Extending Authoring Tools for Location-Aware Applications with an Infrastructure Visualization Layer

Leif Oppermann[1], Gregor Broll[2], Mauricio Capra[1], Steve Benford[1]

[1] Mixed Reality Laboratory, University of Nottingham
Wollaton Road, Nottingham, NG8 1BB, UK
{lxo,mxc,sdb}@cs.nott.ac.uk
[2] Embedded Interaction Research Group, University of Munich
Amalienstraße 17, 80333 Munich, Germany
gregor@hcilab.org

Abstract. In current authoring tools for location-aware applications the designer typically places trigger zones onto a map of the target environment and associates these with events and media assets. However, studies of deployed experiences have shown that the characteristics of the usually invisible ubiquitous computing infrastructure, especially limited coverage and accuracy, have a major impact on an experience. We propose a new approach in which designers work with three layers of information: information about the physical world, information about digital media, but also visualizations of ubiquitous infrastructure. We describe the implementation of a prototype authoring tool that embodies this approach and describe how it has been used to author a location-based game for mobile phones called Tycoon. We then outline the key challenges involved in generalizing this approach to more powerful authoring tools including acquiring and visualizing infrastructure data, acquiring map data, and flexibly specifying how digital content relates to both of these.

Introduction

As location-aware applications continue to emerge and mature so attention increasingly turns to the question of how best to author and configure them. Applications as diverse as tourism, games, navigation and mobile information retrieval require the creation of content in which digital assets – images, sounds, text, video and graphics – are attached to locations in the physical world to be triggered by participants who enter, leave or dwell in them.

Dedicated authoring tools and techniques for location-aware applications have begun to emerge in response. All of the examples produced to date share a 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, locales or hotspots on top of this that are triggered according to a set of basic events such as participants entering and leaving them.

The Mediascape tool (see figure 1, left) from the Mobile Bristol project is one such tool in which designers can place different sized and shaped triggers on a map and associate these with a range of events. Mediascape [1] has been used to support a variety of location-aware applications including: Scape the Hood [2] in which users explored an area of San Francisco, triggering stories as they entered different city blocks and Riot! [3], an audio experience in which participants triggered spatialized sounds as they explored a large city-centre square.

A variant on this approach is the idea of Colour Maps (figure 1, right), in which artists directly paint location triggers onto a map, affording a high degree of flexibility in terms of their size and shape and also enabling them to use existing and familiar paint tools. This approach was used to support the projects Uncle Roy All Around You in which a group of artists defined a trail of text clues around a city for players to follow as part of an interactive performance [4], and Savannah in which groups of six children at a time role played being lions on a virtual savannah that appeared to be overlaid on their school playing field [5].

As a third example, Caerus [6], enables a designer to administer trigger areas that overlay a digital map. The user can add further maps, points of interest and multimedia streams.

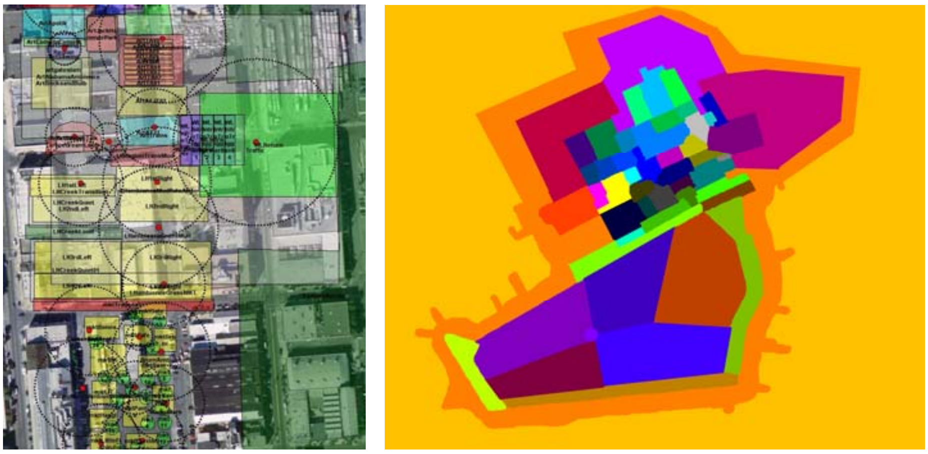

Fig. 1. The design of Scape the Hood in Mediascape (left), The design of Uncle Roy All Around You using Colour Maps (right)

Designing for the uncertainty of positioning and communications

However, published studies of location-aware applications suggesting that there is more to designing them than attaching media assets to a map. 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 communications and limited coverage and accuracy of positioning systems.

Studies of Can You See Me Now? (CYSMN), a game of chase in which online participants were chased through a 3D virtual model of a city by performers who, equipped with PDAs and GPS units, had to run through the actual city streets in order to catch them, showed that limited coverage and accuracy had a major impact on the game [7]. 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. The extent of the problem can be seen on in figure 2 which visualizes the characteristics of GPS and Wi-Fi over the course of a two hour performance on a peninsula in Rotterdam. The light colored areas correspond to locations where performers acquired a GPS fix and were able to transmit it back to the game server over Wi-Fi (i.e., where both Wi-Fi and GPS were available). Light blue shows high estimated GPS accuracy and light green lower accuracy. The dark areas represent areas where either the performers never ventured or some combination of GPS and/or Wi-Fi were not available. Given that the performers ventured widely, there were clearly many areas that were not playable, especially the central thoroughfares between the main buildings (which are shown as back rectangles – the surrounding black area is water where the performers never went). Furthermore, the availability and performance of GPS and Wi-Fi vary over time as objects move into and out of the environment and GPS satellites move over the sky, so that such pictures can in fact be highly dynamic.

This raises a fundamental question for location-aware authoring tools? What is the point of placing assets at physical locations where there is insufficient coverage or performance to reliably access them? Or put another way, how can we enable designers to taken account of the usually invisible ubiquitous infrastructure as well as the nature of the physical environment when authoring an experience. This is the key challenge addressed by this paper.

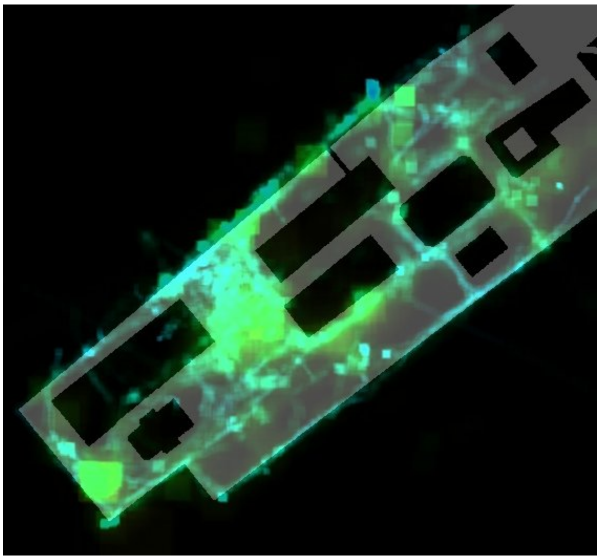

Fig. 2.: A visualization of GPS and Wi-Fi coverage and accuracy from CYSMN

Extending authoring tools to reveal the ubiquitous infrastructure

In our extended approach, we support the designer with knowledge of the physical world and the ubiquitous infrastructure at the same time. The authoring process now involves overlaying information characterizing communications and sensing systems on meaningful backgrounds that facilitate the orientation of the game designer. For geo-coded data this would be a map of the area, for time-stamped data this might be a timeline. For the scope of this paper we will limit the possibilities to interactively visualizing geo-coded data on maps.

Figure 3 summarizes the overall concept behind our authoring tool for location-aware applications. The tool enables designers to work with three layers of information: the *physical environment layer*, with representations of the target physical environment; the *infrastructure layer*, with representations of the ubiquitous 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 GIS data for the physical environment; recorded and predicted information about the coverage and accuracy of communications and sensing systems for the infrastructure layer; and triggers, hotspots, assets and events and the content layer.

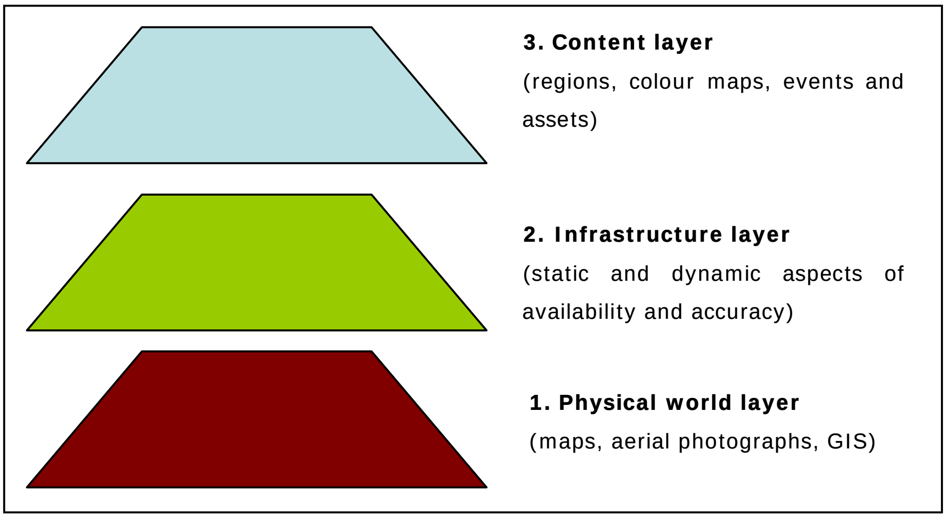

Fig. 3. Authoring Content Triggers based on Infrastructure Visualization

The key innovation here in relation to previous tools is the insertion of the infrastructure layer. This layer helps the designer to understand the spatial spread of otherwise invisible infrastructure (e.g. GSM network, Wi-Fi, GPS) at authoring time.

Designing the authoring tool

Our new authoring tool is designed to work with a large number of samples that measure the characteristics of the underlying infrastructure. Due to potential screen space constraints we decided to implement the visualizations in an interactive way. We have adopted a 'detail-on-demand' approach to visualization following Ben Shneiderman's Visual Information Seeking Mantra: "Overview first, zoom and filter, then details-on-demand" [8].

wxWidgets[1], OpenGL[2] and SDL[3] are used as a platform independent base-system, providing the necessary user input and output channels for our interactive application as shown in the data-flow diagram in figure 4.

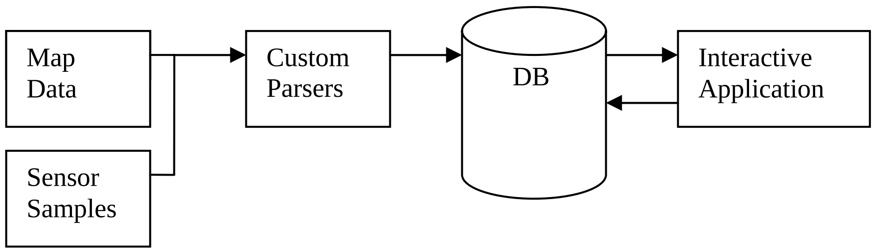

Fig. 4. Data-Flow Diagram

For our maps, we used closed source vector-data from Ordnance Survey[4] and parsed it with FWTools[5] before rendering it with Mapserver[6] to a raster file-format. The sensor samples have been collected using a Place Lab compatible program and parsed from a flat-file storage using our custom parser. All input data for the *physical world layer* and *infrastructure layer* is currently accessible using abstract C++ interfaces. The overall architecture leans towards utilizing web services in the future, which would support the use of different programming languages across different computers for different interaction tasks.

Physical world layer

This layer provides the user with a meaningful background in regard to the target environment. For the current prototype we limited the background to 2D maps in Cartesian coordinates.

Reading common file-formats, the tool also allows for using digitized maps or orthophotos. The loaded bitmaps can then be geo-coded by supplying two points, both in the image coordinate-space and the real world coordinates. Ideally this would be the lower-left and upper-right coordinates of the map.

---

[1] http://www.wxwidgets.org/
[2] http://www.opengl.org/
[3] http://www.libsdl.org/
[4] http://edina.ac.uk/digimap/description/
[5] http://fwtools.maptools.org/
[6] http://mapserver.gis.umn.edu/

Infrastructure layer

The infrastructure layer contains samples of network information tagged with time and/or position. It is projected in the same way as the background map and thus the samples are put into the right context when the two layers are stacked up. This enables the user to reason about the infrastructural data, e.g. decide about whether an area has been sampled densely enough. Figure 5 (middle) shows a generally well sampled area which could eventually need some more data on the northern and western side of the area. Each of the green dots in the image represents a sample (in this case of GSM cell data that has been geo-referenced using GPS).

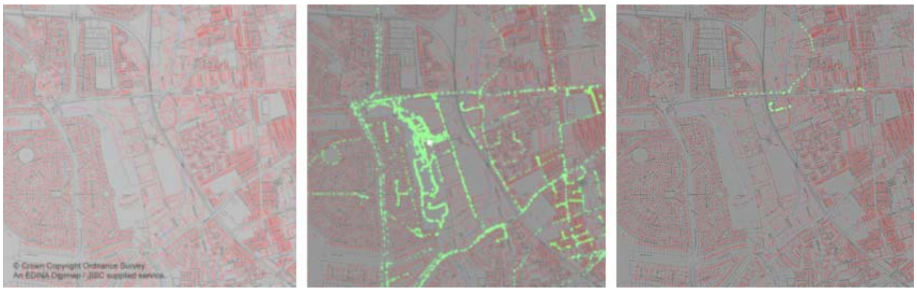

Fig. 5. A map of Nottingham, UK (left), overlaid with GSM sensor-data (middle), view interactively filtered by Cell ID (right)

The samples are referenced by unique ID-strings. If the ID-strings themselves should be used to trigger content in the final experience they will have to be reproducible on the target device by accessing the infrastructure. Examples of sources for such possible ID-strings are given in the following table.

Table 1. Reproducible ID-strings for different infrastructures

GSM	Full Cell ID (MCC_MNC_LAC_CI)
Wi-Fi	BSSID (Mac Address of Access Point)
GPS	Location (latitude/longitude)

Content layer

The interactive visualization itself only acts as a hint to the user. The actual content is defined and refined in the content layer. Here the author can make informed decisions about where to place assets, where the experience should take place or which observed sensors could serve as triggers for events. For example the right image in figure 5 shows the infrastructure filtered to show the possible spread of a single GSM cell across a region of space. This cell could arguably be used to trigger events which have a notion of location but do not require exact latitude/longitude coordinates, e.g. traffic info for the surrounding area or virtual locations in a mobile phone game. The following section presents a game design that works using this principle.

Authoring Tycoon – a game for mobile phones

In order to demonstrate our proposed approach to authoring ubiquitous experiences we have created Tycoon, a location-based multiplayer trading-game for mobile phones. The gameplay in Tycoon is driven by players entering and leaving different mobile phone cells which are associated with different game resources. Choosing which resources to associate with specific cells is therefore a key aspect of designing the game and in turn, requires an understanding of how the cells are distributed across physical space.

The basic idea behind Tycoon's gameplay is a simple producer-consumer-cycle. Players have to roam a designated gaming area (e.g. a park or the centre of a city), collect resources and trade them for objects that earn them certain amounts of credits. The objective of the game is to collect as many credits as possible in order to win the game. Tycoon uses the different physical GSM-cells of a service-provider network within the gaming area and maps one or several of them to virtual locations in the game that represent producers, consumers or neutral areas (figure 6). The game uses the metaphor of a wild-west scenario to communicate its basic mechanisms of collecting resources from producer-locations which are called "mines" and using them to buy objects from consumer-locations which are called "brokers" and have the names of towns or counties in California, USA.

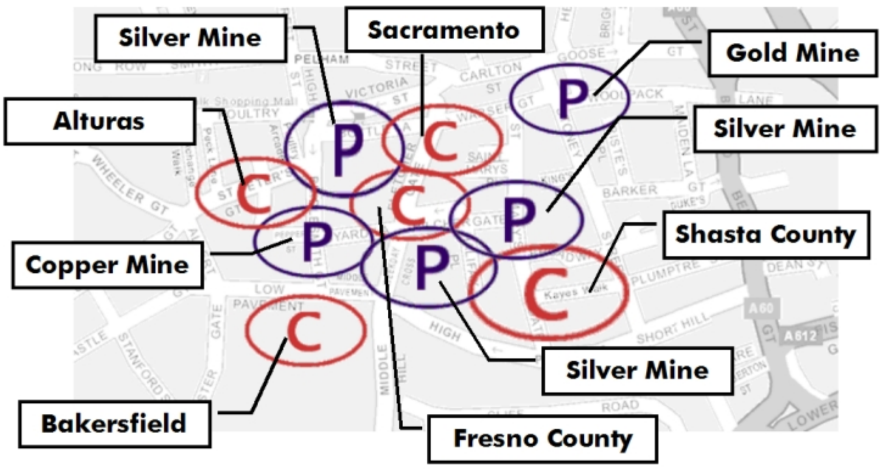

Fig. 6. Example of a Tycoon gaming-area with its mapping of physical locations to consumers (C) and producers (P)

During the game, players have to explore the gaming area and discover the locations of brokers and mines (figure 7, left). After their discovery, players can collect local resources from mines by staying in them for a certain period of time (figure 7, middle). Each mine provides an unlimited amount of one local resource in the game – gold, silver or copper – which differ in their value and the time it takes to collect them. They are called "local resources" since players collect them independently from each other and do not compete for them.

Brokers on the other hand each keep a list of unique global objects, e.g. different buildings in towns or estates in counties (figure 7, right). Players can see these lists after having discovered the location of the corresponding broker. The number of global objects in the game is limited, each one is unique (e.g. there is only one hotel in the city of Bakersfield) and players compete against each other for claiming them. Each object has a certain price in different local resources and a corresponding value of credits. Players can claim global objects from a broker's list after entering his area and earn the objects' credits provided they can afford their price and the desired objects are still available.

Fig. 7. Different screen images from Tycoon's interface: location discovery (left), collecting Resources (middle), claiming objects (right)

Seams in mobile applications

Ubiquitous Computing systems and especially applications for mobile communication and navigation are susceptible to the effects of so-called seams in the underlying technical infrastructure. Seams can be seen as deviations in actual use from a notional ideal of technological continuity or uniformity, including discontinuities in technologies themselves and discontinuity between what actually happens and what the system observes [9]. They are mostly caused by technical limitations and constraints in the heterogeneous infrastructure of a system. They may reveal their effects on the user experience, showing themselves as uncertainties, ambiguities or inconsistencies when users interact with a supposedly seamless system. Common examples for seams in mobile applications are deviations in positioning, unavailability of network services or patchy network coverage. As such, they may come to the users' attention as interruption and loss of services or uncertainty about current position.

Previous research has identified a range of different strategies for dealing with these kinds of seams when designing ubiquitous experiences [10]. These include improving or deploying the underlying technologies so as to *remove* them; carefully designing the experience to *hide* their worst effects from users; carefully *managing* the experience from behind the scenes; *revealing* the presence of seams to users so that they can adapt to them; and even deliberately *exploiting* the seams as part of the design (the approach of 'seamful design' as described in [9]). All of these approaches require that the designer become aware of the presence of seams across their chosen physical environment, which of course is the purpose behind our proposal to add a visualization layer to current authoring tools.

Tycoon follows the approach of exploiting seams that arise from the use of GSM cells as a basis for positioning. Users are normally unaware of their current GSM cell or location while using their mobile phones, as this information is mostly hidden by the system and the handover between adjoining cells is handled seamlessly [11]. While seamless design usually neglects this information, our seamful approach to mobile games reveals the presence of individual mobile phone cells and uses them to drive the overall user experience.

Since location-awareness is crucial for Tycoon's gameplay, we use information from GSM cells for positioning which is sufficient for our approach but also implies several constraints and seams. Figure 8 gives an example of GSM cell propagation and coverage which depends on many factors, is very dynamic and has irregular, fluctuating shapes. Cells do not have fixed boundaries and do not share exact borders with adjoining cells but often overlap with them. These properties can become a problem for mobile applications whose behaviour is dependent on exact positioning.

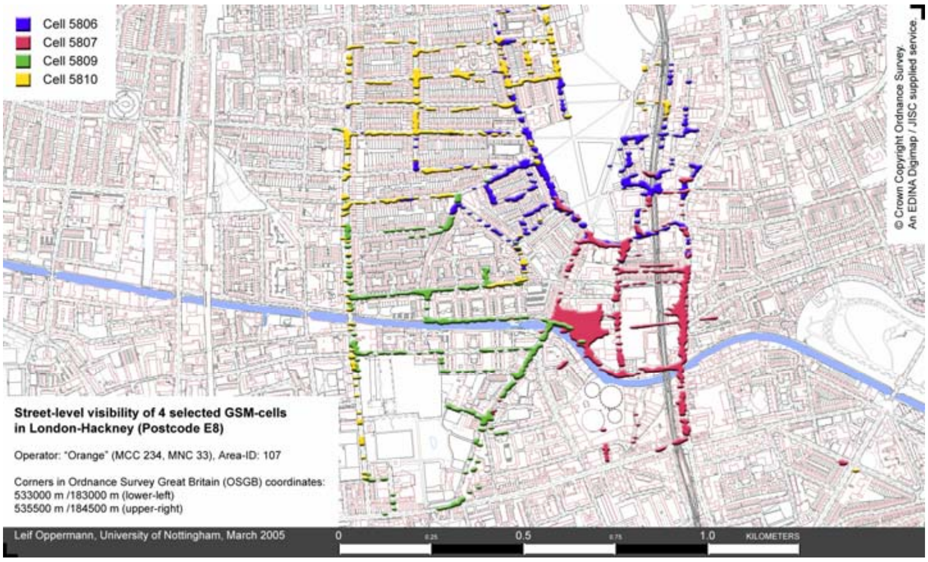

Fig. 8. GSM cell coverage and propagation indicated by samples of cell-IDs and their GPS-position in London

Being a mobile multiplayer game, Tycoon's gameplay is driven by the competition between several players with mobile phones. Tycoon is implemented using a client-server-architecture and players need continuous access to globally shared data on the game-server in order to compete successfully with each other. Therefore mobile Internet-access is most convenient and almost indispensable for accessing the global server and synchronising its shared game-state with mobile clients. A seamless approach would try to guarantee as little inconsistency between the global and all local game-states as possible by frequent updates between the mobile clients and the game-server. Whenever a mobile client locally changes the game-state, e.g. by claiming a globally shared object, the game-server would either have to route this change to all clients or they would periodically have to ask the server for updates on the latest changes. Either way, a considerable volume of expensive GPRS-traffic would be generated. Given the definition of seams in [9], these expenses might also be considered to be self-made, artificial seams, since they are neither caused by technical constraints nor show themselves as uncertainties or ambiguities. But as costly expenses for GPRS-traffic still constrain the uniformity and continuity of mobile applications, we want to treat them as a significant seam.

The frequency of updates between server and mobile clients in order to maintain a globally correct game-state is tightly linked to the probability of data inconsistencies which occur as a consequence of insufficient synchronisation. Inconsistencies emerge when individual clients update data that is shared with other clients through a common server. When a Tycoon client synchronises its locally altered data with the shared server in order to update the globally shared game-state, e.g. after claiming an object, local copies of that data on other clients become inconsistent with the updated game-state and have to be updated accordingly.

Augmenting mobile gameplay with seamful design

Tycoon's game-logic and interface are designed to incorporate and exploit the seams of dynamic cell positioning, expensive internet-access and data-inconsistencies in order to enrich its gameplay.

Tycoon's gameplay is location-dependent and applies positioning through the recognition of unique GSM cell identifiers in a gaming area. As mentioned above, this method of positioning is not very accurate and can comprise considerable deviations. In order to improve navigation and orientation for the user, the main screen of Tycoon's interface always displays the name of the current location, along with the amount of collected resources and earned credits (figure 9, left). When a player changes from one cell into another, an alert is triggered and a notification about them entering a new location is displayed (figure 9, middle). Afterwards the name of the new location is shown on the main screen until the next change (figure 9, right).

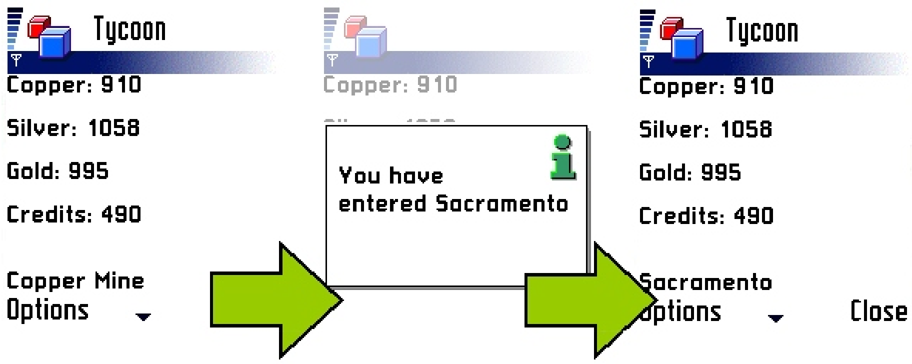

Fig. 9. Tycoon's alert-mechanism for location-visualization

This alert-mechanism is more flexible than a static map of the gaming area, reveals information about fluctuating coverage and propagation of GSM cells, and improves the visualization of locations and their boundaries. Thus the design of Tycoon is intended to decrease players' uncertainties about their position, improve their interaction with the seam of dynamic cell propagation and use it as an important part of game logic.

Instead of providing players with a complete map of the area showing them exactly where to find mines and brokers, we want them to start the game by having to explore the area, gather their own knowledge about it and discover mines, brokers and their locations themselves. This way, players are encouraged to use their spatial knowledge about the gaming-area in order to adopt their own strategies of moving between cells and finding the most efficient tactics of which resources are needed to buy which available objects and where to find them in nearby mines.

During this process of appropriation, players can exploit positioning ambiguities caused by dynamic cell propagation more effectively when they find an area where adjoining cells overlap and easily flip after some time without moving. This issue is especially interesting for designing the gaming-area and assigning mines and brokers to different locations.

Considering locations according to the game-design

Since the game is running on a mobile phone, it is easy to assume the phone visits different locations given that people move over time. We wanted the game to be playable during the players' daily life without neither interfering too much with what they are doing nor requiring to them to visit special places. We therefore decided that locations in the game should be mapped to places that people come across anyway.

Our game-design defines that Tycoon is going to be played by the inhabitants of a city over a long summer weekend to facilitate discovering places. Assuming that people are doing their shopping at weekends we choose the city-center and shopping malls to be the brokers. Parks and recreational areas are defined to be the mines.

The next section shows how the current prototype tool can be used to author Tycoon for these locations.

Authoring for locations in Tycoon

This section shows our prototype design tool can be used to author Tycoon. We show how the infrastructure visualization layer helps to make a GSM based location based game. Using the tool we define cell ID based regions that serve as triggers for virtual locations in the game. In the following, the presented GSM infrastructure data has been collected over a period of two days using one phone and a GPS receiver.

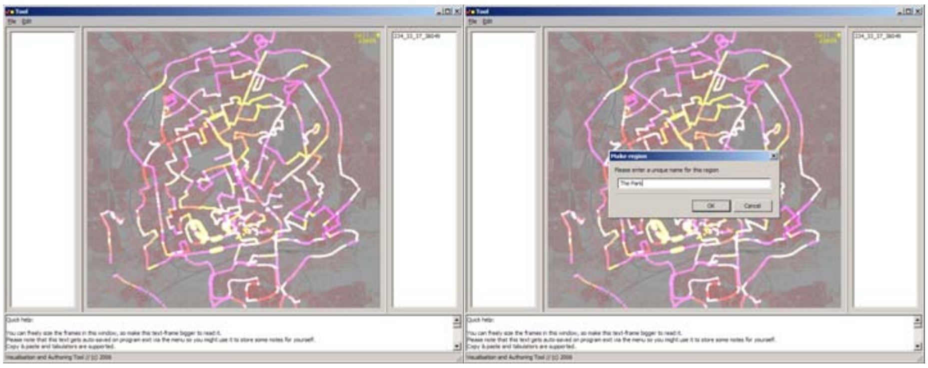

Fig. 10. Initial graphical definition of a region

Figure 10 above shows two screenshots of the authoring tool with a 6 x 6 km view of a city in England overlaid with infrastructure data from the Orange GSM network. Thousands of coloured dots represent where a combination of GSM and GPS data has been sampled. Each dot represents one sample and shows that a particular cell ID has been seen at this specific position. Dots with the same colour represent the same ID-string (here: cell ID). By hovering the mouse over the visualization the user can see which ID is next to the mouse pointer and where else that ID has been observed. The samples with the current ID are highlighted by jumping yellow dots (bold dots in these screenshots). Having animated features allows the user to easily distinguish between samples that belong to the current selection and those that do not. The current selection of one or more IDs is shown in the listbox to the right of the window. The user can store this selection as a region by giving it a unique name (figure 10, right). The region then appears in the region listbox to the left (see figure 11, left).

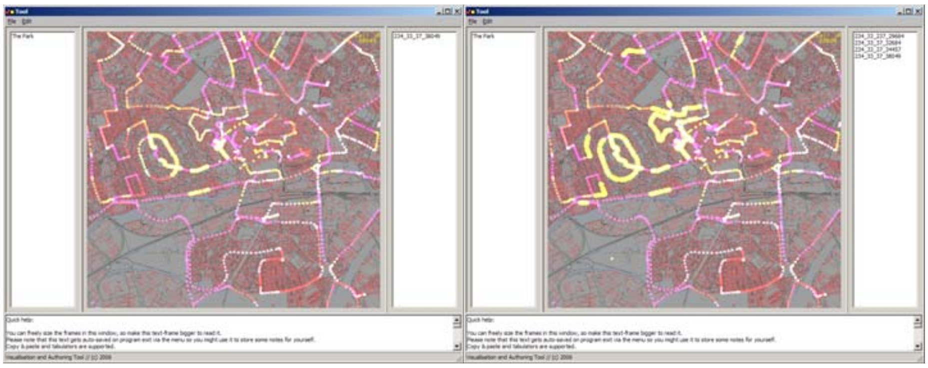

Fig. 11. Close up (left) and refined definition (right)

By zooming the view, the user can look over the details of their selection. Figure 11 shows the same selection as before but with a more close-up view. In this example, the user wanted to select the area around the park in the left part of the image. Although the area seems to be well covered by just one cell, the visualization certainly shows samples with different colours nearby. The user can now modify his selection in the zoomed view and apply any modifications should they be needed.

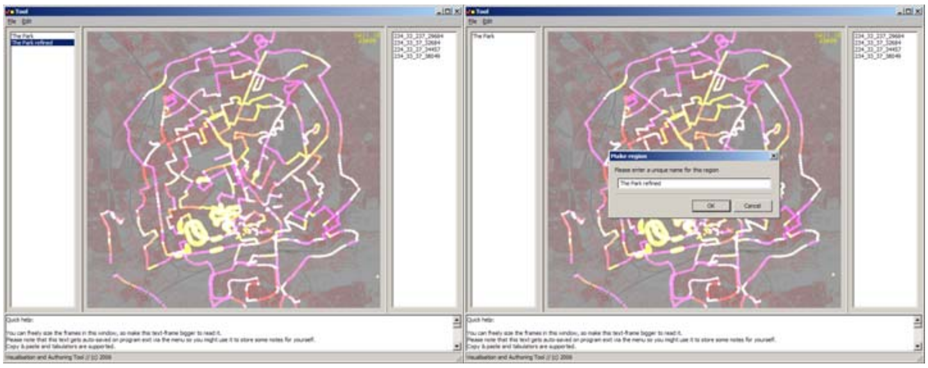

Fig. 12. Zoomed out for overview (left) and save refined region (right)

Figure 12 (left) shows the refined region definition that covers more of the park. There are now 4 Cell IDs in the selection. By zooming out, the selection can be visually verified once again before saving it as a region if satisfied (figure 12, right).

The resulting region definitions (textual ID-strings) can then be post-processed and translated into an application specific format to serve as location triggers in the game.

Deploying the game content to the clients

The Tycoon game client runs on Symbian phones and uses a client-server architecture. The server uses a standard LAMP[7] software installation. The MySQL database on the server is filled by a PHP script parsing the XML region definitions (see figure 13). The same XML-file is also used by another PHP script to configure the clients when they log into the server.

```
<r>
      <rid>4</rid>
      <n>Sacramento</n>
      <t>broker</t>
      <lid>234_33_237_29678</lid>
      <lid>234_33_37_34457</lid>
      <lid>234_33_37_24949</lid>
</r>
```

Fig. 13. Example XML Content Definition (Excerpt)

Figure 13 shows an excerpt from the content definition. It defines the region number 4 called “Sacramento” to be a broker. The region will be triggered when a player’s phone is connected to one of the cells that construct the supplied ID-strings. Figure 14 shows the pr

[7] Lamp = Linux, Apache, MySQL, PHP/Perl/Python

#
#opagation of the regions from the XML file through the server to the clients.

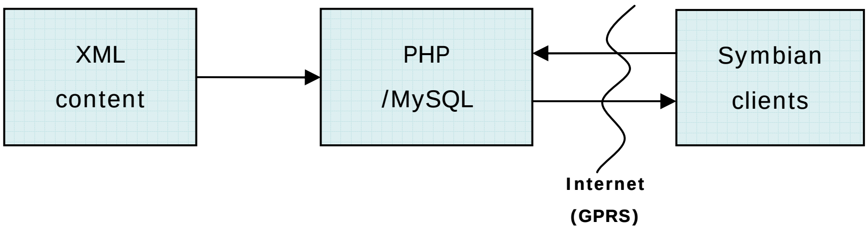

Fig. 14. Propagation of content from server to client

Challenges for further work

We have presented a prototype authoring tool that supports the designer of location-aware applications with interactive visualizations of ubiquitous infrastructure in addition to the typical map background. The prototype has been used to author a location aware game for mobile phones that does not require a GPS device for positioning. However, if we want to generalize our authoring approach and transform it into something that can be used anywhere in the world we need to address some key issues related to our three underlying layers: the maps, the infrastructure and the visualizations.

Issues with maps (physical world layer)

Maps form the background of the authoring view. They are not strictly required but do help give an understanding of the physical area in which an experience will take place and will therefore often be used.

There are two important data-sources for this layer: printed maps/photos and online-services/GIS. Existing printed maps or aerial photos can be used if the coordinates are known or can be measured. The image just needs to be digitized in this case and afterwards geo-referenced.

Online services can be used to acquire digital maps. The advantage of digital maps is that they are already precisely measured. This means that the overlay in the authoring tool will be as precise as possible. Unfortunately most of the services on the market are only available after the payment of a fee. Our visualization tool is based on Digimap data from Edina [12] due an existing subscription made by our university. Edina offers several products that could be added to the visualization as additional information layers, e.g. transport networks, boundaries, buildings, contours lines, etc.

Another (free) service we used is Google Earth, especially its KML markup-language [13]. The KML-language allows us to quickly plot the surveyed data onto a virtual globe and thus visually check the data for validity. The real image of buildings, roads or cities for example helps us to understand the context in which the data was collected.

However, one of the main issues for the future of our approach is scalability. This can bring some difficulties in the visualization process, especially in map acquisition across many different countries. We are conscious that our approach needs to be flexible to interoperate with web based services like *Web Map Service* and *Web Feature Services* as defined by the Open Geospatial Consortium [14]. Both services are Internet operated and return data and/or images of a specific area on request. Also Google Maps and Google Earth will continue to be helpful for what we are doing.

Another important point is to make our tool is able to communicate with GIS packages like the open source GRASS [15] and PostGIS [16] or commercial packages like Erdas Imagine [17] or ArcGIS [18]. More generally, we suggest that authoring tools for location based games are getting closer to Geographical Information Systems (GIS) and will soon begin to merge.

Issues with data collection (infrastructural layer)

Our main source of infrastructure data is self-collected. It has been initially obtained using the Placelab [19] stumbler software but we later moved to our own program called Pystumbler. The program takes a snapshot of the supported networks (i.e. GSM on the phone) about every second and also gets a GPS fix about every few seconds. The file-format is a well formed text-file which is compatible across platforms. We wrote a parser to support this file-format.

Pre-mapping an area using a network stumbler is time-consuming. It can take several hours to map an area in a dense city centre and several days to get a rough picture of the observed infrastructure in wider parts of a city. The ultimate goal of the stumble method would be to know how the characteristics of the infrastructure on every street. This is not feasible with only few people doing the data collection. For this reason this approach is not feasible for games that are to be played on a much larger, national or planetary scale without modifications. It is therefore interesting to see how the data collection process can be interwoven with the game-play or how data-collection can be archived as a by-product of other applications, e.g. games.

In addition to the more sophisticated stumbler setup using GPS we also created a less obtrusive long-term data collector that can run as a background task on a phone for a very long time without requiring any attention. It logs the cell ID once per minute and also every cell ID change up to a granularity of ten seconds. We have tested this software to track the movement of ten persons over the period of a month. The generated log-files contain universally valid cell adjacency information as well as subjective (i.e. only valid for the subject) patterns of movement.

Knowing that other people do similar things, it is also interesting to interoperate with projects like Wigle [20], Place Lab [21] or Mobilife [22] to utilize the data which has already been collected. Infrastructural data collection (also known as ‘wardriving’), especially for Wi-Fi networks has been done for many years. Some of the data is shared through online-databases. The data presented there is usually the averaged position per network ID which means that the data shows less detail than the self collected trails of stumbler data but is potentially still very useful as it would allow authoring for unknown areas if it can be trusted. A verification and modification (cleaning) process will also be needed later on.

Issues with the visualization

Choosing an appropriate visualization style for the data is the key question for any information visualization application. For our current prototype we deliberately decided to not abstract the data in any way (e.g. creating bounded regions from collections of individual sample points) but rather have tried to do as much as we can with the full data in its pure form (i.e. each sample visible on screen). We choose an interactive visualization because it allows the user to make quick visual queries into the data.

The current point visualization is however not enough. Many different views on the same data are possible and the tool should be flexible enough to allow those different views to be set up simultaneously. Not all visualizations will require a high-performance implementation in C++ and it should be possible to access the shared data using other programming languages like Java and Python as well. Also an advanced user should be able to interact with the data through a scripting interface for those occasions where the visualization alone is not sufficient.

Beyond this, we identify two major challenges for visualization. First is representing uncertainty. The collected data will be subject to uncertainties that designers may need to bear in mind. Positioning technologies such as GPS (that underpins the Tycoon data) are subject to error. Furthermore, data sets will be incomplete, requiring designers to interpolate between particular samples. Visualizations may need to convey uncertainty or be able to suggest possible interpolations. A second challenge lies in the time-varying nature of the ubiquitous infrastructure. Visualizations may also need to account for the ways in which coverage and accuracy vary over time as well as over space.

Conclusion

The idea that designers can author location-aware experiences by placing locales or hotspots on maps and other representations of physical spaces is already well established and forms the basis of several existing prototype tools. However, studies of previous experiences suggest that designers also need to be able to account for the characteristics of the usually invisible infrastructure of positioning and communication technologies, especially (but not only) their limited coverage.

We are therefore proposing that authoring tools for location-aware experiences be extended to include visualizations of the likely state of the infrastructure across a selected environment so as to guide designers as to how their intended experience might actually be experienced by end users in practice.

Essentially, we propose that future authoring tools should provide access to three broad layers of information (each of which might consist of several sub-layers): a physical world layer that describes the target physical environment; an infrastructure layer that represents the likely state of the infrastructure across this target environment; and a content layer that defines locales or hotspots that associate events or media assets with geographic locations.

We have produced an initial demonstration of how this general approach can assist the design of a location-based game, and have discussed key challenges that need to be met at each layer in order to develop it further. Our future work will of course involve addressing these challenges. It will also involve refining our current prototype authoring tool and working with external designers in order to elicit their more detailed requirements and to evaluate the longer-term potential of our approach through practical experience.

Acknowledgements

We gratefully acknowledge the support of IPerG, the European Project on Pervasive Gaming, project FP6-004457 under the European Commission's IST Programme (www.pervasive-gaming.org). We are also grateful for additional support from the Equator Interdisciplinary Research Collaboration which has been funded by the UK's Engineering and Physical Sciences Research Council (www.equator.ac.uk).

References

1. Hull, R., Clayton, B., Melamed, T.: Rapid Authoring of Mediascapes. In: Proc. Ubicomp (2004) 125-142
2. Mediascape. Scape the Hood.[Online]. Available: http://www.webtales.us/Scape1.html
3. Mediascape. Riot! - An Interactive Play for Voices.[Online]. Available: http://www.mobilebristol.com/QueenSq.html
4. Flintham, M.: Painting the Town Red: Configuring Location-Based Games by Colouring Maps. In: Proc. ACE (2005)
5. Benford, S., et al.: Life on the Edge: Supporting Collaboration in Location-Based Experiences. In: Proc. CHI (2005) 721-730
6. Naismith, L.: CAERUS Overview. The University of Birmingham (2004)
7. Crabtree, A., et al.: Orchestrating a Mixed Reality Game 'On the Ground'. In: Proc. CHI (2004) 391-398
8. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. In: Proc. IEEE Symposium on Visual Languages (1996) 336-343
9. Chalmers, M. And Galani, A. 2004. Seamful Interweaving: Heterogeneity In The Theory And Design Of Interactive Systems. In Proceedings Of The 2004 Acm Symposium On Designing Interactive Systems, August 2004, Cambridge, Massachusetts, Acm Press, 243-252.
10. Benford, S, Crabtree, A., Flintham, M., Drozd, A., Anastasi, R., Paxton, M., Can You See Me Now?, ACM Transactions on CHI, 13 (1), March 2006.
11. Willassen, S. Y.: A method for implementing mobile station location in GSM. Norwegian University (1998)
12. Edina. (2005) Digimap.[Online]. Available: http://edina.ac.uk/digimap/description/

13. Google. Google API.[Online]. Available: http://code.google.com/apis.html
14. Open Geospatial Consortium Adopted Document Baseline.[Online]. Available: http://www.opengeospatial.org/specs/?page=baseline
15. GRASS Development Team. (2005) Welcome to GRASS GIS.[Online]. Available: http://grass.itc.it/
16. Refractions Research. (2005) What is PostGIS?[Online]. Available: http://postgis.refractions.net/
17. Leica Geosystems LLC. (2004) IMAGINE Virtual GIS. Leica [Online]. Available: http://gis.leica-geosystems.com/Products/files/vgis_ds.pdf
18. ESRI. (2006) ArcGIS Family of Products.[Online]. Available: http://www.esri.com/software/arcgis/index.html
19. LaMarca, A., et al.: Place Lab: Device Positioning Using Radio Beacons in the Wild. In: Proc. Pervasive Computing (2005) 116-133
20. Wigle. [Online]. Available: http://www.wigle.net/
21. Placelab. [Online]. Available: http://www.placelab.org/
22. MobiLife. [Online]. Available: http://www.ist-mobilife.org/