was obviously surprised by the coarse grained granularity of the network, which was much coarser than the other networks. However, this was not a cause for frustration for the artist as he quickly accepted this layout as a fact and continued with his work. He commented that he had already discovered this effect in the initial visualisations (see figures in section starting on page 152) and that he knew that it would happen again.

A positive side-effect of such self-acquired knowledge is that it makes the designers less dependant on the technology experts and also enables them to more clearly express their own requirements. Because of this, feature requests for the supporting tools are more to the point which benefits the swift progression of the overall project.

4.4.2.3 Content Layer

The content layer is very closely related to the supporting technology which was mentioned in the previous section about the infrastructure layer. Technology is the hardware side whereas the content layer is more on the software side. The content triggering mechanism is especially important in this respect as it is responsible for delivering the location-based content at the right location. The content triggering mechanism utilises the chosen infrastructure and generates events under certain circumstances. For example in a GPS-based application this could be done by having trigger-zones defined as circles around points of interest and continuously testing the current GPS position for containment in any of those zones.

4.4.2.4 Summary

In this study we started off using cell ID positioning, a very simple mechanism which triggers when the ID of the currently serving cell is equal to a pre-defined value. Cell IDs are numbers that bear no unit and are almost meaningless on their own, yet they are spatially arranged in a network structure. We devised and evolved a suite of authoring tools that allowed our artists to geo-reference and visualise cell IDs and thereby reveal the structures of GSM networks in a way that was meaningful to them and which allowed them to use cell IDs as the location mechanism for their game. For the final version of the game we also used cellular positioning, but in another way. Instead of determining the currently serving cell on the client handset the final game utilised operator based positioning which essentially does the same, but on the server side (c.f. page 341).

Having done this iteration of work for cellular positioning in GSM networks, it would be interesting to investigate the use of the 3 layer model with a content triggering mechanism that builds on a different wireless technology. The following chapter presents a detailed account of the Rider Spoke study, which relied on Wi-Fi access points for its location service.

5. Second Study: Rider Spoke

This fifth chapter is about the *Rider Spoke* trial which has been designed and developed in collaboration between the performance art group Blast Theory and the Mixed Reality Lab at the University of Nottingham as a final phase showcase for the IPerG project. The author's work on *Rider Spoke* was all about helping the artists in understanding a complex location mechanism that was based on Wi-Fi.

5.1 Overview

This approach of this study is similar to the one used in the Love City study, but this time a different group of artists was supported in working with a different technology. This chapter starts with an overview of the designers' tradition, the author's role in the process, and a general description of the experience. It then presents some technical background, followed by a detailed description of the iterative development process that led towards the premiere and beyond. The description of the *Rider Spoke* development process is based on the author's active participation in development phases two and three (I joined the project at the end of the exploratory phase), and on the minutes, plans, reports, notes and emails that have been written throughout the development. It is also based on some video observations that have been recorded during the setup for Athens.

5.1.1 Designers' Tradition

Blast Theory (19) is an artist group that was founded in 1991 and which is based in Brighton. Having a background in theatre, the group uses new media and technology in their performances to create interactive art. Blast Theory started collaborating with the Mixed Reality Lab at Nottingham University in 1999 for a project called *Desert Rain* (56), and have since then created several widely known location-based experiences, including the award winning *Can You See Me Now?* (Golden Nica for Interactive Art at Prix Ars Electronica 2003, page 37), and *Uncle Roy All Around You* (page 44).

5.1.2 Author's Role

The author helped informing the decision to use Wi-Fi as the technical infrastructure for Rider Spoke by giving a demo of the Wigle Wi-Fi database (14) at the project kick-off meeting. He did not actively take part in the early exploratory work but was subscribed to the mainlinglist and thus the description of that phase (section 5.3.1 and parts of 5.3.2) is based on the minutes, plans, reports and emails that have been written and circulated during that time.

The author joined the project at the end of the exploratory phase to help conducting the network survey and to subsequently support the project with visualisations of the infrastructure to empower the team to tune its location mechanism. He led this initiative by devising the required mobile survey and desktop visualisation workflows in close cooperation with the artists and the core development team in an iterative process. The author integrated an existing graph visualisation system with the Rider Spoke game server and with Google Earth (page 211). He also wrote an extension to the game server that allows simulating and visualising different location mechanism settings for the mobile devices without having to go outside (page 226), and programmed a mobile survey tool with geo-coding capabilities (page 232). The author also documented the participatory design process, especially the artists' experience with the tools.

5.1.3 General Description

Rider Spoke is an interactive location-based experience for cyclists. Players explore the urban environment on their bikes, setting off from and returning to a hosting venue. On their journey, players are invited to record spoken answers to reflective, increasingly personal questions and to hide their answers at chosen locations. They can also search for and listen to other players' answers. The game lasts for an hour and reminds participants when it is about time to return to base.

The game runs on Wi-Fi enabled Nokia N800 handheld computers, which are securely mounted to the handlebars of the bikes (see figure 5.1, left), and has a touchscreen user interface which is implemented in Adobe Flash (figure 5.1, right). The game is designed to be gentle and simple to use in order to fit well with the task of cycling. With safety in mind, players are not required to interact with the device while cycling, and indeed, they are specifically asked not to do this. The narrative of the game is kept together by a game voice, which talks players through the game, asks questions and instructs them to stop so that answers can be recorded. Interaction with the game is deliberately designed to be low key and simple so as to be safe and also not to interfere with the distinctive solitary experience of cycling through the city at dusk, when the game is played.

Figure 5.1: Rider Spoke participant listening to audio content via headphones (left), game device showing a question for the player (right) (Source: Blast Theory)

Rider Spoke is a stand-alone experience. Players are not connected live to one another (although they may of course encounter one another on the streets) or to a central game server. Instead, the content required for each game, including previous players' answers, is preloaded onto their device before they depart. At the end of the game any new content that they have created is collected from their device so that it can potentially be merged into the evolving database of content and subsequently be made available to other players (following an overnight process of review and selection of the best player generated messages).

5.1.4 Game Walk-Through

Players arrive at the hosting venue either on their own bikes or to borrow one. They register at the reception, sign a disclaimer, leave a deposit, get briefed on how to play the game and then go out for an hour with a game device attached to their bikes. A few minutes into the game music starts setting the tone for the experience and a narrator starts talking to the players. The players' first task is to record an answer in which they describe themselves.

5.1.4.1 Recording an Answer

Having received a question, players would then look for a valid place to hide and record their answer to it. A suitable location is defined to be one that a player has chosen and that is also "empty", i.e. not associated with another player's answer.

The players may have to cycle around for a while before the screen shows an iconic representation of a swallow (figure 5.2, left) and thereby signals that a valid hiding place has been found. Stopping their bikes and tapping on the touchscreen, they would reach the next screen (figure 5.2, right) where the question is repeated and an iconic house represents a location where they could record and hide their answer. They also have the option to refrain from recording their answer at this location and go back to the cycling task to look for a different location.

Figure 5.2: Swallow icon denoting a free hiding place (left), hiding content (right)
(Source: (Adams et al., 2007))

Figure 5.3: Recording interface (left), deciding what to do next (right)
(Source: (Adams et al., 2007))

Figure 5.3 (left) shows the interface after the user has selected ‘hide here’ and then recorded some audio content. Options here are to play back their recording, re-record the message or quit recording (‘done’). After having recorded at least one answer, players could decide if they wanted to answer another question or would prefer searching for other players’ answers (figure 5.3, right).

Table 5.1 lists some of the questions that players would encounter on their journey and table 5.2 lists some exemplary answers that were recorded by players.

Question	Scripted text
Describe yourself	*This is one of those moments when you are on your own. You might feel a little odd at first, a bit self-conscious or a bit awkward. But you're alright and it's OK. You may feel invisible tonight but as you ride this feeling will start to change. Relax, don't forget to breathe both in and out and find somewhere that you like, it might be near a particular building or road junction, it might be near a mark on a wall or a reflection in a window. When you have found somewhere you like give yourself a name and describe yourself.*
Window	*I want you to look for flat or a house and find a window that you would want to go through. I want you to stare into that window and tell me what you see and tell me why you want to go through that window.*
Father	*Please, will you tell me about your father. You might want to pick a particular time in your father's life or in your life. Freeze that moment and tell me about your dad: what they looked like, how they spoke and what they meant to you. And while you think about this I want you to find a place in the city that your father would like. Once you've found it stop there and record your message about your father at that moment in time.*
Promise	*You've been riding for a while now. You've answered some of the questions I've asked and you've explored the city. Thank you. I have one last thing to ask of you and when you have answered please can you come back to the Barbican. Will you make me a promise? It might be small, a promise about tomorrow or a friend. It might be something more profound. But, now, tonight, here, make a vow about your intentions. Think for a few minutes. Go somewhere, stop your bike and say your promise out loud into the air.*

Table 5.1: Example questions

Question	Recorded answer
Window	*It's a window in somebody's flat and you can't see in it very well because the blinds are drawn, but there are little cracks, and I want to go in and I want to see what's on the other side.*
Father	*I'm at St John's the Baptist church in Hoxton and I chose this place because I thought my father would like it, as it's a bit calm and relaxing. I admire my father because he's a very intelligent and wise person and he's*

	very calm and in fact he's just had a birthday which I couldn't attend because of work commitments and I feel a bit sorry about that and I hope that I can make up for it by visiting him on his next year's birthday again, and hence I'm choosing this place in a kind of attempt to make up for it.
Promise	*Even though it's almost impossible, I really am going to try my best not to be afraid, cos I've been afraid so much and to make some choices, if for no other reason than in honor of the people that can't make choices at all anymore. For Hope and Danny, and Markus and Mary. Alfred, Lilly, Lewis and Martha, Liz and especially Dan.*

Table 5.2: Examples of player recorded answers

5.1.4.2 Listening to other Players' Answers

Apart from recording their own answers, players can also eavesdrop on other players' answers. Figure 5.4 (left) shows the interface after the player has decided to search for other players' messages. The game has selected three responses to the question 'Describe yourself. What are you like and how do you feel?'. The names of the players who created these answers are displayed along with a hint as to how far away the messages might be found.

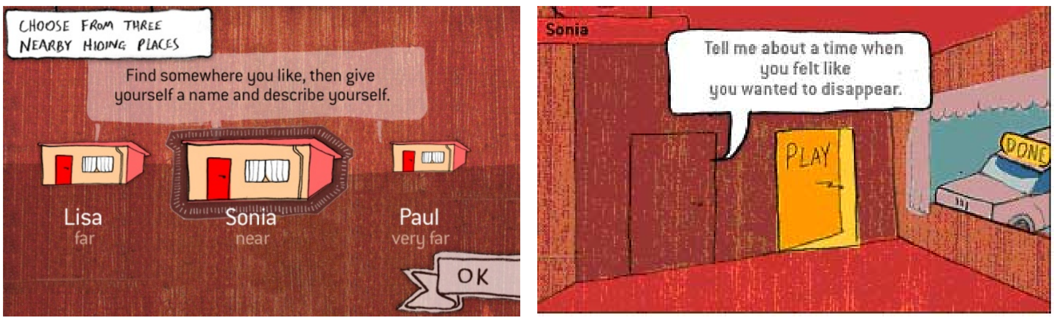

Figure 5.4: Showing nearby answers of others (left), listening to another player's answer (right) (Source: (Adams et al., 2007))

The player could then decide on an answer they would like to listen to, which would take them to the next screen as seen in figure 5.4 (right). In this case the other player is 'Sonia' who has answered the instruction 'Tell me about a time when you felt like you wanted to disappear'. The corresponding question text is repeated and the player could decide to play that piece of audio content or leave the hiding place ("done"). When done, players could decide if they wanted to answer another question or would prefer searching for other players' answers, again.

5.1.5 Public Stagings

Rider Spoke premiered at the Barbican Centre (57) in London on 11th October 2007 where it consequently attracted over 500 players over the course of 8 days, spread across 2 extended weekends (Thursdays – Sundays).

Figure 5.5 shows the Rider Spoke reception area at the Barbican just before the premiere (left) and the rental bikes (right) which could be picked up on the other side of the doors once the players had registered and received their brief at the reception.

Figure 5.5: Rider Spoke reception area at the Barbican (left), rental bikes (right)
(Source: (Chamberlain et al., 2008))

After London, the first international staging of Rider Spoke happened in the centre of Athens, right below the Acropolis. By invitation of the British Council (58) and with support from the local "Friends of the Bicycle" club it attracted 94 players between 13th and 16th of March 2008. A third staging of the experience happened at the Brighton Festival (59) between May 8th and May 11th 2008 and attracted about 400 players. This chapter draws on experience of and feedback from all three of these public stagings.

5.2 Technical Background

On the technical side, Rider Spoke utilises the ubiquitous Wi-Fi access points in urban environments to employ a novel location service in which an evolving graph of so called Wi-Fi fingerprints and adjacency relationships between them is used to build locations and infer the likely proximity of good hiding places and other players' content in relation to the current player's location. One of the reasons why it has been decided to build the location mechanism like this is because in past experiences GPS had proven to be problematic in densely built urban environments. In a city, the limited view to the sky often hinders GPS to work precisely, or sometimes blocks it completely; this would heavily interfere with a game design that relies solely on precise positioning.

Figure 5.6: "Urban canyon" in the old town of Vienna

Figure 5.6 shows such a problematic urban environment where GPS would not work reliably – a so called "urban canyon". This example is certainly an extreme, but even less dramatic urban sceneries are known to cause interference with GPS, as for example revealed by the "Can You See Me Now?" research (Crabtree et al., 2004), which analysed how performers in a GPS-based experience dealt with being disconnected from GPS (c.f. page 110).

A typical approach to counter balance this effect is to build a positioning system that does not require direct line of sight to the satellites in the sky. Several alternative approaches have been previously presented in the literature review (page 83). One of them was the Place Lab project from Intel Research Seattle (42) which presented a hybrid system that could draw on a variety of sources, such as GPS, Wi-Fi, or GSM cell ID. Especially interesting in the context of our work is a temporal coverage experiment where they analysed the availability of the aforementioned technologies over the daily routines of a small set of test-subjects with different occupations (LaMarca et al., 2005). In their experimental setup, GPS was only available in 4.5% (!) of the time, but Wi-Fi was available 94.5% of the time of the subjects and GSM coverage approached 100% (c.f. page 115). Another paper from the same project tried to evaluate the characteristics of different algorithms to support metropolitan scale Wi-Fi positioning (Cheng et al., 2005) with an emphasise on practicality and short calibration efforts.

Similarly, earlier IPerG work for a game called "Hitchers" (page 54) suggested that in order to increase the scale of an experience beyond controlled lab conditions, its calibration process should be incorporated into the experience design itself as this would allow continuous improvement of the underlying data-set while the experience is running (Drozd et al., 2006).

Previous work has been carried out on identifying locations that are meaningful to people. Ashbrook and Starner built and studied a system that learnt relevant locations from traces of GPS data (Ashbrook and Starner, 2002). The "Contextphone" (Laasonen et al., 2004), "Opportunity Knocks" (Patterson et al., 2004), and the social location disclosure application "Reno" (Smith et al., 2005) all allowed inferring locations from traces of GSM data. The common goal of identifying meaningful locations is usually achieved by either machine learning or humans explicitly entering this information. Examples of the former category are the Ashbrook and Starner's GPS experiment and the Contextphone which both tried to automatically extract relevant information by applying heuristics to the gathered input data. Examples of the latter are embodied in the Reno application as well as in Hitchers: users explicitly declare their current location and give it a more or less meaningful name (e.g. "in the lab", "at the bus-stop", "at home").

Many of these location aware applications do not only keep track of the locations that are discovered, but also of the connections between them. Thus, they gradually build up a network, or graph of locations and the routes between them. See figure 2.21 in the Hitchers section (page 54) for an example of such a graph.

A main concern of all the related projects is the behaviour of the utilised location algorithms. This is because handling these alternative algorithms can be considerably more complex than simply using the geographical coordinates that are supplied by a GPS receiver[18]. They are more complex in the sense that more input variables affect the outcome of the algorithm, and it is also more complicated to predict their performance.

The same applies to the Rider Spoke location mechanism, as it is also based on an ever-changing wireless infrastructure. There is a need to understand the behaviour of such dynamic systems, especially for those who make use of them, i.e. the experience designers. Rider Spoke therefore presented itself as a good opportunity to raise the designers' awareness of the underlying wireless infrastructure by supporting them in the development of the experience.

The following sections present a detailed account of our iterative development process. The sections are organised into three major development iterations: initial explorations, development up to the premiere, and modifications that have been applied after the premiere of the experience.

5.3 Development Iteration 1: Initial Exploratory Phase

The main focus at the start of the project was to produce a game design, decide on a strategy for locating players and ultimately come up with a specification for a target platform, as well as a set of tools which would allow the artists and developers to build the game as intended. This early phase of the project was organised as an interleaved series of workshops and tests between the artists and the developers. An internal project manager had been appointed to ensure timely progression through the schedule throughout this phase. The author of this thesis did not actively participate in the early work of this phase, but joined the project when the need for infrastructure visualisations arose (page 199).

5.3.1 Deciding on a Content Triggering Mechanism

One of the project goals from the outset was to target a commercially available mobile gaming device and to explore how IPerG research could be adapted to it. The game idea and hardware requirements for this IPerG final phase showcase

---

[18] GPS is a very complex system in itself, but all of that complexity is hidden from the user.

game underwent considerable changes in the early planning phase and ranged from a computer vision based monster hunt to a GPS-based in-car game for motorists in Los Angeles – which stemmed from an offer for an artist residency to Blast Theory. The idea for an in-car game influenced the work in this phase. With Sony being one of the project partners, it was considered using the Playstation Portable (PSP) console as the game device.

With these broad constraints in mind, attention then turned towards how to best trigger the location-based content under these constraints. A series of tests were conducted that aimed at gaining an understanding of the different possible location mechanisms.

5.3.1.1 Informative War-driving

To get a first idea of the granularity of the Wi-Fi environment in their neighbourhood in Portslade near Brighton, the artists went out with an IBM Thinkpad laptop, with an Orinoco Silver PCMCIA Wireless card and a Garmin GPS mouse. Netstumbler v0.4.0 (60) has been used as the war-driving software. The artists slowly (not exceeding 20 mph) drove around in a car for about 45 minutes, paying attention to the computer as well as the neighbourhood. The software recorded 2365 readings of 307 different Wi-Fi access points at a scan-rate of about 1 Hz. On average they discovered a new access point every 9 seconds, which was considered as a quite good granularity. In addition to these raw numbers, the test led the artists to the discovery of interesting environmental factors that would affect a Wi-Fi based location mechanism, such as:

- ***Area type*** – Industrial areas and main roads/bypasses showed fewer APs.
- ***Density of housing*** – small residential terraced brick houses gave greater density than road with larger detached houses.
- ***Road width*** – wide roads with parks, deep gardens or slip roads between the mapping route and buildings showed fewer hotspots even where they were densely lined with housing.
- ***Apparent expensiveness of an area*** – cheaper areas of Portslade showed a significantly lower density of Wi-Fi.
- ***Wi-Fi transparency of building materials*** – or how far Wi-Fi signals can

carry. In areas with terraced brick housing, access points were not visible beyond a single road suggesting they spilled from the front of houses but were not powerful enough to pass out the back and through the house of the next street. This might not be the case in areas where buildings are made of different materials.

5.3.1.2 Content Triggering Tests

It was clear from the very beginning that the content in Rider Spoke would consist of audio samples. Still aiming to produce an in-car game, the next tests then focussed on trying out GPS and Wi-Fi based content triggering mechanisms in cars, especially how reliably and timely they would trigger. The artists were provided with a piece of software for their Windows laptops which would continuously provide a GPS position as well as a Wi-Fi position, which was based on the averaged geographical positions of all currently visible access points. The application maintained an internal list of known geographical positions of access points for this purpose. The data for this list was self-collected using the same setup as in the informative war-drive and uploaded to an online community website called Wigle which collects geo-coded access points (14). Wigle was then queried for the area of interest and the resulting list of geo-coded access points was downloaded in text format. Introducing this extra step to the process had the advantage that it allowed to profit from any potential third-party data that Wigle might store for the area of interest. The user could select which trigger mechanism should be used (GPS, Wi-Fi, Debug[19]) and the application would use the current geographical position provided by that mechanism to trigger audio content if it falls within predefined circular regions of variable sizes (see figure 5.7). For example, in the more complex case of the Wi-Fi position mechanism this meant that the mobile device continuously scanned the network environment for known access points, averaged their associated geographical positions, and used this absolute position coordinate to test for intersection with any of the circular content regions. Upon intersection with any of those regions it would then trigger the associated audio content.

---

[19] uses pre-defined positions to enable the developer to work from his desk

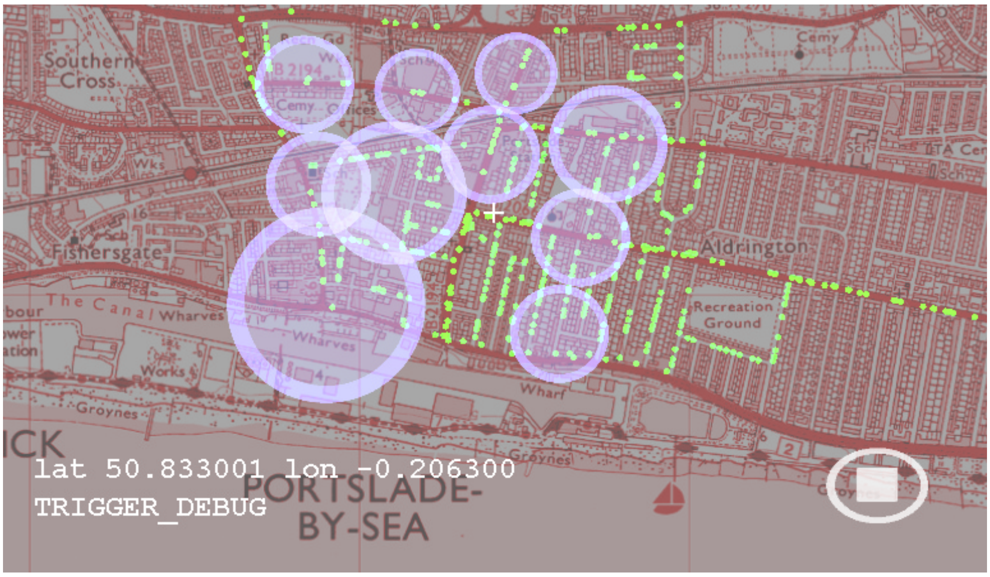

Figure 5.7: Screenshot of Wi-Fi positioning test-application showing 10 content regions (in blue), positions of known access points (in green), and current position (coordinates and small crosshair)[20]

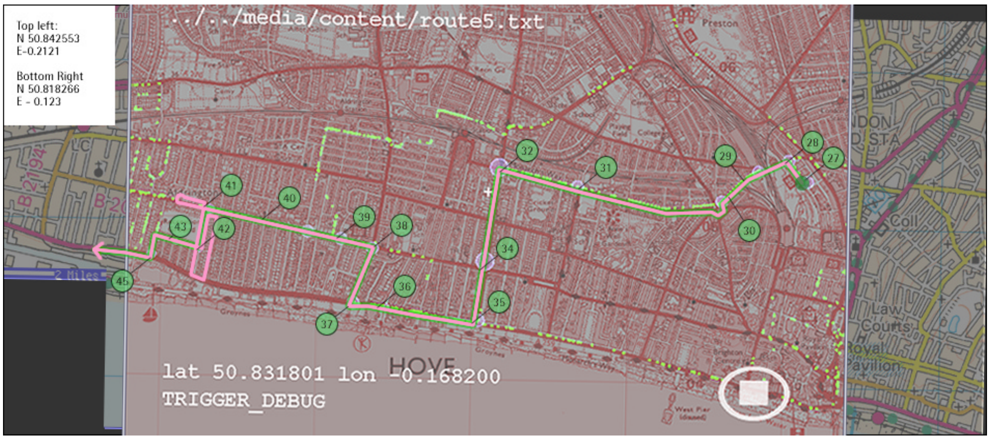

Figure 5.8: Locations of navigational content (green circles), travelled route (pink line)

The artists prepared navigational content pieces which have been assigned to content regions along a planned route (figure 5.8) and which were intended to guide an uninitiated tester from Brighton (on the right) to Portslade (on the left).They then conducted two test runs with this dataset – one for GPS and one for Wi-Fi positioning. In these tests they observed how the tester would cope with the navigational cues, how timely the cues would trigger, and if they triggered.

[20] Figures 5.7 - 5.9 courtesy of Martin Flintham

Findings from the GPS Test

Unsurprisingly, the GPS based content triggering in a car went generally well. Only one of the defined regions did not trigger and this was because the radius of the region was set too small (15 m) to be triggered in the car as it was travelling at 30 mph. On the other hand, too large regions would potentially entail surrounding streets and trigger there unexpectedly. Reflecting on their own practical experience, the artists inferred that there was a trade-off regarding the diameter when using circular content regions.

The artists summarised the following outcomes from the GPS test:

- Hotspots should be centered on the desired point of triggering with a minimum radius of 20m.
- Landmarks described within instructions should guide the driver's attention to the desired direction.
- Descriptions should be brief to keep the actual instruction clear.
- Authored content requires a longer process of revision to account for issues such as timing, subjectivity of descriptions, and proper user testing.
- In some circumstances an instruction will need to be repeated due to external factors, e.g. noise from restarting the engine, the radio volume being mis/reset. Allowance should be made for this.
- Defining circles in this test caused problems because of instructions being triggered on adjoining roads.
- We should look further at how going off route is handled.
- Waiting in traffic or stopping at lights creates unexpected pauses which may be a good opportunity for giving content.
- Instructions which depend on the driver stopping should make allowance for how long this might take. In a residential road a 40 m radius hotspot was not large enough in this case.

An alternative to using static shapes as content triggers for GPS-based experiences could be to use more adaptive shapes, i.e. shapes that change according to user variables, such as speed. Another alternative for improving GPS-based triggering is presented later in the discussion chapter on page 306.

Findings from the Wi-Fi Test

The Wi-Fi based content triggering worked well for regions that contained three or more mapped access points, but it did not work in 4 out of 19 instances. Two instructions failed to trigger because none of the previously mapped access points could be found during the test. Another two instructions failed because no mapped access point fell within the 40 m diameter of the content region.

The tester reported that the instructions generally felt late and that in some occasions it would have been reassuring to know if she still went in the right direction. The artists concluded that the lateness of the instructions did happen because they defined the regions without reference to a Wi-Fi map.

The artists compiled the following outcomes from the Wi-Fi test:

- Limited navigation using Wi-Fi seems to be possible.
- It is suggested the hotspots are centred on the desired point of triggering with a radius set to encompass a minimum of 2 or possibly 3 access points. In the test area this would be a radius of between 20 - 60 m.
- Testing content should be done using the final platform, not NetStumbler or otherwise.
- Authored content requires a longer process of revision to account for issues such as timing and subjectivity of descriptions along with proper user testing.

5.3.1.3 Storylines

The results of the content triggering tests were promising as both positioning mechanisms and the navigational tests have generally worked. Building upon this work the artists now wanted to extend the mechanism and test if storylines, i.e. audio samples grouped into routes, could be used as a game mechanism that guides players around the area (see figure 5.9).

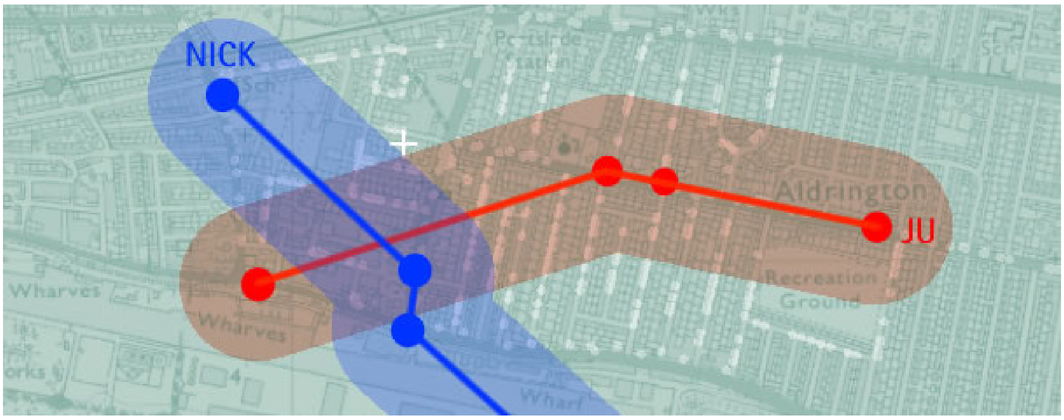

Figure 5.9: Two storylines that players could follow

In this idea, a player would select a story and then move towards the start point where the initial content would be triggered. The player would subsequently be guided towards the next piece of content until they reach the end of the line. If a player on one route happens to cross another storyline, he could choose to change the story and continue with the other story from that point.

From a project management point of view the project was in a state of flux at this time. The artists' residency in Los Angeles did not happen. As the whole in-car game idea stemmed from exploiting the unique car culture in California – which was no longer an option – the underlying game principle has been reassessed. This led to the discovery of cycling as a possible alternative mode of transport for Rider Spoke, much in keeping with a likely premiere in Britain. Consequently this test investigated how the system would work on bikes and so the laptop with the prototype software has been mounted to a bike.

Functionally everything worked as expected: players could select an audio trail they would like to follow, the delivery of audio content was timely and in the right location (with a noticeable delay in the Wi-Fi test), the software guided the way to the next piece of content on the trail and players could change trails as they crossed another story line.

However, the idea of story trails has been abandoned and replaced by a hide & seek metaphor, where user could record and hide their own content at "empty" locations and seek for other players' answers which have been hidden nearby.

5.3.1.4 Making a Decision

Some 5 months before the premiere, a decision needed to be made on the content triggering mechanism. At this stage, several key constraints of the project had shifted slightly:

- The idea for an in-car game in Los Angeles was no longer an option
- The Playstation Portable could not be used for license reasons

The Playstation Portable (PSP) development kit proved to be problematic and not applicable for the development workflow of a location-based experience. The development kit was essentially a complete desktop PC with a PSP attached to it. The PSP could not be detached and moved around for testing, as it would be required for developing a location-based application. Also, the development code could not simply be tested on a separate PSP, as the developers were not allowed to write their own UMD disks due to license restrictions. Getting a UMD with their code on it was theoretically possible, but would have involved a code review at Sony in Japan, which would have taken a considerable amount of time – somewhere in the order of weeks. This was clearly not an option for the developers, as they required much quicker turn-around times in the order of minutes. The only other option for testing location-based PSP code would have been to design a custom mobile carriage (with an independent 220V power supply) for the development kit to move around. As there was no use in such a bulky kit for prototyping, the PSP had been abandoned as a target platform.

The search for an alternative started. The majority of the established console-like devices in 2007 came without GPS but with Wi-Fi, e.g. the Sony Playstation Portable, the Nintendo DS or Windows Mobile PDAs. It was decided that in order to fulfil the original goal of "targeting a commercially available pervasive game device" the use of GPS for positioning had to be ruled out in favour of a Wi-Fi based solution. After a market research into alternative devices it became evident that the Wi-Fi enabled Nokia N800 Internet tablet was going to be used as the game device and that the game was going to be designed for cyclists.

With the new target platform fixed, the content triggering mechanism remained to be chosen. The artist asked for technical guidance in this matter. Precise, i.e. GPS-like positioning was not very important to them, but it was seen as crucial that the content would be triggered in a sensible, location-based way.

Different content placement strategies have been discussed, including:

- Placing content at geographical coordinates (latitude, longitude)
- Placing content at Wi-Fi hotspots by their MAC address (BSSID)
- Placing content at Wi-Fi hotspots by their names (SSID)
- Placing content at locations that are defined by generating a representative "fingerprint" of the Wi-Fi environment at each particular location
- Organising locations in a graph

It was decided that the content trigger mechanism of Rider Spoke was to utilise Wi-Fi fingerprint locations (see section "The Rider Spoke Location Mechanism" on page 208 for a detailed explanation) that are connected and organised in a graph structure. This approach was seen as beneficial for two main reasons:

1. Fingerprints of the Wi-Fi environment would allow defining locations without relying too much on the presence of any particular access point. This was thought to be useable for smoothing out the fluctuating nature of the Wi-Fi environment.

2. The graph structure would allow a notion of nearness between locations, even if their physical position was not exactly known. This was thought to be useable to measure the distance between a player's current location and pieces of content left at other locations, so that a player could be automatically presented with nearby content.

The user generated game content, which consists of audio samples that are recorded by the players while they are outside, would be mapped to locations in a 1:1 fashion, meaning that each location could only hold one piece of content and that each piece of content could only be assigned to one location.

5.3.2 Network Survey and Initial Visualisations

(Author joins the project). The next step then was to conduct a network survey, which was prepared by us and carried out by the artists from Blast Theory. The goal of this exercise was to test the first version of the fingerprint algorithm under real conditions and to collect real data for initial visualisations of the network infrastructure in order to better understand the Wi-Fi fingerprint infrastructure. A further goal was to produce an overall strategy for bootstrapping the game.

5.3.2.1 User Requirements

The artists were interested in the potential granularity of the location mechanism that was going to be built. They were especially interested in finding out about the radii of access points, i.e. the physical area around an access point where a Wi-Fi enabled device would still sense a particular access point. Their secondary focus was to get a sense of the topology of the fingerprint graph, if possible also in relation to the physical area. It was suggested by the artists that it would be useful to have "different lenses", i.e. views, on the infrastructure and that a graph visualisation laid out with a spring algorithm would be regarded as one of these views.

The developers, on the other hand, were specifically interested in a layered spatial visualisation which would allow them to investigate the raw data from the survey in relation to a map of the environment. Any algorithmically expressed assumption based on this raw input data – i.e. the fingerprint and game related information – was asked to be kept on a separate layer, so as to have access to all available raw information for analysis without discarding or hiding any data.

Apart from the requests for separated visualisations of the raw-data and the fingerprint graph, both groups also specifically requested an easy workflow.

5.3.2.2 First Tests of the Mobile Client and Data Collection

The user requirements led to the design of the initial network survey which was conducted as part of the first test of the mobile game client. We decided to keep the workflows for the two different visualisations completely separate:

- The fingerprint data was collected with an early version of the location mechanism running on the actual N800 hardware.
- The geo-coded access point data was collected with a war-driving software for Windows Mobile PDAs called “WiFiFoFum” (WFFF) (61). Version 2.2.12 of the software has been used on a HP hx2700 PDA running Windows Mobile 2003 SE with position data coming from a Haicom HI-406BT Bluetooth GPS receiver with SiRF Star III.

The PDA war-driving software was freely available and ready to be used. Incorporating it into the initial test-setup for the GPS data collection had the advantage that it allowed us to start straight away, without imposing any more work on the core developers by asking them to implement a similarly reliable piece of software for the N800. This meant the developers could concentrate on the more pressing task of getting the location mechanism working on the N800, which was a crucial component of the mobile game client.

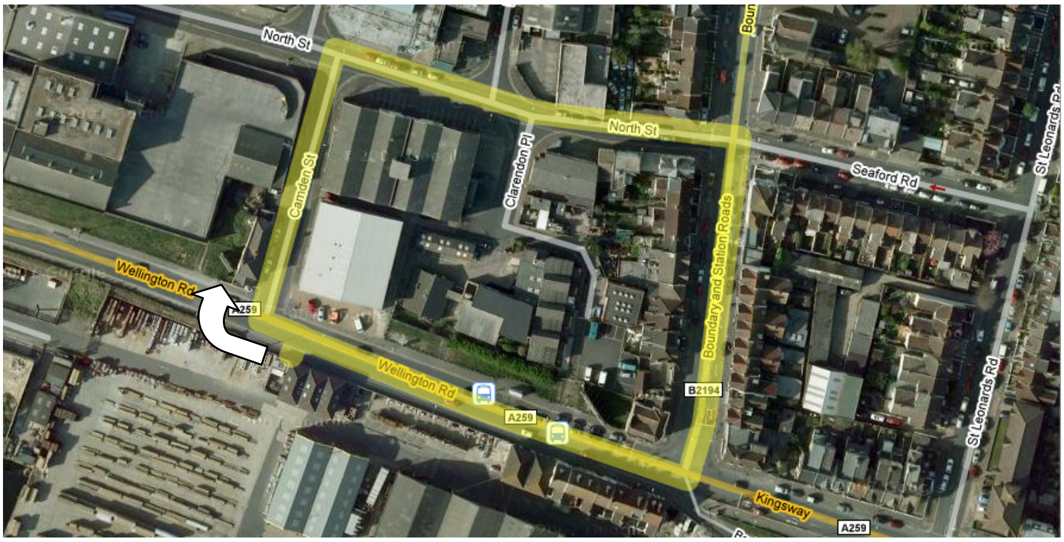

Figure 5.10: Test route in Portslade, UK

A circular test-route (see figure 5.10) was defined and two N800 devices as well as one PDA have been taken along this route multiple times. The route was travelled along in clockwise direction, starting at Wellington Road on the lower left corner of the route (see the arrow). The goal of the test was to assess the new fingerprinting algorithm under real world conditions. Artists and developers wanted to find out if the mobile client could scan for Wi-Fi access points, if it reported those access points on the user interface, if it successfully created fingerprints from those access points, if content could be attached to fingerprints, and if fingerprints could be retrieved when returning to a location.

While the PDA was just used to collect data for later analysis and visualisation, the N800's logged all their data and additionally provided text feedback about the current fingerprint to the testers while they were outside.

The algorithm determining the current fingerprint was quite simple at this stage. It scanned the Wi-Fi environment for 2 seconds, removed those readings that were below a certain (optional) signal quality threshold and then built a hash value from all remaining BSSID strings. Hash values were intended to be unique per location, so if an already known hash value was discovered, the device was assumed to be at a previously visited location. Otherwise a new location was added to the internal graph structure.

Figure 5.11: N800 mounted to bike with elastic straps and tape

Three circular test runs were conducted, the first on foot and the following two runs with the devices mounted onto bikes (see figure 5.11). For each run, the route has been completed more than once. On the first loop every discovered fingerprint was marked. Then on subsequent loops, rediscovered fingerprints have been noted.

The N800's were set to different signal quality settings in order to get a feeling for how different values of this parameter affected the game experience, especially how they influenced the granularity of locations as well as the retrievability of fingerprints (FP) when returning to previously visited locations.

The setup and results of the different runs are summarised in table 5.3.

Test	Dev. 1 Threshold	Device 1 Notes	Dev. 2 Threshold	Device 2 Notes
1 (walk)	none	31 FP on first loop (inc. 5 seen in building), 5 refinds on second loop	-60	7 FP on first loop, 3 refinds on second loop
2	-80	9 FP on first loop (0 in building), 0 refinds on second loop, 1 refind on third loop plus 4 new	-60	0 FP on first loop, no refinds on second loop plus 2 new, 0 refinds on third loop plus 1 new
3	-80	52 FP on first loop (0 in building), 5 refinds plus 11 new on second loop, 6 refinds and 15 new on third loop. Unstable locations.	-60	0 FP on first loop, 0 refinds and 0 new on second loop, 0 refinds and 1 new on third loop

Table 5.3: Settings and notes of the first fingerprint test on the mobile client

Device 1 has been run without signal quality threshold on the first test and afterwards with a quite low value of -80 dBm[21] on tests 2 and 3. This device produced a fair number of fingerprints, but they generally had a bad retrievability, i.e. they were easily produced but not found back on subsequent test rounds. It was reported that the 52 fingerprints from the third test have been discovered in only 16 different locations and that the device had continued to produce new fingerprints even when the tester stopped at a location for longer. Device 2 had a harsher signal quality setting of -60 dBm and consequently produced only very few, sometimes no fingerprints.

To test the effect of speed, a fourth test was conducted where two N800 with the same settings have been mounted onto two bikes. The two bikes went out at the same time and rode alongside each other for three loops at fast, medium and slow speeds. The test resulted in one device constantly producing more fingerprints than the other. It was found that with the tested hashing algorithm there was a significant difference between devices, even when they had identical settings. No conclusion about the effect of speed on the fingerprint algorithm could be made from this test.

---

[21] Wi-Fi signals typically range between -80 dBm and -60 dBm

Retrievability of content has been tested in a fifth test where one person used an N800 to record some audio at a location within 500 m of the start point. Then a second person went out who received guiding instructions (“hotter”, “colder” and “neither hotter nor colder”) from a third person. It took about 10 minutes to navigate the person to a position which was then about 6 – 8 m away from the original location.

Findings from the Mobile Client Test

It was concluded that the 2 second Wi-Fi scan time worked well and that -90 dBm could be a reasonably setting for the signal quality threshold value. From a player’s point of view, the hotter/colder navigation seemed to be working okay and it was confirmed that the game content should be attached to Wi-Fi fingerprints. It was requested to visualise the internal graph structure of fingerprints as this was invisible in this test. Finally, it was suggested that two possible refinements to the fingerprinting algorithm should be tested:

1. use only strongest N access points for a fingerprint (N suggested to be 5)
2. introduce a fuzzy match to the fingerprint so that X % of the access points in a fingerprint need to be visible in order to be reported in that location

5.3.2.3 Initial Visualisations

The data gathered during the mobile client test has been used to generate initial visualisations. These visualisations have then been discussed with the artists and developers which led to the functional requirements for the tools that support the actual game development.

Graph Visualisation of the N800 Log-Files

A visualisation of the game-internal graph of fingerprints was already a clear requirement at this stage. Its feasibility has been confirmed with a proof-of-concept visualisation which was implemented in Java. It processed the mobile devices’ Hessian log-files, transcoded them to XML and visualised them using the freely available JUNG graph library (O’Madadhain et al., 2005) (62). This first version of the graph visualisation was not geo-referenced.

Processing the PDA Log-Files

The PDA log-files (recorded with the "WiFiFoFum" program) contained geo- and time-coded sightings of access points at a scan-rate of 1 Hz, but it did not explicitly contain empty scan-results, i.e. those scans where no Wi-Fi access point could be sighted. However, this data was contained implicitly in the logs due to the presence of time-stamps and a constant scan-rate. It could therefore be reconstructed by subtracting the number of unique time-stamps in the log-file from the maximum time-span covered by the file. This method of revealing implicitly contained empty-samples would potentially miss all non-samples that occurred before the first or after the last logged reading. But as our test route started at a location where Wi-Fi access points could be found, this theoretical problem did not apply.

Histograms

The described algorithm has been implemented to generate histograms showing the number of access points in sight – including none – during the test runs. This visualisation has been written in Python and utilised the freely available Matplotlib (Hunter, 2007) (63). The histograms have been created from all moments where 0 or more access points have been sighted and shows the distribution of number of access points in sight together with their percentage. A fitting curve has been applied whose peak shows the average number of access point in sight over the run. Each histogram is per log-file, i.e. per test run.

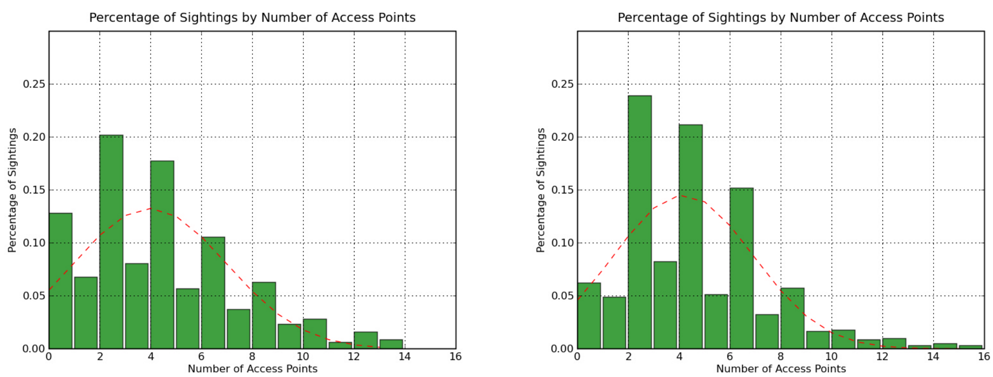

Figure 5.12: Percentage of sightings by number of access points
from two runs on two consecutive days

Figure 5.12 shows the histograms of two runs over the same route on two consecutive days. The number of sighted access points at any moment ranged from 0 to 16, with an average of 4 access points in sight in both runs. There were a significant amount of moments where no access points could be found – roughly around 10% of the time. There is also a visible tendency for peaks of 2, 4, 6 and 8 access points in sight which can be clearly seen in both histograms. These histograms are specific to the route, affected by the speed of data collection (which is ideally constant) and are also specific to the characteristics of the Wi-Fi adapter of the data collection device. They were thought to be useful for a quick visual verification of the log-file data, especially to identify moments of no access points and to find the average number of access points in sight. However, they have not been regarded as essential at this stage by the artists and have thus been discontinued.

Geo-Coded Visualisation

One of the initial user requirements was to enable the artists to get an idea of the radii of access points, i.e. the area in which a particular access point would be visible from the mobile device. This requirement was approached with scatter plots of all access point readings grouped into layers per access point BSSID. The visualisation was presented as a KML-file which could be loaded into Google Earth, where the user could selectively toggle layers of interest on or off (figure 5.13, left).

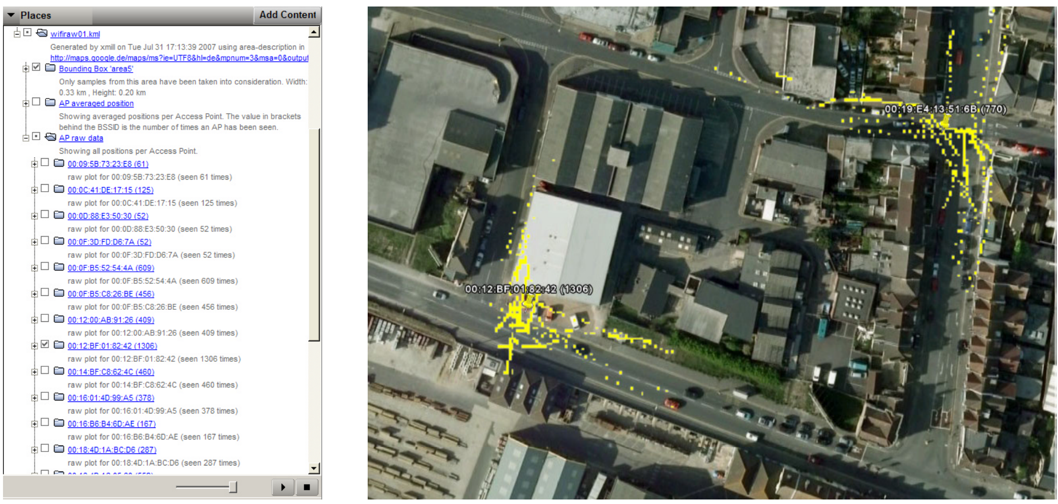

Figure 5.13: Scatter Plots of 2 Access Points in Google Earth (right), menu to toggle layers (left)

An excerpt from this visualisation can be seen in figure 5.13 (right) which shows a map of the test route overlaid with the scatter plots of 2 of the 22 access points that were discovered during the test. Each layer is labelled with the BSSID of the access point and the total number of discoveries of this access point in brackets. It can be seen how the 2 selected access points in the figure are visible around street corners and about 50 m into adjoining streets.

On the one hand this visualisation was perceived as informative in the sense that it showed the spread of access points overlaid onto a map. But on the other hand the visualisation was reported to be too hard to handle in an intuitive, interactive fashion. One specific concern was about its selection mechanism (toggle layers in the menu) being too data-centric. It was seen as more desirable to select layers the other way around, i.e. based on a position or location on the map. This visualisation has been written in Python and also used Matplotlib for generating the scatter plots. The custom data-access and KML export routines have been written with reusability in mind. This has proven to be useful in retrospect, as large parts of the code from this visualisation were still in use in later versions of the tools.

5.3.2.4 Requested Functionality for Tools

The discussion of the initial visualisations at the end of the exploratory phase led to the functional requirements for the tools that should support the development of Rider Spoke up to the premiere.

It was decided that the authoring of content was going to be done using the mobile game clients themselves, so there was no need for a separate authoring tool.

The chief concern on the infrastructure layer was to visualise the fingerprint graph topology, so that the artists could be sure to have a fair number of different locations for the game. This graph should also be used to visualise the locations of content.

The second most important request was to relate the graph topology back to the topography of the area where the game was going to be played. This was seen to be achievable by pinning fingerprints with known geographic coordinates to their positions and then exploiting the structure of the fingerprint graph to approximate positions of those fingerprints that do not have known coordinates. Combining these two requests led to a third: the artists were interested in the physical area that a fingerprint would cover (similar to figure 5.13).

On the physical world layer, it was confirmed to use freely available maps through online services such as Google Earth.

Furthermore, a specific request for a production tool emerged at this stage. The artists envisioned another tool that could be used during the event – not while setting it up – which would visualise the locations of content on the graph as well as the routes of players through the graph as they played the game.

Altogether, it was suggested to visualise data of 4 different kinds:

- Ground truth data, such as geo-coded access points
- Locations, i.e. graph of fingerprints
- Content
- Routes of players

At this stage the initial exploratory development phase ended. The target platform as well as a set of required tools had been defined. Attention now turned towards implementing the game and the required tools in time for the premiere.

5.4 Development Iteration 2: Towards the Premiere

The distributed system architecture of Rider Spoke has been finalised at this stage and the location mechanism has been extended based on the findings from the initial tests. New versions of the infrastructure visualisation tools have been implemented based on the feedback gathered during the first development iteration. The focus of this second iteration was on understanding this mechanism in its full context to ensure a successful premiere.

The description of this main development phase is based on the minutes, plans, reports and emails that have been written during this phase as well as the authors own notes and documents like the "Visual Design Document" – see appendix 6 (page 360). The latter has been written to establish a common understanding of the required visualisation tasks between artists and technicians. The document summarised the different potential visualisation task as perceived by the author after the initial exploratory phase. The artists where then asked to prioritise those tasks based on the anticipated usefulness from their point of view.

This section begins with a description of the new location mechanism, followed by a high level overview of the distributed system architecture. It then introduces the new graph visualisation tools and describes how they have been used up to and during the first outing of the Rider Spoke experience. The section closes with a post-event reflection on the artists' use of the supplied tools.

5.4.1 The Rider Spoke Location Mechanism

The first key idea behind the Rider Spoke location mechanism is that it focuses on Wi-Fi network locations rather than geographical positions[22], i.e. it doesn't rely on GPS coordinates to define the participant's geographical position. Instead, it continuously scans the Wi-Fi network environment and, after each scan, uses the list of discovered access points together with their respective signal strengths to build a Wi-Fi network fingerprint for each physical location. The idea to define locations like this is based on two assumptions:

1. Each fingerprint will be representative for the physical location it was recorded in and will only be reproducible in or near that location. Thus, the fingerprint can be used synonymously when referring to that location.

2. The granularity of locations across the game area can be less than would be possible with GPS as long as it is possible to tell locations apart from each other and supply the game players with a sufficient amount of locations to hide their content.

The second key idea behind the Rider Spoke location service is that it organises the locations in a graph structure. In graph theory, a graph is a set of elements (nodes) that are connected by edges. In the case of Rider Spoke this is easily done

[22] Refer to section 2.1.1 for a definition of terms

for each individual device as it travels through the physical environment over time, traversing through different network fingerprint locations. Starting at an initial location, the fingerprinter algorithm continues to return one location until the network environment has changed in such a way that the list of access points returned by the Wi-Fi scanning process is so much different that the fingerprint algorithm produces a different location. Keeping assumption 1 from above in mind, this also implies that the participant moved physically. In the case of a change of fingerprints, the transition between the previous and the current location will be recognised and stored as an edge in the graph structure.

Figure 5.14 shows the principle behind this graph evolution. The left image contains a very small graph consisting of only two locations and one recorded transition between them. Let the current location of a player in that image be location 2. Then, after some time, the fingerprinter algorithm on the player's device returns a third location, which causes that location (node) as well as the transition (edge) between locations 2 and 3 to be added to the graph (right image). This process is continuously repeated in the main loop of the application so that a player will always be either at one of the known locations or discovering a new one and thus gradually extending the graph. Note that it is also possible to discover only a new edge between already known nodes. In the example below – without adding any new nodes – it would still be possible to discover an edge between locations 1 and 3.

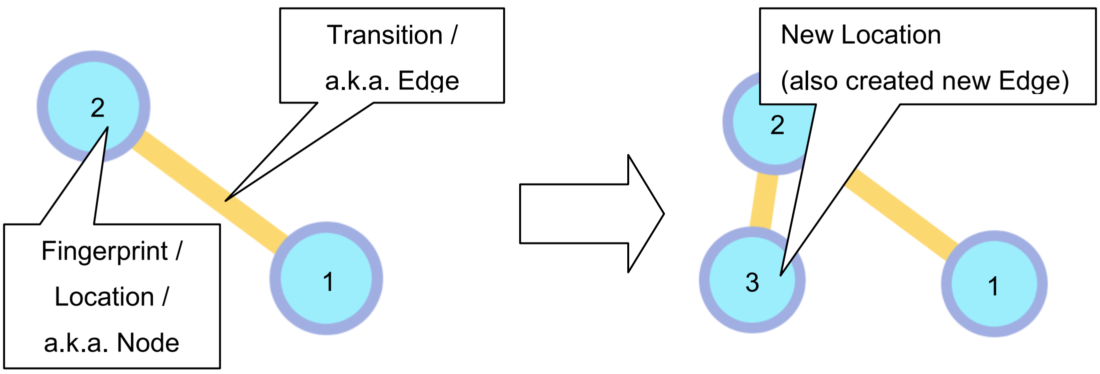

Figure 5.14: Rider Spoke Graph evolving, before (left) and after (right) discovering a new location

Over time, the game devices of each Rider Spoke player will evolve their own personal copy of the graph. Then at a later stage, graphs from different devices will be merged into a common graph which we call the game graph. Connectedness of the game graph is required by design as the Rider Spoke game engine measures distance in graph space to find "nearby" content that could be presented to a participant when that participant decides to listen to other players' answers.

If the game graph would not be fully connected it would mean that content left at certain disconnected locations could not be retrieved – as there would be no known route to it – which was unacceptable for the artists. To ensure that the game graph is fully connected, it is vital to start each device with a base graph that covers the network environment around the start point of the game so that players always start at a known location from which they can roam freely.

5.4.2 A Need for Understanding the Mechanism

Supporting the tuning of Rider Spoke's location mechanism was the major challenge for the author's work in this study. The behaviour of the location mechanism was obscure as the underlying fingerprint algorithm was quite complex. A considerable amount of time has been spent making this mechanism more accessible and understandable for the experience designers and its developers by discussing its inner workings and revealing the resulting graph of locations. Therefore an understanding of the algorithm is necessary for the reader to be able to follow the discussion. This will be kept short in this context, but more details on the algorithm can be found in appendix 5 (page 348).

In short, the fingerprint algorithm is tuned via 3 parameters (**signal quality threshold**, **maximum access points** and **overlap**) which determine how fingerprints are created from the scanned Wi-Fi data. Wi-Fi scanning happens every few seconds and returns a list of Wi-Fi access points, each recorded with their MAC address and the associated received signal strength. The algorithm then processes the list of discovered Wi-Fi access points in several steps. Access points that are below the **signal quality threshold** value are removed straight away, based on the grounds that they represent fluctuating background noise. Then only the strongest N access points are retained, with N being the **maximum access points** value. The resulting list of 1..N access points represent a fingerprint of the current scan results. In a final step, this fingerprint is fuzzy matched against a database of previously recorded fingerprints with the help of the **overlap** value. This value expresses to what extent a fingerprint needs to overlap with a previously known fingerprint in order to either snap to that location, or to create a new location. For example, the lists (1,2,3) and (1,2,3,4,5) have a 60% overlap, and the former would snap to the latter if the overlap was set to 60%. Choosing the highest overlap value of 100% would require the lists to match exactly, whereas choosing lower values allows for some fluctuation in the Wi-Fi scan results, as only a sub-set of access points from a previously known fingerprint need to be found in the current scan-results in order to snap to a location.

The location mechanism as used by Rider Spoke has many advantages that are listed in the appendix 5 (page 351). But it also has the significant disadvantage that it is quite complex and not straightforward to understand! The behaviour of the mechanism is largely dependant on the 3 parameters for the fingerprinter algorithm as well as on the Wi-Fi neighbourhood, which can vary significantly between cities and even between different parts of a city.

Tests of the earlier and simpler version of the location mechanism already verified that its settings have a huge impact on its performance. This was expected to be even more the case with the improved and more complex version of the location mechanism. We therefore sought to integrate a graph visualisation system into the overall distributed Rider Spoke system architecture (c.f. page 343) to help understand the behaviour of our location mechanism.

5.4.3 Visualising the Game-Graph

Visualising the game-graph was intended to help the artists in setting up the game as well as monitoring its progress. All required input data for the visualisations is obtained from the game server XML interface. This is the most important input channel, but there are also alternative means of loading in backup data from the game server (XML-encoded) and the mobile devices (Hessian-encoded).

5.4.3.1 Data Collection

Contrasting the initial network survey, no separate set of tools has been used for data collection in this phase of the project. Instead, all data for the visualisation was collected on the actual mobile game devices and uploaded to the game server. The optional GPS support has been enabled on some of the mobile N800 devices so that they would collect geo-coded fingerprint as well as access point data. This allowed for later fixing parts of the game graph to known geographical positions – a process that we called pinning. Pinning allowed us to visualise game locations in relation to their approximate real world context.

5.4.3.2 Technical Overview

The visualisation of the game graph is implemented in the GUESS Graph Exploration System (Adar, 2006). According to the author's own description on the project website, *"GUESS is an exploratory data analysis and visualisation tool for graphs and networks"*. It *"contains a domain-specific embedded language called Gython (an extension of Python, or more specifically Jython) which supports the operators and syntactic sugar necessary for working on graph structures in an intuitive manner"*. GUESS integrates the JUNG graph library which was already used for the initial visualisations in the exploratory phase. The graphical front end of GUESS uses the Piccolo toolkit (Bederson et al., 2004) and implements a hardware-accelerated zoomable user interface for an infinite 2D plane on which the game-graph is situated.

The Gython programming language is used to implement the Rider Spoke specific functionality of the new visualisation tool. Making use of the language's relationship to Python, most of the code is written in standard Python version 2.2 syntax. Incoming as well as outgoing XML data[23] are handled using the standard Python `xml.dom.minidom` class which is an implementation of the XML Document Object Model (DOM) interface.

[23] Outgoing data is encoded in KML, which is an XML based markup language

5.4.3.3 Functional Overview

When the tool is started, it loads the current game data from the game server via the XML interface, lays out the graph using a radial layout algorithm with the starting location (“FP501_”) at the centre and sets up the zoomable user interface to show the entire graph (see figure 5.15).

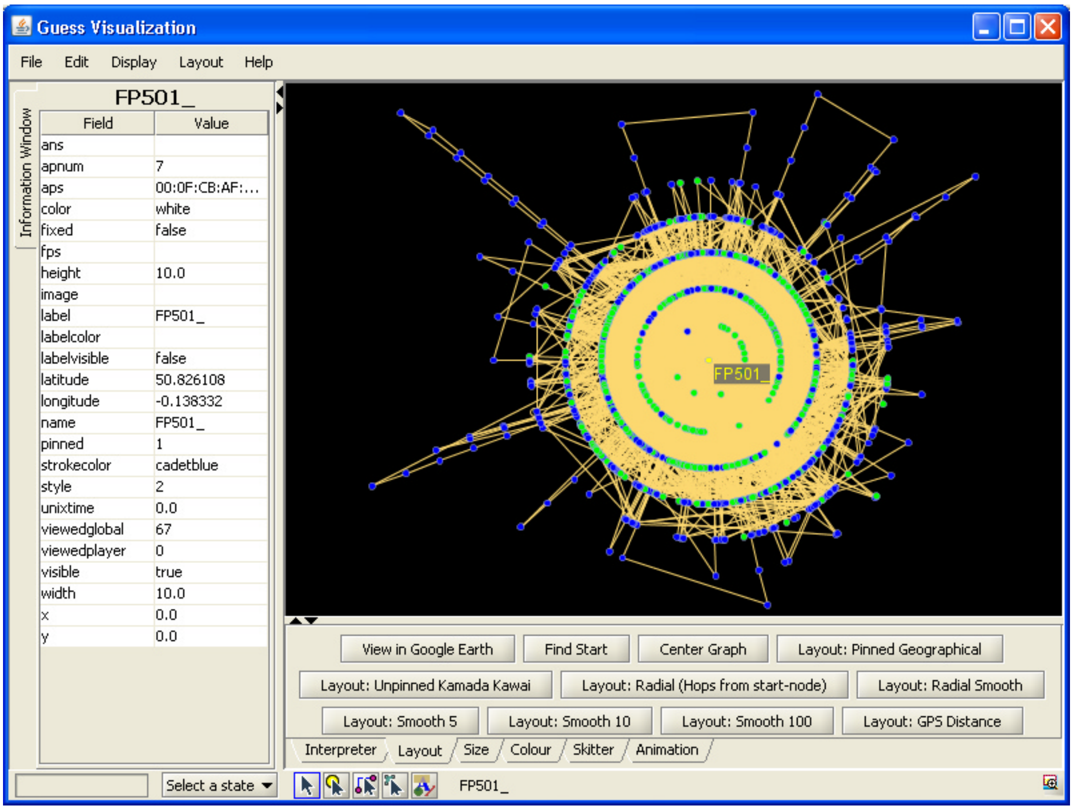

Figure 5.15: Screenshot of the graph visualisation tool in its initial state

The graphical user-interface of the visualisation tool is divided into three parts:

- The central “Graph Visualisation” window
- The “Information Window” to the left
- The “Docking Panel” at the bottom

The **“Graph Visualisation”** window shows the graph in its current layout. The user can pan the visualisation by clicking anywhere on the black background with the left mouse-button, holding the button down and moving the mouse horizontally and vertically. Similarly, the visualisation can be zoomed in and out by clicking and holding the right mouse-button and moving the mouse horizontally. Hovering the mouse over any of the nodes or edges in the graph causes that element's fields to be shown in the "Information Window". Left-clicking on any of the nodes or edges in the graph centres the view on the selected element.

The **"Information Window"** reveals all the element's fields. These fields are key/value pairs where the keyword is of type String. Some of the keywords are reserved by the GUESS system, e.g. "name"[24]. Any unreserved keyword can be used to store user-data. Table 5.4 lists the fields that have been chosen to implement the Rider Spoke visualisation.

Key	Value	Type
aps	Comma-delimited set of access point bssids for this element	String
apnum	Number of access point bssids contained in field "aps"	Integer
latitude	Geographic latitude of this element	Float
longitude	Geographic longitude of this element	Float
pinned	1 if element has known geographic position, otherwise 0	Integer
unixtime	Time of element creation in seconds since 01.01.1970	Float
fps	List of visitors to this location (including its creator)	List
ans	List of visitors that sighted an answer at this location	List
viewedGlobal	Number of visitors to this location (including its creator)	Integer

Table 5.4: Rider Spoke specific fields for nodes in the GUESS graph structure

The **"Docking Panel"** holds the custom user interface elements that have been added for the Rider Spoke graph visualisation. These are buttons which are thematically grouped into tabs: Layout, Size and Colour. There is also a tab for the GUESS internal Gython interpreter which can be used by a programmer to directly interact with the graph data if the supplied facilities are not sufficient.

[24] also: color, fixed, height, image, label, labelcolor, labelvisible, strokecolor, style, visible, width, x, y, size, __edgeid, node1, node2, directed

Using the **“Layout”** docking panel, the graph administrator can change the graph layout between radial, spring and geographical pinned:

- The radial layout option produces a number of concentric rings around the central fingerprint which give an indication of distance in graph space – locations that are one hop away from the starting location are shown on the first ring, locations that are two hops away are shown on the second ring and so forth.
- The spring layout option uses the Kamada Kawai algorithm (Kamada and Kawai, 1989) for drawing the undirected Rider Spoke graph in an aesthetically pleasing, but non-topographic way. Figure 5.16 is laid out using this method.
- The geographical pinned layout pins nodes with a known geographic position to their position and allows exploiting the graph structure to approximate the positions of the other nodes. Figure 5.19 is laid out using this method (and shows the graph in Google Earth, which can be achieved with a single button click in the graph visualisation tool).

All three ways of laying out the graph visually benefit from applying a smoothing algorithm after the initial layout. One way of doing this is to treat the nodes positions and the lengths of the adjoining edges as if they represent the energy in a physical system. This kind of smoothing can be quickly applied to varying degrees by selecting either of the “Smooth” buttons, which call the physicsLayout() function in the GUESS framework[25]. An example of its effect can be seen in figure 5.17, which has been initially laid out using the same spring layout as in figure 5.16, but additionally smoothed using this function.

The buttons on the **“Size and Colour”** panel allow changing these visually most striking node attributes according to different characteristics of the graph such as:

- Number of access points in fingerprint (apnum)
- Number of visitors to fingerprint (viewedGlobal)
- Fingerprints with GPS coordinates

[25] original implementation is attributed to http://www.schmuhl.org/graphopt/

- Fingerprints with content

Different combinations of the colour and size attributes have been defined in cooperation with the artists and coded to a number of quickly accessible buttons.

Finally, there are three buttons that are common to all panels:

- "Find Start", which focuses the zoomable user interface on the element with the lowest name, i.e. the first element.
- "Center Graph", which adjusts the zoomable user interface to show the entire graph.
- "View in Google Earth", which exports the graph in its current layout to a KML-file in the local file system and then launches that file so that it shows immediately in Google Earth. This function is most useful for visualising (partially) geo-referenced graphs.

5.4.4 Practical Use by the Artists

Having introduced the visualisation system, we now focus on how our artists used this tool in practice.

5.4.4.1 Using the Visualisation Tool for Setup

A few days before the premiere, the overall Rider Spoke software was regarded as functional so that the attention turned towards shaping the experience as a whole. A team of artists and technicians moved over to the venue and began setting up the game. This included fine-tuning the fingerprinter settings and assessing the resulting graph-growth in the visualisation tool.

The artists' main concern at this stage was that the overall experience would work well for the end-user, i.e. the paying participants at the Barbican premiere. Transferred to the location mechanism, this meant that there should be sufficient locations in the game where players could hide their content, without having to search for too long for finding a free spot in their limited play time. Consequently, the artists mainly used the graph visualisation tool to verify that there would be enough locations in the graph. They searched for a set of parameters for the fingerprint algorithm that would cause the creation of a fair number of locations. It was also required that locations should be retrievable later on.

Several test-players have been sent out with the mobile devices to collect fingerprint data in the wider area around the venue. After their return, the data has been taken off the devices, merged into the common game graph on the server, visualised using the tool, and then discussed with the team. This discussion included the number of locations in the graph, the connectedness of the graph, and the future graph growth when used during an experience with 500 participants. Promising settings have been used for a second test, where the artists went out to record some content and then subsequently tried to retrieve that content in situ.

Finding content tended not to work at this stage and the artists spent about half a day of trial and error testing different settings for their appropriateness. As none of the settings provided the desired effect of easy retrievability of content in the field, the artists finally decided that the rules for searching for content should be relaxed on the mobile game interface. They then triggered a team discussion to reflect on the settings that were tested so far and pick the most promising one. These settings were then tested once more, the resulting graph eyeballed in the visualisation tool, and finally chosen as the settings for the premiere. They are listed in table 5.5 below.

Parameter	Value
Signal Quality Threshold	-80 dBm
Maximum Number of Access Points per Fingerprint	unlimited
Overlap	60%

Table 5.5: Settings for the Rider Spoke fingerprint algorithm in London

These settings have been applied to a number of mobile devices which have then been taken out again by the test-players in order to generate the initial game graph. The image in figure 5.16 (left) shows a visualisation of the graph after merging two players' journeys. Please note that this is a topological image, i.e. orientation on the image does not reflect orientation on the compass. The graph is fully connected as both players started end terminated at the same place. The route to the lower left seems to have involved going out and coming back on the same route. The route towards the upper right appears to show an outward journey which then came back to the central point via some other way. There are also a few spikes of fingerprints on each route, where a device was only connected to a new fingerprint for a while and then flipped back shortly afterwards. This might or might not indicate that the player actually went on a route similar to this, e.g. in and out of an alleyway. But it is much more likely that it shows the beginning of the interconnectedness of the fingerprint graph. The right image in figure 5.16 visualises the graph an hour later when five players have been out. It can be seen that the fingerprint graph is now more densely interconnected.

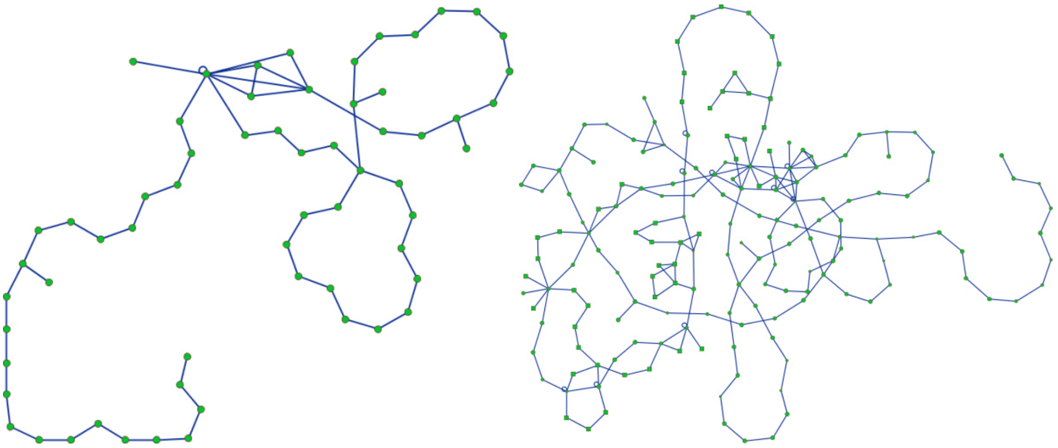

Figure 5.16: Fingerprint topology graph showing merged journeys of 2 (left) and 5 players (right)

Figure 5.17 (left) shows how the graph has further evolved after the journeys of 7 players have been merged into it. When comparing it with the previous state it can be seen that the long loose end from the right side of figure 5.16 (right) cannot be found anywhere in figure 5.17 (left). This is an indicator for the retrievability of fingerprints as one of the later test-players must have retrieved the loose end and subsequently reconnected it to the rest of the graph during their journey.

After a final check of the newly created game graph with the visualisation tool the artists went outside again and used the mobile game client to record some initial answers for the game. This was done so that also the players of the first day would have some audio-content available that they could listen to. Figure 5.17 (right) shows the state of the graph as the game went live. At this stage the graph consisted of 549 fingerprints and 730 edges, with 21 answers already in place (depicted by orange nodes).

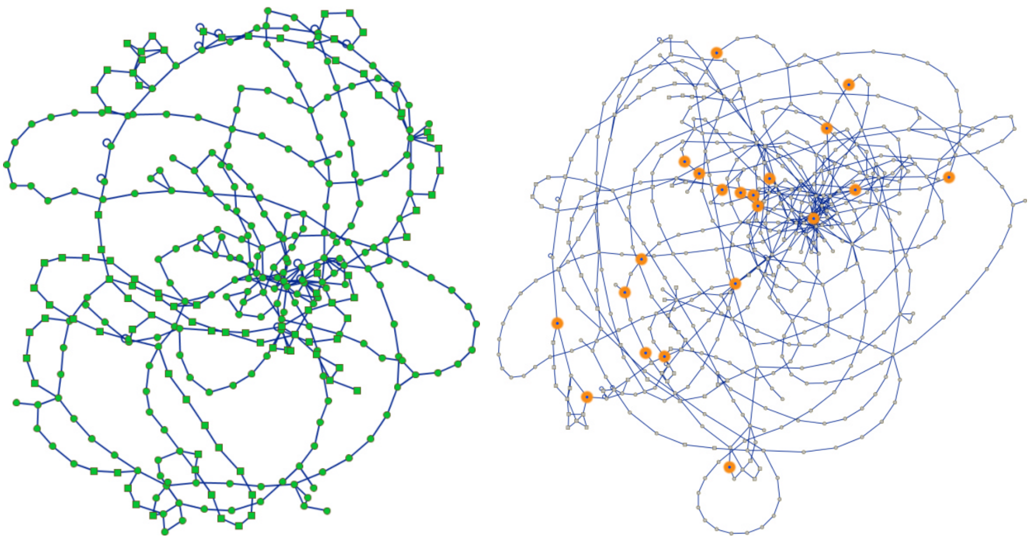

Figure 5.17: Fingerprint topology graph of 7 test-players (left) and before going live with 21 pieces of content placed on the graph (right)

As the game went live the parameters of the fingerprinting algorithm could not be changed anymore. Consequently the setup task ended and the visualisation tool was now only used for orchestration purposes, as we now discuss.

5.4.4.2 Using the Visualisation Tool for Orchestration

The primary use of the visualisation tool while the game was running was to verify that the graph grew as intended and to visualise the location of player generated content on the graph.

Some GPS data had been collected beforehand which could now be used to partially pin the graph to geographic positions. After selecting the graph layout and colouring in GUESS, the graph could be exported to the KML-format and automatically launched in Google Earth in order to make use of the available aerial images and street overlays, as well as the faster display engine and layering facilities of that program.

Figure 5.18 shows a Google Earth view of the area around the venue, with a pin marking the “Barbican Entrance”, and a superimposed street-layer revealing the street network and names of streets. A black layer with variable transparency has been introduced to reduce the contrast of the underlying aerial images and to increase the readability of labels. It can be seen that when leaving the venue onto "Silk Street", players could decide to either cycle to their right towards "Moor Lane", or to their left towards "Beech Street".

Figure 5.18: Area around the "Barbican Entrance" in Google Earth, showing streets

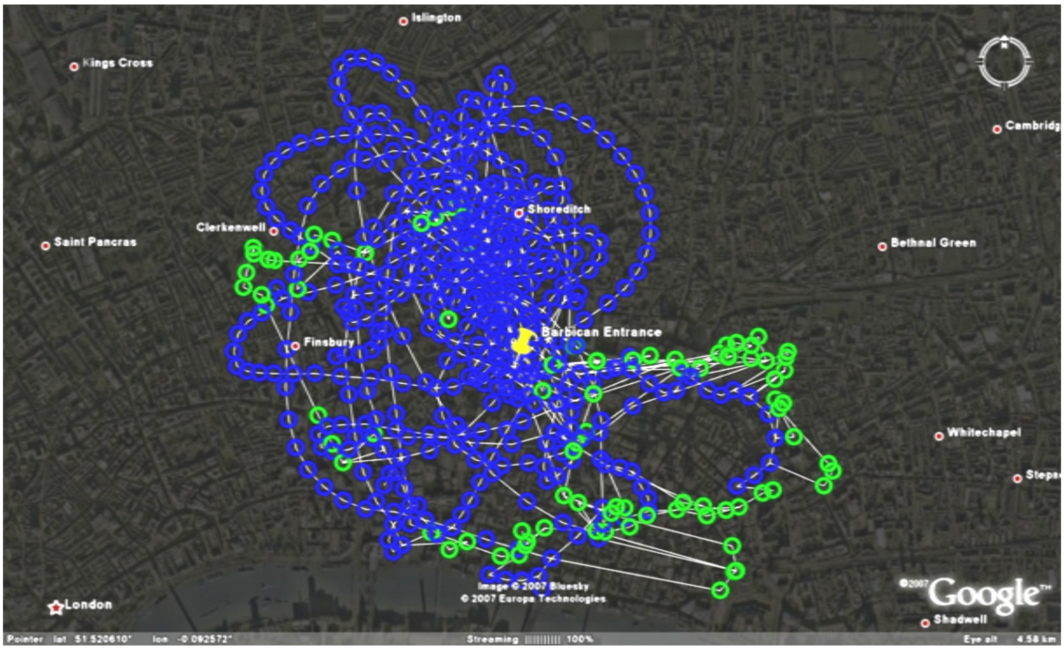

Figure 5.19: Fingerprint overlay in Google Earth (green = GPS, blue = no GPS)

Figure 5.19 shows a partially geo-referenced version of the game graph. In this image, green nodes represent known GPS positions and blue nodes have been filled in between by using a force directed graph layout algorithm. It has to be noted that the positions of the blue nodes on the map bears little reference to their actual positions, which were unknown. This image reveals that we had collected to little GPS data to make a useful geo-reference visualisation of the game graph.

Figure 5.20 uses a different colouring scheme and has a different purpose: it highlights the fingerprints with associated audio-content (in orange, left). The right image additionally highlights the route of a player, marked with yellow pins. This optional layer of information could be enabled by selecting a player's ID from the automatically generated KML.

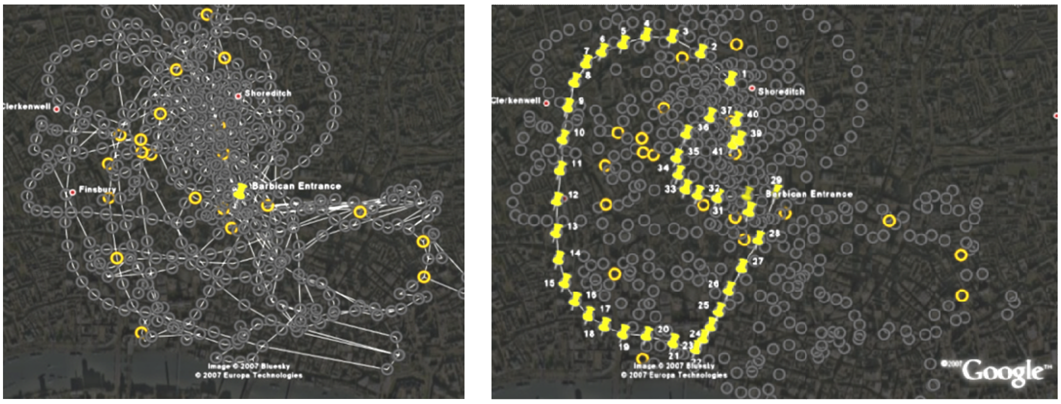

Figure 5.20: Highlighted content (left) and route of a player (right)

Figure 5.21: Using size to show frequency of visits to fingerprints (left), details on demand (right)

Size was also employed as a visual indicator to facilitate understanding the graph in its spatial context. Figure 5.21 (left) uses size to indicate the most frequently seen fingerprints in the graph, where bigger nodes have been more frequently visited by players. By clicking on a node of the graph in Google Earth, a pop-up HTML window appeared which contained the name of the fingerprint, its calculated position and the access point that make up the fingerprint. Figure 5.21 (right) shows these details for a small, i.e. less frequently seen fingerprint ("FP612_"), which consists of 13 access points.

In London, the game was running for five hours each day, for a total period of eight days. At the end of each day, the log-files were taken off the mobile devices, merged into the game graph and visualised using the tool. This was to verify that the overall system worked as designed for.

The artists used this orchestration process especially during the first four days, as they paid very close attention to the graph growth. On day four, they realised that the graph grew much too quickly. They originally expected that the number of locations in the graph would start to stabilise at some point a few days into the staging. This was based on the assumption that many popular real world locations would then already be linked to a virtual location. But this was not the case. In fact, the graph was continuing to grow at a steady and alarmingly high speed. The artists and the team feared that the graph would quickly become computationally unmanageable for the mobile devices and, as a consequence, slow down the player interface too much. To prevent this from happening, it was decided that the graph growth should be stopped completely for the remaining days of the show in London. This was accomplished by sending players out with a fixed data-set from day three and not merging their collected and recorded data back into the database. Although this slightly hampered the idea of user-generated content, it was seen as the only way out of a spiralling graph growth that would have ultimately stopped the game from working altogether.

5.4.5 Post-Event Reflections

The visualisation tools have worked for the London premiere when fed with given data from either the server or the device logs. The graph visualisation provided guidance to the artists and the team when they were setting up the game and tried to find the best settings in their manual process of testing a device outside, then visualising and assessing its collected data using the visualisation tool. The artists' primary concern at this stage was to find "good" settings for the anticipated graph growth based on two metrics:

1. There should be sufficient locations so that players could easily find a new location to hide their content.

2. The existing locations should also be retrievable so that players could find other players content.

Finding these settings proved to be a difficult endeavour as these two metrics appeared to be diametrically opposed to each other: a setting which would generate less locations in the graph would inevitably generate locations that are easier to retrieve and vice versa. Finding an acceptable compromise was especially hard and time-consuming, as it included repeated tests in the field, followed by visualisations and discussions in the studio. One such round-trip to test and discuss a single set of parameters took about an hour to complete; this was much too long. Also, this process was essentially trial and error. The artists articulated that better supporting the setup process was seen as essential.

The second most important request was to better support pinning, i.e. use combined GPS- and Wi-Fi data to geo-reference some graph nodes, unwind the overall orientation of the graph, and put everything on a map of the environment. This was already supported in our software, but it was not embedded deep enough into the overall workflow to efficiently support this feature. The mobile device software could be started with or without Bluetooth GPS support. As the deadline approached time became a premium and compromises had to be made. One of the compromises was to skip handling the Bluetooth GPS receivers during the busy setup time which bought us a quicker turn-around time at the expense of missing out opportunities for collecting GPS-data. The effect of this can be seen in figure 5.19, which shows a partially pinned graph from this period. In that image the blue non-GPS-pinned nodes are routed in big loops around the green GPS-nodes. These loops are artefacts from the force-directed layout algorithm and do not represent actual orientation of the nodes, which is misleading when interpreting the image, and should be fixed. Several possible strategies could be adopted to collect more GPS data:

1. make GPS a part of the game setup

2. use external data from third party sources

3. use external data gathered during own surveys

The first strategy is generally a good way to go and could be implemented by utilising a device type with built-in GPS, where GPS and Wi-Fi positioning could complement each others. However, this wasn't an option for Rider Spoke as the project had made financial commitments and bought many devices of a device type without built-in GPS. The second strategy would exploit existing geo-coded wireless access point databases such as Wigle.net. In fact, Wigle inspired the Rider Spoke location mechanism and one of the prototypes used during the initial exploratory phase could utilise data through a Wigle interface. Finally, the third option would establish a work-flow for a network survey to collect the required data as part of the process without being subject to being skipped if the deadlines are getting too tight. The required geo-coded access point data could be collected on the actual game hardware or on a device with similar characteristics.

The use of the visualisation tools for orchestration purposes was only of minor importance in London and was limited to assessing the graph growth as the game was running. This task is very closely related to the top-priority of finding the best settings for the location algorithm. It turned out that the graph grew faster than desired and generated too many locations. This resulted in too large datasets which needed to be loaded by the unoptimised mobile device software on start-up – a process which took 7 minutes after day one and up to 20 minutes (!) after three days. The team at the reception desk tried to counter-balance this effect by using the booking time-table to start devices well in advance of their anticipated use.

But this measure also had to be balanced against yet another issue: battery-life. Devices had to go out several times during an evening and needed to be charged in between. The artists' own experiments about battery-life resulted in a table which contained minimum charge times based on previous use, e.g. run-time, display brightness and volume settings. These charge times were calculated for switched off devices and switching device back on too early voided the calculations. Still, this happened, as both loading time and charging time had to be managed at the same time in order to send players out. This constant battle caused considerable slow-downs to the check-in procedure at peak times. To some extent, the artists managed to literally buy some time for the reception crew by inviting waiting players for a drink. But at some point the long loading-time and the minimum charging-time interfered too much with the check-in procedure for the players, and some players had been sent out with undercharged batteries, which compromised their experience in a few cases. On day four, the artists made the decision to use a fixed data-set from day three to prevent at least the loading times from growing any further. This drastic measure helped to stabilise the situation and smoothen out the check-in procedure. It was also not noticed by any player. Nevertheless, it became clear that the overall system needed to be much improved to overcome these problems.

The other orchestration related visualisation tasks have been reported to be of minor importance. It was regarded as interesting to visualise historic information, e.g. follow a player's route, see which questions they have answered, find out which locations have been retrieved the most or filter the output based on which question a player answered. But it was seen as more important to first get the setup right. The following section provides an account of our post-premiere developments and improvements to the setup process.

5.5 Development Iteration 3: Post-Premiere

Based on the feedback and observations from London premiere of Rider Spoke it became evident what needed to be improved in the overall system. First and foremost, the artist needed a better tool support for setting up the game. The manual trial and error process from the London outing was very time consuming and – unfortunately – not very insightful. It was thus neither efficient nor effective and demanded for instant remedy. This problem was approached with a simulator tool that allowed testing different settings within a matter of seconds, rather than hours. Second, we also improved the tool support for geo-referencing, so that the map based visualisation could later provide better insight into the places that people visited. In short, the post-premiere development was led by two objectives:

1. A simulator for replaying existing log-data with different settings was needed to speed up the setup process.

2. Geo-coding locations in the graph needed to be more deeply supported in the overall workflow.

New versions of the tools have been developed and tested on subsequent outings of Rider Spoke in Athens and Brighton. This section will introduce the improved tools and describe how they have been used to on these two occasions.

5.5.1 The "What-If" Simulator

The artists wanted to know which settings best support their requirement of having enough locations for players. This requirement was tethered with two additional requirements: locations must also be easily retrievable and there should not be too many locations, so that the mobile clients could handle the resulting amount of data gracefully. The settings for the London outing provided more than enough locations, i.e. too many. In London, the artists had to make the difficult decision of cutting the graph growth completely after day three. This should not happen again and thus the artists were keen to better understand the location mechanism to find settings that would take them through a complete staging.

The "What-If" simulator has been written in order to be able to quickly assess the effect of different fingerprint algorithm parameters on the growth of the fingerprint graph. It allows simulating *what* the graph would look like *if* a certain set of parameters would be used. To do so, the simulator operates on real data which needs to be collected beforehand during a site survey and then uploaded to the machine that runs the simulator. The "What-If" simulator consists of two components:

1. a server-side simulator that replays the logs of selected players (or all) for a given fingerprint algorithm parameter set and returns the resulting graph

2. a desktop analysis program that controls the server-side simulator, collects and assesses the resulting graphs for different permutations of the parameter settings and implements a graphical user interface

The server- and desktop components can be run on a single laptop. This allows taking the simulator on-site, doing a survey and analysing the data straight away without having to be connected to a remote server or going back to the office.

5.5.1.1 The Server Component

The server component of the simulator regenerates fingerprint graphs from scratch based on some input parameters. It is implemented as a Java Servlet and is part of the game server web application which means that it has direct access to the game data. When the server is running, the Servlet can be controlled with a simple HTTP GET request which accepts a number of URL encoded parameters that express which settings should be applied to which player-logs (see table 5.6).

Parameter	Type	Range	Comments
RssiThreshold	Integer	~ -100..-10	Rssi is measured in dBm
Overlap	Float	0..1	0.5 means a 50% match
MaximumAps	Integer	0..n	0 means infinity
Players (optional)	String (comma-separated)	0..n	Names of players whose traces should be replayed

Table 5.6: Parameters accepted by the "What-If" simulator server component

The first three parameters are required by the fingerprint algorithm and work exactly as described in section A5.2.1 on page 348. The optional fourth parameter allows selecting which player-journeys should be taken into account when rebuilding the graph; if omitted, all available player-journeys will be used. The fingerprint algorithm is then chronologically applied to the raw Wi-Fi scan results of the journeys of the selected players, which gradually builds up the graph. This building process usually takes a fraction of a second when applied to the log-files of a typical one day site-survey, but obviously the processing time depends largely on the amount of data and the speed of the server. Once completed the graph is returned as XML in the body of the HTML response.

5.5.1.2 The Analysis Program

The analysis program is the graphical user-interface for the designers. It controls the server-side component, collects and ranks the returned graphs based on different metrics and presents its results in a sortable spreadsheet-view. The program is implemented as a Python desktop application and uses the wxPython cross-platform GUI toolkit to build a familiar looking user-interface. The SQLite database engine is used to implement a local cache.

The analysis program works by allowing the user to generate large numbers of different permutations of the fingerprint algorithm parameter set. The user can employ a graphical wizard to define so called "experiments". An experiment is a configuration that defines which range of values should be simulated for each of the three parameters and how many steps should be used to cover that range. The experiment configuration also contains the player parameter as well as meta-data such as experiment name and notes.

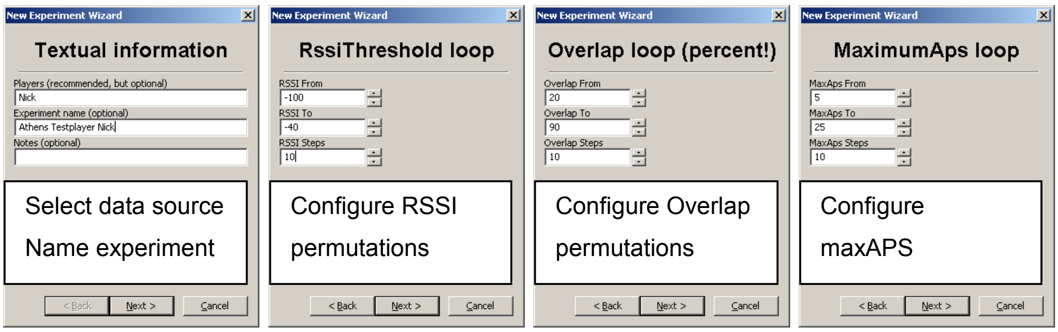

Figure 5.22: Annotated screens of the "New Experiment Wizard", as used to configure data source and parameter permutations for an experiment called "Athens Testplayer Nick"

Figure 5.22 shows the screens of the wizard as it has been used to setup an experiment called "Athens Testplayer Nick" which concentrated on rebuilding different graph-scenarios based on data of that player. The three parameter ranges for the permutation have been defined and each of the ranges is sampled in ten steps. This results in 10x10x10=1000 different fingerprint graphs which will be generated by the server-side simulator and then stored in the local database cache for fast access. Initial generation of the 1000 graphs in this experiment took about 5 minutes with the server running on the same laptop machine. Subsequent access through the cache was then about two orders of magnitude faster.

Once the experiment has been run, all of the returned graphs will be ranked based on several different metrics. Each metric is encapsulated in a ranker class which implements a common abstract interface. Each ranker class is fed with all graph permutations before being asked for its ranking. A ranker class simply ranks each graph according to its encoded metric, e.g. “number of fingerprints in graph”. When queried for the graph rankings, a ranker class sorts its internal results and returns two values per graph in a list of pairs:

1. absolute value of the assessed metric for this graph (e.g. “63”)
2. relative rank of this graph amongst all assessed graphs in percent based on the metric (e.g. “100%”, which means that this graph was the best)

The obtained multi-dimensional results of the experiment are presented to the user in a sortable spreadsheet report view as seen in figure 5.23. Each row in this view represents one fingerprint graph with all its parameters and rankings in different columns. Table 5.7 summarises the columns of the report view.

Column	Meaning
Index	Internal variable (order of creation)
RSSI	Parameter “RssiThreshold”
Overlap	Parameter “Overlap”
MaxAPS	Parameter “MaximumAps”
SampleID	Internal variable (for caching)
%_metric_	Rank of graph amongst all graphs according to _metric_
#_metric_	Absolute value assigned by _metric_

Table 5.7: Columns in the report view

What If Report for Experiment 28: Athens Testplayer Nick

Index	RSSI	Overlap	MaxAPS	SampleID	%: Finge...	#: Fingerprints	%: Edges	#: Edges	%: FP Views / #FP	#: FP Views / #FP
541	-100	83	5	8749	100.0%	63	100.0%	78	020.7%	1.52380952381
542	-90	83	5	8750	095.1%	60	098.7%	77	019.7%	1.5
560	-90	83	11	8768	077.0%	49	081.8%	64	024.2%	1.61224489796
559	-100	83	11	8767	075.4%	48	079.2%	62	022.2%	1.5625
554	-90	83	9	8762	075.4%	48	080.5%	63	024.7%	1.625
548	-90	83	7	8756	075.4%	48	080.5%	63	024.7%	1.625
596	-90	83	23	8804	073.8%	47	079.2%	62	028.6%	1.72340425532
590	-90	83	21	8798	073.8%	47	079.2%	62	028.6%	1.72340425532
584	-90	83	19	8792	073.8%	47	079.2%	62	028.6%	1.72340425532
578	-90	83	17	8786	073.8%	47	079.2%	62	028.6%	1.72340425532
572	-90	83	15	8780	073.8%	47	079.2%	62	028.6%	1.72340425532
566	-90	83	13	8774	073.8%	47	079.2%	62	028.6%	1.72340425532
553	-100	83	9	8761	072.1%	46	077.9%	61	026.6%	1.67391304348
595	-100	83	23	8803	070.5%	45	074.0%	58	025.4%	1.64444444444
589	-100	83	21	8797	070.5%	45	074.0%	58	025.4%	1.64444444444
583	-100	83	19	8791	070.5%	45	074.0%	58	025.4%	1.64444444444
571	-100	83	15	8779	070.5%	45	074.0%	58	025.4%	1.64444444444
565	-100	83	13	8773	070.5%	45	074.0%	58	025.4%	1.64444444444
547	-100	83	7	8755	070.5%	45	075.3%	59	024.6%	1.62222222222
487	-100	76	7	8695	070.5%	45	075.3%	59	024.6%	1.62222222222
481	-100	76	5	8689	070.5%	45	079.2%	62	026.3%	1.66666666667
577	-100	83	17	8785	068.9%	44	079.2%	62	028.7%	1.72727272727
499	-100	76	11	8707	068.9%	44	072.7%	57	026.0%	1.65909090909
488	-90	76	7	8696	068.9%	44	075.3%	59	026.9%	1.68181818182
422	-90	69	5	8630	068.9%	44	080.5%	63	028.7%	1.72727272727
421	-100	69	5	8629	068.9%	44	075.3%	59	032.3%	1.81818181818
536	-90	76	23	8744	067.2%	43	072.7%	57	029.4%	1.74418604651
535	-100	76	23	8743	067.2%	43	074.0%	58	026.6%	1.67441860465
530	-90	76	21	8738	067.2%	43	072.7%	57	029.4%	1.74418604651
529	-100	76	21	8737	067.2%	43	074.0%	58	026.6%	1.67441860465
524	-90	76	19	8732	067.2%	43	072.7%	57	029.4%	1.74418604651
523	-100	76	19	8731	067.2%	43	074.0%	58	026.6%	1.67441860465
517	-100	76	17	8725	067.2%	43	074.0%	58	026.6%	1.67441860465
511	-100	76	15	8719	067.2%	43	074.0%	58	026.6%	1.67441860465
505	-100	76	13	8713	067.2%	43	074.0%	58	026.6%	1.67441860465
500	-90	76	11	8708	067.2%	43	075.3%	59	028.5%	1.72093023256
482	-90	76	5	8690	067.2%	43	071.4%	56	026.6%	1.67441860465
518	-90	76	17	8726	065.6%	42	070.1%	55	027.3%	1.69047619048
512	-90	76	15	8720	065.6%	42	070.1%	55	027.3%	1.69047619048
506	-90	76	13	8714	065.6%	42	070.1%	55	027.3%	1.69047619048
494	-90	76	9	8702	065.6%	42	066.2%	52	021.6%	1.54761904762
493	-100	76	9	8701	065.6%	42	067.5%	53	022.6%	1.57142857143

Figure 5.23: Report view of an Experiment

The report view shows the three graph-defining parameters "RSSI", "Overlap" and "MaxAPS" as columns on the left and the resulting relative graph ranks ("%") and absolute ranking values ("#") per metric on the right side[26]. The report view can be sorted on any column by clicking on the column header. A small triangle will appear to indicate the sorting order which is either ascending or descending. The graph behind each row can be visualised by right-clicking on the row and selecting "View Graph" from a pop-up menu which will launch the existing graph visualisation tool with the selected graph.

For example, figure 5.23 is sorted on descending column "%: Fingerprints" which means that it will display the graph with the most number of fingerprints in the top row (underlined with a dark bar in the image). It can be seen that the highest number of fingerprints found in any of the simulated graph-scenarios was "63". Looking towards the left of the row it can be seen that the parameter settings of this graph were: "-100" (RSSI), "83" (Overlap) and "5" (MaxAPS). This setting also produced the most number of edges in the graph. Comparing the relative

[26] "Index" and "SampleID" coloumns are internal values and can be disregarded by the user

ranks of the graphs in the different columns reveals that there is an almost linear correlation between number of fingerprints and number of edges in the graphs, but other metrics such as "FPViews / #FP" follow a different order.

The number of metrics to be used for ranking the graph scenarios is technically unlimited. Five different metrics have been implemented for the setup of Rider Spoke in Athens and Brighton – they are summarised in table 5.8. The first three metrics (Fingerprints, Edges, Fingerprint sightings) are the most basic as they just count raw entities – the higher the value the better the rank.

Metric	Description
Fingerprints	Ranks the number of fingerprints in a graph
Edges	Ranks the number of edges in a graph
Fingerprint sightings	Ranks the number of fingerprint sighting logged while building the graph. A fingerprint sighting is logged every time the fingerprint algorithm moves a player to a different fingerprint.
Fingerprint discoveries / number of fingerprints	Ranks based on a ratio which is calculated by dividing the sum of all fingerprint discovery values by the total number of fingerprints.
Fingerprint discoveries / number of users	Ditto, but divides by total number of users instead.

Table 5.8: Metrics used for assessing the generated graphs

The last two metrics are more advanced, as they try to express one of the artists' initial verbal requirements: "content and fingerprints must be easy to find". This requirement is modelled by giving higher ranks to those graphs where individual fingerprints are revisited more often. Both of these metrics are based on a member variable of the fingerprint class called "viewedGlobal" which counts how many players, including its original creator, have discovered a particular fingerprint, i.e. the "viewedGlobal" value of a fingerprint is either 1 or higher. Both metrics first sum up the "viewedGlobal" values of all fingerprints in the graph to test and then put the resulting sum into a relation with either the total number of fingerprints or the total number of players by dividing it through the respective value. The resulting unit less ratio value expresses the retrievability of fingerprints (and thereby content) in the ranked graphs, where a higher value means better retrievability. The user can assess the resulting rankings via the sortable report view and make better informed decisions about the parameter settings in less time.

5.5.2 Collecting more GPS Data

It had been identified that more GPS data was needed to better support the geo-coding of fingerprints. The intended workflow of using some of the game-devices in combination with an optional Bluetooth GPS device had not been picked up by the artists for the London premiere when the deadline pressure rose. The Athens and Brighton outings of the game provided two more opportunities to test the applicability of this approach as well as different means of data collection.

5.5.2.1 Using an Existing Data Collection Tool

As already used in the initial exploratory phase (page 199), the “WiFiFoFum” (WFFF) war-driving software for Windows Mobile PDAs (version 2.2.12) has been offered for data collection. The software has again been used on a HP hx2700 PDA running Windows Mobile 2003 SE with position data coming from a Haicom HI-406BT Bluetooth GPS receiver with SiRF Star III.

5.5.2.2 Using a Custom Data Collection Tool

A third way of collecting GPS data for a geo-coded fingerprint graph was tested in Brighton. The Nokia N800 devices have been used to collect raw access point data with the actual game client. Additionally we supplied the artists with a custom data collection tool written in Python for Series 60 that ran on a Nokia N95, a mobile phone with built-in Wi-Fi and GPS. This mobile survey tool was constantly scanning the Wi-Fi environment and logging the visible access points as well as the received GPS coordinates in a time-stamped text format. Inspired by the audio-alerts of the WFFF program it also made use of sound, but in a slightly different way. Rather than playing a sound upon discovery of a new access point it implemented the same Wi-Fi fingerprinting algorithm as the game device and played a notification sound upon the discovery of a new fingerprint. It also played a different sound upon retrieving an already known fingerprint, so that the user could get an initial feeling for the granularity of the fingerprint algorithm and the extent of fingerprints while doing the data collection. Despite these additional audio notifications, the programs main use was of course to collect the required raw GPS and Wi-Fi access point data in a format that could be used later on.

5.5.2.3 Incorporating Wigle Data

In addition to collecting our own survey data, we also incorporated existing GPS and Wi-Fi data from third party sources. As mentioned earlier, the Wigle.net community maintains a database of geo-coded access points that is fed by different individuals, predominantly in the United States and Western Europe. The website can be used to browse through all available access points within a bounding box of interest. Using this facility we implemented a custom parser that transcoded the data from the website into a comma separated value (CSV) format.

5.5.2.4 Local Database

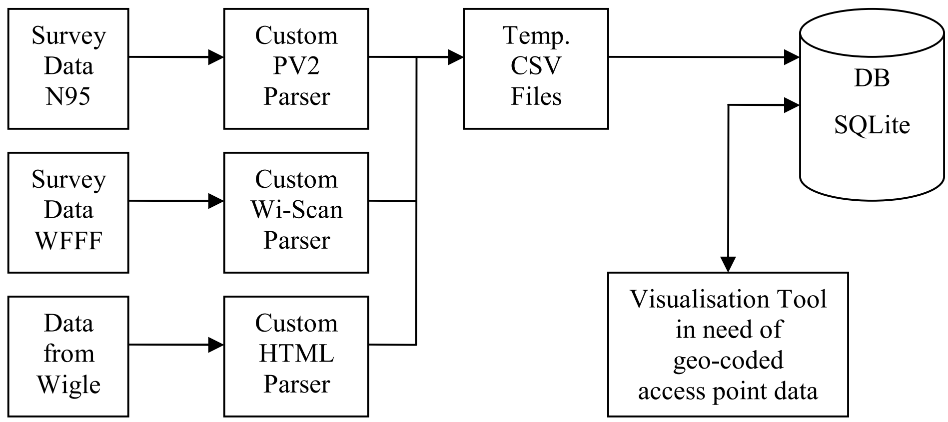

Figure 5.24: Local database unifying data from different sources

A local database has been used in order to unify all the geo-coded access point data coming from different sources, i.e. Wigle.net, WFFF and our own survey tool, as can be seen in figure 5.24. The survey data from our own N95 data collection program are saved in a Place Lab compatible “PV2” text-format, the data from WFFF is stored in the “Wi-Scan” text-format and the Wigle data is stored in a series of HTML files. A conversion script triggers several custom parsers that transform the data into a series of temporary CSV files which satisfy the schema of our database.

The SQLite library (64) has been used to realise the database in a server less fashion. After generating the temporary files, the script then calls the SQLite binary to build the database file from the CSV files. Subsequent access to the data by the visualisation tools is then handled in a unified way via the Python `sqlite3` module.

5.5.3 Setting Up in Athens

The setup for Rider Spoke in Athens began with a preliminary site-survey and feasibility study by the artists' intern one month before the game went live. This was followed by the actual network survey a week before the premiere.

5.5.3.1 Collecting Data

We provided the artists with two setups for data collection. The first setup comprised of an N800 game-device with GPS-enabled game binaries and an external Bluetooth GPS unit. The second setup comprised a Windows Mobile PDA with built-in Wi-Fi, an external Bluetooth GPS receiver and the WiFiFoFum data collection software.

Device	Pro	Con
N800 w. GPS	• actual game device • produces fingerprint data	• might be tricky to set up
PDA w. GPS	• easier to set up • responsive UI with audio feedback upon discovery of APs • can give an idea of the granularity	• not the actual device • fingerprints cannot be produced from the collected data

Table 5.9: Pros and Cons of the two data collection setups for Athens

The reason for providing two alternative solutions for the same task was that the artists wanted to conduct a network survey and feasibility study well in advance of the actual event and allocated some time for it during their site-visit in Athens. For financial and logistical reasons this survey had to be completed in the allocated time and thus we wanted to provide a back-up solution in case of any problems. We had provided them with a written explanation of the pros and cons of the two setups so that they could decide which one to use (see table 5.9).

The artists originally decided to use the N800 setup because the hardware was the actual game device and it was thus most trusted to produce the right kind of data. Unfortunately the artists had to surrender this setup on site when they realised that is was too complicated for them to handle and they could not actually get it to work on their own. The alternative PDA solution was easier to operate and so the artists used it for their site-survey. Although it did not produce exactly the right data – WFFF did not log the signal strength of the discovered access points – the program was helpful to give the artists an idea of the granularity of the Wi-Fi network environment at the site in Athens. This was also helped by a very responsive user interface which could be set to play a sound upon the discovery of a new access point so that the person doing the survey would get immediate audible feedback of the invisible radio environment, similar to a Geiger counter.

Using the WFFF PDA data collection setup, the intern managed to discover 659 access points during 6 hours, spread over 2 days. She was advised to regularly save the data and thus saved to about 40 different files (or one file about every ten minutes). Although the collected data did not contain all the required data it proved to be helpful as it provided an insight into the availability of Wi-Fi in the desired target area north of the Acropolis of Athens, for which no third party data could be found.

The data collection route (see figure 5.25) was designed to provide a fair cut through the different areas most likely to be visited by the players. Athens is quite hilly, especially around the Acropolis to the bottom of the image and it was expected that players would stay on the north side of the hill. The route therefore covered the neighbourhood of the venue around the train station in Thissio to the left and to the top of the image, the tourist area around Monastiraki, Plaka and the ruins of "Ancient Agora" in the centre (c.f. figure caption "Ancient Angora), as well as the Hellenic Parliament near Syntagma Square to the right side of the image. Each yellow pin in figure 5.25 represents one or more access points.

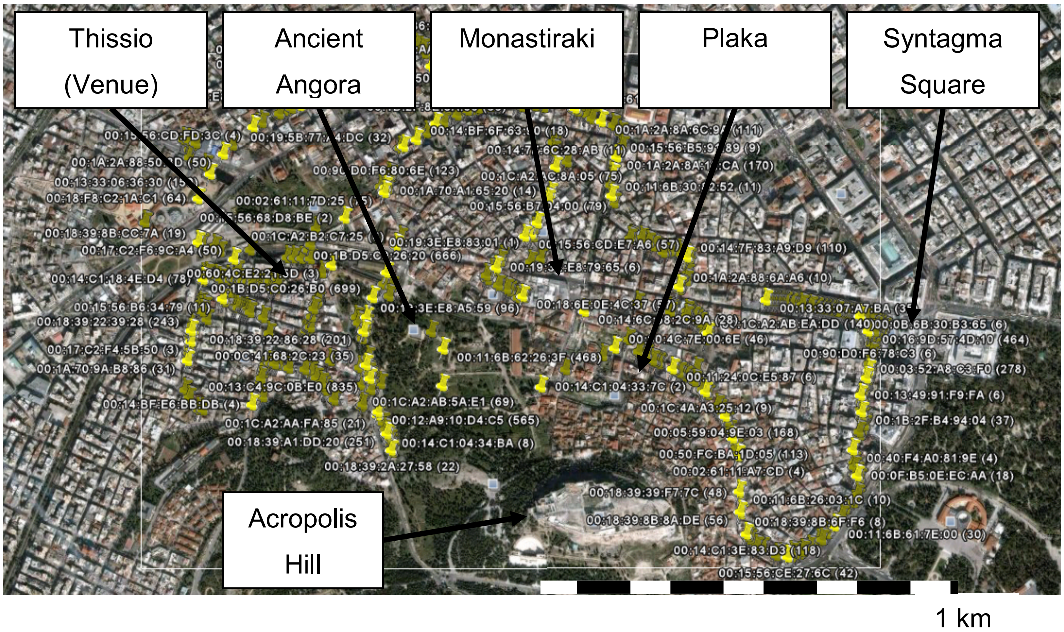

Figure 5.25: Trail of 659 Access Points discovered in Athens

The initial site survey served as an indicator that the target area would have sufficient Wi-Fi access points to make the game work. But the real network survey still needed to be conducted when the whole team moved to Athens for setting up the game a week prior to the premiere.

Data collection for the survey was scheduled for a one hour time-slot and thus several people had to be sent out in parallel to different areas of the city in order to speed up the whole process. The planning of routes had been done on paper maps, with one of the artists proposing and the others agreeing to the proposed routes. Two people were assigned to cover Monastiraki and Plaka to the north of the Acropolis, one person was assigned to cycle around the venue in Thissio and one person was told to strike out in one direction for half an hour in order to get a sense of how the Wi-Fi environment changes between different parts of the city. So the artists tried to cover as much different types of terrain as possible in the available time. All routes were laid out with a bit of overlap between them so that the traces from the different routes could be connected afterwards.

5.5.3.2 Choosing the Settings

After the survey, the artists gathered back in the studio at the venue. All device logs have been uploaded to a local development laptop which housed a copy of the Rider Spoke game server as well as the analysis program. This setup has then been used for choosing the most appropriate settings for Rider Spoke in Athens.

The artists' primary concern was the graph growth. From their experience of the Rider Spoke outing in London, the graph grew much too fast and quickly reached a point where it became unmanageable. For the Athens outing they wanted a much slower growth. The only constraint was that there should always be enough locations for players to leave their answers, as the artists didn't want them to waste several minutes of their one hour play time on finding a free location. With this important requirement in mind, we began to use the simulator.

Exploring Settings for the RSSI Parameter

After a thorough introduction to the analysis tool and the different settings that it covers the artists questioned if the signal strength (RSSI) parameter as it has been used for London would be a good choice for Athens as well. It was their impression that the street layout in Athens was a bit more spacious and with a different mix of houses, and they expected that this would have an effect on the choice of the RSSI parameter. In London, a setting of -80 dBm has been used for the RSSI parameter, meaning that any signal weaker than this was ignored by the fingerprinter algorithm. And in fact: a look into the database revealed that over two-thirds of the samples collected during the survey in Athens were below -80 dBm and would thus be disregarded if the RSSI setting was not going to be changed! The conclusion was that the RSSI parameter should be lowered to -100 dBm to also make use of all those access points whose received signal strength was below -80 dBm.

The Overlap Parameter

There wasn't much discussion about the overlap parameter, which basically means that the artists were quite happy with the setting they chose for this in London (60%). But they used the analysis tool to verify the effect of modifications of this parameter. They concluded that lowering it to below 50% would be bad as the amount of fingerprints generated would be significantly less. Adversely they were frightened of increasing it above 66% as the amount of fingerprints generated in that case would be too much, according to their estimates. It was decided to leave this parameter at 60% for Athens.

Maximum Number of Access Points (BSSIDs)

This parameter was discussed more deeply by the artists. For London it had been set to "unlimited", based on the assumption that one would be at a different location if one found a significantly different mix of BSSIDs, even if they contained BSSIDs from a known fingerprint.

BSSIDs	Overlap	Fingerprint
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	-	A
1,2,3,4,5,6,7,8,9,10	50% of A	B
1,2,3,4,5	50% of B	C
1,2,3	60% of C	C
1,2	50% of C	D
1	50% of D	E

Table 5.10: Generation of fingerprints from the same range of access points (overlap of 60%)

Table 5.10 summarises the assumed effect. For example, if one found a place where the device could receive signals from twenty previously unknown access points (say 1-20) that would generate a fingerprint "A". If subsequently one then found any mix of ten access points from that range (say 1-10) and the required minimum overlap was set to 60% (and 10 out of 20 would only be 50%), one would have found another fingerprint "B". Table 5.10 continues this list down to a single BSSID and shows how the algorithm would generate five fingerprints.

Using the tool the artists came to the conclusion that the "unlimited" setting was not the best choice. While glancing over the report view, still looking for settings that generated a high number of fingerprints, they discovered a visual pattern which attracted their interest. This pattern is highlighted and explained below.

What If Report for Experiment 26: Athens Initial 5 Testplayers

RSSI	Overlap	MaxAPS	%: Fingerprints	#: Fingerprints	%: FP Views / #FP	#: FP Views / #FP
-28	50	7	000.0%	0	000.0%	0
-100	58	7	053.1%	165	009.5%	1.76363636364
-92	58	7	053.1%	165	011.9%	1.84242424242
-84	58	7	047.3%	148	019.2%	2.07432432432
-76	58	7	035.3%	113	007.7%	1.70796460177
-68	58	7	018.5%	64	001.7%	1.515625
-60	58	7	004.8%	24	007.7%	1.70833333333
-52	58	7	000.0%	10	004.4%	1.6
-44	58	7	000.0%	0	000.0%	0
-36	58	7	000.0%	0	000.0%	0
-28	58	7	000.0%	0	000.0%	0
-100	66	7	054.1%	168	007.9%	1.71428571429
-92	66	7	054.8%	170	008.2%	1.72352941176
-84	66	7	048.6%	152	016.8%	2.0
-76	66	7	035.6%	114	007.0%	1.68421052632
-68	66	7	018.5%	64	001.7%	1.515625
-60	66	7	004.8%	24	007.7%	1.70833333333
-52	66	7	000.0%	10	004.4%	1.6

Figure 5.26: Wave-Pattern that caught the artists' attention

Figure 5.26 depicts a report view where all visible rows have the same MaxAPS setting of "7". The encircled column "#: Fingerprints" shows a wave-pattern caused by the different lengths of the numbers written in each cell. Apparently those settings that caused the generation of many fingerprints also took up more typographic space in the "#: Fingerprints"-cell in the report view; unintentionally providing a bar-chart like visualisation that the artists discovered and exploited.

What If Report for Experiment 26: Athens Initial 5 Testplayers

RSSI	Overlap	MaxAPS	%: Fingerprints	#: Fingerprints	%: FP Views / #FP	#: FP Views / #
-100	82	5	100.0%	302	000.0%	1.4602649006
-92	82	5	097.6%	295	000.8%	1.4847457627
-92	82	7	079.8%	243	005.7%	1.6419753086
-100	82	11	079.1%	241	004.2%	1.5933609958
-100	82	7	079.1%	241	004.4%	1.6016597510
-92	82	11	078.4%	239	004.4%	1.6025104602
-92	82	9	078.1%	238	003.9%	1.5840336134
-100	82	9	078.1%	238	005.3%	1.6302521008
-100	74	7	076.0%	232	004.5%	1.6034482758
-92	74	7	075.0%	229	005.9%	1.6506550218
-92	82	13	071.9%	220	007.3%	1.6954545454
-100	82	13	071.6%	219	005.9%	1.6484018264
-92	82	15	070.5%	216	006.6%	1.6712962963
-84	82	5	069.5%	213	006.7%	1.6760563380
-100	82	15	069.2%	212	007.7%	1.7075471698
-92	82	17	068.2%	209	008.8%	1.7416267942
-92	82	23	067.1%	206	007.1%	1.6893203883
-92	82	21	067.1%	206	007.1%	1.6893203883

Figure 5.27: Looking for settings that generated the most fingerprints

The discovery of this coincidence and the closer analysis of those settings that generated many fingerprints led the artists to question the MaxAPS parameter. It appeared to them that the MaxAPS parameter was less important for generating many fingerprints as expected and seems to have a different effect – but they were not sure which one, yet. To check which settings actually produced the most fingerprints they sorted the report on the "%: Fingerprints" column (figure 5.27).

As expected, the setting which caused the generation of the most fingerprints (highlighted by the blue bar in figure 5.27) had a very low RSSI value of -100, which was already a confirmed choice for the artists. The setting also had a high overlap value of 82% (the highest value in the simulated range); which was no surprise, as this parameter was known to increase the granularity of fingerprints. But contrary to the artists' belief the MaxAPS value was very low (5) – it was actually at the lowest end of the simulated range rather than striving for infinity. It appeared to the artists that limiting the number of access points in a fingerprint caused only the strongest of the received signals to be taken into account. The artists called this effect "capping". They imagined it reduces the noise in the generated fingerprint landscape and would also allow empty areas, where there are no strong access points nearby, to be filled with fingerprints generated from more distant access points. This, of course, required a lowered threshold value for the RSSI parameter, which the artists already confirmed.

Parameter	Value
Signal Quality Threshold	-100 dBm
Maximum Number of Access Points per Fingerprint	6
Overlap	60%

Table 5.11: Settings for the Rider Spoke fingerprint algorithm in Athens

Equipped with a feeling for what each parameter actually does, the artists used the report view more autonomously. With a few candidate settings in mind they weighted the number of fingerprints against the ranked retrievability by eyeballing the effect of different settings in the report view and visualising promising candidate settings in the graph visualisation tool. They eventually arrived at the settings for Rider Spoke in Athens which are listed in table 5.11.

5.5.4 Setting Up in Brighton

The setup for Rider Spoke in Brighton started with a network survey, followed by an analysis of the data and the selection of the settings. Due to a streamlined process and an emerging routine, all of this was completed within one day.

5.5.4.1 Collecting Data

The main data collection was done in about one hour by one of the artists. He planned his route so that it would cover different types of terrain, e.g. the city centre of Brighton, suburban areas as well as the sea-front. The artist took two hardware setups on his route:

1. An N800 running the actual game – without GPS
2. A GPS-enabled N95 phone running our custom data collection tool

In addition to the artist, the author of this thesis took a second N95 for a walk into the city centre, collecting data in some of the smaller alleyways and on the main routes down to the seafront. He also collected data during an evening walk along the seafront upon his arrival on the day before the survey.

This two-tier setup proved to be working smoothly. The N95s were running without problems and their audio signals provided an insight into the granularity of the Wi-Fi fingerprint locations while doing the survey, or even just while doing an evening stroll. Our data collection software automatically saved its logs every 10 seconds so that no user-interaction was required. The N800 used by the artist crashed twice during the one hour. This was due to a bug in an optional survey-feature (play audio signals like the N95) which had been implemented in the last minute, but not properly tested. The feature has subsequently been removed again.

5.5.4.2 Choosing the Settings

After the survey the collected data had been uploaded to a local web-server running on a development laptop. It appeared that 1766 different access points have been discovered by the N95s and that the N800 game device, using the same settings as in Athens, generated 120 fingerprints during one hour – or roughly one fingerprint every 30 seconds.

The geo-coded access point data collected by the N95s has been used to geo-reference the fingerprints for the visualisation. This has been done in two passes. The first pass calculates an average position from those access points in every fingerprint whose position is known in the database. This usually results in a significant proportion of fingerprints being geo-referenced already. The second pass then exploits the graph network and calculates positions for the remaining fingerprints by averaging the geo-referenced positions of the neighbouring fingerprints. This geometrical approach is superior to our previous force-directed approach as it avoids repulsion between nodes[27].

Figure 5.28 shows the resulting geo-referenced graph of the one hour data collection survey. In this visualisation green means that nodes have been geo-referenced in the first pass and red that nodes have been geo-referenced in the second pass. The disconnected end of the graph on the bottom left is caused by the crashing of the N800 device during the survey.

The distribution of fingerprints looked fine at first sight. The device had been able to generate fingerprints on all parts of the route, i.e. in the city centre (to the left), in the suburbs (to the right) and at the seafront (at the bottom). But at second sight, it became apparent that there was a cluster of fingerprints in the city centre around Jubilee Street, where dozens of virtual locations referred to a single real location: a busy street corner. Figure 5.29 shows two close-ups of this cluster. The left image is a zoomed-in view from the same session that figure 5.28 was taken from. Clicking on the encircled button takes the user to the Google Earth view of

---

[27] Repulsion of nodes is a desired effect of force-directed layout algorithms which strive for visual clarity rather than correct geo-referenced positioning as it is required in this case.

the same graph (right image), where it can be more closely examined in relation to a meaningful aerial image background. Looking at the number of people visibly sitting in cafés and bars on the street corners, one can deduce that this area is indeed a very popular and busy location.

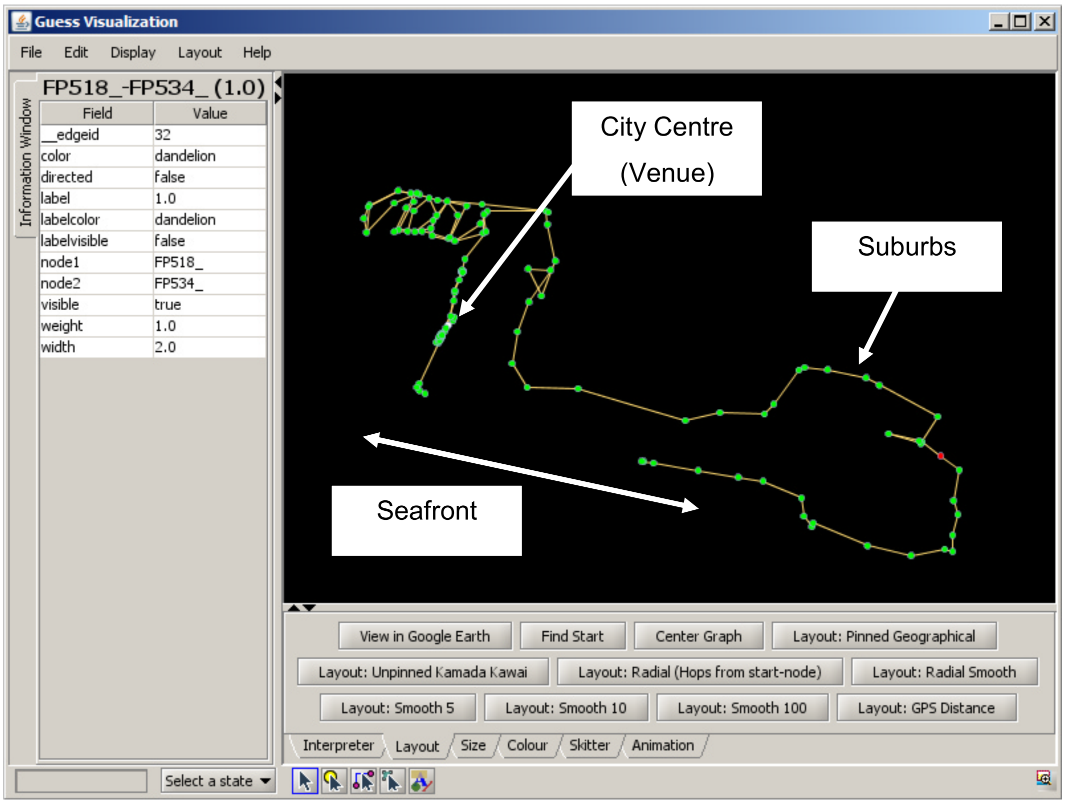

Figure 5.28: Setting 1: geo-referenced graph from the network survey in Brighton (120 FP)

Figure 5.29: Setting 1: cluster of fingerprints visualised in Guess (left) and Google Earth (right)

Apparently, this was the street corner where the artist was kitted up. Kitting up took at most 10 minutes and involved only small physical movements within the range of a few metres around the artist's bike. He started his survey as soon as the devices were turned on and good to go. It thus became evident that with the current settings, and by simply residing in a busy neighbourhood, the fingerprint algorithm would generate a multitude of virtual locations. This was not acceptable, as the artists aimed for a balanced graph which should not contain too many locations and whose locations should be retrievable on the spot. Trying to find a certain virtual location on this street corner with a mobile device running with the current settings seemed impossible. Furthermore, it was to be expected that the graph would grow out of proportion with these settings. This was even more the case, as the starting point of the game was just 100 m to the north of this street corner, so that many players would travel through this spot, potentially creating many more fingerprints.

Altogether, the settings that caused the creation of this graph, although previously used in Athens, were not satisfying for our artist. He already knew how to use the "What-If" simulator and the graph visualisation tools, and was eager to find a better solution. He used our tools to quickly iterate over different settings and arrive at the final settings which have been used for Rider Spoke in Brighton.

The remainder of this section is an account of this process. It is based on notes which contained his chosen settings and the judgements that he has spoken out aloud (settings and comments are summarised in table 5.12). The screenshots have been generated afterwards to avoid interfering with the artist's workflow.

Setting 2 (-90 / 6 / 50)

The artist's first idea was to slightly reduce the overlap (to 50%) as well as slightly increase the required signal strength (to -90 dBm), while leaving the MaxAPS parameter untouched. As the resulting graph was still very crowded, he sought more drastic measures to see some effect.

Setting	RSSI	MaxAPS	Overlap	Comments
1	-100 dBm	6	60 %	Unsatisfactory start settings, 120 fingerprints / h
2	-90 dBm	6	50 %	Still very crowed near Jubilee Str.
3	-50 dBm	6	70 %	Still crowded near Jubilee Str., too sparse around St. James Str. (suburban), largely disconnected!
4	-60 dBm	6	70 %	Ditto
5	-70 dBm	6	70 %	Slightly better, but still too sparse, lots of fingerprints with only one or two access points
6	-80 dBm	6	70 %	Better, but still too sparse in suburban and too crowded near Jubilee Str.
7	-80 dBm	7	70 %	Ditto
8	-80 dBm	7	60 %	Better, said to look similar to London
9	-90 dBm	5	30 %	Jubilee Str. is really clean, while the rest of the route is still nicely connected, selected because this setting was the winner of the “FPViews / #FPs” ranker, artists wanted to have a few more fingerprints
10	-90 dBm	4	40 %	Bad try, quickly rejected
11	-90 dBm	7	40 %	Looks good to the artist, used as the game setting for Brighton, 67 fingerprints / h

Table 5.12: Iteration of settings for Rider Spoke in Brighton as checked by the artist (yellow background denotes settings which will be reviewed in the following)

Setting 3 (-50 / 6 / 70)

The artist raised the bar for the required signal strength to -50 dBm and also increased the overlap value to 70%. The resulting graph was a big surprise. It is depicted in figure 5.30. This graph is bad in two ways:

1. The cluster of fingerprints around Jubilee Str. got even bigger

2. The suburban and seafront part of the route were completely disconnected

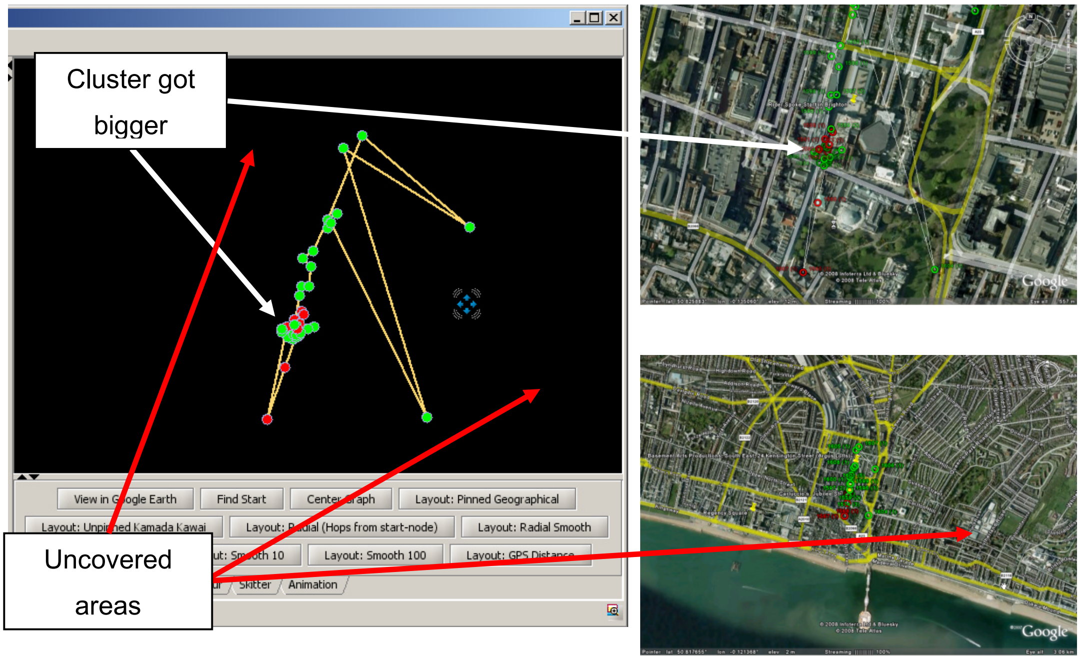

Figure 5.30: Setting 3 - A worst case scenario in Guess (left) and Google Earth (right)

It became apparent that the first issue was caused by increasing the overlap value to 70%. The second issue was caused by requiring a much too strong input signal, which could only be found very close to access points, i.e. in the city centre. This setting classified the majority of the received signals as too weak to be considered and thereby completely disconnected parts of the game area, including the popular seafront. A game with these settings would have been a disaster to play.

Setting 6 (-80 / 6 / 70)

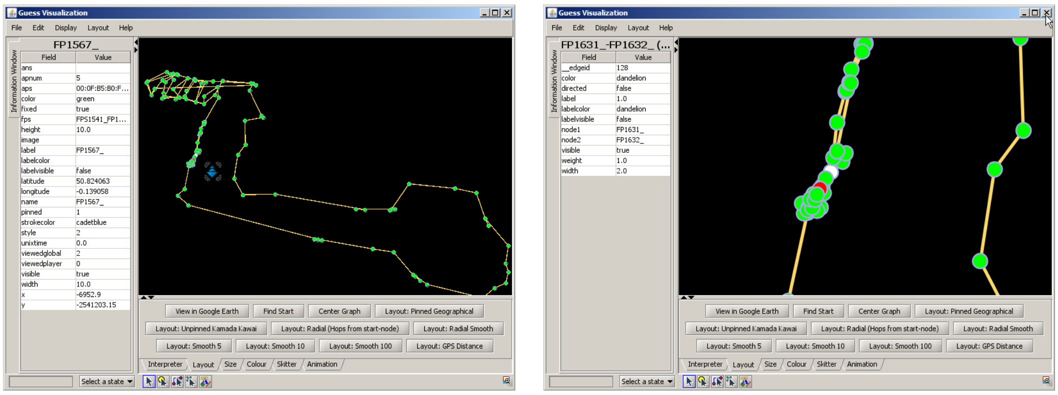

Figure 5.31: Setting 6 - The graph is still too crowded

The artist acknowledged the effect of the signal strength parameter and lowered it to -80 dBm, the same value that has been used in London. He was much more pleased with the appearance of the graph for these settings, but still found it to be too crowded near Jubilee Street (figure 5.31).

Setting 9 (-90 / 5 / 30)

Feeling a bit out of luck with his trial and error approach, he sorted the report view on the ranking in the “FPViews / #FP” column (which expresses retrievability) and visualised the simulated graph for the winning settings (figure 5.32).

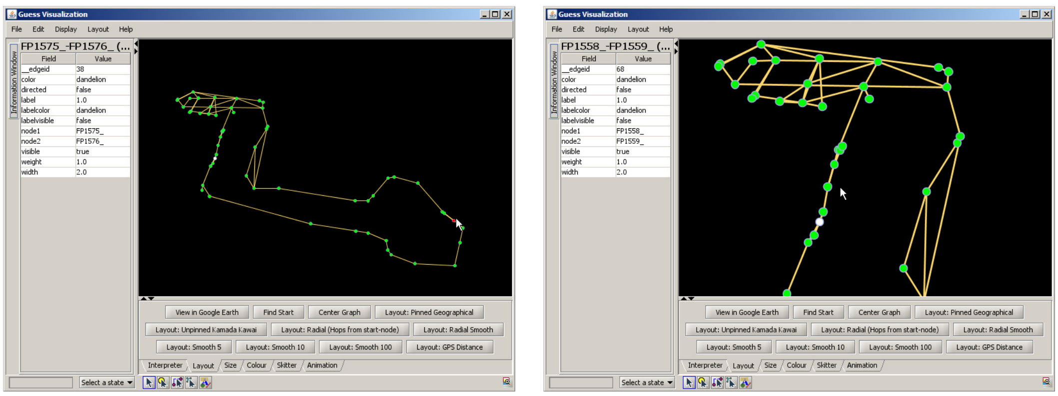

Figure 5.32: Setting 9 - The graph is clean

The Overlap and MaxAPS parameters were much lower than he would have tried, but the layout of the graph was appealing to him: it was very clean around Jubilee Street, yet fully connected in all parts of the city.

Setting 11 (-90 / 7 / 40)

Being generally happy with the previous graph settings, the artist just wanted to generate a few more fingerprints so that players would have potentially more hiding places available. This finally led him to the settings as they are summarised in table 5.13 and the simulated graph layout as depicted in figure 5.33.

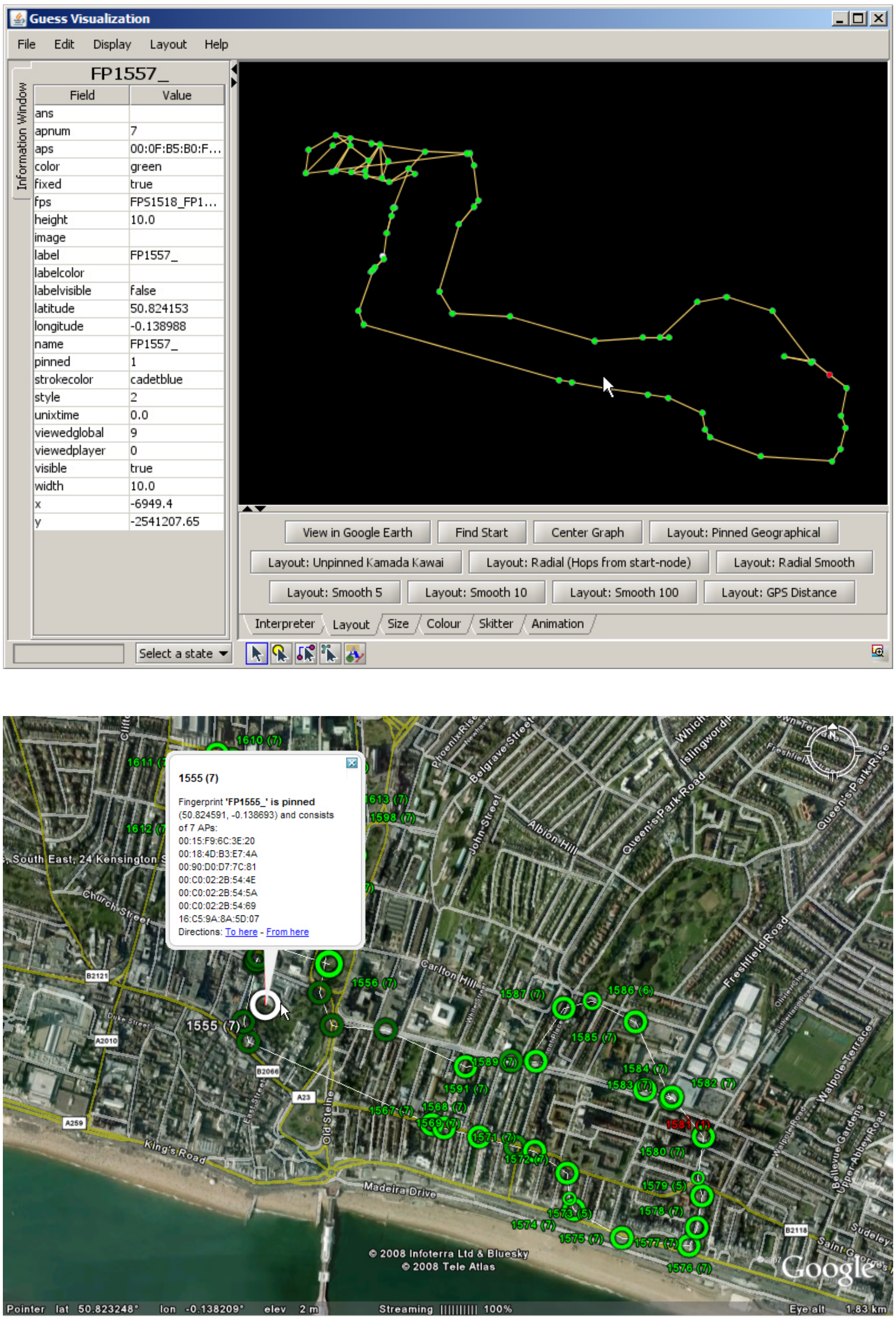

Figure 5.33: Setting 11 - Final simulated graph (top), Google Earth view with node size set to scale according to number of access points in a fingerprint (bottom)

Parameter	Value
Signal Quality Threshold	-90 dBm
Maximum Number of Access Points per Fingerprint	7
Overlap	40%

Table 5.13: Final settings for the Rider Spoke fingerprint algorithm in Brighton

To finally verify that the fingerprint algorithm would make best use of the surrounding Wi-Fi infrastructure, the artist used the custom "Scale: APNUM" function of the visualisation tool to scale the size of each fingerprint icon according to the number of access points it contains (figure 5.33, bottom). It can be seen that the vast majority of the icons visible in the map view are of the biggest size – which represents the capping value of 7 – with only a few slightly smaller icons on screen that represent fingerprints with less access points in them.

5.6 Post-Event Reflections

Rider Spoke and its supporting tools, as presented throughout this chapter, passed through several iterations of development. This process led the game from the exploratory phase to the public premiere in London, and to two further stagings in Athens and Brighton. When looking back on this process, it can be stated that all stagings went well, but that the Athens and Brighton stagings went slightly smoother behind the scenes. While this can certainly be attributed to an emerging routine of all who were involved, it was also due to a generally improved setup workflow. We have seen how the setup workflow evolved with each iteration for the three outings of the game and finally culminated in a suite of tools that allowed the artists to easily set up the game in less than a day.

5.6.1 Workflow in London

For the initial outing, the setup process included discussions, manual tweaking of settings on the mobile devices, tests outside, and visualisations of the collected data back in the studio. This process was performed in several loops until the designers accepted the performance of the game as passable. Setup was a very time-consuming process which caused a hectic rush in the team as the premiere approached. It also ultimately led to the dismissal of GPS data collection which was not integrated tightly into the workflow and could thus be pushed aside to free up some more time. While this was a reasonable compromise in sight of the deadline, it came at the expense of further insights into the workings of the location mechanism and the spread of locations across the game area. It also led to the selection of settings which made the game graph grow so quickly that its sheer size caused slow downs to the game runtime on the mobile devices. This problem became apparent after the third of eight days of performance. By design, the settings could not be changed without loosing all mappings between content and location, which meant that the graph would have continued to grow at high speed for the last five days of the experience. This would have caused further slow downs to the game client and affected the end-user experience in a serious way. Thus, the artists had no other choice than approaching the problem with drastic measures; they decided to cut the graph growth completely by providing players on the last days only with a static dataset from day three. While this ensured a smooth show, it also meant that the content generated by players on the following days was never exposed to anybody, as it was not merged into the common game graph. Although this was not communicated to participants and did not get noticed by anybody, it was still artistically unacceptable. Consequently, the task of estimating graph growth needed to get better tool support, and this was addressed for the Athens outing.

5.6.2 Workflow in Athens

The setup for Rider Spoke in Athens provided an opportunity to use the what-if simulator (page 226) for estimating graph growth. This simulator provided the artists with a tool to better predict the likely evolution of the game graph. The simulator operates on historical data that gets collected by the artists themselves. They plan their data collection routes to provide a fair mix of environments that cover those areas of a city that are deemed most relevant to the game. This process is described in detail on page 233. In essence, the simulator put the artists in a position to better understand the location mechanism of the game. While the London setup resembled more of a trial and error approach on the ground, the setup for Athens actually involved human assessment of simulated graphs for different settings in the studio. The artists could now quickly test the effect of different settings on their own historical data and get an immediate visual response from the system as to *what* the graph would have looked like *if* they had used those settings. This proved to be effective, as the artists could now invest a reasonable amount of time into a single site survey that covers a mix of areas of their own selection and then subsequently reuse that data as often as they liked, without having to go outside again. Thus, the simulator allowed for a more focussed work on the problem of choosing the right settings without frequently interrupting that task. One of the artists was quickly able to use the analysis front-end to the simulator autonomously and through this process gained a better feeling for the location mechanism and the effect of its parameters on the graph growth. This fostered a discussion with the author over dinner, and led to the discovery of an effect that the artist called "capping" (page 353). The artist finally arrived at settings that looked right for him in the simulator. After a final test outside on the mobile devices, these settings were confirmed to be suitable and were used for the public performance. The artists reported that the staging of Rider Spoke in Athens went well. Still, for a spatial analysis, the quality and the amount of the collected geo-referencing data was not right. This problem was approached in a further iteration of the tools for the Brighton outing.

5.6.3 Workflow in Brighton

The Brighton outing of Rider Spoke basically employed the same workflow as in Athens. But we additionally equipped the artists with a Wi-Fi and GPS logging device which collected the data that was required for geo-referencing the graph. The artist took it with him on his site survey, together with the actual mobile game device that collected the main survey data. The data from these two devices was later synthesised before visualisation. The visualisation tools now allowed for the game graph to be eyeballed in reference to aerial images of the area. The process of choosing the right settings is described in detail on page 242.

What is especially interesting about the Brighton setup process is how it revealed the importance of geo-referencing. It was only with the help of the geo-referenced version of the graph that the artist could find balanced settings that produced a fair amount of locations over the area. Without this feature, he would have missed the ill cluster of virtual locations that were created in one physical location next to the start of the game (see figure 5.29, page 242 ff), as well as the apparent badness of the setting where he raised the signal quality threshold to -50 dBm (page 246). Both settings would have resulted in an uneven distribution of locations over the game area and in an unstable location mechanism. This would have caused a bad retrievability of content and ultimately a bad end-user experience.

Reportedly, this was not the case, and the game ran smoothly in Brighton. At one point after a few days, the artists received feedback from their players who found it very hard to find any more free locations for recording their content. This must be seen as a proof that the retrievability of locations in Brighton was very good, as many popular spots had already been occupied by other players. The artists were initially concerned about this feedback, because they aimed to always have enough locations for their players (page 237 ff). But it quickly proved that this scarcity of locations was not a problem of the location mechanism and could be quickly solved with the change of a single variable in the system. The presence of content at different locations is not only controlled by the location mechanism, but also by the quality of the content, as ranked by the artists at the end of each day (page 346). Each piece of content has a rating between 1 and 5 (best), and the Rider Spoke content control system allows setting a minimum quality threshold value that a piece of content needs to satisfy in order to make it into the data-set for the mobile devices. It appeared that the artists had initially used a threshold value of 1, i.e. every piece of content was accepted. By raising the threshold value to 3, they were able to remove a lot of substandard content and thereby freed up many locations and made room for new answers by other players. Thus they effectively increased the quality of the experience on two fronts with the change of a single variable. This marked the end of the Rider Spoke development process, as the artists were now able to control the full system on their own.

This ends the discussions of Rider Spoke as well as the presentation of the core thesis material. The following chapter provides a more general discussion of the lessons that were learned from the two studies.

6. Discussion

This chapter reflects on the feasibility of the conceptual framework, after it has been put into practice and studied on two occasions. The chapter starts with a brief discussion of the benefit of the infrastructure layer, as it has been proposed in chapter 3 (page 118). It then drills more deeply into how we supported the design work of our artists with tools and workflows that linked work in the field with work in the studio (page 256), and how our artists made sense of location in their designs (page 267). This user-centred view is followed by a discussion of the engineering side (page 274), which refers back to the challenges that were set out in chapter 3, and provide ideas for system-level improvements (page 285). Sections on best practices (page 290), and an outlook with recommendations (page 298), close the discussion chapter.

The core idea of this thesis is the authoring of location-based experiences with three layers of information at hand:

1. the *physical world layer*, with representations of the target physical environment

2. the *infrastructure layer*, with representations of the wireless computing infrastructure across the physical environment

3. the *content layer*, with representations of digital media and their associated scheduling rules

From this three layer model, the infrastructure layer is the key innovation of this thesis. A second innovation that facilitated the creation of the infrastructure layer is the abstract location model, as presented in section 3.3 (page 131). Both studies in this thesis implemented location mechanisms that map onto this model. The main benefit of the abstract location model in relation to infrastructure visualisation is that it enabled us to algorithmically determine if certain sensor input data resolved to one location or another. In combination with GPS-based geo-coding, this allowed for visualising the spatial spread of locations, their granularity, and of the transitions and boundaries between them.

The literature review concluded that previous authoring solutions for location-based experiences do not provide sufficient information about the infrastructure to their users, which ultimately caused a gap between what experience designers expect and what wireless technology can provide. It was anticipated that revealing the characteristics of the underlying wireless infrastructures to the experience designers would help to increase their awareness of the technological uncertainties that underpin their work.

6.1 The Benefit of the Infrastructure Layer

The empirical knowledge gained during the development of the location-based experiences Love City (chapter 4, page 141) and Rider Spoke (chapter 5, page 181) suggests that this kind of information about the wireless infrastructure is indeed useful to the designer. Both experiences integrated a wireless location service other than GPS and provided opportunities to study the proposed method in practice as it has been used by professional designers.

6.1.1 For the Designers of Love City

Love City used cellular positioning in GSM networks. The artists wanted to design a location-based experience that would work on technology that people already owned, which led them to the use of mobile phones. Our artists entered the development process with a rather optimistic, perhaps even naïve, view of how accurately and reliably cellular positioning world work. By the end, they had built up a thorough and detailed appreciation of its reality, becoming intimately familiar with variations in coverage and accuracy, both across different locations and different networks, including the ways in which cells overlap. In their own words, working with ubiquitous technologies is: “a bit like fighting with jelly”, i.e. hard to handle, unpredictable and constantly changing. They felt that our tools were a good starting point for experimenting and for learning about the nature of the underlying ubiquitous computing infrastructure. They could then exploit this new knowledge in redesigning their game, for example defining appropriate sizes for virtual locales and adjusting the game narrative.

6.1.2 For the Designers of Rider Spoke

Rider Spoke used a fuzzy fingerprint algorithm for Wi-Fi networks, which was conceived as part of the development process (page 208). The artists wanted to exploit the possibilities for location-based play as offered by commercial mobile game-consoles of the year 2007. These mobile devices generally featured a Wi-Fi adapter but would not have a GPS receiver, and thus it was decided to work under these constraints. The artists were participating in the design process of the location-mechanism through surveys, experiments and feedback, and thus gained empirical knowledge about the Wi-Fi infrastructure early on. But the resulting algorithm was complex, it relied on three parameters that had an unknown effect on the granularity of locations, and it was also inherently non geo-referenced, which altogether made it hard to understand. Provided with visualisation and simulation tools that made the spatial spread and other characteristics of the location mechanism more accessible, the artists were finally able to independently configure the algorithm so that it produced locations of sufficient granularity as they saw fit for their purpose.

6.1.3 On a General Level

Both experiences utilised fairly simple content triggering mechanisms, as they only used their underlying location mechanisms to determine whether a mobile device, and thus its player, was in one location or another. The need for understanding the infrastructure was therefore in both cases limited to the location mechanisms itself and so the supporting tools could focus on this task. Compared to Love City, the Rider Spoke location mechanism was more elaborate and thus more difficult to understand. This was reflected in more iterations of the locations mechanism itself, and also in the tools that were developed to support the designers in reasoning about it, i.e. the graph visualisation tools (page 211) and the what-if simulator (page 226).

This process of understanding a location mechanism seems to be a general theme and it is suggested that tasks and tools that contribute to this understanding should be used by experience designers early on in the development process in order to verify that the desired target infrastructure does indeed work in an acceptable way. The following section reflects more deeply on the tools and tasks that should be supported in a tightly integrated workflow that links work in the field with work in the studio.

6.2 Supporting Design Work

We found out that the idea of visualising the wireless computing infrastructure to the designers is rather complex in practice and raises problems of data access, data capture, visualisation and practical work. It also requires an iterative workflow that moves between working ‘in the field’ and ‘in the studio’. Specifically, we learned that as well as extending PC based authoring tools with visualisation layers, it is also necessary to provide more powerful tools for capturing, testing and annotating data while mobile in the field. This section first elaborates on these two aspects of the integrated workflow, which is most important in the elaboration and construction phases. The section closes with a discussion of the main tasks that mattered to our artists on a conceptual level.

6.2.1 In the Field

We have seen that work in the field combines three different activities, collecting data about the infrastructure, verifying the location mechanism during mobile tests, and possibly adding annotations like descriptions of locations, location mismatches and other experiences.

6.2.1.1 Mobile Authoring Tool

Feedback from the artists in the first study raised the requirement to integrate all of these activities into a single mobile authoring tool so as to improve productivity. Collecting as much of the right data as possible about the desired infrastructure can indeed be seen as a key task for such a tool, and this should be supported in an effortless manner.

For our second study, we had initially fallen back to using existing third-party software for data collection. While this allowed a quick start, it later imposed problems when writing the visualisation components, as the logged data did not contain all the required information.

A later version of the mobile Rider Spoke game client provided optional support for GPS data collection (together with the Wi-Fi data it was already collecting), yet this also tended not to work for practical reasons. First and foremost, GPS-support for external Bluetooth receivers was designed to be enabled at compile-time and required the manual definition of the receiver's MAC-address in a configuration file. Second, most of the mobile devices did not have the GPS-enabled version installed, as it was not used for the actual end-user experience, and thus a dedicated installation via a command-line interface was required if one wanted to collect GPS-data. As a result, the GPS-enabled mobile game client was too complicated to use and could not be handled effectively by others.

While this could be seen as an engineering problem, it actually represents a more profound problem: the required data collection survey was not regarded as essential and thus not sufficiently integrated into the overall workflow from the beginning. As such, it got pushed aside when the deadline pressure rose and no more man hours could be freed for this task.

The data collection workflow needed attention after the premiere. For the second outing of Rider Spoke in Athens, we provided the artists with a choice of two tools for data collection – the GPS-enabled game client and an easier to use third party application (page 234). Although the artists stated that they would have preferred to use the actual game-client for data collection, its complicated handling prohibited this endeavour and they decided to use the third party software (which did not log all required data) instead. Seeing that an effortless integration of GPS-data collection in the mobile game client was out of question, as the construction phase was already over and the developers were engaged in other projects, we finally provided the artists with an easy to use custom data collection program that logged all required GPS and Wi-Fi data, and ran on a separate mobile phone. This program was actually an extended version of the data collection tool that has been previously used in the Love City study – we just added Wi-Fi scanning, logging and a reimplementation of the Rider Spoke location mechanism so that the user could be notified about location changes.

It appears that such a mobile tool could be universally useful for the creators of location-based experiences, as well as to the people who study them. The tool should be easy install and run, provide means of logging as much data about the infrastructure as possible (including GPS, GSM cell ID and Wi-Fi), allow for mobile annotation (including audio and text) and be easily extensible with new technology by a programmer. Indeed, we envisage a further extension to these ideas, for example to support other forms of annotation such as capturing images and video material that could further support design discussions or provide 'assets' for design work. We also found that location-tags would provide a valuable mapping mechanism that works independently of any wireless radio technology, i.e. when using the mobile tool indoors.

6.2.1.2 Practical Use of the Tool

In the Love City development process, we saw how our artists employed multiple forms of transport over multiple site visits, initially driving around three cities so as to establish a 'broad brush' picture of cells and then drilling down into chosen areas on bicycles or on foot. A similar use has been observed in the Rider Spoke study, where the artists pre-planned their survey routes so that they would provide a fair mix of data about the wireless infrastructure from different neighbourhoods. They then went out with several persons on bikes in parallel in order to finish the survey as quickly as possible.

Consequently, we need to design the mobile tool to be usable in cars, on cycles and on foot, for example supporting hands-free operation when driving or cycling. This should include clear feedback about important states of the data collection software. Different background colours of the screen could be used to signal current location, as shown in the Love City study. Furthermore, audible feedback could be used to provide additional information or signal critical errors without distracting the designers from driving safely. Our artists named the following state changes as candidates for audio signals:

- Changing content regions (relates to their content mapping)
- Changing locations (relates to the underlying location mechanism)
- Discovering new locations (ditto)
- GPS problems (might invalidate survey results, needs immediate attention)

While it was hard work, we saw that it was quite possible for small groups of artists to map out their game areas for themselves. This is a feasible overhead for such projects and was not a major source of complaint or criticism. In fact, feedback suggested that detailed surveying of locations brought other benefits in terms of greater familiarity with the physical environment, its character, features and likely routes through it. Such local knowledge will be important in many 'site specific' location-based applications which have to be carefully designed to fit a given environment, e.g. a tourist guide.

6.2.1.3 Four Approaches to Scale Up

While dedicated data collection might be feasible on a small per-project basis, it could become problematic in larger projects. It would therefore be interesting to see how this process could be scaled up to larger or multiple projects.

One approach is to share data online, as we already see with initiatives such as Wigle.net. However, we should be aware that designers might then miss out on local knowledge as discussed above. Perhaps we could extend sharing services to enable the users of published mapping data to contact its providers so as to discuss how it was acquired and possibly draw on their local knowledge. To gain controlled access to the trusted data, it might also be interesting to exploit social networks, where participants manually express their relationships with others, or trust networks, where an algorithm automatically finds partners that are deemed trustworthy.

A second approach is for operators to provide infrastructure data to designers, potentially as a new commercial service, although some may be reluctant to enable people to compare details of their coverage with their rivals.

A third is for participants to generate mapping data as a side effect of another activity, an approach that has been explored in the design of 'seamful games' – which exploit and capture information about the seams in the ubiquitous computing infrastructure (page 49). Bearing in mind how several mobile tasks have been combined in the Love City study, it might also be possible to devise a mobile workflow that generates useful data for the infrastructure *and* physical world layers simultaneously and that maybe even links in with voluntary geographic mapping activities (Goodchild, 2007).

Building on the third idea is the fourth. We have observed in both studies how our artists accurately planned their survey routes on paper maps. If we wanted to scale up the survey process beyond the use by a single co-located group of people, this planning task needs to be better supported to schedule the routes of participants on a potentially planetary scale. To some extent this is already done, e.g. in the OpenStreetMap community (65). Participants of that community organise regular meetings all over the world where they plan to map out black spots on their map. This is usually as a special day activity, where participants meet up in person, socialise, and plan their routes in person. Spatial databases might be used to automatically propose fruitful survey routes to willing participants wherever they may go and whenever they may feel like contributing.

6.2.2 In the Studio

Supporting work in the studio remains important, as artists need to be able to gather together to discuss data, a process which can be supported by using large shared displays running sophisticated interactive visualisations. Indeed, providing such visualisations is probably the major challenge here.

6.2.2.1 Visualisation Tools

In the two studies presented in this thesis, we provided our artists with different kinds of visualisation tools that sought to facilitate the artists' work in similar ways, but with slightly different features in each case.

Table 6.1 provides an overview of the main features as they have been supported in the different tools. The tools in both studies quintessentially provided ways to visualise location-based data in a meaningful reference context. This is a basic requirement for making sense of location-based information, which would otherwise be meaningless[28]. A straightforward way to achieve a meaningful

[28] As an analogy, think of a temperature reading which needs to be mapped to location to be meaningful.

context is to geo-reference the sampled data, i.e. by synthesising it with GPS-data, and displaying the compound information on a map background; this has been supported in both studies. While for Love City this was the only way of seeing the collected data in context, the graph-based location mechanism of Rider Spoke also allowed locations to be viewed in the context of their surrounding locations. Consequently, the supporting tools in that study provided a means of analysing the data in both contexts, graph-based and geo-referenced.

Feature	Love City	Rider Spoke
Geo-referenced data	Yes	Yes (on demand, otherwise graph)
Map background	Yes (preconfigured)	Yes (seamless handling via Google Earth)
Different views	Yes	Yes
Zoomable UI	No	Yes
Interactive overlay	Yes	No
Spread of locations	Yes	Yes
Size of locations	Yes	No
Raw data	Yes	No (discontinued after first prototype)
Abstract data	Yes	Yes
Colour schemes	No	Yes

Table 6.1: Features of visualisation tools in the two studies

Access to relevant map-data was seamlessly handled for the artists in both studies. For Love City, a preselection of maps at different scales has been made and mapped to different hard-coded views that could be changed via function keys. Although this provided a functional interface, the artists' feedback hinted at several improvements, including the need for a freely zoomable user-interface and the need to access both, maps and aerial images, for information gathering.

These suggestions were addressed in the second study, where we provided our second group of artists with a zoomable user-interface and seamless handling of aerial images by using the freely available Google Earth program to display our geo-referenced graphs. This seems to have worked well for the artists in this case as verified by our observations and the absence of complaints regarding this matter. Finally, the Rider Spoke graph visualisation tool supported the use of colour schemes and icon sizes (only visible when exported to Google Earth) to highlight different aspects of the data. This feature originated from feedback in the Love City study, where the artists mandated the use of colour for their mobile tools. In Rider Spoke, the artists could quickly change between different schemes which, for example, used the size of location icons to denote the number of player visits to a location and their colour to differentiate between locations without content (grey) or with player answers (orange). This example scheme provided a quick overview of the most popular locations and whether they are already occupied by player content. This and similar schemes have been used by the artists.

6.2.2.2 Practical Use of the Tools

A main task for such visualisation tools is to support the designers in reasoning about locations. This includes the spatial spread over the target area and possibly the size of locations, although this might sometimes be an uncertain or blurry measure. A main concern for the artists in both studies was the granularity of locations. They wanted to ensure that there were a sufficient number of locations to be used by their players. These locations did not have to be overly precise or small grained – just good enough to support the notion of location in their designs.

Our two groups of artists fought the opposite ends of the spectrum of granularity. On the one hand the designers of Love City had to cope with quite large locations, as commanded by the underlying GSM networks, which forbid the use of a detailed narrative in the game. On the other hand the designers of Rider Spoke faced the problem that the underlying Wi-Fi fingerprint location-mechanism tended to produce more locations than originally anticipated, which ultimately caused slowdowns of the mobile game client and bad retrievability of location-based content.

In the Love City study, we have seen how visualising the coverage and accuracy of cellular positioning was exceptionally complex involving at least twelve distinct layers of data (four networks across three different cities). In order to make sense of this complex picture, our artists needed to switch different layers on and off and also flexibly combine them, sometimes working with a single network across a single city, and at others combining several networks or even all three cities. They also needed to be able to zoom in and out, working at multiple spatial scales. The first version of the infrastructure visualisation tool that was presented to the artists only provided a visualisation of the infrastructure as a series of dots representing sample points (see figure 4.16 on page 159 ff), essentially showing them the 'raw data'. The artists explained that while they found this view to be aesthetically beautiful, it could become visually overwhelming and too confusing to work from alone, especially when looking at coverage across large areas. They commented that the addition of the visualisation of approximate cell location and shapes in the second version (see figure 4.14 - 4.16 on page 164 ff) was greatly appreciated.

In the Rider Spoke study, we have seen how visualising the coverage and characteristics of a new, custom, and therefore unknown location mechanism was even more complex. Being Wi-Fi based, the algorithm relied on an infrastructure that is non-uniformly distributed over space, as Wi-Fi access points are generally put up individually by businesses, institutions or private individuals. This is in contrast to the GSM networks in the first study, which are planned and erected by operators that have a contractual obligation to cover a large percentage of a country's population. Trying to understand a new algorithm that exploits this unstable infrastructure proved to be a complex matter that needed several iterations of work. It was clear from the feedback of the first prototypes that raw-data visualisations would be too overwhelming, especially in regard to controlling the visualisations. To satisfy their own requirement of having enough locations for players, our artists needed to be able to quickly see the effect of different parameters on the resulting game-graph. This was supported through a simulator (see page 226), which replayed log-files of real data with different parameters and thus allowed for quick experiments. A graphical front-end to the simulator eased setting up the different experiments, running them through the simulator and analysing the resulting graphs in a number of ways. Such experiments would easily generate large amounts of data. We employed a sortable spreadsheet view to present analysis results and rankings to the user in a structured way. From any of those rows, the user could directly send the data of interest to a graph-visualisation package for further inspection (see page 213). Furthermore, it was also possible to geo-reference and visualise the graph in Google Earth to see it in relation to background maps. Altogether, this provided the artists with several views on the data. They commented that the simulator was the most crucial addition to the workflow as it proved to speed up the set-up process tremendously – from several days down to a few hours.

The use of Google Earth for map-based visualisations in Rider Spoke also introduced foreseeable functional limitations that were noticed by our artists and that affected their work. The biggest limitation was caused by the mechanism which is used to select information for presentation in this program. Google Earth organises its data hierarchically in layers and folders which can be individually switched on or off in a tree-menu. This interaction metaphor is well known, especially to artists who are familiar with graphic design programs like Photoshop which have a similar concept for organising layers. But with a larger number of layers, this menu-driven metaphor becomes increasingly awkward to handle. This was already noticed in one of the first prototype visualisations for Rider Spoke which produced one overlay image per discovered access point (see page 205). Confronted with several hundred layers of information in Google Earth, our artists commented that this kind of presentation was interesting, but the selection mechanism was too data-centric to be useful. It was regarded as much more desirable to use the mouse on the map-background to indicate a position of interest and then have the visualisation software automatically present the most appropriate layers of information. Such an interactive overlay was only given to the artists in the first study – who liked it – but not to the artists of the second study, who missed it and came to suggest this feature with their own words. An interactive overlay thus appears to be a desirable feature for such visualisation tools, as it eases the selection task.

Sometimes our artists needed to work with more or less raw data, but at others required a far more abstract overview of the situation, even when they knew that this tended to hide some detail. It appears that an abstraction, even when simplified, gives useful general information about broad coverage and general variation in location size, whereas raw data gives details when looking at specific locations close-up. Overall it was found that multiple levels of abstraction are required in the visualisations. This observation is in line with Shneiderman's visual information seeking mantra *"overview first, zoom and filter, then details on-demand"* (Shneiderman, 1996).

6.2.2.3 Do not Rely on Visualisations Alone

In both studies, appropriate information visualisations supported the revelation of the complexities of the underlying wireless infrastructures and provided visual feedback to the artists. But this is not to say that the visualisations alone provided the desired insight; rather they complemented the mobile survey activities and face to face discussions. This point can be further illustrated with an example that is the frequently found in the visualisation literature – but with a different connotation, as we will see.

Figure 6.1 shows the now famous map of the 1854 cholera outbreak in Soho, London. The map was made by John Snow and provides a graphic summary of the results of his study. Snow was a British physician, who researched into the mode of transmission of cholera. He did not support the then-common theory that cholera was transmitted due to the breathing of foul air and rather suspected that the illness is caused by drinking foul water (Snow, 1855). His map shows the number of deaths in each household as lines that are stacked up on the sides of the streets and also the location of the water pumps in the area. The map is focussed on the area around Broad Street (now Broadwick Street), which lies at the centre of the image and has an extent of about 195 m[29]. It can be seen that the death toll around this area is much higher then elsewhere and Snow famously related the transmission of the Soho outbreak of cholera to the water pump on Broad Street. When the water pump was closed the spread of the disease declined.

---

[29] measured from the centres of the adjoining streets after aligning the image in Google Earth

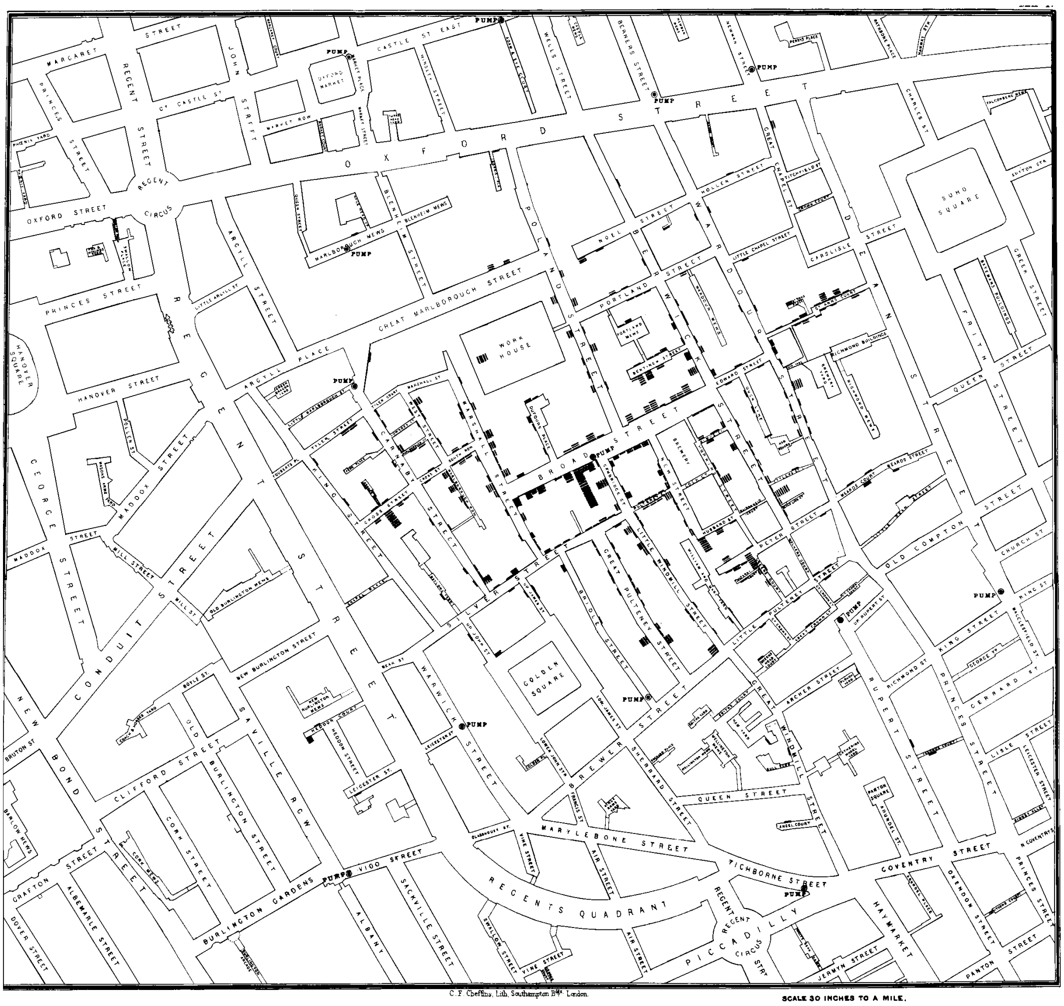

Figure 6.1: Snow's 1854 cholera and water pump map of London Soho (Source: (Mackenzie))

Snow's cholera map is often held as an example to justify different agendas around visual data analysis, as criticised by Koch (Koch, 2004). Koch's interesting article reveals several variations of Snow's original intent to graphically summarise his studies into the intent to advocate the use of visual data analysis by means of a famous example. He basically criticises that, over time and different citations, Snow's map has been mystified from the *result* of his studies into the *source* of his discovery that the cholera outbreak was caused by foul water from the pump at Broad Street. Koch suggests that *"Gilbert, Tufte, Monmonier, and the CDC (the US Centers for Disease Control and Prevention – ed.) sacrificed Snow's brilliantly complex and amazingly thorough thinking – which included mapping – in service of a myth whose ultimate message is that map-making is a science that, without further research, can solve complex problems with nothing more than the possibility of a simple spatial correlation."*

It is with this criticism in mind that Snow's cholera map is hereby presented as a prime example of good visual information design. This visualisation provides a clear summary of Snow's idea and key findings, but it is not the source of his discovery. This is the same as with the infrastructure visualisation tools that our artists used *after* discussing about possible location mechanisms and conducting their own surveys. The visualisation tools contributed to the artists' understanding, but the other tasks were just as necessary to come to a conclusion.

6.2.3 Making Sense of Location

On a high level, the main concern of our artists was about understanding the location mechanism to a degree that allowed them to judge whether there would be enough locations for content in their experience and how this content would be triggered. This section provides some reflections on these main tasks.

6.2.3.1 Understanding Location in the Design

We have found that our designers think differently about locations than engineers do. A typical engineer's view would be to optimise the location-mechanism so that it provides a more and more fine-grained grid of unique locations, which is based on the assumption that a higher precision means a better mechanism. To some extent this is, of course, also of interest to the experience designer, but what really seems to interest them is that the location-mechanism provides *enough* locations for them to attach their content to – no more, no less.

This relates back to the concept of *meaningful locations* (page 125) and to Heidegger's notion of *ready-to-hand* and *present-at-hand*, or the difference between a tool and using a tool (page 11). Heidegger used the example of a "hammer" and the activity of "hammering" to illustrate how the hammer itself facilitates the hammering, which is what people really want to do with the hammer. In this sense, the tools that were presented throughout this thesis represent the hammer that allows experience designers to work with locations in their location-based experiences. Consequently, and this was also observered in the studies, the tools were ultimately not the primary interest of our designers. They just wanted to use the tools to go beyond and realise their experience designs. Seeing them achieve their goals by using the tools must be seen as a positive feedback.

Subjective Locations

In both studies, the underlying location mechanisms have not been optimised for the sole purpose of being as accurate as possible. Rather, they have been used to generate locations in the union between the real and virtual world that were less about coordinate values, such as latitude and longitude, and more about the semantic or context of a location that an individual participant would roam in. These locations then provided the stage for the interaction with the end-user, and it was the artists' intent in their overall design to allow these locations to become meaningful places in the end-user experience.

In Love City, for example, it was important that a player would be in or nearby a real park when the location-mechanism located them in a virtual park. This mitigated the need to have an absolute knowledge about where the player actually was. This kind of fuzzy, yet structured definition allowed the experience to unfold differently for every participant at the places they visit, without having to survey the physical environment in too much detail.

In Rider Spoke, it was important that players would have enough places to hide their user-generated content. It was also important that these places could be rediscovered, so that the content hidden by one player could be found by and played back to another. The absolute knowledge of physical locations was, again, less important, as the questions that players had to answer were phrased in such a way that it mitigated the need to have a really precise location mechanism.

Language

Indeed, a particular strong tool for designers seems to be language, or the narrative of the game. Apart from setting the tone for the overall experience, language can help to frame the intrinsic precision requirement of the experience. During the design sessions for the Love City game, the designers discussed the possibility to use past tense when referring to locations, as this would lower the requirement for the location-mechanism considerably. For example, when locating a player in a virtual park, the game could provide feedback to that player that he *"has been seen at a park"* rather than that *"he is in a park"*. The former provides more room for interpretation and suspension of disbelief, as the player does not actually have to be at the park when the message reaches him; just having been there, or nearby, recently would be enough to uphold a believable mapping between the real and the virtual.

Similarly, the designers of Rider Spoke paid great attention to the phrasing of the questions that players had to answer. These questions were generally verbalised in a way so that it would be the players' choice where to hide their content. The location-mechanism was just used to give players feedback if their chosen location was already occupied, in which case the game told them to carry on and look for somewhere else to hide. References to locations in the Rider Spoke questions were along the lines of:

- *"find a place you like, stop your bike and tell me about yourself"*
- *"find somewhere your father would like, go there, and tell us about him"*
- *"find me a stinking a***hole of a spot and tell me all about it"*

The language of Rider Spoke thus provided its participants with a great structural flexibility to appropriate the space that they were in to become a place that they liked by means of the stories that they recorded. This flexibility allows for narrative immersion of the players and seems to be a good trick to facilitate the creation of places in location-based experiences.

Well Enough

So there seems to be a sweet spot, where designers regard a location-mechanism as able to support their intended user-experience. They can then take it from there and design the rest of the experience in a way that might even smooth out potential roughnesses in the location-mechanism. This observation is in line with the evaluation results of the GUIDE project (page 29), who observed that designers of location-based experiences *"must not be over zealous when deciding to constrain the information or functionality provided by the system based on the current context."* This sweet spot is related to the granularity of locations.

Visualising the spatial spread and extent of locations to the experience designers, as described on page 262, is therefore a key task to help them understand the use of location in their design.

To some extent these observations seem to contradict the design-mantra *"Never leave well enough alone",* as postulated by the influential industrial designer Raymond Loewy (Loewy, 1951). But on closer inspection, this mantra still holds true for the design of location-based experiences, as it is the end-user's experience that ultimately matters, and this does not necessarily require a perfectly precise location-mechanism.

6.2.3.2 Mapping Content to Locations

We have found that the most important task when building a location-based experience is the one of spatial authoring, i.e. mapping content to locations. In essence, this is what all the authoring tools presented in the literature review (page 86) had as a common denominator. Content mapping is the actual configuration task where digital media assets are associated with the locations that are produced by the location-mechanism. For a GPS-only system, this task is already well supported and authors who wish to build an experience in such a way would be well advised to look at tools like Mediascapes (page 89).

The two studies presented in this thesis chose a different path and implemented location-mechanisms that did not use GPS at run-time, and were thus more complex to understand by humans. We facilitated our designers' understanding through multidisciplinary discussions in a participatory design process (page 291), by getting them involved in network survey activities (page 256), and by providing them with appropriate visualisation tools for analysis and reflection in the studio (page 260). In both studies, these measures provided our artists with means to reason about the spatial spread of locations in their design, and also about their possible extent. Both systems generated unique identifiers (IDs) for each location, as proposed in the abstract location model (page 131), and so the task of content mapping was merely to support mapping these location IDs to the IDs of the desired pieces of content.

In the case of Love City, content mapping was supported through a desktop mapping tool (page 164). Here, the artists could visually define regions of interest on an interactive, dynamic grid that was geo-referenced and overlaid on a map of the area. As the visualisation was dynamic, the layout of the grid could change over time, when more survey data became available. However, this did not cause any inconsistencies, as each grid cell represented a uniquely identified location in the database, and it was only the appearance of the cells in the visualisation that possibly changed. This resilience is in stark contrast to the grid mechanism of CAERUS, which invalidated all applied content mappings when the underlying grid changed (page 97).

For the development of Rider Spoke, the visualisation (page 211) and simulation tools (page 226) were used to assess the apparent performance of the location-mechanism based on real survey data, and providing immediate feedback on the effect that different parameters had on the generation of locations. Rider Spoke's workflow skipped the need for a separate content mapping tool, as this task was performed using the mobile game clients. Once the artists knew that the location-mechanism worked to their contentment, they took the mobile devices outside to record some audio content in-situ.

In both studies, data was initially collected in the field and then visualised in the studio. Our artists assessed the likely performance of the location-mechanism based on their own survey results and then subsequently tested their assumptions in-situ, as they went back into the field with a mobile tool. So there is at least one full loop of being outside, coming back, and then going out again. In fact, it was observed that many such iterations were necessary. This is in-line with previous conclusions by Weal et al. (Weal et al., 2006), who highlighted the importance of in-situ authoring for location-based experiences as they found that ideas for location-based content are best triggered on location, but further revisions would need to be applied at the desktop. To facilitate this frequent change of workplaces, a seamless data-link between the mobile and the studio is recommended, possibly with the help of a server.

In summary, we found the following aspects to be beneficial for content mapping:

- A visualisation that fits the data
- Raw as well as abstracted visualisations
- Unique and reproducible identifiers for locations
- Seamless data-transfer between mobile and studio
- A spatial overview, with an overlaid geo-referenced grid of locations
- An interactive selection mechanism that operates on that grid
- An emulator to pre-test the mobile experience from the desk
- A mobile tool for surveying and actual testing

An interactive and abstracted spatial grid of locations that supports grouping of locations into regions seems to be of special interest to the task of content mapping. Although the designers of the Rider Spoke experience applied their pre-authored content in the field, and therefore did not rely on a desktop content mapping application, they came to suggest that such a tool would indeed be beneficial to their work as well. When assessing their players' content, they found that some of the recorded answers were very engaging, yet not location-specific at all. They thus wanted to use such universal and good pieces of content in forthcoming stagings of Rider Spoke. But this was not possible, as their authoring workflow did not support remapping of content to different locations (in another instance of the game) in an intuitive way. We envision that the interactive spatial grid could support this remapping task quite effectively.

The emulator is not intended to replace the actual mobile testing activity, but rather to support the developer in quickly testing alterations in the design without having to interrupt the current workflow at the desk. Mediascapes provides such an emulator. From their experience, an emulator should also try to emulate the characteristics of the wireless infrastructure, as, for example, the task of selecting a location was much easier in the emulator as it was in the field, where GPS jitter complicated the matter. Most software development kits (SDK) for mobile devices nowadays ship with their own device emulators. These emulators usually lack some of the device capabilities, e.g. the camera or the GPS module would not be accessible. It would therefore be interesting to design a workflow that permits using the existing device emulators, and somehow provide the software running on the emulator with the relevant data, such as GPS, from external sources. This could be achieved by providing a separate window on the development machine, where the designer could explore the experience-space by self-declaring their location, e.g. by clicking on a map. An untested, but potentially quite feasible idea to replicate some of the GPS jitter could be to apply some jitter to the mouse pointer that the developer uses to declare the position on the map. An alternative would be to feed in pre-recorded, time-stamped GPS data, and maybe provide means to skip through that data based on a time-line. This idea is actually very similar to the idea of replay-tools, which are covered at the end of the following section.

6.2.3.3 Understanding Places in the Post-Event Analysis

Having previously said that the precision requirement for a location-mechanism in an actual experience is not necessarily very high, it has to be noted that this precision requirement usually increases for a post-event analysis. This is because artists and ethnographers are then interested in finding out about the places people visited. They would like to know how the content was spread out over the city and how it was triggered and experienced by the participants. This is interesting, as it provides insight into the mechanics and dynamics of the experience, and might provide clues as to how participants roamed the space and turned it into places, which can feed back into future designs.

There are several ways to collect the right data to support this iterative process. Buxton and Sniderman (Buxton and Sniderman, 1980) provided an early account of this and summarised that participant observation by researchers, video observation and system logs provide means for non-verbal observations. They also outlined verbal techniques, such as report of impression by the subject, and interrogation by researchers. All these techniques can be applied to the domain of location-based experiences. But as these applications have a particular focus on the use of space, it can be especially beneficial to analyse the system logs, which should contain spatial information, preferably also GPS, if possible. For systems that do not rely on GPS, this can still be achieved by equipping a sub-set of participants, or alternatively the observers, with GPS devices that log their position-trails in a time-stamped format. This information can then later be synthesised and analysed in dedicated programs, such as the Replaytool (Tennent and Chalmers, 2005) or the Digital Replay System (Brundell et al., 2008).

6.3 The Engineering Side

Due to its user-centred approach, this thesis has very much focussed on the user's side, i.e. the view of the experience designers. Our observations and their feedback have provided us with valuable insights into the development process of location-based experiences under realistic and commercial conditions. However, this artistic side does only exist in partnership with the engineering side, or as Burgess put it: *"there is no art without science"* (Burgess, 1999).

This section seeks to provide an overview of some of the practical issues that were faced during the development of the studied experiences and revisits the challenges that were outlined in the framework chapter.

6.3.1 The Challenges Revisited

The framework chapter raised a number of challenges that were foreseeable when the work on this thesis started (page 135). The following text provides a practical account of how we dealt with these challenges.

6.3.1.1 Map Data Access

Unsurprisingly, maps proved to be very useful and meaningful backgrounds that helped putting the collected infrastructure data in the right context of the physical world. There are two main data-sources for maps: analogue (printed maps/photos) and digital (online-services).

Analogue Maps

Printed analogue maps and aerial photos are useful for planning purposes, and we have used them extensively for this purpose during the production of both experiences. It might sometimes be interesting to transfer them into the digital domain, in which case their coordinates should be known or should be measureable. In this case, the image needs to be digitised, aligned, possibly orthorectified (if it was an aerial photo), and can then be geo-referenced and used as a background in an authoring tool. This is often a useful process as a lot of information does not exist in digital form but is available on paper. However, we have not used analogue maps in this respect and instead reverted to the use of digital maps.

Digital Maps

Digital maps (or the data required to compose them) can be acquired from online services. The advantage of digital maps is that they are already precisely measured which means that the display in an authoring tool will be as precise as possible. The process of implementing digital maps usually includes two decisions that need to be made: which data-source to access and how to access it. Examples of protocols and specifications to consider for map data interoperability include:

- ESRI Shapefile (66), a popular vector-data exchange format for geospatial applications.
- Web Map Service (WMS) and Web Feature Services (WFS) are open international standards defined by the Open Geospatial Consortium (67). These protocols are Internet-operated and return data and/or images of a specific area on request. WMS and WFS are specifications and part of an abstract architecture that defines how map data can be obtained in a loosely coupled web service architecture.
- Google KML (68), although as a mark-up language mostly being used for visualisation, KML can also be used to specify the alignment of image files to physical world coordinates by using the Google Earth program, defining an overlay image and taking the saved KML file as meta-data that contains the required coordinates. This can be used as a way to visually geo-code digitised map data.
- Custom formats for map tile servers such as Google Maps, Yahoo Maps, Microsoft Live, or OpenStreetMap.

Most of the digital data-sources on the market are only available after the payment of a fee and usually restrictive in what can be done with the data and how it can be accessed. But other data-sources, such as the currently popular Google Maps and some regional WMS map-servers, are openly available via HTTP-calls which makes it technically possible to access the data according to their protocol. This does not mean that the presented data is public domain and can be freely used for any purpose, but that is a legal question which is not going to be discussed here.

Practical Use

Examples of analogue and digital map data-sources that should be considered for use in authoring tools for location-based experiences include:

- Official digital national map data such as from the Ordnance Survey in the UK which is available via the Edina Digimap service (69). These data-sources offer a variety of maps in a variety of different raster and vector formats. Furthermore they usually offer additional data such as transport networks, boundaries, buildings, etc., which could be added to the visualisation layer, if necessary.
- Official analogue maps from the respective national land registry, including historic maps.
- Freely available[30] satellite images, orthophotos and elevation models such as the NASA Blue Marble (70) or data from the United States Geological Survey (USGS) (71) which is also available via TerraServer USA (72).
- Google Earth/Maps (55) (49) or Microsoft Bing Maps (73) (74) tile data. These services combine data from different sources and make them available via unified interfaces that are straight forward to implement. Although the legal status of using this data in own applications is not clear there are many benefits to do so. Most importantly this data is available for the whole planet and is delivered via high-performance servers that have little downtime.
- Community driven mapping projects, i.e. OpenStreetMap (65).

The two main issues when dealing with maps are access and scalability. These issues usually occur once an authoring tool gets deployed to its users which might not have access to the same data-sources as the developers or live in a different

[30] US government data produced with public funding is freely available by law

part of the world (which is not covered by whatever access mechanism has been chosen). To mitigate these problems, the physical world layer should support a variety of digital data-sources as well as digitised and geo-referenced paper maps, and let the users decide which ones they want to use.

From a programmer's point of view, access to different data sources can be facilitated by incorporating available third party software packages such as MapServer (75) or FWTools (76). In addition, seamless access for the user would also require a flexible and service independent tiling engine. The engine would load map tiles from a variety of different sources and draw them on a geo-referenced, zoom- and scrollable background canvas. Open Layers (77) implements such an engine in JavaScript and can be used in browsers. While this might already be very useful for many projects, we previously identified the need for interactive visualisations which functionally go beyond what is currently possible in web browsers. Building a tiling engine that could be used in standalone applications is therefore advocated.

In addition to access and scalability, there is also a temporal aspect to the use of digital map data. Work on this thesis began before online map data became ubiquitously available with the advent of Google Maps. Our first iteration of tools thus relied on digital data that was acquired from other sources, namely Edina Digimap, which provides our University with all sorts of closed-source data as part of a contractual agreement. Although this service still exists today, the source of data that we drew upon ("Land-Line.Plus") was withdrawn from this service in late 2008. This demonstrates how access to digital data is usually provided "from the top". The same applies to the currently available online map data, as it is, for example, provided by Google Maps. Although such data is very useful for the purpose of building location-based experiences, it is still copyrighted and thus not free in the sense of free speech.

A more sustainable approach appears to be taken by community driven projects like OpenStreetMap (Haklay and Weber, 2008), which generates its own map data "from the bottom up". Although this data-set is not as complete as its commercial counterparts, it has grown considerably over the last 5 years, and now presents a serious alternative to commercial data in some areas of the world. It will therefore be interesting to exploit such community sourced data for future work.

6.3.1.2 Infrastructure Data Collection

This challenge was already discussed in the context of supporting design work in the field (page 256). It was proposed that this task could be parallelised with other activities, such as testing in the field, and might also benefit from work carried out by other user groups, e.g. the Wigle.net community (14), which is collecting geo-reference samples of Wi-Fi data on a planetary scale. It was concluded that a general mobile tool could be devised, which would support data logging, testing, and annotation, and which might prove useful for designers seeking to understand the wireless infrastructure that they are intending to use.

When looking at the data produced by our artists during their mobile network survey activities (see figure 4.7 on page 152 ff), it appears that they have not only sampled the wireless networks quite densely, but also the street network. Such survey data might therefore not only be of interest to projects that collect infrastructure data, such as Wigle, but also to projects that collect map data, such as OpenStreetMap. It does seem to be a promising avenue to investigate how a single mobile workflow could be of benefit to different projects at the same time.

6.3.1.3 Presentation

This challenge was already discussed in the context of supporting design work in the studio (page 260). Features for interactive visualisations were discussed, which included map-based and geo-referenced visualisations, a zoomable user interface and an interactive grid overlay, which could be used for visual grouping of locations. We found that our artists mainly used these visualisations to reason about the spread and size of locations, and for assessing the spread of content over the area. It appeared to be beneficial to provide abstract visualisations, as raw data views were perceived as overwhelming, which was inline with a well known mantra from the domain of information visualisation. Furthermore, it was discussed that one should not rely on visualisation alone (page 265), as they are not the sole key to understanding complex problems. Self-conducted surveys and discussions in a multidisciplinary context were found to be of equal importance.

Support Different Views

In addition to these observations, it appeared that no single visualisation or selection technique would be universally sufficient. Just for the two studies presented in this thesis, we already had at least three different types of visualisations: map-based, graph-based and sortable spreadsheet – all of them with interactive selection mechanisms. We can anticipate other techniques such as time-line visualisations and selections, or cloud visualisations and tag-based selections to be relevant for certain applications. As such, it is advocated to provide many different views on the data and to facilitate the creation of these views. For the spatial views, this might be facilitated by interoperating with geographical information systems (GIS), which are all about visualising and analysing data in a spatial context. Our artists from the first study mentioned that heat map visualisations could be interesting to them; such visualisations are well known from the GIS domain.

Verbal Requirements

Indeed, articulated verbal requirements by the intended target users should form the basis for all visualisation and interaction efforts. It is only by paying close attention to the real user needs that one might possibly come up with a user interface that is not only a fit for the data, but also helps the users in mastering the tasks that they want to accomplish. In the Love City study, the artists wanted to learn about the granularity of the given GSM network infrastructure and how they could use it for developing their idea and deploying a public experience. Consequently, our tools helped them find an answer to this main question.

In Rider Spoke, the artists participated in the development of a custom location mechanism, whose granularity could be tuned by adjusting three parameters. Since the granularity of the location mechanism was therefore in our hands, the artists wanted to find the settings that would satisfy their requirements, which they articulated as:

- There should be enough locations for players to leave content.
- Locations must be easy to find, so that players do not have to spend too much time searching for a place that they could inhabit.

Again, our tools mainly supported the artists in finding answers to these main concerns, and allowed them to make conscious decisions about which settings they wanted to use. In both studies, these requirements were elicited through discussions. Seeing how both groups of artists were chiefly concerned about the granularity of locations in their designs, discussion of the required target granularity early on in future project meetings is recommended.

Adaptive Visualisations

So far, we have only visualised rather static location mechanisms. This is not meant to mitigate the intrinsic uncertainties of the infrastructures that we relied on and the problems and variations that this caused. But Love City, as well as Rider Spoke, both employed location mechanisms that were static at runtime:

- The Love City location mechanism relied on a given network infrastructure granularity which could only be observed, but not adjusted.
- The Rider Spoke location mechanism allowed the configuration of parameters *prior* to running the experience, but demanded constant settings at runtime to maintain its content-to-location mappings.

Appendix 5 discusses how an adaptive algorithm could be an immediate next step for the Rider Spoke fingerprint location mechanism (page 358). Although representing the ever changing nature of such an algorithm is beyond the capabilities of our current visualisation approaches, it would certainly be interesting to consider adaptive visualisations for future work.

Metrics

Suitable metrics are required in order to align the data visualised on the infrastructure layer to the meaningful background on the physical world layer. For geospatial data this would usually be absolute coordinates expressed in geodetic coordinate systems such as:

- the World Geodetic System (WGS84), in world-wide use
- the Universal Transverse Mercator (UTM), in world-wide use
- the Gauss-Krüger coordinate system (GK), as used in e.g. Germany
- the British national grid reference system (OSGB36), as used in the UK

Similarly, a visualisation of temporal data would require a precise measurement of time which would preferably be constant across all participating devices. Although the requirement for a precise timing in a location-based experience could be automatically satisfied with relative ease by obtaining atomic measurements either from a potentially present GPS receiver or via the Network Time Protocol (NTP), this metric often gets severely neglected by relying on the user correctly setting the time on the devices[31].

Contrasting the absolute measurements of space and time, topological and graph-based visualisations reveal relationships between measured entities in relative terms, such as:

- "person 1 is in location A", "person 2 is in location C"
- "A is connected to B" and "B is connected to C", therefore "A and C are connected via B"
- There is a possible "path between person 1 and person 2"

Topological relations can be very useful for location-based experiences as they can describe locations as well as paths through the game space. These visualisations do not rely on a geospatial layout with a background map, but provide additional insights if present.

In the context of this thesis, infrastructure visualisation was deemed most useful for those technologies that are most pervasive amongst today's consumer hardware and thus most likely to be used for location-based experiences: GPS

---

[31] Human-entered time is not a reliable data-source as it is error prone and bears an unknown deviation that could range from a few milliseconds up to several years, depending on how much the user cared about entering the correct time. And even if the local time has been entered accurately, the clock might still be out of sync compared to other devices, as different time-zones and daylight savings settings are complicating the matter.

positioning, wide area cellular telephone phone networks (GSM/GPRS and 3G), wireless local area networks (WLAN a.k.a. Wi-Fi), and potentially also short range communication technologies such as Bluetooth, NFC or RFID. These infrastructures usually need to be surveyed before they can be visualised or used for content definition. The metrics listed in appendix 3 (page 335) could be used for this purpose. In addition to the visual aspects of presentation, we also found that the challenge of presentation is not necessarily limited to visualisations in that sense alone. Outdoor survey activities provided valuable insights into the structures of the city, and Geiger counter like audio interfaces into the invisible wireless networks were seen as useful tools in this context as well.

6.3.1.4 Workflow

The importance of workflows was raised as a challenge in the framework chapter. Indeed, this thesis discusses workflows in the context of organising work *over time* in a multidisciplinary team (page 291), and *across different devices* in the field and in the studio (page 256). While both aspects are important to the outcome of the overall work, the engineer is especially interested in the latter.

Using Equip2

Both of the presented experiences relied on a three tier architecture that linked work in the field with work in the studio via a central server. Communication between the distributed components was facilitated through the Equip2 framework by Greenhalgh et al. (Greenhalgh et al., 2007a), which is briefly introduced in appendix 2 (page 332). Equip2 has been designed with four broad development strategies to support a diverse set of devices with varying capabilities:

- Prioritise strong server support, including web integration
- Migrate functionality from the server to more capable handsets
- Support very flexible communication
- Use a loosely coupled (data-driven and component-based) software approach

Although the generality of Equip2 and its web application framework first imposed an additional burden on the developers to master the API, completing this learning process later paid off, as the features of the system provided effective support to the development.

Equip2 imposes a model-driven development process. We started by defining the application specific core objects for the database model, which was configured in XML. The build step translated these generic definitions into Java classes that could be used to create, manipulate and destroy data in an object-oriented fashion. Communication with a relational MySQL database was handled transparently by the system through the Hibernate object-relational mapper (ORM); developers therefore did not have to write SQL statements by hand, which was seen as a benefit. Further time-savers were the auto-generated administrator web-pages for the server, which provided an easy way to access the server-side database through a browser, and which were always up-to-date with the data-model. So far, these features are inherent to many web application frameworks (page 329). But Equip2 also provides support beyond this core functionality, and reaches out onto mobile devices with its flexible communication strategy.

In Love City

Love City's smart phone client was implemented in Python S60 and made calls to the game-engine on the central server via a compressed and encrypted remote procedure call mechanism (RPC). The final refactored SMS based version of the game could reuse the existing game engine and extend it with new text-based input and output channels that communicated over SMS instead of RPC. The developers also made heavy use of unit tests that covered most parts of the game engine.

In Rider Spoke

Rider Spoke's mobile game client was developed in a mix of three (!) different programming languages: Java, C++, and Flash Actionscript. General Equip2 related communication and housekeeping routines were written in Java. More device dependent code (Wi-Fi, audio, GPS, database) was written in C++, and all user interface related code was written in Flash. This mix has been made to work together by two means. First, the Java code was compiled and then transcoded into C++ code by using a transcoding tool that is part of Equip2. The resulting code could then be compiled again and linked with the native C++ code. Second, the Flash code communicated with the compound C++/Java code over an XML-encoded socket connection, which needed custom protocol-code on the Flash side.

Reflection

While the Equip2 framework provided useful support for building our location-based experiences as it is, we found that it would benefit from another design-iteration. It its current form, Equip2 provides communication support in a flexible way, and across programming languages boundaries. But it seems that while the automatic transcoding (Java to C++) mechanism provides an effective way to cross programming language boundaries (as it achieves the desired effect), it does not support this in an efficient manner (as it is quite complicated to use). By this we mean that the current mechanism achieves this flexibility at the expense of the developers, who have to deal with a less intuitive programming workflow.

We observed that none of the developers of Rider Spoke[32] completely mastered all aspects of the system: the Flash programmer did not know anything about the underlying C++/Java code, and the core programmers did not know much about Flash. Arguably, the Flash programmer should not know anything about the underlying code and just use it as a supporting abstraction layer over a well defined API. But seeing that 4 programmers were necessary to tune the underlying code during the development phase, and that 2 of them ended up writing Actionscript wrapper code for the generic, but unintelligible XML socket interface, it can be surmised that the balance of the system was not, yet, right. It was also discovered that the underlying source-code contained “paths of least resistance”, such as bubble sort and string compares, and therefore showed signs of system imposed bias.

[32] There were 6 programmers in total: 1 for Flash, 1 for the visualisations, and 4 for the system

In their paper about the iterative design process of HCI systems (Buxton and Sniderman, 1980), Buxton and Sniderman argued that such a situation should serve as an indicator for a need to redesign the utilised prototyping tools. They also advocated the use of suitable high-level languages that allow the effective encoding of the concepts that are to be tested. Altogether, it might be beneficial to try to reduce the cognitive load of the developer that deals with the communication parts of a distributed application. One way of doing this would be to take a stable API and wrap it up into libraries that could be used with other high level programming languages.

6.3.2 Ideas for System-Level Improvements

This section provides thoughts on how to improve the overall handling of the distributed system and thereby increase the developers' and artists' performance when prototyping locations-based experiences.

6.3.2.1 Easy Data Synchronisation

The proposed distributed workflow, with work in the field and work in the studio, demands for a tighter integration between the system's components. The artists in our first study could already receive updates from the server with a single menu-selection. But this communication mechanism used custom code that was not universally valid and also did not provide any way to automatically synchronise the collected data back to the server by ways other than the expensive GPRS data-connection. The data synchronisation mechanism in the second study was more basic and required dedicated file-copying from each participating mobile device, and a further manual file-upload to the server which merged all data. From our experience, an ideal data synchronisation service would:

- provide very flexible communication
- facilitate cross-language message routing
- provide some sort of quality of service routing (speed vs. cost)
- blend into the mental background, i.e. no manual file-copying!
- use existing compression standards

To some extend, this is already supported by several existing libraries (page 332), including Equip2 (page 282), and probably many others. But it was found noteworthy that although both studies used supporting libraries and software packages, the process of synchronisation between devices and server was still an issue that was very much present in the mental foreground and demanded dedicated attention. It would be interesting to investigate how this task could be facilitated for future work, so that both developers and users would not have to constantly think about it and instead concentrate on more important tasks.

6.3.2.2 Coordinated Visualisations

Analysing the data in several different, but synchronised views can be regarded as a task that would benefit from easy data synchronisation. We have seen how our artists needed several different views for data analysis. With the Rider Spoke what-if simulator, for example, the artists used a spreadsheet view, a graph view, and a map view, which were all chained together via the file-system. This means that the data analysis would start in the spreadsheet view from which the graph visualisation could be started. The graph visualisation could be used to arrange the layout of the graph, e.g. geo-reference it and apply a certain colour and size scheme, and then pass the graph on to Google Earth for map-based visualisation. While this worked and was quite effective, it can be much improved. It would be much more desirable to have multiple coordinated visualisations that change their views according to the user's selection in one or the other view. A glimpse of this functionality was already present in the current system, as the Guess tool, which was used to build the Rider Spoke graph visualisation provided such a coordination between its textual view and the graph visualisation. While working on graphs with Guess was straightforward, this tool is certainly not the fit for every purpose. Other kind of visualisations will be necessary, and might need to be constructed using other tools, such as Matplotlib (63), or the visualisation toolkit VTK (78).

Making a range of programs to coordinate their views and exchange user focus data is a manageable engineering problem; the more profound problem is how such systems would then be configured, as identified by North and Shneiderman (North and Shneiderman, 2000). Their snap-together visualisations implement a common broker API and can then be freely combined and configured by their users. The Equator Component Toolkit by Greenhalgh et al. (Greenhalgh et al., 2004) provides a similar approach, but is more concerned about configuring complete distributed ubiquitous systems. With this system, users can define the data-flow through different components of the distributed system by configuring a graph. In this graph, the nodes represent the components and the edges (i.e. the connections between the nodes) represent the flow of data between components. Such systems are called node-based systems, and are also well known in other domains.

When trying to learn from the adjacent fields of film- and web-production (pages 287, 299), one can find similar workflows especially in the digital post-production, which deals with combining different digital media into a final, broadcast quality output. The highly specialised (and expensive) compositing programs in the digital post-production domain usually either implement a layer-based or a node-based user interface. Programs worth learning from in regards to node-based workflows include Autodesk Flint/Flame/Inferno, Quantel products, and Apple Final Cut Studio. An example for a layer-based workflow is Adobe After Effects. In the web-domain, Yahoo Pipes is a free node-based editor that allows visual configuration of so called mash-ups, i.e. web-applications that combine data from different sources.

It would be interesting to make use of freely configurable, coordinated and interactive visualisations for future authoring tools. An abstract location model with unique reproducible ID-strings, as proposed on page 131, could serve as a good basis to coordinate the different views.

6.3.2.3 Scripting Interface

Both studies entailed extensive programming work on all sides of the project, from the game engine and server side processes, over the supporting tools to the visualisations. Programming is unavoidable; thus, in order to facilitate the overall development process, it is required to support the programming task as well. Appendix 2 (page 326) provides a pragmatic overview of the software development process, and presents advantages and disadvantages of different programming languages in this context. Previous work in the field of location-based experiences offered scripting support, e.g. Mediascapes (page 89) or Gamecreator (page 100). Both of these examples implemented a domain specific, custom language of a limited scope and also used a visual paradigm for constructing code snippets. While this certainly presents benefits in terms of easy accessibility for non programmers, this approach is questionable at the current state of technology for several reasons.

First and foremost, such visual abstraction should only be built on a stable architecture and not on a moving target, as it is currently the case with architectures for location-based experiences. Reid et al. (Reid et al., 2005a) compared the production process of mediascapes to the film industry; an analogy on which the author fully agrees. Software packages for 3D animation as they are used in the film industry, such as Softimage, Maya or 3ds Max[33], underwent a considerable evolution in this respect over the last ten years. Extensions to such software packages originally had to be written in a native programming language, which was usually C++. This was followed by a generation of tool versions that employed custom scripting languages to facilitate writing extension and building the artists' user interface, as exemplified by Maya's Embedded Language (MEL). Today, custom scripting languages in these tools are replaced by generic programming languages, e.g. Maya now supports scripting in Python and Softimage even supports a range of generic scripting languages that all provide access to the core functionality of the tool. The point to make with regard to developing location-based experiences is that we should learn from the extensive experience that went into the architectures of the film industry's leading software packages, which first stabilised their core functionalities and then made it easier to use them by providing access to these functionalities via scripting languages.

Furthermore, it can also be learnt that custom scripting languages for 3D animation packages have been replaced by generic scripting languages. It appears to be worth considering skipping the use of custom scripting languages for the

[33] remarkably all three are owned by www.autodesk.com at the time of writing

development of location-based experience altogether, and jump straight away to the use of generic scripting languages.

Prechelt (Prechelt, 2000) provides an empirical comparison of seven programming languages, spanning from compiled to interpreted scripting languages. He found that programs written in scripting languages are usually shorter and take significantly less time to develop than programs written in a compiled language. He also observed that albeit their memory requirements are higher than those of C/C++ programs, they are still much lower than those of Java programs, and concludes that scripting languages *"offer reasonable alternatives to C and C++, even for tasks that must handle fair amounts of computation and data"*.

Use scripting languages for developing location-based experiences is therefore recommended. Laurila et al. (Laurila et al., 2006) presented a port of the Python programming language for Nokia S60 mobile phones and proposed its use for rapid prototyping of pervasive applications. Their port provides full access to the phones' core functionality in an abstracted way. This is especially useful, as the various versions of the S60 operating system are known to be difficult to target with C++ code, a problem which has previously prevented even big programs such as the Context Phone (Raento et al., 2005) (79) to continue working on newer generations of phones. The Python S60 interpreter abstracts the program logic from the underlying operating system and thus provides a unified access layer to the various versions of the S60 operating systems. Compared to the Java language, which set out with the similar goal of "write once, run anywhere", Python has the additional benefit of expandability, i.e. missing functionality can be implemented as a module in native code and called from a script.

In addition to providing a unified interface to several versions of the same operating system, a carefully designed scripting interface could also provide a way to tackle the variety of different operating systems that are currently on the market. Easy data synchronisation across device boundaries is a main concern to the overall workflow, and the scripting interface should therefore provide methods for this task. The scripting interface should also provide methods to deal with locations in an abstract way (page 131), and allow easy access to low-level sensors such as GPS or Wi-Fi in a unified way.

6.3.2.4 Flash Interface

From our experience, artists can handle programming and user interface design tasks. This is not to say that all artists are programmers and should program, but that they can pick it up if it is a desirable skill for them, or that they would at least know someone to hire for the job, so that they can articulate their ideas. This was observed in our two studies, where both groups of artists hired an external Flash programmer to work in close cooperation with them and also to work with us on a technical level. It was also found in the literature review, where location-based authoring tools such as Mediascapes or CAERUS made heavy use of Flash to deliver multimedia content.

It appears that Flash is a tool that designers know particularly well and that they like to use. In the Love City study, the artists and their programmer built a complete graphical mock up of the mobile client even before work on the game engine started. In the Rider Spoke study, the complete mobile user interface was designed by the artists and their programmer; a process, which was supported by an underlying communication system that was written in compiled languages.

Seeing how well artists can handle scripting in software packages like Flash, Director or Maya, and also seeing that Adobe is already providing a specialised version of Flash for mobile phones with an estimated user-base of 800 million mobile devices in early 2009 (80), it is anticipated that this platform will gain further importance in the future. The provision of a Flash interface for developing location-based experiences is therefore recommended in addition, or as part of the above mentioned scripting interface.

6.4 Best Practices

The previous two sections of this discussion covered the artistic and the engineering side separately. While this was necessary to reach a certain focus, the two sides need, of course, to be brought together in order to deliver a successful project. This section provides some reflections on this topic, including some surrounding factors.

6.4.1 Opportunity

An opportunity is a favourable circumstance that facilitates progress. Location-based experiences, like any other project, benefit from good opportunities. They provide the right framing for the development and incentives to complete on time. We have seen how our two artist groups utilised the opportunity of presenting their works at art events. These events provided time-frames for public presentation and thus also implicit deadlines for delivering the works. This pattern can also be seen in the reviewed examples and seminal works (page 25), which were all associated to similar opportunities for better exposure. Opportunities come at different scales and types, and need not only be art events. Small scale opportunities such as course works or hobby projects, and large scale opportunities such as multi-national research projects, can both provide valuable incentives to facilitate progress. Of course, funding is always a major incentive to conduct a certain project. As location-based experiences have not, yet, hit the mass market, funding will continue to be a good, if not the best opportunity for development.

6.4.2 Development Process

Many software development processes have been devised over time and this dissertation does not want to add another one. This section simply reflects on the overall organisation of our work and puts it in relation to existing approaches. The only noteworthy difference between our process, as implemented in the two studies, and a traditional software development process from the text-book is that we, as the researchers, were not solely interested in building a piece of software on time, but also in empirically measuring the users' reaction to it and how it hopefully empowered them to accomplish tasks that were previously not achievable. We did not build final software products, but rather learned from prototypes to envision the future. Our process was thus more reminiscent of an iterative design process for human-computer interaction, as outlined by Buxton and Sniderman (page 4), and in the tradition of cooperative prototyping (page 9).

We facilitated this intention by devising a general user-centred, participatory design process (Shneiderman, 2002: 53) where people with different backgrounds would frequently meet and discuss each others results on a symmetrical level, adopt the plan based on the latest findings and define the actions for the next iteration of work. Such a multidisciplinary approach has been described as *"a key factor of high impact"* user-centred design (Vredenburg et al., 2002) and is the established practice of work in the Mixed Reality Lab. If possible, different strands of work for the artists and the engineers have been designed to be carried out in parallel, but this was not always possible, e.g. when an experiment needed a certain time to setup before it could be conducted. Although this did not prevent the artists from doing their work – there was always something else that needed to be done – it effectively meant that adjustments to the system could sometimes take precedence and cause locks before the next iteration of work could commence. As such our development process allows for "Action Research" (Lewin, 1946) where adjustments to the system are made *between* iterations and then the users' performance is monitored *during* iterations.

Upon reflection it appeared that this methodology also integrates well with the established "Unified Software Development Process" or in short "Unified Process" (UP) (Jacobson et al., 1999). The Unified Process is a process framework which decomposes a software development process into a set of iterations that are of limited duration and that fall into one of the categories: inception, elaboration, construction or transition. Several iterations could fall into each category, but the project sequentially moves forward through these categories from inception to transition. Most time is spent in the elaboration and construction phases whereas the inception phase is required to be relatively short. In the case of our two studies, the following mapping could be applied between UP categories and project iterations:

- Inception: Develop game idea, potentially pitch project with funders
- Elaboration: Exploratory phase (gain experience, produce requirements)
- Construction: Build the prototypes, build the final system
- Transition: Deploy to users / stage the experience

By organising project activities around themed workshops, prototypes and other milestones, we provided the basis for doing Action Research and Unified Process development at the same time. As previously stated, this was important to us as we aimed to a) build tools that could effectively support artists when working with invisible wireless infrastructures and b) study the process.

	Inception	Elaboration	Construction	Transition
Staff	Brainstorming for ideas and target platform	Finalising ideas, target platform, architecture and location mechanism	Implementation	Runtime support Game-mastering Orchestration
Mobile	Inspiration	Survey Testing	(User-) Testing Survey Configuration	Runtime Front-End
Desktop	Ditto	Visualisation Analysis (Configuration) (Content mapping) Programming	Visualisation Analysis Configuration Content mapping Programming	Potentially visualisation
Server	Ditto	Data-Hub (Data-Model)	Data-Hub Data-Model	Data-Hub Data-Model Back-End
Third Party Sources	Ditto	Software Maps Libraries Data import	Software Maps Libraries Data import	Software Maps Libraries Data import

Table 6.2: Project resources and focus of work over the full development process

Table 6.2 provides an overview of project resources and work focus over the full development process. It uses the aforementioned categories from the Unified Process to differentiate between different stages of the project. Development starts with brainstorming for a broad plan in the **inception phase**. This phase only needs to outline a general idea and possible target platform(s), and should be kept short.

The **elaboration phase** was especially crucial for us. Equipped with an overall idea of the project from the inception phase, we set up the artists to do their own surveys, which in turn helped them to gain an understanding of the proposed location mechanism and prompted them to more clearly define their requirements for us. At the same time, work at the desktop was predominantly aimed at visualising and analysing the results from the mobile surveys, and at programming the overall system, which is an unavoidable task that has been mainly handled by us. The programming task was aided by utilising and appropriating third party sources such as software, maps, software libraries and any other readily available data that might be imported. At this stage, the server merely acted as a data-hub (using existing software), but work on a data-model for the final application might have already been undertaken. The elaboration phase ends with a clear set of requirements for the final architecture and especially the location mechanism.

These requirements then drive the **construction phase** which might consist of several milestones and ends with a fully usable system. In the case of Love City, this development iteration consisted of 2 milestones. Development towards milestone 1 transformed the game from an idea into a first fully workable system. The resulting prototype was then user-tested and evaluated which led to adjustments in the system design. In this case the game needed to be rewritten from a thick-client smartphone version to a thin-client SMS version to accommodate the target groups' needs for an accessible game that did not require any specific type of mobile phone – this was done for milestone 2. Ideally the resulting second system would have been user-tested again, but this step had to be skipped for time reasons so that the game has been deployed to the actual users immediately after completion. In the case of Rider Spoke, the construction phase completed without any alterations to the design.

With a complete system at hand, the experience is finally staged and deployed to the end-users in the **transition phase**. No more programming and configuration should happen at this point as all attention now turns towards shaping the end-user experience and thus the personnel is occupied with providing run-time support, orchestration and game-mastering.

Communication between stakeholders is commanded at all stages of the process, but from our experience this is especially the case between project phases, where we have always met in person and discussed the state of work. This is especially important before entering the construction phase, to ensure that the final system meets everybody's expectations and that no time is wasted with unnecessary work. It can be recommended to formalise this step and produce a written document that provides a sufficiently detailed account of the architecture, functionality and upcoming tasks from an engineer's perspective. This document must be reviewed by the experience designers who should check that it entails all required functionality and who might prioritise certain tasks based on their anticipated usefulness. In a commercial context, this design document could be part of the contract to clearly mark the end of the elaboration phase, as any redesign at a later stage will result in additional cost and effort.

In retrospect, it appears that splitting the whole development process into different iterations was a good idea as it allowed more focussed work on smaller chunks of work one at a time and – more importantly – constant revision of the overall plan. This idea was certainly not invented by us, but it can also not be attributed to a single process methodology. It is, for example, immanent to software development methodologies such as "Rapid Application Development" (Martin, 1991) or "Scrum" (Beedle and Schwaber, 2001) and is also at the heart of "Kaizen", the Japanese philosophy of continuous improvement (Imai, 1986). Developers of location-based experiences with a similar scope as Love City or Rider Spoke would be well advised to read up on process frameworks and consider organising their work in a similar fashion. But this should only be done to a level that facilitates communication and planning, and does not encumber the creative flow.

6.4.3 The Artistic Umbrella

When looking at our two studies and several of the reviewed examples, it becomes apparent that an "artistic umbrella" can be beneficial for a better, i.e. more positive reception of the work. This point can be illustrated by comparing two examples from the literature review: Noderunner (page 42) and the Defcon WarDriving Contest (page 39). Both had a similar theme, but Noderunner was much better received.

The legacy of Noderunner is two-fold. On the one hand it is a cleverly conceived and well documented piece of free artistic work which was internationally recognised through its reception of the reputable Golden Nica award and its subsequent demonstration at the Ubicomp conference in 2003. This caused researchers to build on its idea and also generated positive press coverage about the work. On the other hand, however, the game mechanics of Noderunner are at least ethically questionable and probably illegal in several countries, including the United States where it was originally staged. The problem with Noderunner is that participants have to piggyback on unsecured wireless networks in order to score in the game and thereby consciously utilise bandwidth that they have not paid for. While the author reserves himself a liberal position on this, it has to be noted that under different circumstances, i.e. in a less artistic environment, Noderunner might have been seen in a very different light. This becomes clearer when comparing Noderunner with WarDriving. As described in the literature review, the DefCon 10 WarDriving Contest organisers were aware of the legal and ethical dangers of their game which led them to an appropriation of its design. This was certainly influenced by the context in which the game was going to be staged – at the world's biggest hacker conference. Despite their effort of providing an ethically sound and law obeying game that just challenged the presence of Wi-Fi hotspots in public space and made people aware of security issues that accompany this technology, it initially received a lot of negative press. The organisers anticipated this feedback, or as they wrote in their book: *"the press has a tendency to point out primarily the negative aspects of any so-called 'underground' event"* (Hurley et al., 2004: 265). Since then various press articles reported about wardrivers who utilised unsecured wireless access points to gain free Internet access. The wardrivers got noticed and were subsequently arrested by the police (Leary, 2005). One of these cases is about a man who snatched wireless bandwidth from a coffee shop in Vancouver (Blass, 2006). Apparently the same "offence" of utilizing a coffee shop's unsecured wireless network was committed by a participant of Noderunner. This action was even willingly documented with a timestamp (06/22/04 03:18:30 pm) and a photograph (see figure 2.10, right, page 42 ff), and archived on the Noderunner website.

It can be assumed that the artists behind Noderunner knew that their game was staged in the legal grey area that circumvents any new technology. In their statement for the Golden Nica they wrote: *"As artists, we combined game design with the existing culture of the open wireless movement. Instead of creating an artificial game environment, we tapped into the revolution that was already happening around us. Our goal was not just to contribute to a new genre of public* *art, but also to actively engage the general public in a vital cultural and technological transformation. Noderunner is continually reinvented by the citizens who build the network and run the streets. The game is an entrance point to the political and social movements behind wireless. We offer Noderunner as celebration of free and open wireless connectivity and as a symbol of the city's cultural flexibility and potency"* (17).

It appears that Noderunner's designers were far less concerned about ethical and legal constraints for their work than their fellow DefCon counterparts; or rather they were more prepared to challenge the boundaries of what is acceptable and what is not. Demonstrating such an *"anything goes"* (81) attitude when dealing with location-based experiences is probably the biggest contribution of Noderunner. Seeing that the project was highly acclaimed for its vision at the Prix Ars Electronica, it can be learnt that such challenging positions seem to be more acceptable under an open artistic umbrella than they are under more prestressed conditions.

6.4.4 Legal Issues

Regardless of the approach one chooses to take when staging a location-based experience, be it artistic or not, there are legal and ethical issues that need to be considered when dealing with others. Although this might appear boring and uncreative, it is in fact crucial and needs to be reflected in the overall plan. This is because things sometimes just go wrong, and could even lead into disputes over financial losses or bodily harm. Preparations for such worst-case scenarios had of course been conducted in connection with the presented location-based experiences, e.g.:

- Waiver liabilities: the DefCon WarDriving contest required participants to sign a legal disclaimer that waivered the hosting venue from any liabilities.
- Secure assets: Uncle Roy All Around You asked for a deposit and ID in return for handing out the expensive mobile kit.
- Participants' safety: Rider Spoke provided a detailed security briefing as well as safety accessories such as helmets and high visibility jackets. Still,

it required participants to sign a legal disclaimer and leave their credit card details as a deposit.

- Data protection: Love City participants had anonymous online identities and their real data has not been disclosed to any third parties.
- Privacy: Love City monitored participants' movements over a prolonged period of time by tracking their mobile phones. By law, this required an explicit opt-in as well as the possibility to easily opt-out at any time.

This is not to say that legal disclaimers should be used as a licence for carelessness; to the contrary! Great care must be taken when designing experiences for people to ensure that no one gets harmed. This is also required by ethical research committees at many universities.

6.5 Outlook

The artists' key objective in the early phase of authoring was to gain an understanding of the granularity of the utilised wireless network infrastructure and its potential performance as a location mechanism. This is because the network's granularity, and thus its expected positioning accuracy later in the experience, informed the notion of locations in the design. The artists needed an awareness of the infrastructure, so that they could make sense of the sensing system (Bellotti et al., 2002). Or in other words: the artists wanted to know if the networks would allow them to do what they had in mind. Authoring tools and workflows for location-based experiences must support the users in finding an answer to this question.

In addition to everything that has been said so far in this thesis, the following recommendations might provide some further guidance and inspiration.

6.5.1 Playful Process

A playful creative process was important to the artist, in fact they regarded the fluid relationship between technology and creativity as an asset of our collaboration. To a degree this can be achieved through project management (e.g. workshops, design documents, milestones, bug-trackers, etc.) and communication, but this fluidity also has to be embodied in the user-interfaces of the tools that are being used for the different tasks, and in the development environment that is used for prototyping the tools and the system.

6.5.2 Learn from Others

Even though location-based experiences are a rather young category, there is already a body of literature on the subject. Chapter 2 (page 20) tried to provide a fair overview and summary of the field, but will have undoubtedly left out several relevant papers and pieces of work. The interested reader might find the following references (mostly recent textbooks) useful for further research and self-study: pervasive games (Montola et al., 2009a, Borries et al., 2007), games in general (Salen and Zimmerman, 2003, Huizinga, 1939, Björk and Holopainen, 2005, Gilbert, 1989), human computer interaction (Dix et al., 1993, Shneiderman, 2002, Greenfield, 2006, Harper et al., 2008, Dourish, 2001, Preece et al., 2002, Raskin, 2000), rapid prototyping of user interfaces (Smart et al., 2005, Rappin and Dunn, 2006), and rapid prototyping on mobile phones (Scheible and Tuulos, 2007).

It is also always interesting to sneak a peek into adjacent disciplines. The Mediascapes team (page 89) already noted the similarity of the field to the world wide web (Hull et al., 2004) and the film industry (Reid et al., 2005a), and proposed that much can be learnt from the way how work is done in those domains. The analogy to the web is especially interesting in this context, as many location-based experiences might want to employ the same network infrastructure. On a technical note, it can be stated that choosing HTTP for data-transport is usually a safe bet, as HTTP requests will pass through most networks without being blocked by firewalls. On a more general note, it can be stated that the democratisation of publishing on the web was facilitated by standards (read: HTTP and HTML) and a multitude of authoring tools and hosting services of different complexity that empowered everybody to use it. Web pages can also be very useful interfaces for location-based experiences on many different levels, including front end (users), back end (administrators and orchestrators), and documentation and dissemination of the project (public). In short: there is a lot that can be learned from the web.

Then there is, of course, the computer game industry, whose production workflows nowadays already resemble that of the film industry, with its finely distinguished roles in the production process. This might be the future for the development of location-based experiences as well. But, and one might say luckily, at the moment everything resembles much more of the infancy of the computer games industry in the 1980s and early 1990s: small teams, small budgets, short development times, and a lot of creative freedom.

Above and beyond these adjacent disciplines, we want to add that the field of geographic information science (GIS) is another contestant for closer inspection. In fact, we believe that some of our workflows, such as mobile surveys and visualisation of spatial data, are so similar to work in GIS that much can be learnt from that field and some tools might possibly even merge in the near future.

6.5.3 Use What is Available

From our own experience, one would be well advised to do a market research and use what is available, unless there is a compelling reason not to do so. This applies to hardware, software, infrastructures, and game designs likewise. One of the conclusions from the literature review was that building location-based experiences using off-the shelf components was already possible ten years ago. Since then the situation has only improved. The creative part is usually more in the combination of existing components, practices and ideas, than in the development of something revolutionary. It is this evolutionary process of combination and appropriation, which turns the whole thing into something new.

6.5.4 Build Up Experience

We have observed how understanding the wireless infrastructure was a learning process that had much to do with the accumulation of experiences over time, i.e. the German connotation of the word experience. This clarifies that the development of location-based experiences is a craft that has to be learnt. Fitting tools for this craft should therefore support the learning process of its users.

7. Summary and Conclusions

This thesis set out to facilitate the development of location-based experiences in general. Specifically, it took a user-centred approach to make the designers of location-based experiences more aware of the limited availability and varying performance of the wireless networks that they are using in their works. The scientific method was for the author to first become an expert in building location-based experiences and its prevalent practices, to learn about the designers' tradition and learn their language, and then design changes in their workflows in a cooperative prototyping process. This was evaluated using qualitative research methods applied to the development processes of commercially commissioned location-based experiences which were built in collaboration with professional experience designers.

7.1 Summary

Chapter 1 sat the course for this thesis and formulated the overall goal. The presented methodology was grounded in third generation participatory design, i.e. cooperative prototyping. The chapter provided a condensed overview of this design tradition, including some of the philosophy behind it.

Chapter 2 provided the foundation for this work. It first clarified relevant terms that were used throughout this document and then presented examples of and seminal work for location-based experiences. This information was then analysed to highlight key aspects in the presented examples, such as production categories, recurring game-designs, and technological choices. The literature review continued with a section on how to practically build such location-based experiences from available hardware, software, and infrastructure components. This led to a detailed review of the state of the art in authoring tools for location-based experiences and their underlying principles: background map, external content creation, content trigger zones and events. A discussion of the problem of uncertainty, which circumvents any project that relies on wireless infrastructure, led to the motivation for this thesis: how can we enable designers to take into account the usually invisible wireless computing infrastructure as well as the nature of the physical environment while authoring an experience?

Chapter 3 provided the conceptual framework for this thesis. It proposed to reveal the characteristics of the wireless infrastructure to the designer of an experience during the authoring process. This was thought to increase the designer's awareness of potential technical issues and thereby provide scope for modifications to the design before the experience gets deployed to the public. The framework essentially proposed to support the designer of a location-based experience with three layers of information: a *physical world layer*, with representations of the target physical environment, an *infrastructure layer*, with representations of the wireless computing infrastructure across this area, and a *content layer*, with representations of the associated digital media, content trigger zones and events. The infrastructure layer is the main contribution of this thesis; a full paper about it has been presented at the Ubiquitous Computing conference in 2006. The following section about location mapping motivated separating the notion of "location" from the related terms "position" and "place". This led to the presentation of an abstract location model that could be used for triggering content in future location-based experiences. The framework chapter closed with four challenges that needed immediate attention and a plan of action to pursue the practical thesis work.

Chapter 4 presented the first of the two studies that support this thesis. It was about the development of a location-based experience for mobile phones that linked three cities via one virtual city. After a brief outline of the artistic idea and the resulting end-user experience, this chapter provided a detailed account of the actual practical work that led to the successful delivery of the project. A user-centred participatory design process elicited the artists' requirements for tools that supported their work in the field and in the studio. The artists gave feedback on different iterations of these tools and thereby helped shaping them into a form that provided the required functionality. The artists highlighted the importance of a fluid workflow that seamlessly linked work in the field with work in the studio. Both types of work were necessary and contributed to the artists' understanding of the wireless infrastructure in relation to the physical environment. The artists needed to be able to converge the available information in order to appropriate it for their design. The abstraction of the infrastructure visualisation and authoring tool was seen as indispensable for this process.

Chapter 5 presented the second study of this thesis which was about the development of a location-based experience for cyclists. This work was conducted with a different group of artists who used a different wireless infrastructure to support their design. The chapter started with a brief general description of the end-user experience and elaborated on the technical background. It then continued with a detailed account of the iterative development process that spanned several prototypes and three public stagings of the experience in different cities. Similar to the first study, our artists engaged in a distributed workflow that combined work in the field with work in the studio. Contrary to the first study, the location-mechanism of this experience could be configured by the artists to provide different levels of granularity for locations. Configuration was a rather complex process. In the end, it involved the use of a simulator tool that operated on self-collected survey data. The simulator presented the artists with an analysis interface where they could explore the effects that different input parameters had on the performance of the location-mechanism. Analysis work was supported through visualisations of the resulting spread of locations and the connections between them in a graph visualisation package and in a geobrowser. Across the three public stagings of the experience it could be observed how the artists initially employed a trial and error approach to configuration. With the emergence of a routine, and supported by better tools, they eventually became more autonomous and could finally make informed decisions on how to best configure the location-mechanism to support their design.

Chapter 6 provided a reflection on the two studies and a broader discussion of the overall thesis topic. It highlighted the importance of understanding the technology as the foundation for design, and how this can be facilitated by the infrastructure layer. The discussion elaborated on how our artists made practical use of the provided tools for their work in the field and in the studio. It summarised desirable key features for such tools and provided ideas for scaling up the mobile survey process for use in larger projects. A summary of Koch's criticism on visualisations led the discussion onto the broader stage of how to make sense of location. Our two studies confirmed that visualisations or tools alone won't solve the problem of uncertainty for anyone. It also requires multi-disciplinary discussions and practical work to build up an own appreciation of the problem for the task at hand. The key task for our two artist groups was to understand location in their design, i.e. with regard to the underlying technology, and how to best map their content to locations. The discussion listed several aspects that can facilitate content mapping if embodied in the supporting tools. A brief presentation of post-event analysis methods closed the user-centred part of the discussion. The following engineering centric section revisited the challenges that were outlined in the framework chapter and discussed ideas for system level improvements. The discussion chapter closed with reflections on best practices regarding opportunity, the development process, the artistic umbrella and legal issues, and with an outlook that presented recommendations.

7.2 Reflection on Key Principles

The key principles of the work in this thesis were the three layer model for infrastructure visualisation, the abstract location model, and our artist-led participatory design approach. This section briefly reflects on these principles.

7.2.1 Three Layer Model

The three layer model facilitated understanding the wireless infrastructure through the implementation of its infrastructure layer. Although both projects worked with different technologies, they also both relied on GPS for geo-coding and thus accepted GPS as their ground truth. This was an acceptable compromise for the work that was carried out in both studies, as the anticipated precision of the location mechanisms was at least an order of magnitude worse than GPS alone. Nevertheless, visualising the performance of GPS itself, potentially with the help of building models as proposed by Steed et al. (page 112), appears to be an interesting avenue if one was after a higher precision and better understanding of that location service.

Going beyond the technology that was studied for this thesis, it is also interesting to visualise the characteristics of other pervasive infrastructure technology, e.g. RFID (82), and thus make it easier to understand and design for these proximity-based technologies.

7.2.2 Abstract Location Model

The abstract location model facilitated implementing the infrastructure visualisations, as it provided a way to distinguish between different locations and visualise their spatial layout. Moreover, such an abstraction also facilitates changing the sensing technology of a location-based experience without having to redesign other parts of the application logic. This was seen in effect in the Love City study, where the final version of the experience used a different technology to sense the location of the participants. The abstract location model thus facilitates the appropriation of space in the context of pervasive technology. Looking back at Dourish's distinction of space and place, where he proposed to be flexible about the structure of space to allow designing for place (page 127), we found that our artists were comfortable with the granularity of locations that was provided to them by their chosen technologies. The structural flexibility of the abstract location model ultimately allowed our experience designers to better concentrate on their designs, rather than concentrating on the technology.

Besides the main thesis work, the author also wrote a reference implementation of the abstract location model called Pytheas, which is available at Sourceforge (83). Since then, this implementation has been used to drive another location-based experience for cyclists called "The Sillitoe Trail" (Rowland et al., 2009).

7.2.3 Participatory Design

The work presented in this thesis was designed to be carried out in a cooperative prototyping process, as described in the methodology section (pages 9,13). By building on the designers' tradition we obtained valuable feedback on our tool prototypes from early on in the development process. In fact, the development of certain features was driven by the experience designers, which sometimes meant a sudden death to features that were perceived useful by the author, but were perceived less useful by the designers. Conversely, the implemented features in our prototypes were more to the point for the immediate task at hand, and sometimes the experience designers even surprised us with requirements that we would not have thought of.

In a way, our cooperative prototyping process was one of designing for designers, as our end-users were also concerned with designing something for their end-users at the same time: the location-based experience. There were thus two design processes going on in parallel, our cooperative design process and the artists' experience design process. In order to facilitate speaking about this overall approach to design, one might call it an "artist-led approach", as the artists' curiosity and urge to deliver is a major driver behind this interdisciplinary process.

The artist-led approach is an established way of working in the Mixed Reality Lab, which has been creating, touring, and studying mixed reality performances in collaboration with professional designers for more than ten years. As a final step that goes beyond what has been done in this thesis, the Mixed Reality Lab also finally closes the loop by analysing the end-users' experience with a mixed reality performance and feeds that back into the design.

7.3 Ideas for Future Work

The following ideas appeared upon reflection of the work that has been done for this thesis. They have not, yet, been implemented, but seemed to be too interesting to let them slip away, and might thus present pointers for future work.

7.3.1 Improved GPS Triggering

Current GPS-based experiences trigger their content based on a point-intersection with pre-defined shapes. Due to the low sampling rate of the GPS receivers, which only provide an updated position about once per second, and the system immanent GPS jitter, this sometimes causes content to be skipped. The problem is that one GPS reading might position a participant on one side of the content triggering region, and then the next reading would appear on the other side of the region; thus the content has been effectively missed. The observation of this phenomenon usually leads to a discussion about appropriate sizes for the content triggering regions. We have observed this for ourselves in the Rider Spoke study, where the artists conducted a test of GPS-based trigger regions (page 194) and reported that one region was too small and was therefore missed as they travelled past it with their car.

Upon reflection, it appeared that there might be another solution to this problem. If the GPS triggering mechanism would not operate on a single reading, but on a history of readings, it could test for intersection with content triggers against lines or even triangles. Applied to the above example, a simple line constructed from two GPS readings on either side of a trigger zone would cut through the zone and cause it to trigger. To compensate for additional jitter, it might be interesting to construct a triangle of the last three readings and use that for intersection tests.

7.3.2 Work on Edges

On a similar note, previous work on a location-based experience called "Savannah" (Benford et al., 2005) has found out about the importance of edges in relation to content trigger zones. Savannah was a game where groups of school children played lions in a virtual savannah. In order to successfully hunt their prey, participants had to be in the same virtual location and start a coordinated attack. It was found that this mechanism was sometimes undermined by human factors. Members of the group would discover a new location, notify their partners, and team up with them in order to coordinate the attack. Apparently these gatherings frequently occurred on the edges between two adjacent regions, sometimes causing members to be located in different regions and thereby causing the attack to fail. In the case of this multi-player game it was proposed to extend the handling of locales to embody a more dynamic approach that takes the presence of group members into account.

Similar discussions were observed in our two studies, where talk about the granularity of locations and the fluctuating nature of the wireless networks frequently led to a discussion about overlap and blurry boundaries. It might be interesting to design for edges in the trigger mechanism right away. So instead of testing for containment in location (a) or (b), the trigger would set off when the location changes from a to b (a-b). Work on analysing edges has been presented before (Nurmi and Koolwaaij, 2006), but it would be interesting to see how such a system could be used by designers.

7.3.3 Tagging

This practice of applying freely chosen keywords to locations is often referred to as collaborative tagging or 'folksonomy'. In the online-world, folksonomy is one of the main principles behind the current advent of social "Web 2.0" applications where users are completely free to apply any tag to any (online) location. These tags act as meta-data and are used to organise and manage digital information (Smith, 2008). Given that a sufficient number of people come across and tag the same location in a similar way, its importance is automatically raised and it reaches a higher ranking. The theory behind this collective opinion is the so-called "wisdom of crowds", as described by Surowiecki (Surowiecki, 2005). A typical example of tagging on the social web is the delicious bookmark service (84), where people use tags to organise and share their bookmarks instead of putting them into hierarchical structures like before in their browsers. Another example of social tagging are *games with a purpose*, as described by Ahn and Dabbish (Ahn and Dabbish, 2008). They describe games that contain tasks which are easy to solve for humans, but hard for computers. Their games are designed so that data collected as part of the gaming activity could be used to aid solving computational problems later on. Their well known ESP-Game, for example, challenges two players to agree on a vocabulary for describing pictures that they are looking at. The players don't know each others and have no way of communicating other than providing tags that fit the presented images in a given time frame. Players get points for matching tags and are thus motivated to provide an emergent ontology for images.

It would be interesting to investigate how similar tagging mechanisms could be devised to produce useful data for location-based experiences. Matyas et al. (Matyas et al., 2008) and the Hitchers game (page 54) provided initial approaches towards this idea, but much more work seems to be necessary.

7.3.4 Transfer Knowledge

As much as we can learn from the success of the web, it also seems adequate and timely for researchers in the field of location-based experiences to transfer some of their knowledge back into the web domain. As the social web is already a reality, it seems like the next big thing to happen is the location-based social web.

The W3C currently drafts their geolocation API specification (85), and all major players of the web, as well as countless start-ups, recently revealed their web products for this emerging market, e.g.: Mozilla Geode, Yahoo Fire Eagle, Nokia Plazes, Loki, Rummble, Dodgeball and Google Latitude. There are also new technical platforms which provide hybrid mobile software location services that consists of cellular positioning, Wi-Fi and GPS, such as Navizon or Skyhook Wireless (used in the iPhone). And finally, first deals for hybrid communication and positioning chipsets have recently been announced, e.g. by SiRF and Broadcom. So now that hybrid techniques feed back into mass market hardware, ubiquitously available position readings on mobile devices are only a matter of time. This thesis previously discussed the importance of working with location rather than position (page 124), and argued that many location-based applications care too much about position and too little about location. As the upcoming W3C geolocation API and the new hybrid chipsets will most likely operate on geographical coordinates, it seems timely to investigate how this new breed of applications and devices could still benefit from a more abstract notion of location, before the common approach to building location-based applications finally hardens to be geographical position only.

A second point for improvement of current web domain practices informed by our work would be to update tagging mechanisms with clean and collate functionality, as we have used it in our studies, e.g. with the Rider Spoke content ranking mechanism (page 346). It appears that current tagging mechanisms on the web, as they are used by Flickr, Delicious, and many others, provide flexible means to organise web pages with freely chosen tags. Over time, this provides an emergent ontology that classifies the tagged content. But the problem is that due to misspellings or the use of different words for related content, similar content might be tagged in ambiguous ways, which reduces the usefulness of tagging. A clean and collate functionality might help to improve this situation.

7.3.5 More Layers

Our artists suggested extending the use of the current visualisation tools to provide live visualisations of what is going on in the system when the experience is running. Doing so would close the feedback loop for them and allow for orchestration in the sense of Crabtree et al. (Crabtree et al., 2004).

This revelation of the players' positions and activities effectively constitutes another layer in our conceptual model (page 121). In addition to the three layers that are already present (physical world, infrastructure, and content), this fourth layer would basically foster the understanding of social practice in the physical space. Apparently, this was also a fruitful comment from one of the anonymous reviewers of our Ubicomp paper (Oppermann et al., 2006). A social layer could reveal the routes that players take and the places that they inhabit while the experience is on. Such a layer could possibly nurture a better understanding of space and its embedded human practice, as outlined by Dourish in his book (Dourish, 2001). Thus, a social layer could yield interesting results, not only for the runtime staff, but also for sociologists that are studying the experience. Examples of projects that reveal this kind of information are Cabspotting (86), which reveals taxi routes through San Francisco, and the work of Girardin et al. (Girardin et al., 2007), who revealed tourist movements by analysing large sets of publicly disclosed geo-coded photos.

On a different level, this layer might even just become part of the experience itself, as some kind of remote presence is created for an audience through shared displays or the Internet. This use was previously demonstrated in other projects, e.g. *Can You See Me Now?* (page 37). To a lesser extent, this has also been used by us for the dynamic visualisation of player positions on the *Love City* public website (page 145), and in a projected visualisation of players' positions and heart rate frequencies in *Heartlands* (page 370).

A fifth layer might contain situated annotations of the experience. We have explored this idea for an annotation layer in the Love City authoring workflow, where our artists could record geo-referenced audio samples while out and about (page 169). Similar to a social layer, an annotation layer could be interesting to staff, sociologists and the audience.

Finally, a sixth layer might try to visualise participants' emotional and bodily reactions to an experience, i.e. how people are feeling in various places and situations. Although measuring such information can be hard, several projects are already charting out the territory. Examples include the aforementioned Heartlands, the Biomapping project (Nold, 2004), and the Thrill laboratory (Schnädelbach et al., 2008).

7.3.6 GIS Integration

Geographic Information Systems (GIS) could provide support for the authoring of location-based experiences. The following presents two ideas for this.

We have seen how our artists in both studies pre-planned their survey routes on paper and then went out to capture data along these routes. The paper map supported them in reasoning about different areas that they would deem interesting to capture. While we do not want to replace human reasoning about places in the experience design, it can be foreseen that this planning process for surveys would only work well for small groups of people, and therefore needs to be computationally supported when scaling up the survey process. Integration of our survey exercises with community driven data collection projects like OpenStreetMap was previously proposed (page 276). Those projects are supported by thousands of users who are all planning their journeys individually. While this is part of the fun and freedom that makes up their experience, they probably would not mind some better support. Using GIS software, it would be feasible to automatically reveal areas that have not been surveyed. Such white spots on the map could then be sent to participants who are willing to help, but would not want to produce duplicate work and waste their hours. This could even be supported in-situ with a mobile tool that sends its current geographical position to a server and receives back a map of unknown territories that would need to be mapped.

A second idea would benefit from additional data layers that might be available in a GIS tool. Seeing how our artists wanted to design for places that people would visit, it seems to be interesting to research how this manual process could be computationally supported as well. Data about the transport networks, land use, points of interest, or even demographics could be useful and interesting in this respect. A tool that integrates such data sources might be able to automatically propose potential hotspots for humans and thus facilitate the design process, as well as the survey process. As an example, imagine an experience that needs to link certain points of interest in a city via the public transport network. Of course its designers would want to do their own research and discuss the requirements with their funders. But the designers' work would probably benefit from a tool that automatically proposes routes between points of interest from an analysis of the geospatial topology.

7.3.7 Temporal Variation

Our artists also raised the knotty problem of temporal variation on the infrastructure layer; whether and how the infrastructure varies across time, e.g. due to congestion, and how to show this effectively. This theme was not explored further by this work, but appears to be an interesting avenue for further research. It might be interesting to apply time-based filters to the sampled data-sets, so that only samples that satisfy certain conditions will be visualised. Similarly, a time-line based visualisation might reveal interesting trends regarding location granularity and the use of locations by players.

On a social layer, temporal variation has to do with availability and interruptibility of individuals. The design of Day of the Figurines (Flintham et al., 2007) tried to cater for such temporal patterns. The game has been further studied to find out if player interruptibility and engagement could be modelled (Fischer and Benford, 2009).

7.3.8 Gaps and Conflicts

Our artists in the first study reported that gaps in the infrastructure (i.e. areas of no coverage) can actually provide interesting opportunities that might be exploited for storytelling and gaming. It was therefore suggested to provide another visualisation of the infrastructure that would focus on highlighting the gaps in infrastructure coverage.

On the content layer, a similar observation was made in the second study. As Rider Spoke only allowed one piece of content per location, conflicts would arise where several players recorded answers in a place that was previously marked as free (such conflicts occurred as the game was played off-line and so the notion of occupied locations did not propagate through the system in realtime). It appeared that a tool which visualised conflicts and allowed copying content from over-crowded to empty locations would be helpful for managing such conflicts.

Finally, and taking the previous ideas for social and temporal visualisations into account, it might also be interesting to highlight temporal and social gaps in those layers as well.

7.3.9 Ideas First

Overall, our current approach is led by the data. We conduct survey activities, and visualise the results in ways that fit the data in order to facilitate the designers understanding for the technology that they are using.

In the Love City study, we saw how our artists made good use of a Voronoi visualisation of cell locations. This visualisation was a good fit for the data, as it decomposed the space into areas that are most likely to be covered by the cells that takes precedence at that location, a technique that has apparently also been used for a network planning tool by the operator T-Mobile (Mende et al., 1998).

In the Rider Spoke study, the visualisation matched its data because the abstraction is in the graph, whose construction is dynamically determined on the ground, based on some parameters and the presence of wireless access points in range of the mobile devices. Our supporting tools helped the artists to understand the effects of different settings on the generation of the graph through simulation at the desktop and testing in the field.

However, our artists proposed that it might be useful to explore how this approach to authoring can be reversed, starting the design process with 'ideas first' and then seeing how these might fit onto the infrastructure. For example, the user could begin by drawing desired locations on a map and the editor would reveal how well this idea could be supported by underlying positioning and communications systems. Alternatively, a mobile workflow could include a user visiting key areas and associating abstract locations (names and/or tags) with signal snapshots. This process would require the user to visit a range of locations at the desired target granularity, e.g. room level accuracy. Operating on the sampled data, the system could then try to analyse whether the desired accuracy could be achieved. Based on this system feedback the user would then refine their design to accommodate what is possible with the underlying infrastructure.

7.4 Conclusions

So what can be learned from all this? First of all, it appeared that learning about the technology and tools early on in the process was a good idea. Our doing was concerned with making the technology disappear into the mental background of our designers by first making it perceivable throughout the development phase. This prevented having the technology show its true face at runtime, when it could be too late to factor a potential misconception back into the design under the critical eyes of the end-users (whose suspension of disbelief just broke). Thus, our approach to authoring location-based experiences was in the tradition of seamful design (Chalmers and Galani, 2004) and ultimately ensured a smoother end-user experience.

Second, it appeared that experience designers don't want to be concerned with the tools that they are using – and this includes the wireless infrastructure – when they are designing a location-based experience. Knowing about the tools was rather seen as a necessary skill that needed to be mastered than an end in itself. They wanted to have their tools *ready-to-hand*, rather than *present-at-hand* (page 11, 267). We have also seen how no single tool was enough to support all work, thus we needed a suite of tools, or a toolbox. Such toolbox must integrate with the tools that the designers already use and like to use, so any new tool for them must fit their *tradition* to allow for a *transcendence* in the the sense of the Scandinavian design tradition (page 11). Building a location-based experience will almost certainly involve the need to go outside and do a survey or even the authoring in situ (Weal et al., 2006). But not everything in a creative process can be done outside, thus authoring tools must also continue to support studio work, preferably tightly integrated with the mobile tools to ensure a good workflow (page 256).

Third, designers can really start working once everything is in place for them to build upon, for they know what to do. Designers effectively can reach through the layers of infrastructure to realise their vision and design compelling end-user experiences. Knowing about specific characteristics of the utilised infrastructure helps them appropriating their design. Ultimately, the technology does not even matter for a good experience. It is merely a transportation channel for the experience, and it is not even the only one, as other parts of the script, the people the end-users are with, and the world around them are all highly influential factors. This thesis was all about minimising the technological noise that circumvents the design of location-based experiences.

On a final note, the author would like to draw the reader's attention to the balance of art, craft and science. User-centred design often separates these roles, but they must really be seen as a unity, as all sides are important to the final experience. Bill Buxton highlighted this holistic point of view in his book about sketching design (Buxton, 2007) and in a talk that he gave at Princeton University (Buxton, 2009); he challenged to rethink *"who or what is a designer, and who is not"*. Edward Tufte provided a similar view as he argued for information visualisation that *"graphical competence demands three quite different skills: the substantive, statistical, and artistic"* (Tufte, 1983: 87). In a way, these diverse skills are united in multi-disciplinary projects as we conduct them. But to go even further, one should revive the Bauhaus manifesto of Walter Gropius (Gropius, 1919), who highlighted the importance of craftsmanship as the origin of art. Gropius negated the existence of art as a self-contained profession that could exist without craft. Rather, his down-to-earth vision required an equality of art and craft. Through unity and intent, a new guild should conceive and create the (architectural) structures of the future, which should be built by a million hands.

It is tempting to see Gropius' vision as an analogy to the evolution of the Internet, which was also conceived by a few, but built by many. In the belief that the same will happen to the mobile Internet, it is hoped that this thesis contributed a small step towards facilitating the development of location-based experiences.

Appendix 1 – Information Design

"A picture is worth a thousand words" – this often used aphorism nicely demonstrates the primary goal of information design: to prepare information for efficient human consumption. Of course, as a proverb, the sentence is also a simplification of a much more complex matter – in this case cognition.

Encyclopædia Britannica defines cognition as *"the process involved in knowing, or the act of knowing, which in its completeness includes perception and judgement"* (87). The concept of cognition is used across many different disciplines, including psychology and computer science. It lies at the centre of Human-Computer Interaction (HCI), the study of interaction between humans and computers (Dix et al., 1993), and is relevant to designing any kind of computer technology (Winograd and Flores, 1986). Any human-computer interaction has to be abstracted into messages which are then sent over the available communication channels, i.e. the human senses and the computer's input- and output-channels.

Shannon generalised this process in his communication model (Shannon, 1948), as shown in figure A1.1. According to Shannon, an information source produces a message that gets passed on to a transmitter. The transmitter encodes the message into a signal and feeds it into the communication-channel. The channel (the box at the centre of the figure) is the medium which transports the signal and passes it on to the recipient. On its way, the message might get corrupted by noise, resulting in a potentially different signal. The receiver decodes this signal back into the original message and passes it on to the destination for consumption.

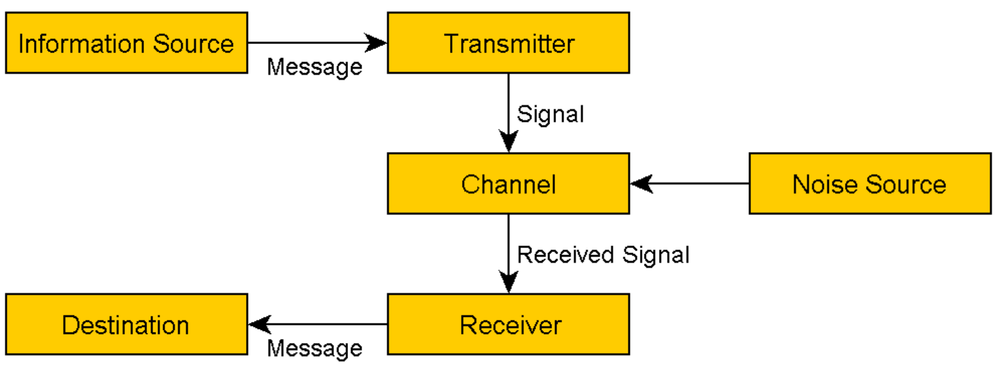

Figure A1.1: A general communication system according to Shannon

Shannon's work is of foundational importance to the field of information theory, which deals with the engineering side of communication, i.e. signal processing, data-transmission, data-compression and the like. It does however not deal with semantics! Shannon himself stated that *"these semantic aspects of communication are irrelevant to the engineering problem."* (Shannon, 1948: 1).

The importance of semantics in communication was later stressed by Schramm (Schramm, 1954). His communication model differs from Shannon's through the addition of an interpreter step: in addition to Shannon's successful delivery of the message, Schramm also requires that the message is to be perceived and understood in the intended way. This additional requirement pays tribute to the human cognition (*"which in its completeness includes perception and judgement"*) and lays the foundation for information design.

Pettersson states that *"information design practitioners seek to make complex information easier to understand for the intended receivers"* (Pettersson, 2002) and Tufte closes his remarkable book about the visual display of quantitative information (Tufte, 1983) as follows:

> *"What is to be sought in designs for the display of information is the clear portrayal of complexity. Not the complication of the simple; rather the task of the designer is to give visual access to the subtle and the difficult – that is, the revelation of the complex."*

Information design itself is agnostic to the medium, i.e. not limited to visual information as initially implied by the proverb at the start of this section. Horn provided the following general definition:

> *"information design is defined as the art and science of preparing information so that it can be used by human beings with efficiency and effectiveness [..]"*

Horn's definition stresses the importance of efficiency and effectiveness as the distinguishing values from other kinds of design (Horn, 1999). Note how his definition is targeted at the receiver of the information and does not state anything about the effort that is required for preparing the information to achieve the goal.

The following sections provide a few examples of good information design. Due to the spatial nature of location-based experiences, this selection concentrates on visual communication and leaves out other strands such as accessibility (Aycinena, 2008), rhetoric (Davis, 1999), writing (Strunk and White, 1962), or media (McLuhan, 1964).

A1.1 Charts

Charts show numerical data in a graphical form. They allow for quick visual inspection of the data and can sometimes even reveal the unexpected. An example for the former is the common use of chart to visualise election results or public opinion polls in the media. Figure A1.2 shows the result of a weekly public opinion poll conducted by the German television channel ZDF, conducted on the 24 October 2008. They asked the question: "which political party would get your vote if there would be an election next Sunday?" The resulting chart (left image) arranges the percental distribution of votes in a semicircle; this mimics the actual layout of the Bundestag (right image). This visualisation is efficient as it immediately reveals the distribution of votes without much mental work. This chart also makes good use of colour as it uses the official colours of the political parties, which are part of their corporate designs and thus widely known.

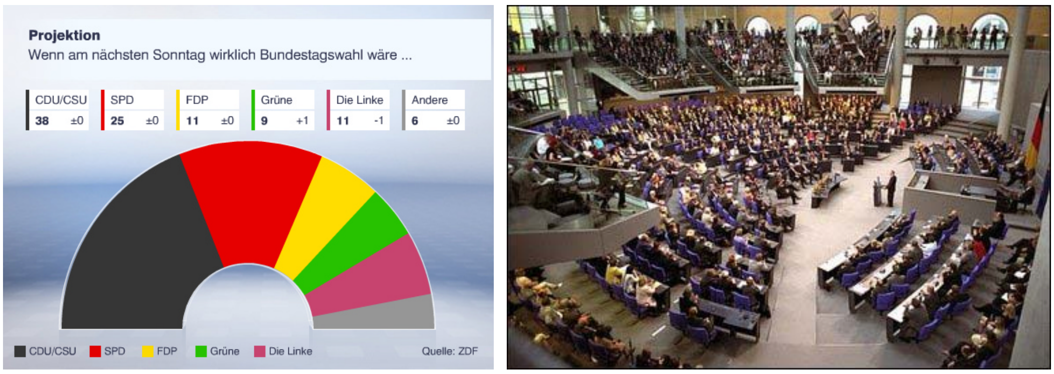

Figure A1.2: Visualisation of an opinion poll for the German Bundestag (left), Layout of the actual Bundestag (right) (Sources: (88), (89))

An example for revealing the unexpected through the use of charts was given by Anscombe, an English statistician. He advocated the use of visualisations for statistical analysis at a time (1973) when the general belief in the field, according to his own judgement, was that *"numerical calculations are exact, but graphs are* *rough"* (Anscombe, 1973). Anscombe argued that "*a computer should make **both** calculations **and** graphs. Both sorts of output should be studied; each will contribute to understanding.*" He designed 4 data-sets to support his argument; they are today known as the Anscombe Quartet (Table A2.1).

X1	Y1	X2	Y2	X3	Y3	X4	Y4
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

Table A1.1: Anscombe Quartet

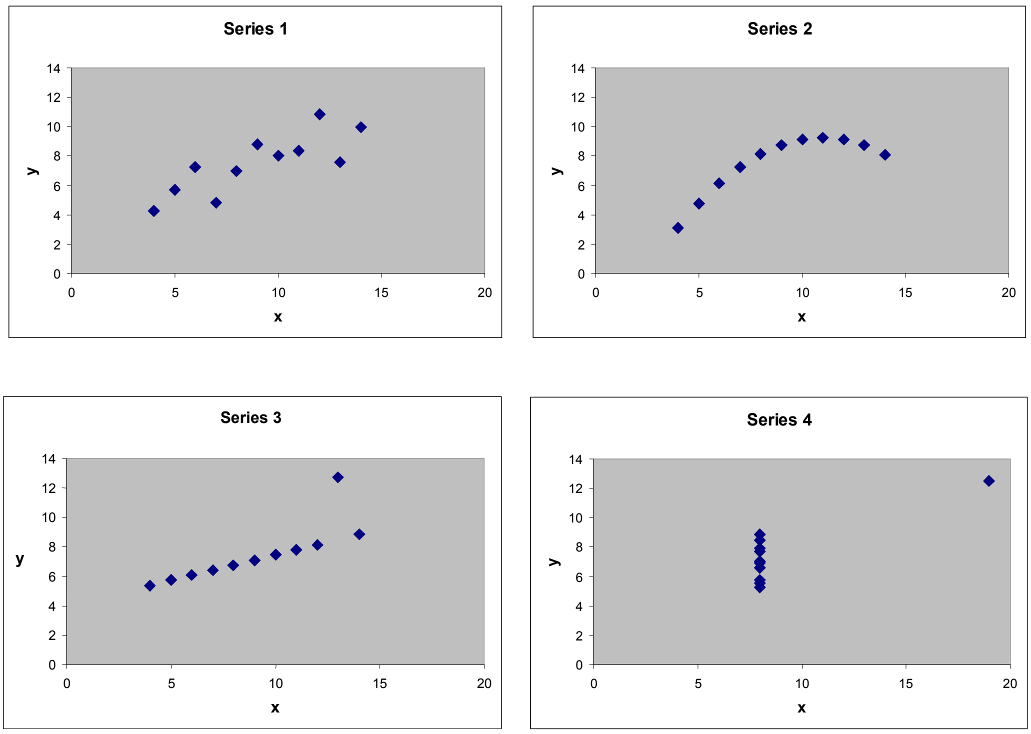

Figure A1.3: Visualisations of the Anscombe Quartet

Each data-set consists of eleven (x/y) pairs, which, when analysed numerically, would yield identical statistical results, i.e. the same mean values for x's and y's (9/7.5), the same equation for the regression line (y = 3 + 0.5 x), etc. Judging from the statistical results alone, one would imagine the data-sets to be very similar. It is only by looking at the visualisations of these data-sets that one would discover that they are actually very different (figure A1.3).

A timeline is a specialised chart that presents a chronology of events or measurements. This is usually done by assigning time to one of the axes of the chart, often the x-axis, and the measured value to the other axis. Timelines facilitate the understanding of historic data and are often used to reveal trends or patterns. Figure A1.4 shows two images from a report on UK house prices (90) done by Nationwide, the UK's biggest building society with about 15 million customers. The report is based on data from Nationwide's own loan approvals to home owners. The left image shows the development of house prices between 1978 and 2008 in absolute terms as raw data and with a fitted curve. The right image shows the annual change in house prices expressed in relative percentage for the period of 2005-2008. Both images clearly show the trend of declining UK house prices, starting at around early 2008.

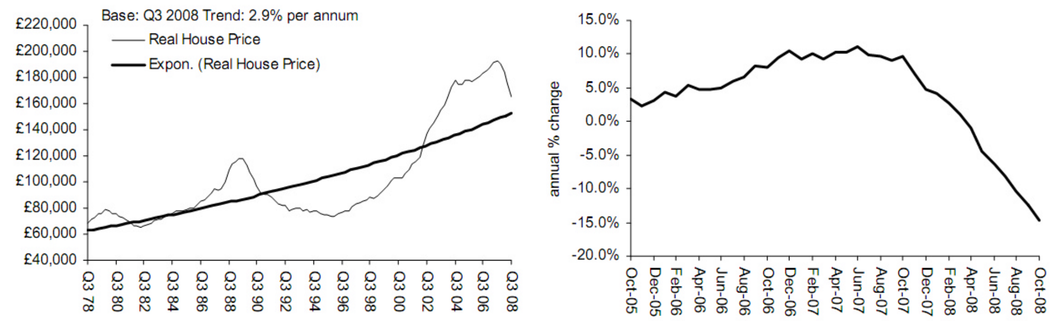

Figure A1.4: Long Term Real House Price Trend (left), Annual % Change in House Prices (right) (Source: (90))

A1.2 Maps

Maps are visual representations of spatial relations. Although a map could represent any kind of spatial data, one usually refers to something as a map if it shows the topographic relation of objects on the surface of the earth. According to Linke (Linke, 2003), a map differs from nature in 4 aspects – it is: scaled down, flat, abstracted and annotated.

Figure A1.5: Ebstorf map (Source: (91))

The Ebstorf map has been chosen as an example of early mapmaking (figure A1.5). This 13th century map, whose name stems from the Benedictine monastery in Ebstorf near Lüneburg, Germany, where it was preserved until its discovery in 1830, was the largest of the mediaeval mappa mundi with an overall measurement of 3.58 x 3.56 metres. The map cannot be precisely dated but is believed to have been produced between 1230 and 1250 (Torge, 2007) or around 1300 (Stercken, 2008) and has been destroyed in the 1943 bombings of Hannover.

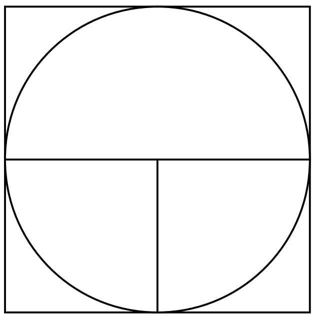

Figure A1.6: T-O map

The map shows a circular representation of the inhabited lands from a European and thus Christian perspective with Jerusalem at its centre, Asia at the top, Europe at the bottom left and Africa at the bottom right. This typical division of early European maps is called a T-O map, where T denotes the divisors of the continents (the Mediterranean and the rivers Nile and Don) and O denotes the ocean that surrounds them (see figure A1.6). Asia is placed at the top of the map because the sun rises in the east. This map orientation is typical for the Medieval and is actually the origin for using the word orientation with regards to maps, as orient is derived from the Latin word “oriens”, meaning “east” (lit. “rising”) (92), and thus orientation in this sense just signifies the relation of the depicted objects around the Mediterranean sea to the east. This cartographic depiction of the world is not to be solely measured against the geographic accuracy of the modern age (Stercken, 2008). Siebold summarises: *“The old adage that a picture is worth a thousand words is well documented and supported by this map. The Ebstorf cartographer accomplishes here what early medieval writers like St. Isidore of Seville attempted to do with mere words, to describe the sum total of accumulated knowledge about man's habitat, the world, resulting in a comprehensive pictorial encyclopaedia, albeit through the rather narrow focus of a religious context. In many ways it does portray the world as it was seen by a great many medieval Europeans“* (Siebold, 1998). In conclusion, the author of the Ebstorf map made extensive use of abstraction and annotation as a means to transport the message but did not try to be geographically accurate. The map must rather be seen as a cognitive map (Görz, 2006) that summarises the contemporary worldview.

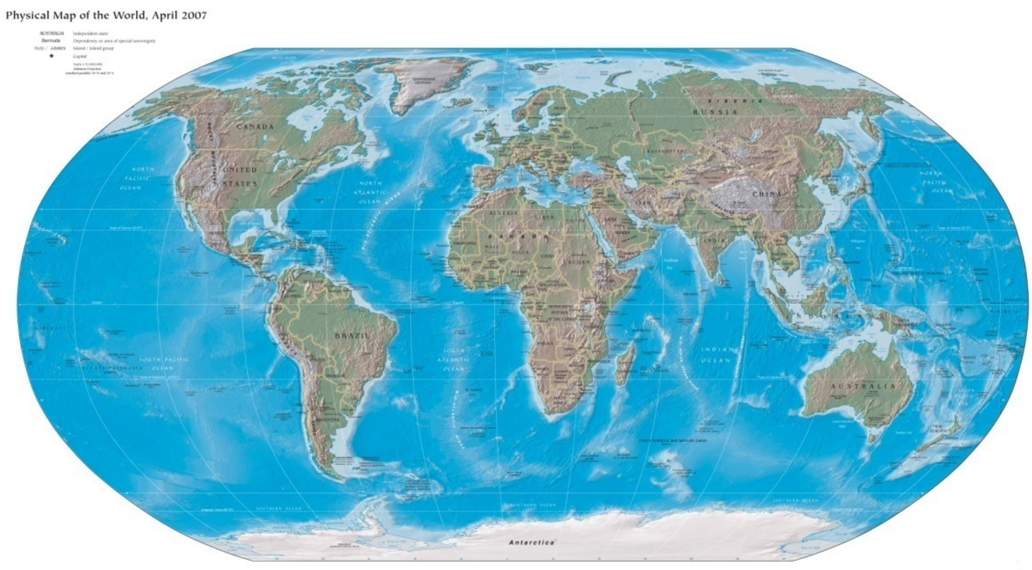

Figure A1.7: Physical Map of the World, April 2007 (Source: (93))

Geographic accuracy is the primary concern of geodesy, which Friedrich Robert Helmert defined as "the science of the measurement and mapping of the earth's surface" (from the year 1880, cited in (Torge, 2007)). Geodetic work led to the production of accurate maps, where distances could be measured on the map and related back to reality. The relation between distances on the map and on the ground is called the scale and it is usually given in the legend of the map. Figure A1.7 shows a map of the world which had an original scale of 1:35.000.000 (1 cm on the map equalled 350 km on the ground). However, this scale could not be preserved for reproduction in this dissertation as the original size of the electronic map document was 44 x 34 US inches – roughly 118 x 86 cm – and the map image had to be scaled down even further in order to fit onto this page with its printable column width of 14 cm. As a result the scale is now by a factor of 118/14 or 8.4 times smaller and 1 cm on the map now equals roughly 3000 km on the ground, but only when measured at the equator. This limitation is due to the distortion of distances on the map. This is a principal problem which arises from the fact that maps are flat but the earth is not. As a consequence a range of map projections have been introduced to overcome this mismatch and allow the measurement and reproduction of a three-dimensional body on a flat surface. Linke provides a pragmatic introduction to map projections and geodetic systems (Linke, 2003).

A second consequence from the flatness of maps is that height has to be represented otherwise, e.g. through colour, lines or shading. Figure A1.7 uses both colour and shading to represent the earth's relief. The map has also been abstracted, i.e. simplified and generalised, in order to preserve those features that were most important for the intended purpose of producing a physical map of the world. In addition, it has also been annotated with names for key geographic features such as capitols, states, islands, deserts, or seas.

A final example for visual excellence in map-making is the London Tube map. The Tube is another name for the London Underground, a metro system of 268 stations and approximately 400 km of track. It is one of the busiest metro systems in the world and carried over a billion passengers in 2007 (94). The London Tube map is the schematic overview of this transportation network and depicts its stations, lines and fare zones (see figure A1.8).

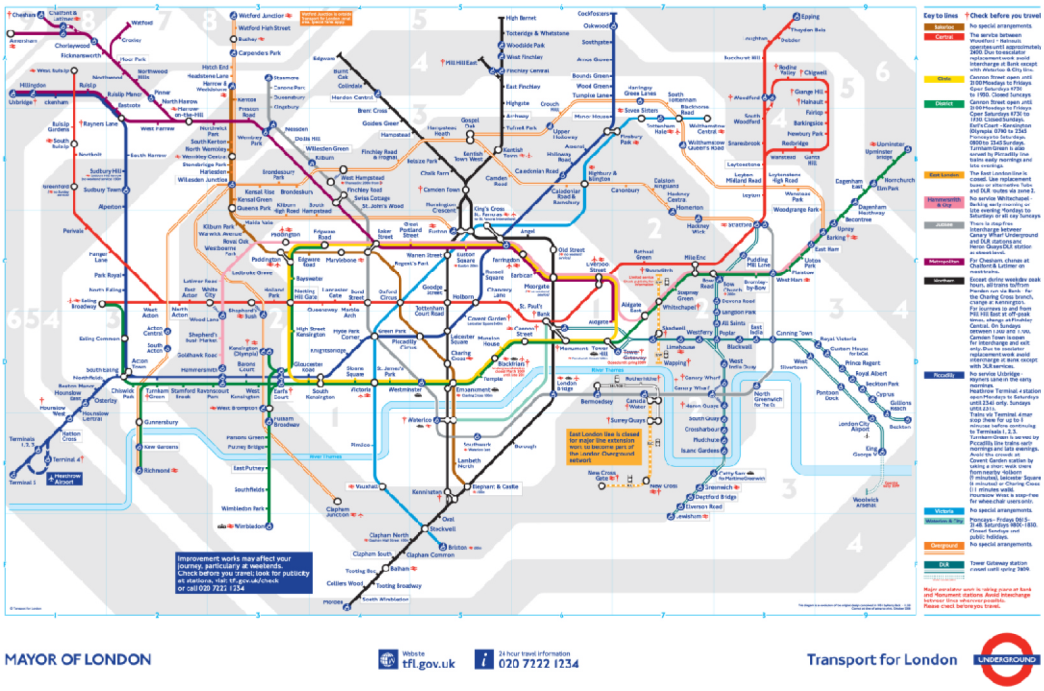

Figure A1.8: London Tube Map (Source: (95))

In order to avoid visual overload due to the amount of information to be portrayed, the map-makers sacrificed geographic accuracy in favour of clarity. The map emphasises on the topology of stations, i.e. it expresses adjacency relations between stations in an abstract visual style where distance and orientation are generally not to scale – although the map also tries to resemble geographic layout where possible. The map makes clever use of colour to denote different lines, such as the Central line (depicted in red) or the Circle line (depicted in yellow). It also uses colour to demark the different fare zones, which are blended into the background in a light grey. Stations that are served by multiple lines are depicted as bold white circles, whereas others that are just served by a single line are marked as little hooks. Imagining that a typical user would know the station name of his destination as well as his current station's name, route planning is basically finding the two stations on the map and tracing the lines between them.

Altogether, the London Tube map is optimised for revealing the most relevant network information at a glance, so that it could be used efficiently and effectively by the tube's many visitors without having to ask for assistance (Leboff and Demuth, 1999). As a schematic map of a transportation network the Tube Map even resembles the Ebstorf map, whose author also advocated its use for travellers. Rosien pointed out that the author of the Ebstorf map annotated the upper-right corner of the map as follows: "it can be seen that [this work] is of no small utility to its readers, giving directions for travellers, and the things on the way that most pleasantly delight the eye."(Scafi, 1999: 64, Rosien, 1952) (96).

Appendix 2 – Software Development

This appendix is meant as a guide for those who intend to programm their own location-based experience. The information presented herein is known to age very quickly and ultimately presents the author's subjective judgement rather than an all encompassing survey. But it has been compiled with great care and it is hoped that it will prove useful for some people. Use it with a grain of salt!

Developing distributed interactive applications such as location-based experiences is a complicated task because any distributed system architecture consists of a wealth of technologies that need to be mastered, e.g. HTML, XML, PHP, SQL, Java, Python, C++, widget toolkits, server administration, etc.. Learning the ropes on all sides of this architecture – mobile clients, server and desktop components – can be a daunting experience due to this technological diversity. Apart from a possible desktop component for orchestration, analysis or visualisation, the most important components are clearly the mobile clients and the server. Development for these profiles is usually constrained in several ways. This appendix tries to summarise the most important software development aspects from a practitioner's point of view and give a brief overview of some existing libraries and frameworks that can support the developers in their work. The selection of material and its presentation and judgement are based on the author's own practical experiences. Consequently they can not be bias free. But it is believed that this appendix can at least provide some initial guidance for the uninitiated reader so that he could make a well informed decision based on his own tradition and use situation.

A2.1 Developing for Mobile Devices

How to program for a mobile device, such as a PDA or a mobile phone, is largely dependant on the operating system in use on the desired device. The operating system is usually either Windows Mobile, Symbian OS, or a Unix-like system (this includes Apple iPhone OS, Palm webOS and Google Android). These operating systems all support a range of programming languages with the most commonly used at the moment probably being Java, C/C++/Objective C, Flash, Python and those supported by the .NET framework. They all have certain advantages and disadvantages, when seen in the context of developing mobile applications, which are briefly summarised in table A2.1 and described below.

The selection of programming languages for this comparison was guided by the availability of a stable release. Many other interesting languages like Ruby currently emerge for mobile platforms, but their release states are too early to be considered for productive use and have therefore been omitted from the table.

	Java	C/C++/Objective C	Flash	Python	.Net
Stable Release	Yes	Yes	Yes	Yes	Yes
Low-level Access	Limited	Full	None	Good	Good
Multimedia Support	Fair	High	High	Fair	High
Code Performance	High	Highest	Fair	Fair	High
Programmers Performance	Fair	Slow	High	High	Fair

Table A2.1: Comparison of programming languages and frameworks for mobile phones

A2.1.1 Programming Languages

Java can be seen as the de-facto standard for mobile phone development. Java offers a relatively high code performance as well as a fair programmer's performance. When used on mobile phones (J2ME), it does however only have limited access to device capabilities, e.g. it can access the camera and a Bluetooth GPS but cannot access the cell ID. The full range of device capabilities can be accessed when programming in C/C++/ObjectiveC. This choice also offers the highest code performance and a very good multimedia support. Basically anything goes, but this is all at the expense of the programmer's performance. This is especially true for Symbian OS C++, which is very different from standard C++. Flash is situated at the opposite end of the productivity spectrum. It offers a well rounded package with very good multimedia support and high programmer productivity. Because it is generally liked by designers, choosing Flash for a project usually also yields very good looking results. The big limitation of using Flash for mobile clients is that it does not support access to low-level device capabilities. Similar to Flash, Python's strength lies in the high programmer's performance. The language also has fair code performance and multimedia support on mobiles. One advantage of Python is that the programmer's performance is not bought at the expense of flexibility regarding access to low-level device capabilities. For example with Python S60 most capabilities are supported, including GPS, camera, cell ID, Wi-Fi, Bluetooth and accelerometer. If needed, missing capabilities can also be implemented in C++ and provided to Python as modules (note that this is not possible with Java or Flash). The last entry in table A1.1 is ".Net". This is not a programming language but a framework which comes with a Common Language Runtime (CLR), much like the Java Virtual Machine. Microsoft .Net supports running code generated by a number of programming languages, including C#, Visual Basic and Iron Python. It is basically a very good choice if one can accept the significant platform lock-in to Windows (with Mono also Linux). This means that .Net is a good choice for PDA based mobile clients but less so for mobile phones, where the market share of Windows is just too small and .Net is not supported by the majority of the phones.

A2.1.2 Device Capabilities

Access to low-level device capabilities is important for many location-based games. If it is a project requirement, e.g. to obtain a mobile phone network cell ID and use that for coarse grained locations, then J2ME and Flash would be effectively ruled out. But as some people still wanted to use their preferred language in this context, they came up with interesting workarounds. The influential Place Lab project from Intel Research Seattle and the University of Washington based their framework on Java and provide the low-level capabilities via a helper program which is written in Symbian C++ and serves its data via a local socket (42) to the main application. Similarly, the Flyer framework for Flash Lite (the mobile phone specific version of Flash, (80)) provides access to local capabilities via a helper program which is written in Python and also serves its data via a socket (97). As a general rule it can be assumed that lower-level languages like C have access to low-level device capabilities and can be used to provide these to higher level languages. Android must be seen as the exception to this rule because its API provides access to the hardware from Java (not J2ME), but prevents lower-level native languages from doing so (98).

A2.1.3 Web-Based Applications

Another approach to developing distributed applications is the web-based approach, where all the program logic resides on the server-side. Here the mobile clients would connect to the server via a mobile web-browser or by other means (e.g. SMS, MMS, or Email), change states on the server and deliver the content back to the user. One of the studies presented in the thesis, Love City, took this approach and even incorporated location by utilising a commercial operator-based positioning service. While this is certainly an interesting alternative for certain cases, it only represents an optimisation for very thin clients and is not the solution for all mobile location-based projects. Nevertheless, developing for servers is of course a core activity when building location-based experiences.

A2.2 Developing for Servers

Server-side development is usually grounded in existing practices in the work environment and personal experiences of the developers, and probably most commonly done in PHP, Java, Perl, Ruby, Python or ASP.Net. Especially PHP is a popular choice with beginners because of its apparent ease of use and wide-spread market acceptance which is expressed by its availability at most web-hosting companies.

Scalability and stability are areas where Java, esp. the Enterprise Edition (J2EE) excels. However, it is the author's opinion that the lengthy compile-deploy-run cycle of Java Servlet programming can take too much time while prototyping. To some extent this effect can be mitigated by incorporating more formalism and applying unit tests to the code so that "compile-deploy-run" turns into "compile-test" and contract violating code fails before being deployed to the server. The more formalistic development workflow of Java is geared towards writing more reliable code from the outset, which is of course a Good Thing, as it leads to a more reliable production system. It can however turn into a Bad Thing when the configuration formalism takes over and burns development time which could be otherwise spent on the application prototype itself.

One way to tackle excessive configuration and still maintain a good style is to utilise an existing web application framework, of which there are plenty. Many of these frameworks inherit the Model-view-controller (MVC) pattern, which clearly separates the data from its presentation and thereby eases the development (Reenskaug, 1979). One example of such a MVC framework is the Zend Framework for PHP. More recently, frameworks like Ruby on Rails (short: Rails) concentrated on optimising the programmer's performance by adopting the so-called "convention over configuration" design paradigm in addition to implementing the MVC pattern. This paradigm basically provides sensible default behaviours for reoccurring tasks which would otherwise require the programmer to repeatedly write boilerplate code. Frameworks that support the "convention over configuration" paradigm only require the programmer to explicitly modifiy configuration files if the application's behaviour is desired to be different from the coding convention. Originally introduced by Rails, the "convention over configuration" paradigm was quickly adopted by other frameworks such as Grails, Django or Turbogears. A good overview of more than 100 web frameworks for different programming languages is maintained on Wikipedia (99).

A2.3 Distributed Architecture

Many location-based experiences are designed to be used by several users simultaneously. This single requirement introduces a lot of complexity to the software internals of the experience. Much of that complexity is due to a distributed architecture which involves mobile devices. Applications on stationary devices can usually rely on a fast and low latency Internet connection in order to work and would consequently just pop the user out if that connection was temporarily down. Applications for mobile network devices, on the contrary, have to cater for temporal disconnection. They also usually have to work with higher latency networks like GPRS, although the advent of 3G and the ever increasing availability of public Wi-Fi access points will mitigate this problem in the future. In addition to disconnections and high latency problems, the synchronisation of states across the distributed system and a persistent data-storage remain key issues that have to be addressed.

A2.3.1 General Architecture

Many mobile location-based experiences implement their own custom architecture by tying together available standard components for distributed applications and filling the gaps with custom code. A typical approach would incorporate a client-server architecture that communicates over Hypertext Transfer Protocol (HTTP) using a standard Linux, Apache, MySQL, PHP/Perl/Python (LAMP) installation on the server-side.

Figure A2.1 shows an example of such an architecture which can be broken down into four main components:

1. mobile units, running a purpose built user interface, using a positioning technology and incorporating some kind of network communication
2. a global web server for data storage and retrieval
3. an alternative public interface to the data-storage, e.g. a community website or a spectator client for non-players
4. an admin interface that allows modifications to the database

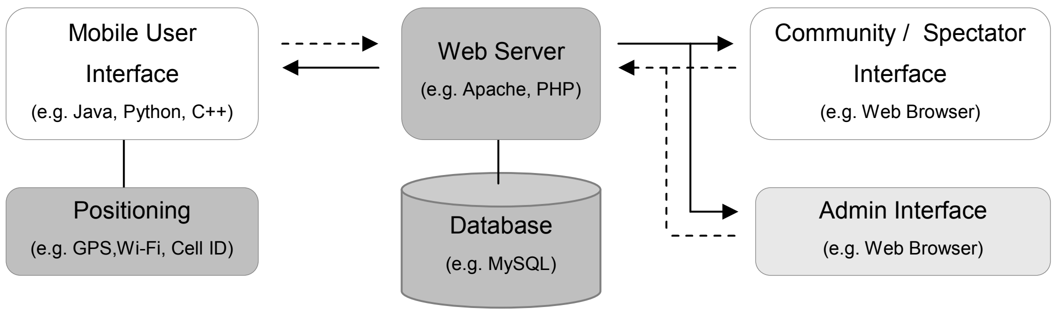

Figure A2.1: Example architecture of a custom mobile location-based game using standard components (Source: (Paelke et al., 2007))

Communication between components is done via HTTP requests (depicted as dashed arrows in the figure) to the web server and its responses (the solid arrows in the opposite direction). The web server hosts dynamic pages (e.g. PHP) and a database for persistent data-storage. Due to the success of the World Wide Web, HTTP is available on almost every network connection. HTTP requests also pass through most firewalls and proxies of corporate, educational or private networks since this is a requirement for web browsers to work. This makes it easy to deploy the public and admin interfaces based on HTTP and makes HTTP an almost ideal choice for many applications. The only major downside of HTTP is that it does not allow messages to be pushed from the server to the client, as each communication has to be initiated by the client.

This general client-server architecture is the most common approach for building connected location-based experiences. Different projects tailor this architecture to their specific needs and might chose to omit components (e.g. there might be no community interface) or implement them differently (e.g. use a different database or server), but the overall architecture remains the same. An example for a project that implemented this architecture with all its components is the Hitchers game (page 54), which used Java/C++ on the mobile devices, cell ID for positioning, GPRS for communication, PHP/MySQL on the server and provided community pages and an admin interface in a web browser.

A2.3.2 Supporting Libraries

While it is perfectly feasible to implement the complete software architecture from scratch using standard components, this would result in much work being spent on coding custom communication and data-storage routines. Albeit this might be an effective approach, as the target can be reached, it would definitely be a very inefficient approach, as it would replicate a lot of work that has been previously done and thus require more development time than necessary. There are numerous software packages that strive to simplify the development of distributed applications by taking the burden of writing custom communication code off the programmer. This section provides a very brief overview of some libraries that have been successfully used to support location-based experiences in the past. This selection was mainly guided by the examples reviewed in chapter 2.

COTERIE (MacIntyre and Feiner, 1996) is a modular toolkit written in Modula-3 which was designed to support rapid prototyping of distributed applications. It also provides a high-level distributed graphics library which allows direct access to the scene graph. It was developed at Columbia University and presented in 1996. COTERIE has been used in the MARS project (page 28).

Elvin (Segall and Arnold, 1997) is a commercial solution that implements a content-based publisher-subscriber pattern in a client-server architecture and provides bindings for several programming languages. The server can push messages to the clients as they occur. A unique feature of Elvin is the content-based message routing which means that the recipients of a message are determined based on the messages' content. Elvin was first presented in 1997 and for example used to support an educational location-based game called Savannah (Benford et al., 2005).

MORGAN (Ohlenburg et al., 2004) is an extensible component-based framework that relies on the CORBA middleware for network communication (100). It provides a producer-consumer approach for passing information around the distributed system and an API in C++. A feature of MORGAN is its integrated render engine which supports multiple scene-graph APIs. MORGAN is a product of Fraunhofer FIT and was presented in 2004. It was used to build Epidemic Menace (page 53).

PDIS, the personal distributed information store, was a Scandinavian project in 2003/2004 which researched into next-generation data synchronization between several devices (Rimey, 2004). PDIS is written in Python and implements a distributed XML database which can be updated at any side and which is automatically replicated throughout all components of the system. PDIS was developed at the Helsinki Institute for Information Technology in close cooperation with leading industry partners, including Nokia, Fathammer and Hewlett-Packard. PDIS is freely available under an open-source license (101).

PART, the Pervasive Applications Runtime, is a light-weight Java library that specifically focuses on event-based communication and data synchronisation for handheld and embedded devices. It has been developed as part of the IPerG research project (2004-2008), where it was used to support several game prototypes, including a technology-supported Live Role Playing Game called Momentum (Hansson et al., 2007). PART is freely available under an open-source license (102). PIMP, the Pervasive Interactive Mobile Platform, is another Java library that has been developed under the IPerG umbrella. It provides object-oriented interfaces for ubiquitous technology, such as GPS or RFID, and can be run on small devices as well as desktops and servers. PIMP uses PART for its internal communication and is also available under an open source license (103).

EQUIP (Greenhalgh et al., 2001) is a framework for developing distributed interactive systems. It is a cross-platform, cross-language middleware for data-sharing between networked clients that employs a producer-consumer approach. Consumers can use pattern-based techniques to register their information interests with the data-space and stay up-to-date with any changes through server pushed messages. A new version called EQUIP2 was introduced in 2007 which focuses on better support for mobile experiences (Greenhalgh et al., 2007b). EQUIP and EQUIP2 are predominantly Java-based. A C++ version of Equip (version 1) exists but proved too difficult to maintain over a longer period of time and has thus fallen behind in features. EQUIP2 approached this cross-language maintenance problem by concentrating on a single core development version in Java and deriving support for other programming languages, including C++, PHP or Python, from that code base. EQUIP2 has been used in Love City and Rider Spoke, the two studies presented in this thesis. EQUIP2 is freely available under an open-source license (104).

Appendix 3 – Metrics for Pervasive Technology

The following tables are based on practical experiences and are not necessarily exhaustive. They can give an idea of interesting metrics that can be used for producing location-based experiences on current consumer mobile hardware.

Metric	Description	How to obtain	Frequency	Units
Latitude, Longitude, Elevation	Position of the GPS receiver as calculated from GPS signals	Query internal or external GPS receiver. Device dependent API or the NMEA protocol.	~ 1 Hz	Degrees as Float (Latitude, Longitude) Metres as Float (Elevation)
Speed, Course	Calculated speed and course of movement (difference between current and previous measurement	"	"	Km/h as Float (Speed) Degrees as Float (Course)
Time	Precise current time as received from the GPS satellites	"	"	Seconds since 01.01.1970 UTC as Integer or Float, so called "Unix time"
Time of Fix	Actual time of the last valid position update	"	"	String (hhmmss)
Date	Date	"	"	String (ddmmyy)

Table A3.1: GPS Metrics

Metric	Description	How to obtain	Frequency	Units
MCC	Mobile Country Code	Device dependent API on some phones, e.g. S60, Android	0.1 Hz Note: empirically chosen	Integer
MNC	Mobile Network Code	"	"	Integer
LAC	Location Area Code	"	"	Integer
CI	Cell Identity	"	"	Integer
Signal Strength	Reception quality	" (on few phones)	"	dBm as Integer or Float

Table A3.2: GSM Cell ID Metrics (0..n)

Metric	Description	How to obtain	Frequency	Units
BSSID	MAC Address of Access Point	Device dependent API	~ 1 Hz	String
SSID	Name of Access Point (might be interesting but it is not unique, e.g. many "BT Home Hubs")	"	"	String
Signal Strength	Reception quality Note: negative value, the closer to 0 the better	"	"	dBm as Integer or Float
Security Mode	Name of encryption scheme, e.g. WPA, WEP, WPA2, none	"	"	String
Connection Mode	Infrastructure or ad-hoc	"	"	String
Channel	1 of 14 pre-defined channels in the 2.4 GHz band, range: 1-14	"	"	Integer

Table A3.3: Wi-Fi Metrics (0..n)

Metric	Description	How to obtain	Frequency	Units
GPS Time	Precise current UTC time as received from the GPS satellites. Note: does not include time-zone offsets or daylight savings	Query internal or external GPS receiver. Device dependent API or the NMEA protocol.	~ 1 Hz Note: Although only updated about once per second this time source can be used to a higher precision with the help of internal timers.	Seconds since 01.01.1970 UTC as Integer or Float, so called "Unix time"
Device Time	Local Time from the mobile device. Note: Error prone as often entered by humans.	Device dependent API	Device dependent	Device dependent, often seconds since 01.01.1970 as Integer or Float, so called "Unix time"

Table A3.4: Time Metrics

Appendix 4 – Love City Details

After user testing the graphical smartphone prototype (page 173), Love City went into the final development phase where a new SMS-based user interface was created. This appendix provides an account of the distributed system architecture that drove the second prototype and the final version of the game.

Refactoring the code from a fat client architecture to a thin client architecture was a major piece of work. This major change in the application design has been kept manageable by utilising an architecture that addresses the diversity of mobile phones in its design (page 282). This allowed re-implementing the user-interface over SMS without having to rewrite any of the other bits of existing server code that were not affected by that change, i.e. the game-engine.

A4.1 Distributed System Architecture for 2 Types of Users

The final version of Love City was designed to be played via SMS using any off-the-shelf mobile phone. Sending messages and locating players was done with the help of an external service aggregator which provided unified gateways to the different network operators.

A4.1.1 Central Web-Application

The distributed Love City application, as depicted in figure A4.1, provided interfaces for two different user profiles: players and administrators. Support for these two profiles was provided via a web application running on a central server. The web application was implemented in Java and used the Spring framework for configuration and dependency injection. The application ran in an Apache Tomcat Servlet container on a Linux server. Persistence of game data was gained through an EQUIP2 object-oriented "data-space" which was backed by a MySQL relational database (see "Game DB") via the Hibernate Object Relational Mapping (ORM) system. An EQUIP2 data-space (c.f. page 332) provided object-oriented database access, publish-subscribe event system functionality, and communication across platforms – in this case Java, JSP and Python for S60[34].

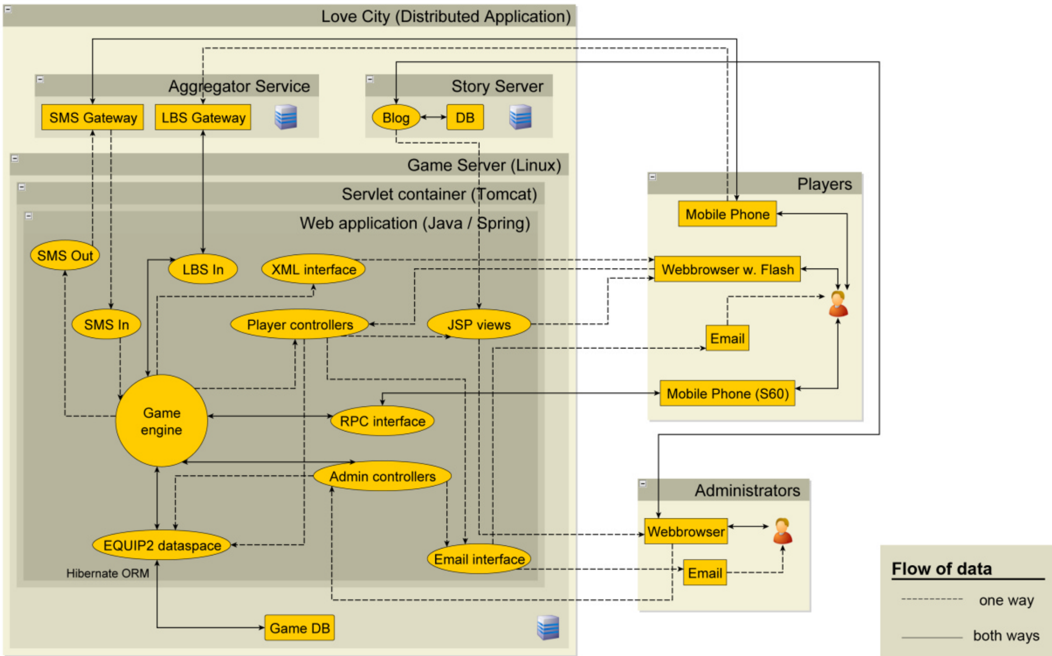

Figure A4.1: Love City architecture overview with main components and channels of communication (source: (Oppermann et al., 2007))

The Love City web application was the central point of service for all components of the system and stored all data. All communication flew through the game-engine (depicted as a circle in the figure) via a number of interfaces such as in- and out-going SMS, a location based service for obtaining phones positions (“LBS In”), a remote procedure call (RPC) interface for the smart-phone version of the game (as described on page 173), a raw XML data-interface for the interactive Flash map, Email, and web-pages generated by JSP views in combination with their attached Java/Spring controllers.

This system architecture is now going to be explained for both user profiles, starting with the administrators (see figure A4.1, right).

[34] Note that the Python for S60 mobile phone prototype has not been used in the final outing of the game; but would have worked as it was feature complete and fully integrated into the system.

A4.1.2 Admin Interfaces

The password-protected operator website provided the artists with simple statistical views of the game as well as pages to operate the game. The game operator could:

- Moderate all incoming player messages (operate as a mechanical Turk[35])
- Ban players that violated the code of conduct
- Send messages to individual players, all players at a location, and all players in the game

Statistical views of player and game data were prepared to allow game operators to keep an eye on information such as the current number of players registered for the game, the number of SMS sent and received, and messages waiting for moderation. Furthermore, information pages about each player could be accessed via a list of all players currently registered for the game, containing the players' real names, agent names and the number of messages that they have received. By clicking on a player in this list the artist could view information about the player's personal details, agent details, bonds made and offspring belonging to this player.

In addition to this core functionality, a blog system had been installed on a separate server and linked into the website via an RSS feed. The blog was used by the artists to write content for the website's newsticker. Finally, an email notification service was implemented which forwarded incoming messages to a number of configurable accounts. One of the artists used this in combination with an email-to-SMS service to stay informed about activities in the game.

A4.1.3 Player Interfaces

Players communicated with the game via SMS messages. In addition, an interactive website provided a visual overview of the game (as described on page 145) and players could also be contacted by email (which was mainly used for registration purposes). Communicating with players via SMS and especially

[35] The Mechanical Turk was an apparent chess-playing machine of the 18th century which later proved to be operated by a human hidden inside the automaton

tracking their locations using their mobile phone operator networks required the service of a specialised third party aggregator company. Our service provider enabled us to send and receive text messages to and from mobile phones on any GSM network using one central “SMS Gateway” web service account with a designated 5 digit short code telephone number.

All SMS messages were routed through the game-engine via the SMS gateway. This was for three reasons:

1. To protect the players' privacy by keeping their phone numbers secret from each others
2. To allow the game-engine to automate most of the control flow to allow the use of a content filter for the messages
3. The 5 digit short code telephone number was the same on all participating networks and thus easy to communicate regardless of the network each individual player was using

Every message sent to the short code number was forwarded to the game-engine where any messages from unknown mobile phones were directly replied with a short help text which explained how to sign up for the game. Every message that did not start with a valid command automatically triggered the delivery of a short help message. Messages had to begin with a valid command from the list of commands as summarised in table A4.1.

Command	Function
WHO	Find out where you are in Love City and list players in your area, e.g. “You are in the Infirmary of the Beloved with: BEAN [D], MONKEY [L], BLOB [N]” (D for Derby, L for Leicester, N for Nottingham)
WHERE	find out more about the current location
SEND	+ players name + message to send a message of love to another player, e.g. “SEND monkey you are my sunshine"
ACCEPT	to accept a message received from another player
REJECT	to reject a message received from another player

OFFSPRING	to find your offspring and create bonds through them
TELL	to give your offspring a message to send, e.g. “TELL offspring1 you make me happy when skies are grey"
TICKLE	to remind your offspring to send a message (this will trigger your offspring to choose a player in their vicinity and send them a message)
SCORE	to receive an update on your score
BLOCK	to block another player from sending messages to you (this will be confirmed by us with email)
STOP	to leave Love City and stop all game play (this will remove you from the game immediately)

Table A4.1: Love City SMS commands

In order to complete the game bonds, players had to accept or reject the messages that they were sent. As this was a core mechanism of the game, players were reminded about it in every delivered message from another player (figure A4.2) so that they would immediately know what to do next. If the game-engine had to deliver several messages to a player they were queued up and delivered one by one only after the user responded to the previous message or it timed out.

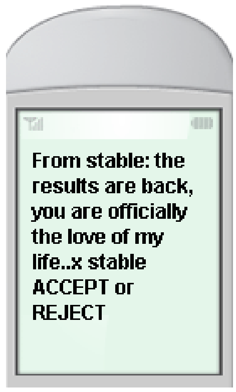

Figure A4.2: Incoming SMS (Source: (Greenhalgh et al., 2007a))

A4.2 Final Location Mechanisms

Locating players in the final version has been achieved through operator-based positioning, channelled through a commercial service aggregator, and triggered every time a player explicitly interacted with the game by sending a SMS.

A4.2.1 Operator Based Positioning

This positioning method needs the individual’s permission to track their phone which we obtained during the registration process. Players had to opt-in to tracking their phone by sending the JOIN command and they could opt-out at any time by sending STOP. The position readings obtained from the service were delivered in World Geodetic System 1984 (WGS84) latitude and longitude coordinates, mimicking the behaviour of a GPS device. The final game regions were exported from my editor as vector-meshes (as seen in figures 4.14 - 4.16, page 164 ff) in WGS84 coordinates. I had to build and export one content mesh per network operator (containing all three cities) and used these meshes to determine the players' location within the game by intersecting them with the players' reported positions.

A4.2.2 Random Positioning

Due to problems out of our control – the person responsible for signing new positioning contracts left the operator company – players could only be tracked on 3 out of the 4 UK GSM networks. Instead of limiting the game to be only played on the remaining 3 networks, the artists decided to randomly generate positions for those players that could not be located otherwise.

Appendix 5 – Rider Spoke Details

This appendix provides a detailed description of the distributed system architecture behind Rider Spoke and how it is used by four types of users.

This appendix also provides a more detailed description of the location mechanism than the one contained in chapter 5. Finally, it presents some ideas for improving the algorithm based on the author's experience from working with it for a long time.

A5.1 Distributed System Architecture for 4 Types of Users

Rider Spoke is designed to be played using a mobile device which is mounted to the handlebar of the player's bike. At the end of each day of play, the user generated content and the log-files of the more than 30 mobile devices get synced onto a central server by the staging administrator. At this stage the locally evolved fingerprint graph from each device gets merged into the common game graph, which can be visualised and inspected by the graph administrator. The artists can moderate and rank the user generated content via a ranker interface by using a web browser. At the start of the next day, a copy of the current game state (game graph and all content above a quality threshold) gets copied back onto the mobile devices and the game can start again.

The distributed Rider Spoke application as depicted in figure A5.1 provides interfaces for four different types of users: players, staging admin, content ranking artists and graph admin. Support for these four profiles is provided via a web application running on a central game server. The web application is implemented in Java and uses the Spring Framework for configuration and dependency injection. The application runs in an Apache Tomcat Servlet container on a Linux server (see the box labelled "Game Server (Linux)" to the middle-right of figure A5.1). Persistence of game data is gained through an object-oriented "EQUIP2 dataspace" (c.f. page 332) which is backed by a MySQL relational database (see "Game DB") via the Hibernate Object Relational Mapping (ORM) system.

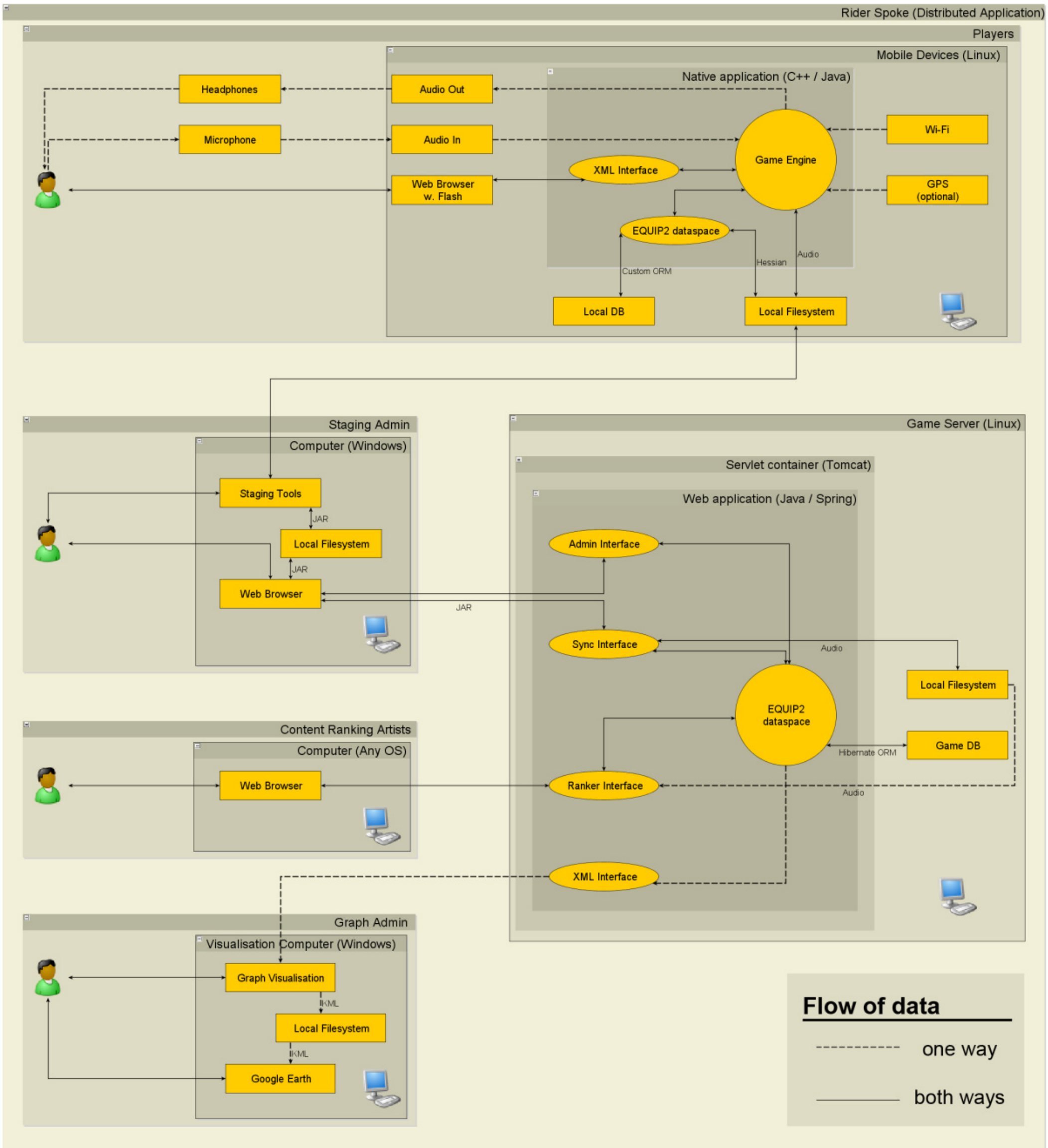

Figure A5.1: Rider Spoke overview with main components and channels of communication

Rider Spoke's four user profiles can be divided into one player profile and three administrative profiles. The Rider Spoke web application is the direct point of service for all three admin profiles and naturally also holds all user generated content from the players after it has been uploaded to the game server by the staging admin. In order to keep the description of the overall architecture comprehensive it is going to be explained per user profile, starting with the players (see figure A5.1, top-left) and then traversing vertically down the figure through the other user profiles.

A5.1.1 Players

Players are equipped with Nokia N800 Linux Internet tablets mounted to the handlebars of their bikes. They can interact with the game user interface (which is implemented in Flash and hosted in a full-screen browser window) via the device's touchscreen. They are also equipped with a combined headphone and microphone which is connected to the N800 and by which they can listen to the game's audio content and record their own answers to the game's questions. The audio playback and recording is directly managed by the central game engine and audio-files get stored in the local file system. The game engine is written in C++ and Java[36] and interfaces to the Flash user interface application via an XML socket. It also performs the Wi-Fi scanning using the device's built-in wireless adaptor and applies the Wi-Fi fingerprinting algorithm to the scan-results. There is an optional support for Bluetooth GPS devices, which can be used to apply position readings to fingerprints and access points. However, most players are sent out without GPS devices. All data from the game engine is managed in a local Equip2 dataspace. Crucial parts of the dataspace, i.e. those parts which are needed to restart the mobile application without loosing important information such as the graph, are persisted into a local Berkeley DB using a custom object-relational mapping method. Apart from this persistence mechanism, all dataspace state changes, i.e. all data, gets also written to Hessian-encoded (105) log-files in the local file system. These files and the recorded audio-files are collected by the stage admin at the end of each day, when all the mobile devices are back and their data needs to be merged to the game server.

A5.1.2 Staging Administrator

The staging administrator (see figure A5.1, upper-middle) is responsible for putting content on the devices at the start of each day and getting content off the devices at the end of each day. Data from and to the players' mobile devices always needs to go via the machine of the staging admin. To accomplish this task, the stage admin utilises the "Staging Tools" which semi-automate the process of accessing the local file systems of the mobile devices via SSH connections.

---

[36] Equip2 related code is written in Java, compiled, translated into C++ source-code (via the Equip2 Javatrans program), then compiled and linked with the rest of the C++ code.

Copying Data from the Mobile Devices to the Game Server

Relevant audio- and log-files get copied to the stage administrator's machine and audio-files are automatically transcoded from PCM format into MP3. That data is packed into one Java Archive (JAR) per device, resulting in as many JAR-files as there were devices in use during the day. These JAR-files are then either manually uploaded to the "Sync Interface" on the game server by using a standard "Web Browser" or batch uploaded using some custom Windows scripts.

Copying Data from the Game Server to the Mobile Devices

When the content has been approved by the content ranking artists (who are described in the following section), it can be downloaded from the Game Server as a single JAR-file which contains all relevant audio files and a Hessian-encoded representation of the new dataspace. Using the "Staging Tools", this JAR-file is unpacked and its contents are distributed to all mobile devices. Finally, a local program is run on each N800, which updates the persistent dataspace with the data contained in the Hessian file.

The staging administrator also has direct access to the game server via an online "Admin Interface" which can be used to investigate the internal server status, manipulate database entries, backup the database and manage media files.

A5.1.3 Content Ranking Artists

A content ranking mechanism has been implemented to deal with two problems that arise from the way content is generated and assigned to locations in Rider Spoke. The first problem is that content generated by users is of varying quality and might range from very good to downright unacceptable. The second problem is that in Rider Spoke only a single piece of content is allowed at each location. Over the course of each day multiple players might have recorded their answers at a location which was marked as "free" at the last update of the mobile devices. This results in the conflict which needs to be resolved and this can be done by ranking the content. For locations with multiple answers, only the best answer will go back onto the devices. Content is ranked on a scale of 1 to 5, where 1 means bad and 5 means excellent.

+date-	+fingerprint-	+player-	+length-	+sightings-	filename	+rank-	set rank
Wed Oct 10 18:37:57 BST 2007	FP1001_ move	Martin	70704	0	listen	3	1 2 3 4 5
Thu Oct 11 21:36:13 BST 2007	FP1002_ move	hoifaiwu	44640	0	listen	unranked	1 2 3 4 5
Thu Oct 11 17:58:32 BST 2007	FP1005_ move	Hannah	222336	0	listen	unranked	1 2 3 4 5
Thu Oct 11 17:58:32 BST 2007	FP1005_ move	Hannah	222336	0	listen	unranked	1 2 3 4 5
Wed Oct 10 17:44:48 BST 2007	FP1007_ move	Lucy	23040	0	listen	5	1 2 3 4 5
Thu Oct 11 19:24:11 BST 2007	FP1009_ move	Anne	264672	0	listen	unranked	1 2 3 4 5
Thu Oct 11 19:24:11 BST 2007	FP1009_ move	Anne	264672	0	listen	unranked	1 2 3 4 5
Thu Oct 11 19:05:17 BST 2007	FP1015_ move	Anne	114912	0	listen	unranked	1 2 3 4 5
Thu Oct 11 18:51:02 BST 2007	FP1015_ move	Nat	84960	0	listen	unranked	1 2 3 4 5
Thu Oct 11 19:05:17 BST 2007	FP1015_ move	Anne	114912	0	listen	unranked	1 2 3 4 5
Wed Oct 10 18:34:18 BST 2007	FP1017_ move	Joe	64080	2	listen	3	1 2 3 4 5
Wed Oct 10 18:39:53 BST 2007	FP1020_ move	Joe	44064	0	listen	3	1 2 3 4 5
Thu Oct 11 19:19:47 BST 2007	FP1021_ move	Dan Lamont	38304	0	listen	unranked	1 2 3 4 5
Wed Oct 10 18:52:15 BST 2007	FP1026_ move	Joe	30672	1	listen	4	1 2 3 4 5
Wed Oct 10 19:03:21 BST 2007	FP1032_ move	Joe	62496	0	listen	3	1 2 3 4 5
Wed Oct 10 19:17:25 BST 2007	FP1039_ move	Joe	22752	0	listen	4	1 2 3 4 5
Thu Oct 11 17:23:26 BST 2007	FP1053_ move	Emma	11664	0	listen	unranked	1 2 3 4 5
Thu Oct 11 19:47:35 BST 2007	FP1055_ move	julia	167904	0	listen	unranked	1 2 3 4 5
Thu Oct 11 19:58:57 BST 2007	FP1059_ move	julia	103680	0	listen	unranked	1 2 3 4 5
Thu Oct 11 21:01:25 BST 2007	FP1060_ move	julia	62496	0	listen	unranked	1 2 3 4 5
Thu Oct 11 21:15:39 BST 2007	FP1061_ move	Pete	113040	0	listen	unranked	1 2 3 4 5
Thu Oct 11 19:34:16 BST 2007	FP1061_ move	lucianne	13968	0	listen	unranked	1 2 3 4 5
Thu Oct 11 20:12:58 BST 2007	FP1062_ move	lucianne	54432	0	listen	unranked	1 2 3 4 5
Thu Oct 11 18:01:28 BST 2007	FP1062_ move	Lucy	77328	0	listen	unranked	1 2 3 4 5
Thu Oct 11 18:01:28 BST 2007	FP1062_ move	Lucy	77328	0	listen	unranked	1 2 3 4 5
Thu Oct 11 18:25:15 BST 2007	FP1062_ move	Nat	93168	0	listen	unranked	1 2 3 4 5

Figure A5.2: Online Content Ranking Interface (Source: (Adams et al., 2007))

To rank the content the "Content Ranking Artists" (see figure A5.1, lower-middle) utilise the online content ranking interface, a screenshot of which is included in figure A5.2. It shows a spreadsheet view of all user generated content that has been uploaded to the game server. The rows of this view can be sorted on any of the columns that have a "+" and "-" sign in the column header (e.g. "date" , "player" or "length"). White rows denote content which has already been ranked and will go into the next dataset for the mobile devices. Blue rows denote content that has not been ranked, yet. Red rows (not seen in the figure) would denote content that has been ranked and will not go into the dataset because their ranking value is below the quality threshold. To accomplish their task the content ranking artists would pick any blue row and listen to the audio sample recorded by the user. They would then set the rank of this piece of content according to their judgement and turn their attention towards the next blue row. The task is completed when there are no more blue rows on the interface. This process can be parallelised by having multiple artists using the online ranking interface from multiple computers.

A5.1.4 Graph Administrator

The graph administrator (see figure A5.1, lower-left) is responsible for verifying the growth of the game graph and visualising other graph related aspects. He does so by utilising the "Graph Visualisation" which gets a copy of the latest game graph via an "XML Interface" from the game server. The graph visualisation is an interactive application with a zoomable user interface which enables its user to visually inspect all properties of the graph. Certain aspects of the game graph, such as the location of content, the number of access point in fingerprints or the number of times a fingerprint has been seen by players, can be emphasised through colours and size of the nodes. If GPS data has been collected, the graph visualisation can fix, or pin those fingerprints that have a known position to their geographical position and exploit the graph structure to approximate positions of the remaining fingerprints in between those anchor points. Finally, the graph can be exported to a KML file in the local file system which gets automatically loaded into "Google Earth" where the graph can be inspected in relation to the physical environment, making full use of the map data that is available through Google Earth as well as its advanced node placement algorithm, annotation features and support for layers.

A5.2 Rider Spoke's Fingerprint Location Mechanism

The Rider Spoke Wi-Fi fingerprint location mechanism, which is not the work of this author, is loosely based on prior work in the field of ubiquitous computing (Bahl and Padmanabhan, 2000, LaMarca et al., 2005, Cheng et al., 2005, Rekimoto et al., 2007). While it shares the same principle of piggy-backing on an infrastructure that is "already out there", it does so in a slightly different way. Whereas the referenced works ultimately produce absolute position coordinates, the Rider Spoke location mechanism works with proximity, i.e. relative coordinates only. It thus only expresses whether one is "here" or "there", and organises that information in a graph structure, so that routes between different locations can be calculated.

The following sections provide a detailed account of the inner workings of the final Rider Spoke location mechanism as it is used in the touring experience. It also presents the author's reflections on the algorithm after having gained an own appreciation of its characteristics from having worked on making the algorithm more accessible to experience designers.

A5.2.1 Algorithm Details

Rider Spoke generates locations through Wi-Fi fingerprinting and organises them in a graph structure. This graph consists of locations (nodes) and the connections between them (edges). Both nodes and edges in Rider Spoke are generated on the mobile game devices using a Wi-Fi fingerprinting algorithm which takes three parameters and works as follows:

1. get a list of currently visible Wi-Fi access points through scanning
2. take a subset of that list based on a signal quality threshold value (e.g. -80 dBm, **parameter 1: signal quality threshold**)
3. only maintain top N access points from that filtered list (e.g. up to 5 access points, **parameter 2: maximum access points**)
4. Fuzzy match the resulting fingerprint against already known fingerprints with adjustable "fuzziness" or overlap (e.g. 60%, **parameter 3: overlap**), then either:
    a. Snap player to a previously known location in the graph
    b. Or create a new location in the graph
5. Eventually create an edge between the players previous and current locations (if not already known)

To illustrate this location mechanism, consider the following example. A mobile device is turned on and thus has no own history of locations. But it has access to a local database of historic fingerprints which have been previously recorded by others. Initially, the mobile device performs a Wi-Fi scan which returns a list of discovered access points and their measured signal strengths. Let this list contain 10 elements (1-10) of which the first nine (1-9) have signals stronger than -80 dBm and the other one (10) has a signal strength below the threshold value. Step 2 of the algorithm filters the list based on the threshold value and only keeps elements that have a signal strength value above the threshold value. It thus only keeps elements 1-9. Step 3 of the algorithm is set to only accept lists of a certain maximum length, or alternatively remove redundant elements from their tail. Let that parameter be set to 5, so that our list of nine elements in now shortened to only contain the strongest five elements (1-5). Step 4 now takes this filtered list of five elements and searches for candidate locations from the historic database. It finds two locations, A (3-7) and B (1-4) that somehow overlap with the test list. Table A5.1 summarises how the algorithm performs the fuzzy matching step and arrives at a conclusion about the current location.

Check for location A (3-7), 5 elements		Check for location B (1-4), 4 elements	
Intersection	Percent	Intersection	Percent
(3,4,5)	3 out of 5 = 60%	(1,2,3,4)	4 out of 4 (!) = 100 %

Table A5.1: Fuzzy matching test list (1-5) against two known locations

The fuzzy matching in step 4 is performed with a required overlap value of 60%, i.e. at least 60% of a fingerprint from the database must be present in the current test list in order to snap to that location. In our example, location A met that requirement, as it had a 60% overlap, and has thus been taken as a candidate location. But location B yielded a 100% overlap. As 100 is more than 60, location B took precedence over location A and thus location B has been chosen as the current location. Step 5 has been skipped, as the device had just been turned on and had no previous location. If the next run of the location mechanism yielded a different location, say C, a connection between B and C would be noticed and inserted as an edge into the graph.

In addition to continuously performing this algorithm and logging its results, the mobile devices also log all scanning data, i.e. the data that went into the fingerprinter algorithm. This allows for a later analysis and reuse of these logs. All devices work with the same set of parameters (threshold, maximum access points, and overlap) for the complete duration of the event. Based on these parameters, each device individually extends its local copy of the fingerprint graph while it is outside with players. At the end of each day, all data (graph, user generated content, log-files) from every device gets merged onto a central server, resulting in a growing common graph of fingerprints (i.e. locations in the game) and connections between them. This merged graph is universally valid for the current parameter set and gets copied back onto the devices before they are given out on the next day.

It is possible that two different players have discovered the same fingerprint and have left content at that location. In this case the merging server application will flag a conflict for this location which can be resolved by the artists through a web-based orchestration interface where they can decide which answer stays at the given location and could also manually move answers to other locations.

An important limitation of the location mechanism as it has been implemented for Rider Spoke is that the algorithm parameters need to remain static over the course of the experience. This is because the player content is attached to fingerprint locations that evolve over time as described above. Changing the algorithm parameters would void the content to fingerprint mapping, resulting in a loss of content which was unacceptable for the artists. This makes it important to be able to understand and predict the behaviour of the algorithm with particular parameter settings.

It is theoretically possible to generate new content to fingerprint mappings, but this has not been implemented for Rider Spoke for various reasons, one of them being that the artists' conflict resolutions are only valid for a particular set of parameters. This means that parts of their orchestration work would be discarded every time the parameter set changes.

A5.2.2 Important Aspects

The Rider Spoke location mechanism builds on the previously mentioned related work in a number of ways. The most important aspects of the mechanism from developers and designers perspective were:

1. It focuses on location rather than on position, i.e. it doesn't rely on having an exact coordinate reading (like 51.826, 10.763).

2. It recognises the pervasiveness of Wi-Fi APs in built-up city environments and its superior temporal coverage (page 115) and therefore bases its location mechanism on this technology rather than on GPS.

3. It requires only a short initial calibration effort as it continuously improves the underlying data-set while the experience is being run with

hundreds of players, each of them scanning the wireless environment and contributing to a common stock of knowledge about it.

4. It utilises Wi-Fi fingerprints to allow the system to continue to work even when access points disappear from the network environment.

5. It records transitions between locations (fingerprints) and records them in a graph to allow a notion of proximity between locations.

6. It keeps track of how often locations are recovered in the graph and thereby allows identifying popular and reliable locations as well as the less known locations.

7. It visualises the characteristics of the supporting technical infrastructure to the designers to allow them to understand and tweak the system altogether (more on this in chapter 5).

8. It can be deployed to readily available commercial pervasive devices. Consoles such as the Sony Playstation Portable (PSP), Nintendo DS or mobile devices like PDAs or the Nokia N800 all incorporate Wi-Fi, whereas they do not include GPS as a standard or widely used add-on.

9. It can be commercially viable to issue large amounts of position requests at a very cheap price (or, as in this case: for free).

10. It can work under conditions where GPS can not.

A5.2.3 Reflections on the Location Mechanism

The author has mainly dealt with visualising the location mechanism to facilitate the artists understanding of it. This process was necessarily tethered with gaining an own understanding of the algorithm. This section provides a brief account of noteworthy aspects of the algorithm that emerged from feedback of the artists as well as through the authors own reflections.

A5.2.3.1 Stabilising the Location Mechanism

During the Athens setup, the artists discovered the positive effect of capping (page 238 ff) on the granularity and spread of locations. By capping they meant limiting the maximum number of access points that would be considered for forming a fingerprint of a location. This was seen as a way to increase the stability and spread of locations. Capping, in combination with a lowered signal quality threshold, caused a more homogenous creation of fingerprints of a similar size (near the capping limit). This works, as the reduced signal quality threshold initially allows more access points to be kept after scanning – even those whose signals are very weak. From the resulting list, capping then only picks the best access points to be used for fingerprinting. This combination of measures thus effectively first widens the search range and then picks only the strongest access points, which results in a better granularity of locations.

This approach differs from the first outing of Rider Spoke in London. Then, it was seen as desirable to not limit the number of access points that would be considered in the fingerprinter, on the grounds that a fingerprint consisting of 20 access points would be significantly different from one that had, say, 8 access points in it, and should therefore denote a different location.

Upon reflection, it appears that capping the number of access points in a fingerprint is a good idea for two reasons. First, the Wi-Fi infrastructure is continuously fluctuating, and the availability of individual access points changes all the time. For example, the situation in the evening hours (when the game is played) might be very different from the situation during the days (when the setup is usually done), because offices might switch off their wireless routers whereas individuals might turn them on. Second, it appears that the fingerprint algorithm does not balance correctly between fingerprints that consist of largely varying numbers of access points.

Bias for Smaller Fingerprints

While the first reason represents common sense, the second reason needs some explanation. Upon closer inspection of the algorithm, it became evident that the algorithm clearly favours smaller fingerprints over fingerprints that contain a large number of access points. This is because the algorithm calculates the intersection between the list of current scan results and the list of access points of every fingerprint in the data base. This overlap is expressed as a percentile ratio which reaches 100% if the reference list, i.e. a fingerprint from the database, is a complete subset of the current scan results. This implies that the overlap can also reach 100%, even if the list of current scan results is much bigger then the reference list. The algorithm thus introduces bias towards smaller fingerprints. Table A5.2 illustrates this behaviour of the algorithm for an overlap setting of 50%. The rows in the table represent the order in which the scan results appeared over time. Please note that the BSSIDs have been converted to integers to make the table easier to read[37].

BSSIDs	Overlap	FP	Comments
1	**-**	**A**	**No FP known, yet. Create initial FP "A" with (1)**
1,2	100% of A	A	
1,2,3	100% of A	A	
1,2,3,4	100% of A	A	
1,2,3,4,5	100% of A	A	
2,3,4,5	**0% of A**	**B**	**0% < 50%, Create new FP "B" with (2,3,4,5)**
3,4,5	75% of B	B	
4,5	50% of B	B	Compare with two rows below
5	**25% of B**	**C**	**25% < 50%, Create new FP "C" with (5)**
4,5	50% of B 100% of C	C	Compare with two rows above

Table A5.2: The fingerprint algorithm is biased towards smaller fingerprints

In this example, the initial scan result is a single BSSID (1). At this point the database of known fingerprints is empty and thus the initial fingerprint A is created, which consists of a list of one access point (1). The four consecutive scan results generate increasingly long lists that all contain the initial BSSID (1), and

[37] They would normally be in the form of "00:50:56:C0:00:01, 00:1F:00:C2:22:23"

are thus regarded as perfect matches, even though they are increasingly different. The next list (2,3,4,5), although very similar to the previous row, is a 0% match to A, as it does not contain (1) anymore. This forces the creation of a second fingerprint B. As the number of access points in the test list is gradually reduced, the algorithm continues to match fingerprint B, but with a decreasing overlap value. This works as intended. When the list is reduced to a single BSSID (5), a third fingerprint C is generated. This leads to an interesting observation in the last row of the table. The fingerprint algorithm now returns a different fingerprint for the same input data (4,5), as it did before the creation of the fingerprint C. This is because C provides a 100% match to the list (4,5), whereas B provided only a 50% match. This example illustrates the dominance of fingerprints that consist of few access points.

Remove Bias

There are several ways to tackle the bias problem. One way to remove the bias could be to modify the algorithm so that a reported 100% match would literally mean that the tested lists are completely equal. This could be achieved by comparing the length of the two lists of access points that are tested in the overlap calculation and taking the longer one as the divisor when calculating the overlap ratio. This appears to be a promising approach, but it has to be noted that its effect on the overall granularity of locations was not, yet, verified.

Limit Number of Access Points in a Fingerprint

A more obvious and tested approach is to limit the number of access points in a fingerprint. Capping, as discovered by the artists, does exactly that: it limits the maximum number of access points in a fingerprint in order to produce a more evenly distributed fingerprint landscape.

In addition to limiting the maximum number of access points in a fingerprint, introducing a minimum number of access points per fingerprint could be a second measure to achieve a more balanced landscape of fingerprints. A similar discovery was made early on after a Wi-Fi test during the exploratory phase (page 195), where the artists concluded that trigger zones should consist of at least 2, maybe

3, access points in order to trigger locations more reliably upon discovery of any one of the contained access points. Reflecting on the author's own practical experience of working with the Rider Spoke fingerprint algorithm it is proposed to only allow generating a new fingerprint if it would contain at least a certain number of access points. This is thought to break the dominance of smaller fingerprints in favour of a more balanced distribution of fingerprints. More specifically, it should provide some room for fuzzy matching, as 100% matches to smaller fingerprints will occur less often, thereby increasing the influence of the overlap parameter that controls fuzzy matching.

Blacklist Access Points

Another way to achieve a more balanced spread of locations with the current algorithm would be to introduce a black list of access points. This could be useful as certain access points have a far wider range than others, e.g.:

- Higher power access points, repeaters, etc.
- Access points mounted at an elevated position
- Access points mounted in a narrow alleyway or tunnel (reflection)

With the current algorithm, fingerprints that contain those access points would be dominant over others. It would thus be interesting to disregard such access points when building fingerprints. Building the black list could be unintuitive and probably needs some tool support. This demands for geo-referencing and could then be done in two ways:

1. Visualisation of coverage area and manual selection of those access points that appear to cover too large an area (possibly mouse-driven)

2. Calculation of coverage area and automatic selection of those access points that cover an area larger than a certain threshold value (possibly with human verification)

MAC addresses matching 00:00:5E:00:01:XX, with XX being any number between 00 and FF hex, should be blacklisted by default, as this template address is reserved for virtual routers by RFC 3768 of the Internet Engineering Task

Force. Other addresses should probably be added as well, including the elusive 00:00:00:00:00:00, which was frequently observed in the Rider Spoke logs.

Filter Access Points

In addition, or as an alternative to blacklisting wide range access points it also seems necessary to search and remove virtual access points before generating fingerprints. Larger networks often incorporate a load-balancing approach where several virtual MAC addresses internally resolve to the same physical access point. While this provides benefits to the network itself, it introduces bias to the fingerprint algorithm which might end up generating several fingerprints that rely on the presence of the same physical access point. Virtual access points on a physical device usually share the first 5 bytes of the 6 bytes MAC address with their virtual neighbours. The log-files of Rider Spoke revealed many such virtual access points, e.g. 02:20:A6:9C:46:XX, with XX being a range of at least 6 different byte values. These access points all resolved to the same physical location on King Williams Street in London (51.5115,-0.0873). It is thus proposed to treat related virtual access points as one. As a first step, this might be achieved by treating only the upper 5 bytes of a MAC address as significant. As this measure alone might introduce different kinds of errors, e.g. by collating access points that are physically disparate, this filter should somehow be restricted to a certain area. Applying it in situ and only to the current scan result might give good results. Otherwise, more complex filters that define areas based on a history of scan results should be considered.

A5.2.3.2 Judging the Location Mechanism

It might appear that finding the "best" settings for the Rider Spoke fingerprint algorithm for each staging is a classic optimisation problem. It would thus be possible to find an optimum solution to this problem in an automated fashion. But this would require a ranking formula that quantifies the apparent goodness of any given solution. We do not have such a formula.

Instead, we rely on the human intellect of the person using the tool (page 228) to judge the fitness of a particular setting. This task is supported by rankings on individual aspects of each graph, e.g. number of fingerprints or retrievability of locations. The user can employ these rankings and weight them according to their personal judgement in order to find a good solution. It might be possible to formalise this process and ultimately come up with a ranking formula that could be used for automation. But it is questionable if this is worth the effort for two reasons. First, the algorithm is far from being perfect at its current state and thus any formula ranking the algorithm's performance will have to be changed sooner or later. Second, we actually like to employ the human intellect when considering the performance of the underlying wireless infrastructure as a matter of principle.

A5.2.3.3 Optimising the Algorithm

An immanent possibility for improvement is to optimise the performance of the fingerprint algorithm in its current form. This seems to be necessary, as a frequent cause of player frustration was the apparent slowness of technology, i.e. the interaction delay on the mobile devices (Chamberlain et al., 2008). This slowness has several causes, one of which is the inner loop of the fingerprint algorithm. The current implementation of the algorithm uses string arrays and bubble sort in time critical sections such as the overlap calculation between two fingerprints. This can be improved by using a faster sorting algorithm and, more importantly, by replacing all strings operations with integer operations. This can be achieved through a mapping function that maps strings to integers before the application enters any time-critical section, so that the inner loops can operate on faster data-types.

A5.2.3.4 Adaptive Algorithm

The current algorithm is static, as it takes a fixed set of parameters that does not change over the life-time of a staging. These settings thus need to be a good fit for a mix of areas that players might visit. It was discussed if the algorithm could be made adaptive, so that it would somehow configure itself based on its current context. Although an adaptive algorithm would not suit the content mappings in the current design of Rider Spoke, it would certainly be an interesting route for further research into fingerprinting algorithms that could be deployed on a larger scale. The most straightforward approach for the current algorithm would be to keep the signal strength parameter (RSSI, page 348) variable. This parameter is most significant to the search radius, whereas the other parameters mostly define the granularity of locations in the fingerprint graph. Initially starting with a higher value, say -50 dBm, and then gradually lowering it in case of empty scan results seems like a plausible avenue.

Appendix 6 – Rider Spoke Visual Design Document

The document in this appendix has been written by the author and circulated to the designers and the core developers of Rider Spoke. The purpose was to reach a common understanding of the location mechanism and the required visualisation tasks. The document is based on preceeding workshops, practical experience of the team, and initial visualisations that have been discussed in the group. Based on this document, the designers decided on the core functionality of the infrastructure visualisation and prioritised its different features.

Leif Oppermann, 24.09.2007, with annotations by Nick Tandavanitj from Blast Theory, 27.09.2007 (yellow being important, "I" means Nick).

A6.1 Definition of Terms

A6.1.1 Rider Spoke and Graphs

Rider Spoke is a game that is played on bikes, using a Wi-Fi fingerprinting mechanism to locate players and game-content in a network graph. A graph is a collection of element (nodes) that are connected by edges. Figure A6.1 shows the most basic graph which contains two nodes (depicted in light blue) and one edge (depicted in orange) connecting those two nodes.

The content in Rider Spoke is located at nodes and the transitions between those nodes are logged as edges. Both nodes and edges are generated on the mobile devices using a Wi-Fi fingerprinting mechanism.

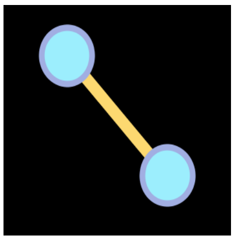

Fig. A6.1: The most basic graph consists of 2 nodes and 1 edge between them

Note: For Rider Spoke, the term location can be used synonymously with a node in the graph.

A6.1.2 Rider Spoke's Fingerprinting Mechanism

In Rider Spoke, the nodes of the graph are created by the Wi-Fi fingerprinting mechanism which works as follows:

1. take the list of currently visible Wi-Fi access points from the device
2. take subset of that list based on a signal quality threshold value (e.g. -80 dBm, decibel meters)
3. only maintain top N from that filtered list (e.g. up to 5 access points)
4. Fuzzy match the resulting fingerprint against known fingerprints with adjustable "fuzzyness" (e.g. 66%), then either:
    a. Snap player to that location in the graph
    b. Or create a new location in the graph

A6.2 Visualising the Network / Game Infrastructure

The visualisation tool seeks to inform the artists and researchers about the state of the game, the location of content and the routes individual players would take through the graph space. A series of views and supporting programming tasks can be derived from this.

View 1: Show the location of all Fingerprints (Pinning)

Functional description:

This view allows having a look at the granularity of fingerprints (how they spread) across the area. This could also guide the artists in thinking about how big an area the fingerprint could cover and how precise (or imprecise) the wording should be when speaking about fingerprinted locations.

Additional Details:

The network graph as generated by the fingerprinting mechanism does not carry any notion of spatial relation to the physical environment. To establish this relationship it is required to collect GPS information using a subset of the mobile devices. This part is already working and it is possible to visualise the locations of fingerprints on their connecting edges on map backgrounds in Google Earth.

Possible highlights:

- number of access points in fingerprint (done)
- pinned or unpinned (done)
- number of global visits to this fingerprint
- last seen

- seen in percentage of rides

Comments:

- Nick: get rid of fingerprint numbers (e.g. 513) in node-labels
- Nick: change colours for unpinned from red to darkgreen or blue (done)
- Nick after noticing that nodes can be pushed in the viewer: would be nice to send those user-pinned fingerprints back to the server
    - Martin follows up: this could be used to move content during conflict resolution
- I think the priority after pinned/unpinned is actually to see locations of all answers on the map
- Second, to show the 'visibility' of all answers on the graph – ie. showing all fingerprints that would give a match as a location of an answer with a given fingerprint matching setting
- Third, to mark those which would have been found/seen by subsequent players with a given fingerprint matching setting
- Finally to filter the answers based on what question they answer

I think View 4 described below is essentially View 1 with the ability to see how changes made to fingerprint settings would change the graph as shown in View 1

View 2: Visualise Route per Player

Functional Description:

This view allows following the routes of individual players through the graph space. Having the graph pinned to the physical environment as required by view 1 facilitates understanding the player's route.

Additional Details:

This information required for building this view (fingerprint seen by player at time) is now available in the FPSighting objects in the dataspace as of webapp build 0.2 (23.09.07).

Possible highlights:

- colorise player's nodes and edges

- playtime (maybe as width of lines or colour or numbering consecutive stops?)
- special locations (maybe through different icons?):
    - player found content (technote: through FPSighting)
        - allow playback of Question audio file (use Question.audioFilename)
    - player answered content (technote: use PASighting.playerAnswerId and use that to look up the player's location from Player_Answer. fingerprintId)
        - allow playback of question / answer (use Player_Answer.filename)
    - none locations
        - graph of #apnums over time per player/ride
        - percentage of time over #apnums

Comments:

- Nick on playtime: first number the steps, then absolute time, then maybe relative playtime in pop-up
- Nick on player found content: show questionText rather than playing audio
- Nick on selection of players/routes: browse list of players as in webapp **(>>>see task 2)**

On reflection I think showing individual players is further down the track

View 3: Visualise per Day

Functional description:

Requirement came up while thinking about the daily de-briefing session at the end of each day. Blast Theory wanted to have a tool to view what has happened during the day.

- Which content had been discovered and which hasn't?
- Where did players go?

Additional Details:

Show the graph as before, but highlight the discovered places and those which haven't been discovered. Also listviews could be helpful to achieve this.

Comments:

- Not very specific so far. We discussed how one could compare inconsistencies between two days but didn't come to a conclusion.

View 4: Show Alternative Graph for Different Settings

Functional Description:

Generate a different graph of fingerprints based on different settings for the fingerprinting mechanism.

Additional Details:

This requires recalculating every fingerprint decision based on the raw access point data. This can be done, since we log the required data and could reconstruct the graph using different settings for the fingerprinting mechanism. But it needs development and considerations to deal with several million records of data which accumulate over the course of the game. (>>> **see task 1**).

Comments:

- Martin and Nick: this is the most important task at the moment.
- I think this means using the raw AP log to visualise the fingerprint graph as in view 1 with different settings, ie: threshold, maximum number of aps included

View 5: Visualise Content per Question

Functional Description:

Display only Answers to a certain Question. The answers could be spread all over the game-area and this view would allow following the spread of each individual question's answers.

Comments:

- None

View 6: Visualise Ground Truth Data

Functional Description:

Show which access points would be visible from a specified location, allowing to investigate the ground-truth layer.

Additional Details:

This has been partially solved with the layer visualisation (see page 205). That approach has been data centric, but the idea is to get that done in a position centric way. Could use the floating canvas to implement that.

Comments:

- Nick: Not so important

A6.3 Tasks to Support the Generation of the Views

Task 1: Building the Alternative Graph

This is a technical decision. Leif is going to write simulation code.

Options to do it:

1. Replicate database to front of house machine.
   Pros: Would allow for disconnection and slow network.
   Cons: Would mean lots of double work.
2. Calculate alternative graphs on server on request by front of house machine.
   Pros: less work, no duplication
   Cons: would not support disconnection, server might choke on huge number of APSightings (30 devices x 3 games a day x 8 days x 10000 APSightings per 1 hour game 1 hour = > 7 Million records) without indexing.

Task 2: Selection Mechanism

A selection mechanism is required to support views 2 and 3 (per player and per day). In its simplest form this ought to be a list selection generated from the database, e.g. name, time (human readable). The list would show the name first, but be sorted on time.

Appendix 7 – Author's Involvements

The evolution of the thesis at hand has been mainly driven by project involvements in the European Integrated Project on Pervasive Games (IPerG) and the European Project on Interactive Storytelling for Creative People (INSCAPE), which allowed the author to become an expert in the field. Practical experience has been gained through involvement in the making of several location-based experiences in collaboration with the artists from Active Ingredient (AI) and Blast Theory (BT). This appendix provides a brief overview of practical experiences that contributed to this thesis.

A7.1 IPerG

The European Integrated Project on Pervasive Games (IPerG) (106) was a ten partner, 42 month project with a total funding of 6 million Euro from the European Commission's IST Programme (107) and a total project cost of 9.84 million Euro that started in September 2004. The IPerG project sought to explore new technologies to support the creation of content for high quality Pervasive Games.

Figure A7.1: IPerG project logo

The author has been involved in several work packages over the course of the project, including the *Tools* work package, and the major final phase showcase *Cultural Console Game* which produced a game called "Rider Spoke" under the artistic lead of Blast Theory (108). Rider Spoke's development process embodied the approach of visualising the wireless network infrastructure to the authors of an experience and is studied in detail in chapter 5 (page 181).

A7.2 INSCAPE

The European Integrated Project on Interactive Storytelling for Creative People (INSCAPE) (109) was a thirteen partner, 48 month project with a total funding of 7.75 million Euro from the European Commission's IST Programme (107) and a total project cost of 12.89 million Euro that started in September 2004. The INSCAPE project sought to produce tools and workflows for creative professionals who want to work with the latest Information Society Technology to produce interactive stories.

Figure A7.2: Inscape project logo

The author has been involved in the project through the development of the location-based mobile phone game "Love City" (110) in collaboration with artists from Active Ingredient. This process is covered in detail in chapter 4 (page 141).

A7.3 Other Involvements

In addition to the above mentioned experiences, the author also benefitted from involvements in other projects, which are briefly summarised below.

A7.3.1 City as Theatre

First experiments for this thesis took place as part of the IPerG *City as Theatre* showcase, which sought to produce artistic games that take place online and on the street. We organised a series of workshops to investigate the possibilities of locating players without using GPS.

As part of this, we conducted an experiment where we visualised scatter plots of mobile phone cell IDs on digital maps, and used these to track the path of two mobile phone users as they were roaming the pre-mapped area (see figure A7.3). Although the final game produced by this showcase (Flintham et al., 2007) did not use any location technologies (it used pure virtual locations, like a multi-user dungeon, but adapted itself to the daily routine of its players based on usage patterns and time), this early experiment can be seen as the starting point of this thesis, as it facilitated the production of the author's first visualisations of wireless networks.

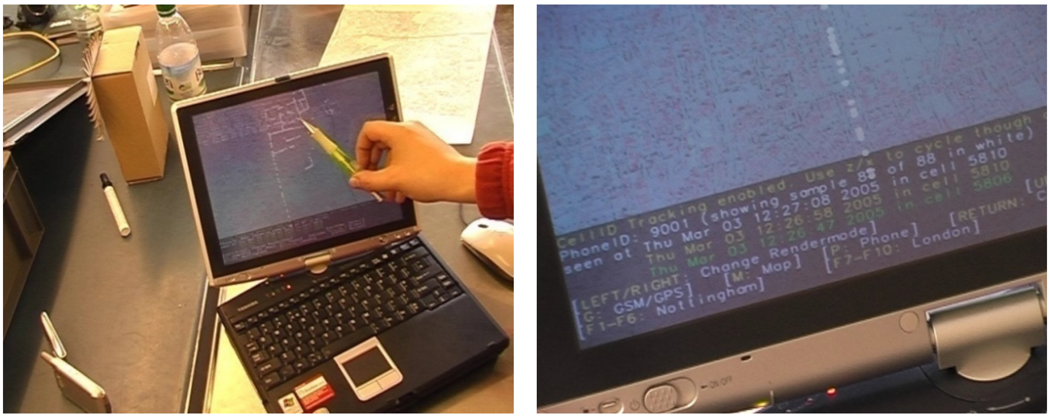

Figure A7.3: Tracking participants based on mobile phone cell IDs

A7.3.2 Crossmedia Gaming

Epidemic Menace was an IPerG showcase game from the *Crossmedia Gaming* showcase which has been first played on the Fraunhofer campus in Sankt Augustin in Germany on August 24th and 25th of 2005. The game utilised a variety of different technologies and gaming devices to allow diverse interaction possibilities for the players (Lindt et al., 2005).

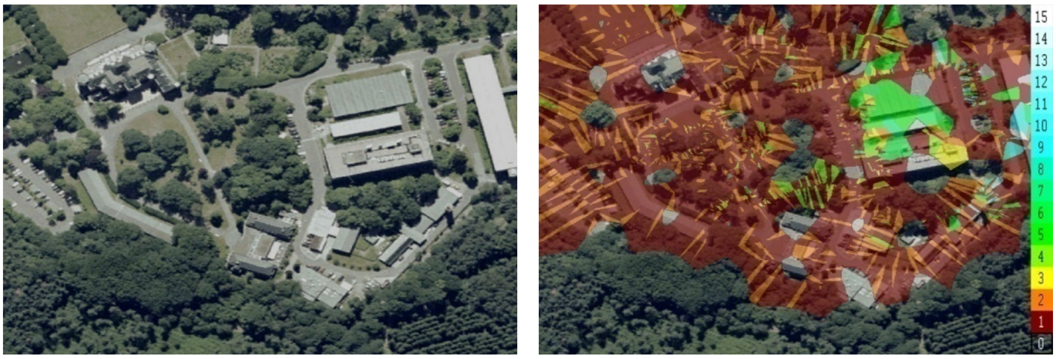

Figure A7.4: Map of the game area (left) and visualisation of Wi-Fi coverage (right)

Epidemic Menace utilised GPS for positioning and Wi-Fi for communication. In order to clarify that the outdoor gaming area would have sufficient Wi-Fi coverage to support the game, the author collected GPS and Wi-Fi infrastructure data on site and used that data to produce a map-based visualisation that showed the approximated number of available Wi-Fi access points at any given point on the gaming area (figure A7.4). The colours in the overlay represent actual number of access points in view and holes denote a lack of sample data at those areas.

A7.3.3 Tycoon

Tycoon was a location-based multiplayer trading-game for smart phones which built on the experiences gained from the early experiments. It incorporated a seamful design which basically means that it focused on making inevitable technical limitations, such as cell coverage, more apparent, rather than trying to hide them (Broll et al., 2006).

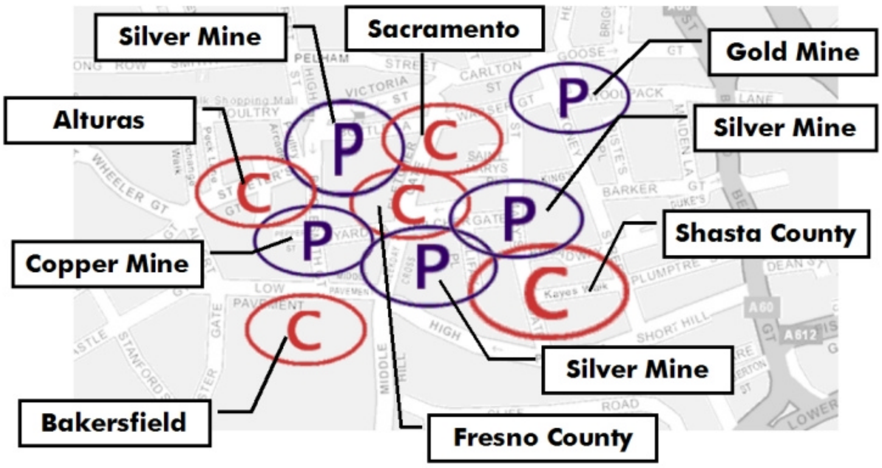

Figure A7.5: Example of a Tycoon gaming-area with its mapping of physical locations to consumers (C) and producers (P) (Source: (Oppermann et al., 2006))

A particularly interesting aspect in relation to this thesis is how the game dealt with dynamic cell coverage. The game uses a wild-west scenario where players have to find natural resources and trade them for goods. According to this idea, the game knows two types of locations (see figure A7.5):

1. Mines (Gold, Silver, Copper) which act as producers and are marked as P
2. Brokers which act as consumers and are marked as C

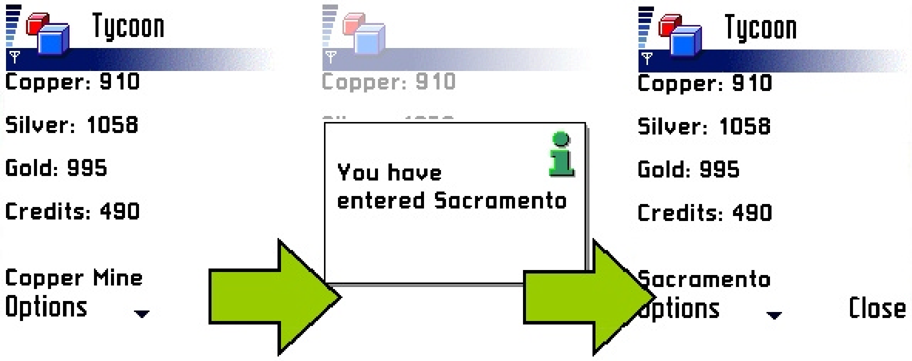

Figure A7.6: Tycoon’s alert-mechanism for location-visualisation
(Source: (Oppermann et al., 2006))

Tycoon used GSM cell IDs to determine in which location the players are. Whenever a player changes from one cell location into another, an alert is triggered and the player gets a notification about the change of locations (see figure A7.6). Due to technical limitations, cell ID positioning is very coarse. This issue was addressed by appropriating the game narrative: natural resources are limited by their very nature and a player would never know exactly where to find them. Therefore it would not be a problem if the location definition was not very exact and would arguably even reflect the way resources are found in reality.

In order to be able to map cell IDs to different locations in the game area, they have to be geo-referenced and made accessible by the game designers. The visualisation program from the *City as Theatre* workshops was extended with authoring capabilities. This first version of an interactive visualisation and authoring tool allowed the game designer to define cell ID based regions that served as triggers for virtual locations in the game. The results of this project have been disseminated as a full paper at the Ubicomp conference 2006 (Oppermann et al., 2006).

A7.3.4 Heartlands (a.k.a. ‘Ere be Dragons)

Heartlands is a GPS and heart-rate based multi-player game where players explore an unknown virtual world by roaming the real world and keeping their heart-rate in a recommended range based on their age (Davis et al., 2007) (111).

Figure A7.7: Scenes from the landscape of Heartlands, showing different characters of terrain resulting from the player's heart-rate performance (left, middle). The opening screen showing a players individual heart-rate band (right)

The game is played in sessions of 20 minutes in which the players are free to walk anywhere they want. Players are equipped with a PDA with GPS and a heart-rate sensor. During their walk, they can monitor their progress – the heart-rate does immediately affect the appearance of the virtual world on their PDA's: green trees and flowers do only appear when the player exercises a bit and stays within the optimum range. If the players exercise too much, their virtual world will turn into a dark forest, if they do too little, their world will turn into a desert and eventually start fading away (see figure A7.7).

Initially a single-player game developed by AI with the help of Hewlett Packard (HP), the author helped to transform Heartlands into a client-server based multiplayer experience. He designed and implemented the communication system, wrote the first versions of a spectator interface that got projected into the hosting venues, and provided tools for setting up the game. Planning and implementing these major extensions in collaboration with the artists has helped the author to immerse himself with their development practice. The work on this project in its current form is finished and it has already been touring for several years, including events in Nottingham (2005), Berlin (2006), Cambridge, San Francisco, Tampere, Sao Paulo (all 2007). Two highlights of the project were:

1.) HP commissioned Active Ingredient and the Mixed Reality Lab to present *Heartlands* at the Game Developers Conference (GDC) 2007 in San Francisco – the major event of the game industry

2.) *Heartlands* won the first prize at the international “Nokia Ubimedia MindTrek Awards 2007” in Tampere from over 140 submissions

A7.3.5 Boxed Pervasive Games

This final phase IPerG showcase researched the idea of turning pervasive games into customer products that could be sold “in a box”. As such, it sought to package a pervasive game development process into a suite of integrated hardware and software components which included GPS and Bluetooth enabled mobile phones, RFID readers, a runtime environment for the phones, example games, and an online authoring and orchestration application called “Gamecreator” (page 100) which was used to build these games.

The author supported this work during its early phase by discussing with its developers and running a mobile authoring user-test with a focus group at a secondary school on Alingsås, Sweden. We tested how in-situ mobile authoring could be integrated into the overall authoring workflow of the box. The pupils received GPS enabled mobile phones running a program which allowed them to roam the schools surrounding area for an hour and annotate places of interest with self-recorded pieces of audio. Back in the school, these annotations have then been put on an interactive map (Google Earth) and made accessible for the pupils for discussion. This early exercise has informed the final workflow of the IPerG Boxed Pervasive Games showcase as well as the IPerG Cultural Console Game showcase with its game “Rider Spoke”. In addition, the author also wrote a production tool which retrieved the contents of the final game server and packaged it for offline use.

Appendix 8 – List of Figures

Figure 2.1: The outdoor Cyberguide application, hardware 27
Figure 2.2: User with full kit, augmented view, browser view 29
Figure 2.3: Lancaster GUIDE distributed architecture 30
Figure 2.4: Lancaster GUIDE hardware, application 31
Figure 2.5: ARQuake player, virtual monster in real environment 33
Figure 2.6: Botfighters artwork with inset web- and mobile interface, Battletech artwork 36
Figure 2.7: A CYSMN runner checking his PDA for nearby online players. An online player in proximity to a runner in the 3D virtual world 37
Figure 2.8: Heatmap of 16 Million geo-referenced wireless access points from Wigle, trail of one team from DefCon 10 WarDriving Contest 40
Figure 2.9: Players found an access point, spectator interface 43
Figure 2.10: Weblog, player piggybacking on a Starbucks hotspot 43
Figure 2.11: Mobile user interface w. map overview, 3D virtual world 45
Figure 2.12: Player in the office, actor in limousine 46
Figure 2.13: Mobile interface: map, radar, collecting items, trading items 47
Figure 2.14: Aerial view of Queen Square, participant with headphones 48
Figure 2.15: Treasure interface, showing the position of players, coins and network availability 49
Figure 2.16: State Garden Show site, position update via infrared 50
Figure 2.17: Borrowing a PDA, roaming the botanic garden 52
Figure 2.18: Listening to audio content, browsing for additional content 52
Figure 2.19: An AR player surrounded by virtual viruses, an audio-player captured a virus, a stationary indoor player guiding the mobile players 54
Figure 2.20: Searching for, picking up and answering a hitcher 55
Figure 2.21: The Hitchers cell ID graph visualisation tool 56

Figure 2.22: Highlighting cells that players have labelled 'Kista', 'SICS' and 'ICE' by being shaded. Is ICE a place within SICS which in turn, is within Kista? ... 57
Figure 2.23: Screenshots of Insectopia ... 59
Figure 2.24: An abandoned nuclear reactor served as the headquarter for the game, The EVP device mounted to a bed and hosted in a room at the site, Player in action ... 61
Figure 2.25: The Steele device with built-in Bluetooth GPS, the Thumin glove with built-in Bluetooth RFID reader in use by a player ... 62
Figure 2.26: Technical overview of the node game in Momentum ... 63
Figure 2.27: Locations of REXplorer, interaction through gestures ... 64
Figure 2.28: Overview map, content selection menu, content presentation ... 66
Figure 2.29: ARToolkit marker, barcode, semacode ... 82
Figure 2.30: Two examples of using Google Maps for defining locations ... 86
Figure 2.31: MARS Authoring Tool with 3D model of Columbia Campus, three defined locations, current user position and two image snippets ... 88
Figure 2.32: Mscape production workflow ... 90
Figure 2.33: Spatial layout of triggering regions ... 91
Figure 2.34: Action editor ... 91
Figure 2.35: Colourmaps from Uncle Roy All Around You and Can You See Me Now? ... 93
Figure 2.36: CAERUS Desktop Authoring Tool, showing "Map Tasks" ... 95
Figure 2.37: Definition of a symbolic region with associated content for the "Walled Garden" and defining its spatial spread on the map ... 96
Figure 2.38: Three screens of additional HTML pages provided for the "Walled garden" ... 97
Figure 2.39: Yamamoto with background floor plan in orthogonal view and with multi-floor model in perspective view ... 98
Figure 2.40: Campus overview with several buildings ... 99
Figure 2.41: Locations in the Zoo of Saarbrücken ... 100
Figure 2.42: Created a rectangular area in Gamecreator ... 102

Figure 2.43: Associated asset to in-game location 103
Figure 2.44: Map-Based spatial content editing in Mercury Platform 104
Figure 2.45: Content editing opened from the spatial administration pane ... 104
Figure 2.46: Mobile authoring tool 105
Figure 2.47: State diagram with two states and two transitions 109
Figure 2.48: GPS and Wi-Fi coverage and accuracy from CYSMN Rotterdam 112
Figure 2.49: Visualisation of predicted GPS availability 113
Figure 3.1: Conceptual model of current location-based authoring tools 120
Figure 3.2: Conceptual model of extended location-based authoring tools with infrastructure layer 121
Figure 3.3: Abstract Location Model 132
Figure 4.1: Defining shared areas in all 3 cities and linking them to a virtual space 143
Figure 4.2: Large Screen in Nottingham Market Square, Registering players in Derby 144
Figure 4.3: Love City website, pin-pointing a player on the map 146
Figure 4.4: Love City website, showing player details 146
Figure 4.5: Photos for Locations “Shopping’s Soft Centre” and “Tender Station” 147
Figure 4.6: Survey equipment, Surveying the streets of Nottingham in a car 153
Figure 4.7: GPS samples from the survey (6 x 6 km) 154
Figure 4.8: Visualisation of cell samples of the closest cell to Nottingham train station on all 4 GSM networks, manually combined view to estimate the spread of the shared area 154
Figure 4.9: Paper map of one of the three cities (Derby) annotated with regions 156
Figure 4.10: Regions of different scale marked on a digital map of Nottingham in Photoshop 157
Figure 4.11: Raw visualisation of samples from 18 cell IDs in Derby, without map 160
Figure 4.12: Map of Derby, overlaid cell centres, highlighted cell 163

Figure 4.13: Data flow between components 164
Figure 4.14: Overview of Love City game area at 40 x 40 km 165
Figure 4.15: Initial result of region selection
and revised version with fewer cells 166
Figure 4.16: Authoring Love City using abstract and raw sample view 168
Figure 4.17: Data Collection tool for Nokia S60, recording an audio tag 170
Figure 4.18: Main menu and location screens of the graphical prototype 173
Figure 5.1: Rider Spoke participant listening to audio content via
headphones, game device showing a question for the player 183
Figure 5.2: Swallow icon denoting a free hiding place, hiding content 184
Figure 5.3: Recording interface, deciding what to do next 184
Figure 5.4: Showing nearby answers of others,
listening to another player's answer 186
Figure 5.5: Rider Spoke reception area at the Barbican, rental bikes 187
Figure 5.6: "Urban canyon" in the old town of Vienna 188
Figure 5.7: Screenshot of Wi-Fi positioning test-application showing 10
content regions, positions of known access points, and
current position (coordinates and small crosshair) 193
Figure 5.8: Locations of navigational content, travelled route 193
Figure 5.9: Two storylines that players could follow 196
Figure 5.10: Test route in Portslade, UK 200
Figure 5.11: N800 mounted to bike with elastic straps and tape 201
Figure 5.12: Percentage of sightings by number of access points
from two runs on two consecutive days 204
Figure 5.13: Scatter Plots of 2 Access Points in Google Earth,
menu to toggle layers 205
Figure 5.14: Rider Spoke Graph evolving,
before and after discovering a new location 209
Figure 5.15: Screenshot of the graph visualisation tool in its initial state 213
Figure 5.16: Fingerprint topology graph showing
merged journeys of 2 and 5 players 218
Figure 5.17: Fingerprint topology graph of 7 test-players and before going
live with 21 pieces of content placed on the graph 219

Figure 5.18: Area around the "Barbican Entrance" in Google Earth, showing streets ... 220
Figure 5.19: Fingerprint overlay in Google Earth ... 220
Figure 5.20: Highlighted content and route of a player ... 221
Figure 5.21: Using size to show frequency of visits to fingerprints, details on demand ... 221
Figure 5.22: Annotated screens of the "New Experiment Wizard", as used to configure data source and parameter permutations for an experiment called "Athens Testplayer Nick" ... 228
Figure 5.23: Report view of an Experiment ... 230
Figure 5.24: Local database unifying data from different sources ... 233
Figure 5.25: Trail of 659 Access Points discovered in Athens ... 236
Figure 5.26: Wave-Pattern that caught the artists' attention ... 239
Figure 5.27: Looking for settings that generated the most fingerprints ... 239
Figure 5.28: Setting 1: geo-referenced graph from the network survey in Brighton (120 FP) ... 243
Figure 5.29: Setting 1: cluster of fingerprints visualised in Guess and Google Earth ... 243
Figure 5.30: Setting 3 - A worst case scenario in Guess and Google Earth ... 246
Figure 5.31: Setting 6 - The graph is still too crowded ... 247
Figure 5.32: Setting 9 - The graph is clean ... 247
Figure 5.33: Setting 11 - Final simulated graph, Google Earth view with node size set to scale according to number of access points in a fingerprint ... 248
Figure 6.1: Snow's 1854 cholera and water pump map of London Soho ... 266
Figure A1.1: A general communication system according to Shannon ... 316
Figure A1.2: Visualisation of an opinion poll for the German Bundestag, Layout of the actual Bundestag ... 318
Figure A1.3: Visualisations of the Anscombe Quartet ... 319
Figure A1.4: Long Term Real House Price Trend, Annual % Change in House Prices ... 320
Figure A1.5: Ebstorf map ... 321
Figure A1.6: T-O map ... 321

Figure A1.7: Physical Map of the World, April 2007 322
Figure A1.8: London Tube Map 324
Figure A2.1: Example architecture of a custom mobile location-based game using standard components 331
Figure A4.1: Love City architecture overview with main components and channels of communication 338
Figure A4.2: Incoming SMS 341
Figure A5.1: Rider Spoke overview with main components and channels of communication 344
Figure A5.2: Online Content Ranking Interface 347
Figure A6.1: The most basic graph consists of 2 nodes and 1 edge between them 360
Figure A7.1: IPerG project logo 366
Figure A7.2: Inscape project logo 367
Figure A7.3: Tracking participants based on mobile phone cell IDs 368
Figure A7.4: Map of the game area and visualisation of Wi-Fi coverage 368
Figure A7.5: Example of a Tycoon gaming-area with its mapping of physical locations to consumers and producers 369
Figure A7.6: Tycoon's alert-mechanism for location-visualisation 370
Figure A7.7: Scenes from the landscape of Heartlands, showing different characters of terrain resulting from the player's heart-rate performance. The opening screen showing a players individual heart-rate band 371

Appendix 9 – List of Tables

Table 2.1: Chosen mobile devices,
positioning- and communication technologies 70
Table 2.2: Key characteristics
of selected positioning technologies and techniques 85
Table 2.3: Place Lab time-coverage study for 3 wireless technologies 116
Table 3.1: Position, Location, Place 128
Table 4.1: First messages received on the phone 148
Table 4.2: Place description,
received after sending the WHERE command 148
Table 4.3: Player generated content 149
Table 4.4: Examples of the daily “Love City News” 150
Table 4.5: Authored content sent to players at specific locations 150
Table 4.6: Summary of suggested functionality for authoring tools 161
Table 4.7: Commands and functions of the Love City authoring tool 167
Table 4.8: Cell ID count of selected regions
for one city (Nottingham) across all GSM networks 169
Table 4.9: Cell ID count of selected regions
for one network (Orange UK) across all participating cities 169
Table 5.1: Example questions 185
Table 5.2: Examples of player recorded answers 186
Table 5.3: Settings and notes
of the first fingerprint test on the mobile client 202
Table 5.4: Rider Spoke specific fields
for nodes in the GUESS graph structure 214
Table 5.5: Settings for the Rider Spoke fingerprint algorithm in London 217
Table 5.6: Parameters accepted
by the “What-If” simulator server component 227
Table 5.7: Columns in the report view 229
Table 5.8: Metrics used for assessing the generated graphs 231
Table 5.9: Pros and Cons of the two data collection setups for Athens 234

Table 5.10: Generation of fingerprints
from the same range of access points (overlap of 60%)............. 238
Table 5.11: Settings for the Rider Spoke fingerprint algorithm in Athens..... 240
Table 5.12: Iteration of settings for Rider Spoke
in Brighton as checked by the artist.. 245
Table 5.13: Final settings for the Rider Spoke
fingerprint algorithm in Brighton 249
Table 6.1: Features of visualisation tools in the two studies 261
Table 6.2: Project resources and focus of work
over the full development process 293
Table A1.1: Anscombe Quartet 319
Table A2.1: Comparison of programming languages
and frameworks for mobile phones 327
Table A3.1: GPS Metrics... 335
Table A3.2: GSM Cell ID Metrics (0..n).. 335
Table A3.3: Wi-Fi Metrics (0..n).. 336
Table A3.4: Time Metrics.. 336
Table A4.1: Love City SMS commands... 341
Table A5.1: Fuzzy matching test list (1-5) against two known locations 350
Table A5.2: The fingerprint algorithm
is biased towards smaller fingerprints 354

Online Resources

1. Game. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Game
2. Experience the Experience! [Online] [Verified: 14 August 2009.] http://www.monochrom.at/experiences/
3. Poème électronique (Le Corbusier). *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Po%C3%A8me_%C3%A9lectronique
4. Lancaster GUIDE Homepage. *Lancaster University.* [Online] [Verified: 14 August 2009.] http://www.guide.lancs.ac.uk/
5. A Short History of Geocaching: May 2000. [Online] [Verified: 14 August 2009.] http://www.guysnamedkim.com/geocache/geocache_history.html
6. The History of Geocaching. [Online] [Verified: 14 August 2009.] http://geocaching.gpsgames.org/history/
7. Quake Homepage. *id Software.* [Online] [Verified: 14 August 2009.] http://www.idsoftware.com/games/quake/quake/
8. Daydream Success Case. *Nokia.* [Online] May 2005. [Verified: 14 August 2009.] http://www.nokia.com/NOKIA_COM_1/Developers/Success_Stories/Case_Studies/Games/pdf_files/successcase_daydream.pdf
9. Distinction for Net Vision / Net Excellence: BotFighters. *Prix Ars Electronica.* [Online] 2002. [Verified: 14 August 2009.] http://web.archive.org/web/20061021014415/http://www.aec.at/en/archives/prix_archive/prix_projekt.asp?iProjectID=11697
10. Classic Battletech: Quick-Start Rules. *Catalyst Game Lab.* [Online] [Verified: 14 August 2009.] http://www.classicbattletech.com/downloads/CBT_Quick_Start_Rules.pdf
11. Can You See Me Now? Homepage. *Blast Theory.* [Online] [Verified: 14 August 2009.] http://www.canyouseemenow.co.uk/
12. Golden Nica for Interactive Art: Can you see me now? *Prix Art Electronica.* [Online] 2003. [Verified: 14 August 2009.] http://web.archive.org/web/20070202221339/www.aec.at/en/archives/prix_archive/prix_projekt.asp?iProjectID=12455
13. WarGames (1983). *The Internet Movie Database.* [Online][Verified: 14 August 2009.] http://www.imdb.com/title/tt0086567/synopsis
14. WiGLE - Wireless Geographic Logging Engine. [Online] [Verified: 14 August 2009.] http://www.wigle.net/
15. DEF CON 10 WarDriving Results. [Online] [Verified: 14 August 2009.] http://www.defcon.org/html/defcon-10/dc-10-post/defcon-10-wardrive-post.html
16. Noderunner Homepage. [Online][Verified: 14 August 2009.] http://noderunner.omnistep.com/game.php?gameid=27
17. Golden Nica for Net Vision / Net Excellence: Noderunner. *Prix Ars Electronica.* [Online] 2003. [Verified: 14 August 2009.] http://web.archive.org/web/20060826235739/http://www.aec.at/en/archives/prix_archive/prix_projekt.asp?iProjectID=12443
18. Uncle Roy All Around You Homepage. *Blast Theory.* [Online] [Verified: 14 August 2009.] http://web.archive.org/web/20080209185320/http://www.uncleroyallaroundyou.co.uk/

19. Blast Theory. [Online]
[Verified: 14 August 2009.] http://www.blasttheory.co.uk/
20. Uncle Roy All Around You Project Homepage. *Blast Theory.* [Online] [Verified: 14 August 2009.] http://www.blasttheory.co.uk/bt/work_uncleroy.html
21. Mogi Homepage. *Newt Games.* [Online] [Verified: 14 August 2009.] http://web.archive.org/web/20080116223117/http://www.mogimogi.com/
22. gpsOne Product Homepage. *Qualcomm.* [Online] [Verified: 14 August 2009.] http://www.qctconnect.com/products/gpsone.html
23. Riot! 1831 - An Interactive Play for Voices based in Queen Square. *Mobile Bristol.* [Online] 2004. [Verified: 14 August 2009.] http://www.mobilebristol.com/QueenSq.html
24. MOSES - The Mobile Service and Exploration System. [Online] 2004. [Verified: 14 August 2009.] http://www.i3a.fh-trier.de/activities/projects/moses/moses_en.html
25. CAERUS Overview. *University of Birmingham.* [Online] 2006. [Verified: 14 August 2009.] http://portal.cetadl.bham.ac.uk/caerus/default.aspx
26. MobiMissions Project Homepage. *Futurelab.* [Online] 2007. [Verified: 14 August 2009.] http://www.futurelab.org.uk/projects/mobimissions
27. Insectopia. *Interactive Institute.* [Online] 2006. [Verified: 24 August 2009.] http://iperg.sics.se/iperg_games7.php
28. REXplorer Homepage. *Regensburg Experience.* [Online] 2007. [Verified: 14 August 2009.] http://www.rex-regensburg.de/
29. REXplorer Homepage. *RWTH Aachen.* [Online] 2007. [Verified: 14 August 2009.] http://hci.rwth-aachen.de/rexplorer.html
30. au. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Au_(mobile_phone_operator)
31. Single Story Building. *Blast Theory.* [Online] 2005. [Verified: 14 August 2009.] http://www.blasttheory.co.uk/bt/work_single_story_phone.html
32. I Love Bees. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/I_Love_Bees
33. Global Positioning System. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Gps
34. NMEA Data. *GPS Information.* [Online] [Verified: 1 September 2009.] http://www.gpsinformation.org/dale/nmea.htm
35. Real Time Location System (RTLS). *Ekahau.* [Online] [Verified: 14 August 2009.] http://www.ekahau.com/
36. PlaceEngine. *Koozyt.* [Online] [Verified: 14 August 2009.] http://www.placeengine.com/en
37. WLAN-Navigationssystem leitet Nürnberger Fußgänger. *heise online.* [Online] 14 January 2008. [Verified: 14 August 2009.] http://www.heise.de/newsticker/meldung/101765
38. Forum Nokia PRO Catalog, Applications for People, Volume 2: Games. *Nokia.* [Online] [Verified: 26 August 2009.] http://www.nokia.com/NOKIA_COM_1/Developers/Applications/Application_Catalogs/Games_Volume_2.pdf
39. Visby Under. *Interactive Institute.* [Online] [Verified: 26 August 2009.] http://www.tii.se/trgl/Projects/195.html

40. Near Field Communication. *Nokia.* [Online]
[Verified: 1 September 2009.] http://www.forum.nokia.com/Technology_Topics/Mobile_Technologies/Near_Field_Communication/
41. OpenCV. *Wikipedia.* [Online]
[Verified: 27 August 2009.] http://en.wikipedia.org/wiki/OpenCV
42. Place Lab - A privacy-observant location system. *Intel Research Seattle.* [Online] 2006. [Verified: 14 August 2009.] http://www.placelab.org/
43. WikiMapia. [Online] [Verified: 14 August 2009.] http://www.wikimapia.org/
44. Brüssel-Guide, Rundgang Zentrum. [Online] [Verified: 14 August 2009.] http://www.bruessel-gui.de/bruessel/rundgang_zentrum.html
45. Scape the Hood. [Online] 7 October 2005.
[Verified: 14 August 2009.] http://www.webtales.us/Scape1.html
46. Dreamweaver Product Homepage. *Adobe.* [Online] [Verified: 14 August 2009.] http://www.adobe.com/products/dreamweaver/
47. Mscape Homepage. *Hewlett-Packard.* [Online]
[Verified: 14 August 2009.] http://www.mscapers.com/
48. Create-A-Scape. *Futurelab.* [Online] 2007.
[Verified: 14 August 2009.] http://www.createascape.org.uk/
49. Google Maps. *Google.* [Online]
[Verified: 14 August 2009.] http://maps.google.com
50. Mscape2.5: The New StateMachine Object. [Online] 2008.
[Verified: 14 August 2009.] http://wiki.mscapers.com/bin/view/Main/WhatsNewInMscape25#The_New_StateMachine_Object
51. Active Ingredient. [Online]
[Verified: 21 September 2009.] http://www.i-am-ai.net/
52. Love City 2007. *Three Cities Create and Connect.* [Online] 19 January 2007.
[Verified: 14 August 2009.]
http://www.threecitiescreate.org.uk/News/Love_City_2007/
53. MobiLife - European Integrated Project (FP6). [Online] December 2006.
[Verified: 14 August 2009.] http://www.ist-mobilife.org/
54. World Wind. *NASA.* [Online]
[Verified: 14 August 2009.] http://worldwind.arc.nasa.gov/
55. Google Earth. *Google.* [Online]
[Verified: 14 August 2009.] http://earth.google.com/
56. Desert Rain. *Blast Theory.* [Online] 1999. [Verified: 21 September 2009.]
http://www.blasttheory.co.uk/bt/work_desertrain.html
57. Rider Spoke in London. *Barbican.* [Online] 11-21 October 2007.
[Verified: 14 August 2009.]
http://www.barbican.org.uk/theatre/event-detail.asp?ID=6071
58. Rider Spoke in Athens. *British Council.* [Online] 13-16 March 2008.
[Verified: 14 August 2009.]
http://www.britishcouncil.org/greece-arts-and-culture-rider-spoke.htm
59. Rider Spoke in Brighton. *Brighton Festival.* [Online] 8-11 May 2008.
[Verified: 14 August 2009.]
http://www.brightonfestival.org/Event_Details.aspx?eid=2728
60. Netstumbler Homepage. [Online]
[Verified: 14 August 2009.] http://www.netstumbler.com/
61. WiFiFoFum. *Aspecto Software.* [Online] 2007.
[Verified: 21 September 2009.] http://www.aspecto-software.com/rw/

62. Java Universal Network/Graph Framework Homepage. [Online] [Verified: 14 August 2009.] http://jung.sourceforge.net/
63. Matplotlib Homepage. [Online] [Verified: 14 August 2009.] http://matplotlib.sourceforge.net/
64. SQLite Homepage. [Online] [Verified: 14 August 2009.] http://www.sqlite.org/
65. OpenStreetMap. [Online] [Verified: 14 August 2009.] http://www.openstreetmap.org/
66. Shapefile Technical Description. *ESRI.* [Online] July 1998. [Verified: 14 August 2009.] http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf
67. OpenGIS Standards and Specifications. *Open Geospatial Consortium (OGC).* [Online] [Verified: 14 August 2009.] http://www.opengeospatial.org/standards
68. KML. *Google.* [Online] [Verified: 14 August 2009.] http://code.google.com/apis/kml/
69. Digimap Collections Description. *EDINA.* [Online] [Verified: 14 August 2009.] http://edina.ac.uk/digimap/description/
70. Blue Marble. *NASA.* [Online] [Verified: 14 August 2009.] http://earthobservatory.nasa.gov/Newsroom/BlueMarble/
71. Maps, Imagery, and Publications. *United States Geological Survey.* [Online] [Verified: 14 August 2009.] http://www.usgs.gov/pubprod/
72. TerraServer-USA. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/TerraServer-USA
73. Bing Maps. *Microsoft.* [Online] [Verified: 14 August 2009.] http://www.bing.com/maps/
74. Bing Maps for Enterprise. *Microsoft.* [Online] [Verified: 14 August 2009.] http://www.microsoft.com/maps/
75. MapServer. *University of Minnesota.* [Online] [Verified: 14 August 2009.] http://mapserver.org/
76. FWTools: Open Source GIS Binary Kit for Windows and Linux. [Online] [Verified: 14 August 2009.] http://fwtools.maptools.org/
77. OpenLayers: Free Maps for the Web. [Online] [Verified: 14 August 2009.] http://openlayers.org/
78. VTK - The Visualization Toolkit. *Kitware.* [Online] [Verified: 14 August 2009.] http://www.vtk.org/
79. ContextPhone Homepage. *University of Helsinki.* [Online] [Verified: 14 August 2009.] http://www.cs.helsinki.fi/group/context/
80. Flash Lite. *Adobe.* [Online] [Verified: 14 August 2009.] http://www.adobe.com/products/flashlite/
81. Paul Feyerabend. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Paul_Feyerabend
82. Immaterials: The ghost in the field (video). *Touch.* [Online] [Verified: 15 October 2009.] http://www.nearfield.org/2009/10/immaterials-the-ghost-in-the-field
83. Pytheas. *Sourceforge.* [Online] [Verified: 19 October 2009.] http://sourceforge.net/projects/pytheas/
84. Delicious. *Yahoo.* [Online] [Verified: 14 August 2009.] http://delicious.com/
85. Geolocation API Specification. *W3C.* [Online] 30 June 2009. [Verified: 14 August 2009.] http://dev.w3.org/geo/api/spec-source.html
86. Cabspotting. [Online] [Verified: 14 August 2009.] http://cabspotting.org/

87. cognition. *Britannica Online Encyclopedia.* [Online] [Verified: 14 August 2009.] http://www.britannica.com/EBchecked/topic/124474/cognition
88. Politbarometer Online. *ZDF.* [Online] 24 October 2008. [Verified: 14 August 2009.] http://www.zdf.de/ZDFmediathek/content/Politbarometer_zum_Anklicken_vom_24.10./411324
89. Bundestag. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Bundestag
90. House Price Index. *Nationwide.* [Online] 30 October 2008. [Verified: 14 August 2009.] http://www.nationwide.co.uk/hpi/historical/Oct_2008.pdf
91. Ebstorfer Weltkarte. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://de.wikipedia.org/wiki/Ebstorfer_Weltkarte
92. Orient. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Orient
93. The World Factbook. *Central Intelligence Agency.* [Online] [Verified: 14 August 2009.] https://www.cia.gov/library/publications/the-world-factbook/docs/refmaps.html
94. London Underground. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/London_Underground
95. Maps. *Transport for London.* [Online] [Verified: 14 August 2009.] http://www.tfl.gov.uk/tube/maps/
96. The History of Cartography Project Homepage. *University of Wisconsin.* [Online] [Verified: 14 August 2009.] http://www.geography.wisc.edu/histcart/
97. Open Source Python Framework for Flash Lite Developers. [Online] [Verified: 14 August 2009.] http://code.google.com/p/flyer/
98. Android. *Wikipedia.* [Online] [Verified: 4 September 2009.] http://en.wikipedia.org/wiki/Android_(operating_system)
99. Comparison of web application frameworks. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Comparison_of_web_application_frameworks
100. Common Object Request Broker Architecture. *Wikipedia.* [Online] [Verified: 14 August 2009.] http://en.wikipedia.org/wiki/Corba
101. PDIS - Personal Distributed Information Store. *Helsinki Institute for Information Technology (HIIT).* [Online] 31 December 2004. [Verified: 14 August 2009.] http://pdis.hiit.fi/pdis/
102. PART - Pervasive Applications RunTime. *Swedish Institute of Computer Science (SICS).* [Online] 9 October 2007. [Verified: 14 August 2009.] http://part.sourceforge.net/
103. PIMP - Pervasive Interactive Mobile Platform. *Swedish Institute of Computer Science (SICS).* [Online] 12 February 2008. [Verified: 14 August 2009.] http://pimp.sourceforge.net/
104. Equip2 Homepage. *University of Nottingham.* [Online] 12 June 2007. [Verified: 14 August 2009.] http://equip.sourceforge.net/equip2
105. Hessian Binary Web Service Protocol Homepage. *Caucho.* [Online] [Verified: 14 August 2009.] http://hessian.caucho.com/
106. IPerG - European Integrated Project on Pervasive Gaming (FP6). [Online] [Verified: 14 August 2009.] http://iperg.sics.se
107. Cordis - Sixth Framework Programme (FP6). *European Commission.* [Online] [Verified: 14 August 2009.] http://cordis.europa.eu/fp6/

108. Rider Spoke. *Blast Theory.* [Online] [Verified: 14 August 2009.] http://www.blasttheory.co.uk/bt/work_rider_spoke.html
109. INSCAPE - European Integrated Project on Interactive Storytelling for Creative People (FP6). [Online] [Verified: 14 August 2009.] http://www.inscapers.com/
110. Love City. *Active Ingredient.* [Online] (Work > Mobile > Love City). [Verified: 14 August 2009.] http://www.i-am-ai.net/
111. 'Ere be Dragons. *Active Ingredient.* [Online] [Verified: 14 August 2009.] http://web.archive.org/web/20070326104131/www.i-am-ai.net/erebedragons/

References

ABOWD, G. D. (1999) Software Engineering Issues for Ubiquitous Computing. *International Conference on Software Engineering.* Los Angeles, California.

ABOWD, G. D., ATKESON, C. G., HONG, J., LONG, S., KOOPER, R. & PINKERTON, M. (1997) Cyberguide: A mobile context-aware tour guide. *Wireless Networks,* 3**,** 421-433.

ADAMS, M. (2003) "On Your Mark, Get Set, Unwire! (Interviewed by Xeni Jardin)" [Online] [Verified: 14 August 2009.] http://www.wired.com/gadgets/wireless/news/2003/12/61721.

ADAMS, M., FLINTHAM, M., OPPERMANN, L., BENFORD, S. & GHELLAL, S. (2007) "IPerG Deliverable D17.2: Game Design Document - CCG Rider Spoke" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D17.2-Game-design-document-CCG-Rider-Spoke.pdf.

ADAR, E. (2006) GUESS: A Language and Interface for Graph Exploration. *CHI.* Montréal, Canada.

AHN, L. V. & DABBISH, L. (2008) Designing Games with a Purpose. *Communications of the ACM,* 51**,** 58-67.

ANSCOMBE, F. J. (1973) Graphs in Statistical Analysis. *The American Statistician,* 27**,** 17-21.

ASHBROOK, D. & STARNER, T. (2002) Learning Significant Locations and Predicting User Movement with GPS *ISWC.* Seattle, Washington.

AVISON, D., LAU, F., MYERS, M. & NIELSEN, P. A. (1999) Action Research. *Communications of the ACM,* 42**,** 94-96.

AYCINENA, P. (2008) Access For All. *Communications of the ACM,* 51**,** 12-14.

AZUMA, R. (1997) A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments,* 6**,** 355-385.

AZUMA, R., BAILLOT, Y., BEHRINGER, R., FEINER, S., JULIER, S. & MACINTYRE, B. (2001) Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications,* 21**,** 34-47.

BADEN-POWELL, R. (1908) *Scouting for Boys*, Oxford University Press.

BAHL, P. & PADMANABHAN, V. N. (2000) RADAR: An In-Building RF-based User Location and Tracking System. *IEEE INFOCOM.* Tel-Aviv, Israel.

BALLAGAS, R., KRATZ, S., YU, E., BORCHERS, J., WALZ, S. P., FUHR, C., TANN, M. J. & HOVESTADT, L. (2007) REXplorer: A Mobile, Pervasive Spell-Casting Game for Tourists. *CHI.* San Jose, California.

BALLAGAS, R., KUNTZE, A. & WALZ, S. P. (2008) Gaming Tourism: Lessons from Evaluating REXplorer, a Pervasive Game for Tourists *Pervasive Computing.* Sydney, Australia.

BALLAGAS, R. & WALZ, S. P. (2007) REXplorer: Using Player-Centered Iterative Design Techniques for Pervasive Game Development. IN MAGERKURTH, C. & RÖCKER, C. (Eds.) *Pervasive Gaming Applications - A Reader for Pervasive Gaming Research.* Shaker.

BANNON, L. (1991) From human factors to human actors: the role of psychology and human-computer interaction studies in system design. IN GREENBAUM, J. & KYNG, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates.

BARKHUUS, L., CHALMERS, M., TENNENT, P., HALL, M., BELL, M., SHERWOOD, S. & BROWN, B. (2005) Picking Pockets on the Lawn: The Development of Tactics and Strategies in a Mobile Game. *Ubicomp.* Tokyo, Japan.

BEDERSON, B. B., GROSJEAN, J. & MEYER, J. (2004) Toolkit Design for Interactive Structured Graphics. *IEEE Transactions On Software Engineering,* 30**,** 535-546.

BEEDLE, M. & SCHWABER, K. (2001) *Agile Software Development with Scrum*, Prentice Hall.

BELLOTTI, V., BACK, M., EDWARDS, W. K., GRINTER, R. E., HENDERSON, A. & LOPES, C. (2002) Making Sense of Sensing Systems: Five Questions for Designers and Researchers. *CHI.* Minneapolis, Minnesota.

BENFORD, S. (2005) "Future Location-Based Experiences" [Online] [Verified: 14 August 2009.] http://www.jisc.ac.uk/whatwedo/services/techwatch/reports/horizonscanning/hs0501.aspx.

BENFORD, S. (2007) Can You See Me Now? Chasing the Virtual. IN BORRIES, F. V., WALZ, S. P. & BÖTTGER, M. (Eds.) *Space Time Play.* Birkhäuser.

BENFORD, S., ANASTASI, R., FLINTHAM, M., DROZD, A., CRABTREE, A., GREENHALGH, C., TANDAVANITJ, N., ADAMS, M. & ROW-FARR, J. (2003) Coping with Uncertainty in a Location-Based Game. *IEEE Pervasive Computing,* 2**,** 34-41.

BENFORD, S., CRABTREE, A., FLINTHAM, M., DROZD, A., ANASTASI, R., PAXTON, M., TANDAVANITJ, N., ADAMS, M. & ROW-FARR, J. (2006) Can You See Me Now? *ACM Transactions on Computer Human Interaction,* 13**,** 100-133.

BENFORD, S. & FAHLÉN, L. (1993) A Spatial Model of Interaction in Large Virtual Environments. *European Conference on Computer-Supported Cooperative Work (ESCW).* Milan, Italy.

BENFORD, S., FLINTHAM, M., DROZD, A., ANASTASI, R., ROWLAND, D., TANDAVANITJ, N., ADAMS, M., ROW-FARR, J., OLDROYD, A. & SUTTON, J. (2004a) Uncle Roy All Around You: Implicating the City in a Location-Based Performance. *ACE.* Singapore.

BENFORD, S., GREENHALGH, C., TOLMIE, P., RODDEN, T., FLINTHAM, M., OPPERMANN, L. & REEVES, S. (2009) Lessons from Touring a Location-Based Experience. *(unpublished).*

BENFORD, S., ROWLAND, D., FLINTHAM, M., DROZD, A., HULL, R., REID, J., MORRISON, J. & FACER, K. (2005) Life on the Edge: Supporting Collaboration in Location-Based Experiences. *CHI.* Portland, Oregon.

BENFORD, S., SEAGER, W., FLINTHAM, M., ANASTASI, R., ROWLAND, D., HUMBLE, J., STANTON, D., BOWERS, J., TANDAVANITJ, N., ADAMS, M., FARR, J. R., OLDROYD, A. & SUTTON, J. (2004b) The Error of Our Ways: The Experience of Self-Reported Position in a Location-Based Game. *Ubicomp.* Nottingham, England.

BICHARD, J.-P. & WAERN, A. (2008) Pervasive Play, Immersion and Story: designing Interference *3rd International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA).* Athens, Greece.

BJERKNES, G., EHN, P. & KYNG, M. (1987) *Computers and Democracy: A Scandinavian Challenge*, Avebury.

BJERVER, M. (2006) Player Behaviour in Pervasive Games - using the City as Game Board in Botfighters. Stockholm, Royal Institute of Technology.

BJÖRK, S. & HOLOPAINEN, J. (2005) *Patterns in Game Design*, Charles River Media.

BLASS, E. (2006) "Wardriver arrested for snagging coffee shop signal" [Online] [Verified: 14 August 2009.] http://www.engadget.com/2006/06/23/wardriver-arrested-for-snagging-coffee-shop-signal/.

BØDKER, S., EHN, P., SJÖGREN, D. & SUNDBLAD, Y. (2000) Co-operative Design: perspectives on 20 years with 'the Scandinavian IT Design Model'. *NordiCHI.* Stockholm, Sweden.

BØDKER, S., GREENBAUM, J. & KYNG, M. (1991) Setting the Stage for Design as Action. IN GREENBAUM, J. & KYNG, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates.

BØDKER, S. & GRØNBÆK, K. (1991) Design in Action: From Prototyping by Demonstration to Cooperative Prototyping. IN GREENBAUM, J. & KYNG, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates.

BORRIES, F. V., WALZ, S. P. & BÖTTGER, M. (2007) *Space Time Play*, Birkhäuser.

BRADSKI, G. & KAEHLER, A. (2008) *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly.

BRAVERMAN, H. (1974) *Labor and Monopoly Capitalism: The Degradation of Work in the Twentieth Century* Monthly Review Press.

BROLL, G., BENFORD, S. & OPPERMANN, L. (2006) Exploiting Seams in Mobile Phone Games. *PerGames Workshop.* Dublin, Ireland.

BROLL, G., KECK, S., HOLLEIS, P. & BUTZ, A. (2009) Improving the Accessibility of NFC/RFID-based Mobile Interaction through Learnability and Guidance. *MobileHCI.* Bonn, Germany.

BRUNDELL, P., TENNENT, P., GREENHALGH, C., KNIGHT, D., CRABTREE, A., O'MALLEY, C., AINSWORTH, S., CLARKE, D., CARTER, R. & ADOLPHS, S. (2008) Digital Replay System (DRS): A Tool for Interaction Analysis. *Workshop on Interaction Analysis, International Conference for the Learning Sciences (ICLS).* Utrecht, The Netherlands.

BURGESS, L. (1999) "Address to the International Dance and Technology Conference 1999" [Online] [Verified: 14 August 2009.] http://artscool.cfa.cmu.edu/~burgess/essay.html.

BUXTON, B. (2007) *Sketching User Experiences: Getting the Design Right and the Right Design*, Morgan Kaufmann.

BUXTON, B. (2009) "From the Materialistic to the Experiential (Talk at Princeton University, April 2009)" [Online] [Verified: 14 August 2009.] http://phillychi.acm.org/?p=144.

BUXTON, W. & SNIDERMAN, R. (1980) Iteration in the Design of the Human-Computer Interface. *13th Annual Meeting, Human Factors Association of Canada.*

CAMERON, L. & ULMER, D. (2004) *The Geocaching Handbook*, Globe Pequot Press.

CAPRA, M., RADENKOVIC, M., BENFORD, S., OPPERMANN, L., DROZD, A. & FLINTHAM, M. (2005) The multimedia challenges raised by pervasive games. *ACM International Conference on Multimedia.* Singapore.

CARLUCCIO, D., LEMKE-RUST, K., PAAR, C. & SADEGHI, A.-R. (2006) E-Passport: The Global Traceability Or How to Feel Like a UPS Package. *Workshop on Information Security Applications.* Jeju Island, Korea.

CHALMERS, M., BARKHUUS, L., BELL, M., BROWN, B., HALL, M., SHERWOOD, S. & TENNENT, P. (2005a) Gaming on the Edge: Using Seams in Pervasive Games. *PerGames Workshop.* Munich, Germany.

CHALMERS, M., BARKHUUS, L., BELL, M., BROWN, B., HALL, M., SHERWOOD, S. & TENNENT, P. (2005b) Gaming on the Edge: Using Seams in Ubicomp Games. *Advances in Computer Entertainment (ACE).* Valencia, Spain.

CHALMERS, M. & GALANI, A. (2004) Seamful Interweaving: Heterogeneity in the Theory and Design of Interactive Systems. *Designing Interactive Systems.* Cambridge, Massachusetts.

CHAMBERLAIN, A., BENFORD, S., FLINTHAM, M., REEVES, S., OPPERMANN, L. & MARSHALL, J. (2008) "IPerG Deliverable D17.3: Final Report from Rider Spoke" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D17.3.pdf.

CHAMBERLAIN, A. & OPPERMANN, L. (2006) The Digital City: Sex, Cams and Scams. *Exurban Noir Workshop, Ubicomp.* Orange County, CA.

CHENG, Y.-C., CHAWATHE, Y., LAMARCA, A. & KRUMM, J. (2005) Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. *MobiSys.* Seattle, Washington.

CHEVEREST, K., DAVIES, N., MITCHELL, K., FRIDAY, A. & EFSTRATIOU, C. (2000) Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. *CHI.* The Hague, The Netherlands.

CHEVEREST, K., MITCHELL, K. & DAVIES, N. (1998) Design of an Object Model for a Context Sensitive Tourist GUIDE. *International Workshop on Interactive Applications of Mobile Computing.* Rostock, Germany.

CLINTON, B. (2000) "Statement by the President regarding the United States' decision to stop degrading Global Positioning System Accuracy" [Online] [Verified: 14 August 2009.] http://clinton3.nara.gov/WH/EOP/OSTP/html/0053_2.html.

COHEN, J.-L. (2006) *Le Corbusier, 1887-1965: The Lyricism of Architecture in the Machine Age*, Taschen.

CRABTREE, A., BENFORD, S., TOM RODDEN, GREENHALGH, C., FLINTHAM, M., ANASTASI, R., DROZD, A., ADAMS, M., ROW-FARR, J., TANDAVANITJ, N. & STEED, A. (2004) Orchestrating a Mixed Reality Game 'On the Ground'. *CHI.* Vienna, Austria.

CURRY, P. (2007) Quake: The Turning Point of 3D Games. IN BORRIES, F. V., WALZ, S. P. & BÖTTGER, M. (Eds.) *Space Time Play.* Birkhäuser.

DAVIES, N., CHEVEREST, K., MITCHELL, K. & FRIDAY, A. (1999) 'Caches in the Air': Disseminating Tourist Information in the Guide System. *IEEE Workshop on Mobile Computing Systems and Applications.* New Orleans, Louisiana.

DAVIS, M. (1999) Rhetoric, technical writing, and ethics. *Science and Engineering Ethics,* 5**,** 463-478.
DAVIS, S. B., MOAR, M., JACOBS, R., WATKINS, M., SHACKFORD, R., CAPRA, M. & OPPERMANN, L. (2007) Mapping Inside Out. IN MAGERKURTH, C. & RÖCKER, C. (Eds.) *Pervasive Gaming Applications - A Reader for Pervasive Gaming Research.* Shaker.
DIX, A., FINLAY, J., ABOWD, G. D. & BEALE, R. (1993) *Human-Computer Interaction (Third Edition)*, Prentice Hall.
DOURISH, P. (2001) *Where the Action Is: The Foundations of Embodied Interaction*, MIT Press.
DOURISH, P. (2006) Re-space-ing place: "place" and "space" ten years on. *CSCW.* Banff, Alberta, Canada.
DOW, S., MACINTYRE, B., LEE, J., OEZBEK, C., BOLTER, J. D. & GANDY, M. (2005) Wizard of Oz Support throughout an Iterative Design Process. *IEEE Pervasive Computing,* 4**,** 18-26.
DREHER, T. (2008) "Sammeltipp" [Online] [Verified: 14 August 2009.] http://iasl.uni-muenchen.de/links/TippSammel1-3.html.
DROZD, A., BENFORD, S., CHAMBERLAIN, A., WRIGHT, M. & TANDAVANITJ, N. (2006) Hitchers: A Scalable Location-Based Game that Explores the Seams in Cellular Positioning. *Ubicomp.* Orange County, California.
EHN, P. (1993) Scandinavian Design: On Participation and Skill. IN SCHULER, D. & NAMIOKA, A. (Eds.) *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates.
EHN, P. & KYNG, M. (1991) Cardboard Computers: Mocking-it-up or Hands-on the future. IN GREENBAUM, J. & KYNG, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates.
EIRUND, H., MÜLLER, B. & SCHREIBER, G. (2000) *Formale Beschreibungsverfahren der Informatik. Ein Arbeitsbuch für die Praxis* B.G. Teubner Verlag.
EISSFELLER, B., TEUBER, A. & ZUCKER, P. (2005) Indoor-GPS: Ist der Satellitenempfang in Gebäuden möglich? *Zeitschrift für Vermessung,* 130**,** 226-234.
ERICSSON, M. (2003) Enchanting Reality: A Vision of Big Experiences on Small Platforms. *Level Up Conference.* Utrecht, The Netherlands.
FEINER, S., HÖLLERER, T., GAGAS, E., HALLAWAY, D., TERAUCHI, T., GÜVEN, S. & MACINTYRE, B. "MARS - Mobile Augmented Reality Systems " [Online] [Verified: 14 August 2009.] http://graphics.cs.columbia.edu/projects/mars/.
FEINER, S., MACINTYRE, B., HÖLLERER, T. & WEBSTER, A. (1997) A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. *International Symposium on Wearable Computing.* Cambridge, Massachusetts.
FISCHER, J. E. & BENFORD, S. (2009) Inferring Player Engagement in a Pervasive Experience. *CHI.* Boston, Massachusetts.
FISCHER, J. E., LINDT, I. & STENROS, J. (2007) Evaluation of Crossmedia Gaming Experiences in Epidemic Menace *Pergames.* Salzburg, Austria.
FLINTHAM, M. (2005) Painting the Town Red: Configuring Location-Based Games by Colouring Maps. *Advances in Computer Entertainment Technology (ACE).* Valencia, Spain.

FLINTHAM, M., ANASTASI, R., BENFORD, S., DROZD, A., MATHRICK, J., ROWLAND, D., OLDROYD, A., SUTTON, J., TANDAVANITJ, N., ADAMS, M. & ROW-FARR, J. (2003a) Uncle Roy All Around You: Mixing Games and Theatre on the City Streets. *Level Up Conference.* Utrecht, The Netherlands.

FLINTHAM, M., ANASTASI, R., BENFORD, S., HEMMINGS, T., CRABTREE, A., GREENHALGH, C., RODDEN, T., TANDAVANITJ, N., ADAMS, M. & ROW-FARR, J. (2003b) Where On-Line Meets On-The-Streets: Experiences With Mobile Mixed Reality Games. *CHI.* Ft. Lauderdale, Florida.

FLINTHAM, M., BENFORD, S., HUMBLE, J., TANDAVANITJ, N., ADAMS, M. & ROW-FARR, J. (2005) "I Like Frank: A Mixed Reality Game for 3G Phones" [Online] [Verified: 14 August 2009.] http://www.amutualfriend.co.uk/html/papers.html.

FLINTHAM, M., SMITH, K., BENFORD, S., CAPRA, M., GREEN, J., GREENHALGH, C., WRIGHT, M., ADAMS, M., TANDAVANITJ, N., FARR, J. R. & LINDT, I. (2007) Day of the Figurines: A Slow Narrative-Driven Game for Mobile Phones Using Text Messaging. *PerGames.* Salzburg, Austria.

FLOYD, C. (1984) A Systematic Look at Prototyping. IN BUDDE, R., KUHLENKAMP, K., MATHIASSEN, L. & ZULLIGHOVEN, H. (Eds.) *Approaches to Prototyping.* Springer.

FLOYD, C. (1987) Outline of a paradigm change in software engineering. IN BJERKNES, G., EHN, P. & KYNG, M. (Eds.) *Computers and Democracy - a Scandinavian Challenge.* Avebury.

GILBERT, R. (1989) "Why Adventure Games Suck And What We Can Do About It" [Online] [Verified: 14 August 2009.] http://grumpygamer.com/adventuregames.

GIRARDIN, F., FIORE, F. D., BLAT, J. & RATTI, C. (2007) Understanding of Tourist Dynamics from Explicitly Disclosed Location Information. *International Symposium on LBS & TeleCartography.* Hong Kong, China.

GOODCHILD, M. F. (2007) Citizens as sensors: the world of volunteered geography. *GeoJournal,* 69**,** 211-221.

GÖRZ, G. (2006) Kognitive Karten des Mittelalters. Digitale Erschließung mittelalterlicher Weltkarten. IN BURCKHARDT, D., HOHLS, R. & PRINZ, C. (Eds.) *Geschichte im Netz: Praxis, Chancen, Visionen - Beiträge der Tagung .hist 2006.* Berlin, Clio-online und Humboldt Universität.

GRAAS, F. V. & BRAASCH, M. S. (1994) Selective Availability. IN PARKINSON, B. W. & SPILKER, J. J. (Eds.) *Global Positioning System: Theory and Applications.* American Institute of Aeronautics and Astronautics.

GRANT, L., BENFORD, S., HAMPSHIRE, A., DROZD, A. & GREENHALGH, C. (2007a) MobiMissions: The Game of Missions for Mobile Phones. *PerGames.* Salzburg, Austria.

GRANT, L., DAANEN, H., BENFORD, S., HAMPSHIRE, A., DROZD, A. & GREENHALGH, C. (2007b) MobiMissions: the game of missions for mobile phones. *ACM SIGGRAPH educators program.* San Diego, California.

GREENBAUM, J. & KYNG, M. (1991a) *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates.
GREENBAUM, J. & KYNG, M. (1991b) Introduction: Situated Design. IN GREENBAUM, J. & KYNG, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates.
GREENFIELD, A. (2006) *Everyware: The Dawning Age of Ubiquitous Computing*, Peachpit Press.
GREENHALGH, C., BENFORD, S., DROZD, A., FLINTHAM, M., HAMPSHIRE, A., OPPERMANN, L., SMITH, K. & TYCOWICZ, C. V. (2007a) Addressing Mobile Phone Diversity in Ubicomp Experience Development. *UbiComp.* Innsbruck, Austria.
GREENHALGH, C., BENFORD, S., DROZD, A., FLINTHAM, M., HAMPSHIRE, A., OPPERMANN, L., SMITH, K. & TYCOWICZ, C. V. (2007b) EQUIP2: A Platform for Mobile Phone-based Game Development. IN MAGERKURTH, C. & RÖCKER, C. (Eds.) *Concepts and Technologies for Pervasive Games - A Reader for Pervasive Gaming Research.* Shaker.
GREENHALGH, C., IZADI, S., RODDEN, T. & BENFORD, S. (2001) "The EQUIP Platform: Bringing Together Physical and Virtual Worlds" [Online] [Verified: 14 August 2009.] http://www.cs.nott.ac.uk/~sxi/papers/tech01a.pdf.
GREENHALGH, C., IZADI, S., TAYLOR, I. & MATHRICK, J. (2004) "A Toolkit to Support Rapid Construction of Ubicomp Environments" [Online] [Verified: 14 August 2009.] http://equip.sourceforge.net/ect/java/docs/developer/papers/ect-intro.pdf.
GROPIUS, W. (1919) "Bauhaus Manifest" [Online] [Verified: 14 August 2009.] http://www.bauhaus.de/bauhaus1919/manifest1919.htm.
GÜVEN, S. & FEINER, S. (2003) Authoring 3D Hypermedia for Wearable Augmented and Virtual Reality. *ISWC.* White Plains, New York.
HAKLAY, M. & WEBER, P. (2008) OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing,* 7**,** 12-18.
HANSSON, P., ÅKESSON, K.-P. & WALLBERG, A. (2007) "IPerG Deliverable D11.9: Second generation core platform (Work package 11: Enhanced Reality Live Role-playing)" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D11.9-Second-generation-core-platform.pdf.
HARMON, K. (2004) *You are here: personal geographies and other maps of the imagination*, Princeteon Architectural Press.
HARPER, R., RODDEN, T., ROGERS, Y. & SELLEN, A. (2008) "Being Human: Human Computer Interaction in 2020" [Online] [Verified: 14 August 2009.] http://research.microsoft.com/en-us/um/cambridge/projects/hci2020/download.html.
HARRISON, S. & DOURISH, P. (1996) Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems. *CSCW.* Boston, Massachusetts.
HAUPERT, J. (2005) "Entwicklung eines interaktiven Grafikeditors zur hierarchischen geo-referenzierten Umgebungsmodellierung (Bachelor Thesis)" [Online] [Verified.
HAUPERT, J. (2007) "Ein Werkzeug zur Modellierung und Simulation von Benutzerassistenzsystemen in Instrumentierten Umgebungen (Master Thesis)" [Online] [Verified.

HEIDEGGER, M. (1927) *Being and Time (English translation 1962, Paperback edition 2008)*, Harper Perennial

HELIÖ, S., LJUNGSTRAND, P., WAERN, A., LINDT, I., WETZEL, R. & ÅKESSON, K.-P. (2008) "IPerG Deliverable D14.4: Box Evaluation Report (Showcase - Boxed Pervasive Games)" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D14.4.pdf.

HIGHTOWER, J. (2003) From Position to Place. *Workshop on Location-Aware Computing.* Seattle, Washington.

HIGHTOWER, J. (2004) The Location Stack. University of Washington.

HIGHTOWER, J., CONSOLVO, S., LAMARCA, A., SMITH, I. & HUGHES, J. (2005) Learning and Recognizing the Places We Go. *UbiComp.* Tokyo, Japan.

HÖLLERER, T., FEINER, S. & PAVLIK, J. (1999) Situated Documentaries: Embedding Multimedia Presentations in the Real World. *International Symposium on Wearable Computers.* San Francisco, California.

HORN, R. E. (1999) Information Design: Emergence of a New Profession. IN JACOBSON, R. (Ed.) *Information Design.* London, MIT Press.

HUIZINGA, J. H. (1939) *Homo Ludens: A Study of the Play-Element in Culture*, Beacon Press.

HULL, R., CLAYTON, B. & MELAMED, T. (2004) Rapid Authoring of Mediascapes. *Ubicomp.* Nottingham, England, Springer.

HUNTER, J. D. (2007) Matplotlib: A 2D Graphics Environment. *IEEE & AIP Computing in Science & Engineering,* 9**,** 90-95.

HURLEY, C., THORNTON, F., PUCHOL, M., MILNER, M. & ROGERS, R. (2004) *WarDriving: Drive, Detect, Defend: a Guide to Wireless Security*, Syngress Publishing.

IMAI, M. (1986) *Kaizen*, McGraw-Hill.

JACOBSON, I., BOOCH, G. & RUMBAUGH, J. (1999) *The Unified Software Development Process*, Addison-Wesley Professional.

JOFFE, B. (2004) "Mogi: Interview on Wireless Watch Japan" [Online] [Verified: 14 August 2009.] http://wirelesswatch.jp/2004/07/02/mogi-socially-connected-gps-gaming/.

JOFFE, B. (2005) Mogi: Location and Presence in a Pervasive Community Game. *Ubicomp Workshop on Ubiquitous Gaming and Entertainment.* Tokyo, Japan.

JOFFE, B. (2007) Mogi: Location Based Services - A Community Game in Japan. IN BÖRRIES, F. V., WALZ, S. P. & BÖTTGER, M. (Eds.) *Space Time Play.* Birkhäuser.

JONSSON, S. & WAERN, A. (2008) The Art of Game-Mastering Pervasive Games. *Advances in Computer Entertainment (ACE).* Yokohama, Japan.

KAMADA, T. & KAWAI, S. (1989) An algorithm for drawing general undirected graphs. *Information Processing Letters,* 31**,** 7-15.

KATO, H. & BILLINGHURST, M. (1999) Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. *International Workshop on Augmented Reality* San Francisco, California.

KOCH, T. (2004) The Map as Intent: Variations on the Theme of John Snow. *Cartographica,* 39.

KUHN, S. & MULLER, M. J. (1993) Participatory Design. *Communications of the ACM,* 36**,** 24-28.

KÜPPER, A. (2005) *Location-based Services: Fundamentals and Operation,* Chichester, Wiley.
KYNG, M. (1994) Collective Resources meets Puritanism. *Scandinavian Journal of Information Systems,* 6**,** 85-96.
LAASONEN, K., RAENTO, M. & TOIVONEN, H. (2004) Adaptive On-Device Location Recognition. *Pervasive.* Linz / Vienna, Austria.
LAMARCA, A., CHAWATHE, Y., CONSOLVO, S., HIGHTOWER, J., SMITH, I., SCOTT, J., SOHN, T., HOWARD, J., HUGHES, J., POTTER, F., TABERT, J., POWLEDGE, P., BORRIELLO, G. & SCHILIT., B. (2005) Place Lab: Device Positioning Using Radio Beacons in the Wild. *Pervasive Computing.* Munich, Germany.
LANTZ, F. (2007) Conqwest: An Urban Treasure Hunt. IN BORRIES, F. V., WALZ, S. P. & BÖTTGER, M. (Eds.) *Space Time Play.* Birkhäuser.
LAURILA, J., TUULOS, V. & MACLAVERTY, R. (2006) Scripting Environment for Pervasive Application Exploration on Mobile Phones. *Pervasive.* Dublin, Ireland.
LEARY, A. (2005) "Wi-Fi cloaks a new breed of intruder" [Online] [Verified: 14 August 2009.] http://www.sptimes.com/2005/07/04/State/Wi_Fi_cloaks_a_new_br.shtml.
LEBOFF, D. & DEMUTH, T. (1999) *No Need to Ask! Early Maps of London's Underground Railways*, Capital Transport Publishing.
LEPETIT, V. (2009) "Computer Vision Research Projects and Publications" [Online] [Verified: 27 August 2009.] http://cvlab.epfl.ch/~vlepetit/.
LEWIN, K. (1946) Action research and minority problems. *Journal of Social Issues,* 2**,** 34-46.
LINDT, I., OHLENBURG, J., PANKOKE-BABATZ, U., GHELLAL, S., OPPERMANN, L. & ADAMS, M. (2005) Designing Cross Media Games. *PerGames Workshop.* Munich, Germany.
LINDT, I., OHLENBURG, J., PANKOKE-BABATZ, U., PRINZ, W. & GHELLAL, S. (2006) Combining Multiple Gaming Interfaces in Epidemic Menace. *CHI.* Montreal, Quebec, Canada.
LINKE, W. (2003) *Orientierung mit Karte, Kompass, GPS,* Herford, Busse + Seewald GmbH.
LOEWY, R. (1951) *Never Leave Well Enough Alone*, Econ.
LONG, S., AUST, D., ABOWD, G. D. & ATKESON, C. G. (1996a) Cyberguide: Prototyping Context-Aware Mobile Applications. *CHI.* Vancouver, Canada.
LONG, S., KOOPER, R., ABOWD, G. D. & ATKESON, C. G. (1996b) Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. *ACM International Conference on Mobile Computing.* Rye, New York.
LONG, S., PINKERTON, M. & ABOWD, G. "CyBARguide Project Notes" [Online] [Verified: 14 August 2009.] http://www.cc.gt.atl.ga.us/fce/cyberguide/cybarguide/CyBARguide.html.
LOOSER, J., GRASSET, R., SEICHTER, H. & BILLINGHURST, M. (2006) OSGART - A pragmatic approach to MR. *Industrial Workshop at ISMAR.* Santa Barbara, California.
LOVELESS, A., DENNING, T., FISHER, T. & HIGGINS, C. (2008) Create-A-Scape: Mediascapes and curriculum integration. *Education and Information Technologies,* 13**,** 345-355.

MACINTYRE, B. & FEINER, S. (1996) Language-Level Support for Exploratory Programming of Distributed Virtual Environments. *User Interface Software and Technology.* Seattle, Washington.
MACKENZIE, J. "GIS Analyses of Dr. Snow's Cholera Map " [Online] [Verified: 14 August 2009.] http://www.udel.edu/johnmack/frec480/cholera/cholera2.html.
MARTIN, J. (1991) *Rapid Application Development*, Macmillan.
MATYAS, S., MATYAS, C., SCHLIEDER, C., KIEFER, P., MITARAI, H. & KAMATA, M. (2008) Designing location-based mobile games with a purpose: collecting geospatial data with CityExplorer. *Advances in Computer Entertainment Technology (ACE).* Yokohama, Japan
MCLUHAN, M. (1964) *Understanding Media - The Extensions of Man*, Routledge & Kegan Paul Ltd.
MENDE, W., OPPERMANN, E. & HEITZER, L. (1998) Mobile Radio Network Management Supported by a Planning Tool. *Network Operations and Management Symposium (NOMS).* New Orleans, Louisiana.
MONTOLA, M. (2005) Exploring the Edge of the Magic Circle: Defining Pervasive Games. *Design Automation Conference.* Copenhagen, Denmark.
MONTOLA, M., STENROS, J. & WAERN, A. (2009a) *Pervasive Games: Theory and Design*, Morgan Kaufmann.
MONTOLA, M., STENROS, J. & WAERN, A. (2009b) Philosophies and strategies of pervasive larp design. *Larp, the Universe and Everything.* Knutepunkt.
MOORE, G. (2005) "Excerpts from A Conversation with Gordon Moore: Moore's Law (Video Transcript)" [Online] [Verified: 14 August 2009.] ftp://download.intel.com/museum/Moores_Law/Video-Transcripts/Excepts_A_Conversation_with_Gordon_Moore.pdf.
NAISMITH, L., SHARPLES, M. & TING, J. (2005) Evaluation of CAERUS: A Context Aware Mobile Guide. *mLearn.* Cape Town, South Africa.
NOLD, C. (2004) "Biomapping" [Online] [Verified: 14 August 2009.] http://biomapping.net/.
NORTH, C. & SHNEIDERMAN, B. (2000) Snap-together visualization: can users construct and operate coordinated visualizations? *Int. J. Human-Computer Studies,* 53**,** 715-739.
NURMI, P. & KOOLWAAIJ, J. (2006) Identifying meaningful locations. *MOBIQUITOUS.* San Jose, California.
O'MADADHAIN, J., FISHER, D., SMYTH, P., WHITE, S. & BOEY, Y.-B. (2005) "Analysis and Visualization of Network Data using JUNG" [Online] [Verified: 14 August 2009.] http://jung.sourceforge.net/doc/.
OHLENBURG, J., HERBST, I., LINDT, I., FRÖHLICH, T. & BROLL, W. (2004) The MORGAN framework: enabling dynamic multi-user AR and VR projects. *ACM Symposium on Virtual Reality Software and Technology.* Hong Kong, China.
OHLENBURG, J., LINDT, I. & PANKOKE-BABATZ, U. (2006) Report about the Crossmedia Game Epidemic Menace. *PerGames.* Dublin, Ireland.
OKABE, A., BOOTS, B., SUGIHARA, K. & CHIU, S. N. (2000) *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Wiley.
OPPERMANN, L. (2007a) Extending Authoring Tools for Location-Aware Applications with an Infrastructure Visualisation Layer. *Pervasive (Doctoral Colloquium).* Toronto, Canada.

OPPERMANN, L. (2007b) Towards a User-Centric Location-Based Service Approach for Mobile Edutainment and Tourism Applications. *International Conference on Humans and Computers.* Düsseldorf, Germany.
OPPERMANN, L. (2008) On the choice of programming languages for developing location-based mobile games. *Mobile Gaming Workshop, Informatik 2008.* Munich, Germany.
OPPERMANN, L. (2009) An Abstract Location-Model for Mobile Games. *Mobile Gaming Workshop, Informatik 2009.* Lübeck, Germany.
OPPERMANN, L., BROLL, G., CAPRA, M. & BENFORD, S. (2006) Extending Authoring Tools for Location-Aware Applications with an Infrastructure Visualization Layer. *UbiComp.* Orange County, California.
OPPERMANN, L., JACOBS, R., WATKINS, M., SHACKFORD, R., TYCOWICZ, C. V., WRIGHT, M., CAPRA, M., GREENHALGH, C. & BENFORD, S. (2007) Love City: A Text-Driven, Location-Based Mobile Phone Game Played Between 3 Cities. IN MAGERKURTH, C. & RÖCKER, C. (Eds.) *Pervasive Gaming Applications - A Reader for Pervasive Gaming Research.* Shaker.
OPPERMANN, L., KOLEVA, B., BENFORD, S., JACOBS, R. & WATKINS, M. (2008) Fighting with Jelly: User-Centered Development of a Wireless Infrastructure Visualization Tool for Authoring Location-Aware Experiences. *Advances in Computer Entertainment Technology (ACE).* Yokohama, Japan.
PAELKE, V., OPPERMANN, L. & REIMANN, C. (2007) Mobile Location-Based Gaming. IN MENG, L., ZIPF, A. & WINTER, S. (Eds.) *Map-based Mobile Services: Design, Interactivity and Usability.* Springer.
PAELKE, V., REIMANN, C. & STICHLING, D. (2004) Kick-Up Menus. *CHI.* Vienna, Austria.
PARKINSON, B. W. (1994) Introduction and Heritage of NAVSTAR, the Global Positioning System. IN PARKINSON, B. W. & SPILKER, J. J. (Eds.) *Global Positioning System: Theory and Applications.* Global Positioning System: Theory and Applications.
PATEL, S. N., KIENTZ, J. A., HAYES, G. R., BHAT, S. & ABOWD, G. D. (2006) Farther Than You May Think: An Empirical Investigation of the Proximity of Users to Their Mobile Phones. *UbiComp.* Orange County, California.
PATTERSON, D. J., LIAO, L., GAJOS, K., COLLIER, M., LIVIC, N., OLSON, K., WANG, S., FOX, D. & KAUTZ, H. (2004) Opportunity Knocks: a System to Provide Cognitive Assistance with Transportation Services. *UbiComp.* Nottingham, England.
PEITZ, J. (2006) "IPerG Deliverable D9.8B: Game Design Document – "Insectopia"" [Online] [Verified: 24. August 2009.] http://www.pervasive-gaming.org/Deliverables/D9.8B-Game-Design-Document-Insectopia.pdf.
PERLIN, K. & FOX, D. (1993) Pad - An Alternative Approach to the Computer Interface. *ACM SIGGRAPH.* Anaheim, California.
PETTERSSON, R. (2002) *Information Design: An Introduction* John Benjamins Publishing Company.
PIAS, C. (2007) Play as Creative Misuse: Barcode Battler and the Charm of the Real. IN BORRIES, F. V., WALZ, S. P. & BÖTTGER, M. (Eds.) *Space Time Play.* Birkhäuser.

PIEKARSKI, W. "About The ARQuake Project" [Online] [Verified: 14 August 2009.] http://wearables.unisa.edu.au/arquake/.

PIEKARSKI, W. "Tinmith Augmented Reality System Homepage" [Online] [Verified: 14 August 2009.] http://www.tinmith.net/.

PIEKARSKI, W. (2004) Interactive 3D Modelling in Outdoor Augmented Reality Worlds. *Wearable Computer Lab, School of Computer and Information Science.* University of South Australia.

PIEKARSKI, W. & THOMAS, B. (2002) ARQuake: The Outdoor Augmented Reality Gaming System. *Communications of the ACM,* 45**,** 36-38.

POULSEN, K. (2002) "War driving by the Bay" [Online] [Verified: 14 August 2009.] http://www.securityfocus.com/news/192.

PRECHELT, L. (2000) An Empirical Comparision of Seven Programming Languages. *IEEE Computer,* 33**,** 23-29.

PREECE, J., ROGERS, Y. & SHARP, H. (2002) *Interaction Design: Beyond Human-Computer Interaction,* New York, John Wiley & Sons, Inc.

RAENTO, M., OULASVIRTA, A., PETIT, R. & TOIVONEN, H. (2005) Context Phone: a prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing,* 4**,** 51-59.

RAPPIN, N. & DUNN, R. (2006) *wxPython in Action*, Manning.

RASHID, O., BAMFORD, W., COULTON, P., EDWARDS, R. & SCHEIBLE, J. (2006) PAC-LAN: Mixed-Reality Gaming with RFID-Enabled Mobile Phones. *ACM Computers in Entertainment,* 4.

RASKIN, J. (2000) *The Humane Interface - New Directions for Designing Interactive Systems*, Addison-Wesley.

REENSKAUG, T. (1979) "Models - Views - Controllers (Xerox Parc Technical Note)" [Online] [Verified: 14 August 2009.] http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html.

REID, J., CATER, K., FLEURIOT, C. & HULL, R. (2005a) "Experience Design Guidelines for Creating Situated Mediascapes" [Online] [Verified: 14 August 2009.] http://www.hpl.hp.com/techreports/2005/HPL-2005-181.html.

REID, J., HULL, R., CATER, K. & FLEURIOT, C. (2005b) Magic Moments in Situated Mediascapes. *Advances in Computer Entertainment Technology (ACE).* Valencia, Spain.

REKIMOTO, J., MIYAKI, T. & ISHIZAWA, T. (2007) LifeTag: WiFi-based Continuous Location Logging for Life Pattern Analysis. *International Symposium on Location- and Context-Awareness (LOCA).*

RIMEY, K. (2004) "Personal Distributed Information Store (PDIS) - Final report" [Online] [Verified: 14 August 2009.] http://pdis.hiit.fi/pdis/report-2004/pdis-final-report.pdf.

ROSIEN, W. (1952) *Die Ebstorfer Weltkarte*, Niedersächs. Amt f. Landesplanung u. Statistik.

ROWLAND, D., FLINTHAM, M., OPPERMANN, L., KOLEVA, B., CHAMBERLAIN, A., MARSHALL, J., BENFORD, S. & PEREZ, C. (2009) Ubikequitous Computing: Designing Interactive Experiences for Cyclists. *Mobile HCI.* Bonn, Germany.

ROYCE, W. W. (1970) Managing the Development of Large Software Systems. *IEEE WESCON.*

SALEN, K. & ZIMMERMAN, E. (2003) *Rules of Play: Game Design Fundamentals*, MIT Press.

SAVIDIS, A., ZIDIANAKIS, M., KAZEPIS, N., DUBULAKIS, S., GRAMENOS, D. & STEPHANIDIS, C. (2008) An Integrated Platform for the Management of Mobile Location-Aware Information Systems. *Pervasive Computing.* Sydney, Australia.

SCAFI, A. (1999) Mapping Eden: Cartographies of the Earthly Paradise. IN COSGROVE, D. (Ed.) *Mappings.* Reaktion Books.

SCHARL, A. & TOCHTERMANN, K. (2007) *The Geospatial Web - How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society,* London, Springer.

SCHEIBLE, J. & TUULOS, V. (2007) *Mobile Python: Rapid prototyping of applications on the mobile platform*, Wiley.

SCHILIT, B. N., ADAMS, N., GOLD, R., TSO, M. M. & WANT, R. (1993) The PARCTAB mobile computing system. *Workshop on Workstation Operating Systems.* Napa, CA.

SCHILIT, B. N., ADAMS, N. & WANT, R. (1994) Context-Aware Computing Applications. *Workshop on Mobile Computing Systems and Applications.* Santa Cruz, California.

SCHILIT, W. N. (1995) A system architecture for context-aware mobile computing. New York, NY, Columbia University.

SCHLIEDER, C., KIEFER, P. & MATYAS, S. (2006) Geogames: Designing Location-Based Games from Classic Board Games. *IEEE Intelligent Systems,* 21**,** 40-46.

SCHNÄDELBACH, H., EGGLESTONE, S. R., REEVES, S., BENFORD, S., WALKER, B. & WRIGHT, M. (2008) Performing Thrill: Designing Telemetry Systems and Spectator Interfaces for Amusement Rides. *CHI.* Florence, Italy.

SCHNEIDER, G. & GREVING, M. (2005) MOSES - The Mobile Service and Exploration System. *Pervasive Computing (Video Sessions).* Munich, Germany.

SCHÖN, D. S. (1983) *The Reflective Practitioner: How Professionals Think in Action (Paperback edition 1991)*, Ashgate.

SCHRAMM, W. L. (1954) How Communication Works. *The Process and Effects of Mass Communication.* The University of Illinois Press.

SEAGER, W. & FRASER, D. S. (2007) Comparing Physical, Automatic and Manual Map Rotation for Pedestrian Navigation *CHI.* San Jose, California.

SEGALL, B. & ARNOLD, D. (1997) Elvin has left the building: A publish/subscribe notification service with quenching. *AUUG.* Brisbane, Australia.

SHANNON, C. E. (1948) A Mathematical Theory of Communication. *Bell System Technical Journal,* 27**,** 379-423.

SHNEIDERMAN, B. (1996) The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. *IEEE Symposium on Visual Languages.*

SHNEIDERMAN, B. (2002) *Leonardo's laptop: human needs and the new computing technology*, MIT Press.

SIEBOLD, J. (1998) "Cartographic Images Homepage" [Online] [Verified: 14 August 2009.] http://www.henry-davis.com/MAPS/.

SMART, J., HOCK, K. & CSOMOR, S. (2005) *Cross-Platform GUI Programming with wxWidgets*, Prentice Hall.

SMITH, G. (2008) *Tagging: People-powered Metadata for the Social Web*, New Riders Press.
SMITH, I., CONSOLVO, S., LAMARCA, A., HIGHTOWER, J., SCOTT, J., SOHN, T., HUGHES, J., IACHELLO, G. & ABOWD, G. D. (2005) Social Disclosure Of Place: From Location Technology to Communication Practices. *Pervasive.* Munich, Germany.
SNOW, J. (1855) *On the Mode of Communication of Cholera (2nd edition),* London, John Churchill.
SÖDERLUND, T. (2005) "Stories from the Frontier of Mobile Gaming" [Online] [Verified: 14 August 2009.] http://web.archive.org/web/20070813185129/http://www.gamesandstorytelling.net/Soderlund-lecture-2005.html.
STAHL, C. & HAUPERT, J. (2006) Taking Location Modelling to new Levels: A Map Modelling Toolkit for Intelligent Environments. IN M.HAZAS, J.KRUMM & STRANG, T. (Eds.) *2nd International Workshop on Location- and Context-Awareness.*
STENROS, J., MONTOLA, M., WAERN, A. & JONSSON, S. (2007) "IPerG Deliverable D11.8 Appendix C: Momentum Evaluation Report" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D11.8-Appendix-C-Momentum-Evaluation-Report.pdf.
STERCKEN, M. (2008) "Rezension von: Hartmut Kugler: Die Ebstorfer Weltkarte. Kommentierte Neuausgabe in zwei Bänden, Berlin: Akademie Verlag 2007" [Online] [Verified: 14 August 2009.] http://www.sehepunkte.de/2008/05/12698.html.
STEWART, S. (2007) I Love Bees. IN BORRIES, F. V., WALZ, S. P. & BÖTTGER, M. (Eds.) *Space Time Play.* Birkhäuser.
STRUNK, W. & WHITE, E. B. (1962) *The Elements of Style*, Macmillan.
STRUPPEK, M. & WILLIS, K. S. (2007) Botfighters: A game that surrounds you. IN BORRIES, F. V., WALZ, S. P. & BÖTTGER, M. (Eds.) *Space Time Play.* Birkhäuser.
SUCHMAN, L. A. & TRIGG, R. H. (1991) Understanding Practice: Video as a Medium for Reflection and Design. IN GREENBAUM, J. & KYNG, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates.
SUROWIECKI, J. (2005) *The Wisdom of Crowds*, Anchor Books.
TENNENT, P. & CHALMERS, M. (2005) Recording and Understanding Mobile People and Mobile Technology. *First International Conference on e-Social Science.* Manchester, England.
THOMAS, B., KRUL, N., CLOSE, B. & PIEKARSKI, W. (2002) Usability and Playability Issues for ARQuake. *International Workshop on Entertainment Computing.* Tokyo, Japan.
THORESEN, K. (1993) Principles in Practice: Two Cases of Situated Participatory Design. IN SCHULER, D. & NAMIOKA, A. (Eds.) *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates.
TORGE, W. (2007) *Geschichte der Geodäsie in Deutschland*, de Gruyter.
TUFTE, E. R. (1983) *The Visual Display of Quantitative Information,* Cheshire, CT, Graphics Press.
VARSHAVSKY, A., CHEN, M., LARA, E. D., FROEHLICH, J., HAEHNEL, D., HIGHTOWER, J., LAMARCA, A., POTTER, F., SOHN, T., TANG,

K. & SMITH, I. (2006) Are GSM phones THE solution for localization? *IEEE Workshop on Mobile Computing Systems and Applications.* Semiahmoo Resort, Washington.

VREDENBURG, K., MAO, J.-Y., SMITH, P. W. & CAREY, T. (2002) A Survey of User-Centered Design Practice *CHI.* Minneapolis, Minnesota.

WAERN, A. (2008a) "IPerG Deliverable D14.5 Appendix: Gamecreator Tutorial (Showcase - Boxed Pervasive Games)" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D14.5_a_a.pdf.

WAERN, A. (2008b) "IPerG Deliverable D14.6: Final Report (Showcase - Boxed Pervasive Games)" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D14.6.pdf.

WAERN, A., BENFORD, S., GOETCHERIAN, V., ÅKESSON, K.-P., SÖDERLUND, T., BJÖRK, S., MÄYRÄ, F., HOLOPAINEN, J., SCHÄFER, L. & GHELLAL, S. (2004) IPerG Position Paper. *PerGames Workshop.* Vienna, Austria.

WAERN, A., LINDT, I., WETZEL, R. & ÅKESSON, K.-P. (2008) "IPerG Deliverable D14.5: Final version of boxes and the internally produced game (Showcase - Boxed Pervasive Games)" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D14.5.pdf.

WAERN, A. & STENROS, J. (2007) "IPerG Deliverable D11.7: Game Design, Momentum (Work package 11: Enhanced Reality Live Role-playing)" [Online] [Verified: 14 August 2009.] http://iperg.sics.se/Deliverables/D11.7-Game-Design-Momentum.pdf.

WAGNER, F., SCHMUKI, R., WAGNER, T. & WOLSTENHOLME, P. (2006) *Modeling Software with Finite State Machines: A Practical Approach*, Auerbach Publications.

WALZ, S. P., BALLAGAS, R., BORCHERS, J., MENDOZA, J., KRATZ, S., WARTMANN, C., FUHR, C., TANN, M. J., SHIN, D. Y., HAMEED, B., BARDOS, L. & HOVESTADT, L. (2006) Cell Spell-Casting: Designing a Locative and Gesture Recognition Multiplayer Smartphone Game for Tourists. *PerGames Workshop.* Dublin, Ireland.

WANT, R. (2006) An Introduction to RFID Technology. *IEEE Pervasive Computing,* 5**,** 25-33.

WEAL, M. J., HORNECKER, E., CRUICKSHANK, D. G., MICHAELIDES, D. T., MILLARD, D. E., HALLORAN, J., ROURE, D. C. D. & FITZPATRICK, G. (2006) Requirements for In-Situ Authoring of Location Based Experiences. *MobileHCI.* Helsinki, Finland.

WEBSTER (1961) *Webster's Third New International Dictionary of the English Language,* Springfield, G. & C. Merriam Company.

WEIDMANN, B. (2009) Telephone conversation between Leif Oppermann and Brigitte Weidmann, project manager of Regensburg Experience (08. January 2009), cited with permission.

WEISER, M. (1991) The Computer for the 21st Century. *Scientific American,* 265**,** 66-75.

WEISER, M. (1994) "Building Invisible Interfaces (UIST 1994 Keynote talk)" [Online] [Verified: 14 August 2009.] http://www.ubiq.com/hypertext/weiser/UbiHome.html.

WEISMAN, J. (1998) The stories we played: building BattleTech and Virtual World. IN DODSWORTH, C. (Ed.) *Digital Illusion: Entertaining the Future with High Technology.* Addison-Wesley.

WETZEL, R., LINDT, I., WAERN, A. & JOHNSON, S. (2008) The Magic Lens Box: Simplifying the Development of Mixed Reality Games. *3rd International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA).* Athens, Greece.

WINOGRAD, T. & FLORES, F. (1986) *Understanding Computers and Cognition: A New Foundation for Design*, Addison-Wesley.

WOZNICKI, K. (2002) "Telepolis: Alle Handys dieser Zelle vernetzt Euch!" [Online] [Verified: 14 August 2009.] http://www.heise.de/tp/r4/artikel/11/11918/1.html.

WRIGHT, M., OPPERMANN, L. & CAPRA, M. (2007) Visualizing Data Gathered by Mobile Phones. IN DODGE, M., TURNER, M. & MCDERBY, M. (Eds.) *Geographic Visualization: Concepts, Tools and Applications.* Wiley.

WYNN, E. (1991) Taking Practice Seriously. IN GREENBAUM, J. & KYNG, M. (Eds.) *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates.