Exploiting Seams in Mobile Phone Games

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

[1] Institute for Informatics, Embedded Interaction Research Group,
Amalienstr. 17, 80333 München, Germany
gregor@hcilab.org
[2] Mixed Reality Laboratory, The University of Nottingham,
Nottingham, NG8 1BB, United Kingdom
{sdb, lxo}@cs.nott.ac.uk

Abstract. Seamful design is a novel approach to Ubiquitous and Mobile Computing systems that reveals and exploits inevitable technical limitations and boundaries as well as their effects on seamless interaction rather than hiding them. In this paper we want to introduce its general ideas, point out common seams on the mobile phone platform and show how people might accommodate to seams and use them for their own advantage as useful features of an application. We apply the approach to seamful design to our own location-based mobile game called Tycoon and incorporate different seams into its gameplay for better interaction and usability through the exploitation and appropriation of these seams.

1 Introduction

Despite the efforts of seamless design, integration and interaction, Ubiquitous Computing systems still have technical limitations and boundaries. Especially devices and applications for mobile communication and computing suffer from patchy network coverage, fluctuating signal strength, deviations in positioning and the generally limited resources provided by mobile devices. Their users experience these limitations indirectly as sketchy and slow mobile internet access, variations in the quality of speech transmission or even loss of a connection in cold spots in network coverage.

The concept of seams and seamful design is a rather new approach to Ubicomp systems whose design is usually ruled by the notion of seamlessness. Both seamless and seamful design are inspired by Mark Weiser's vision of Ubiquitous Computing and the idea of "literally visible, effectively invisible" [1] tools that don't intrude on the user's consciousness but let him focus on a task and not the tool itself (see [1]).

Chalmers and MacColl point out that "this notion of invisibility has been translated into requirements for seamless integration of computer system components, as well as the interactions supported by those components" [2]. As a result, seamless design knits the different heterogeneous and distributed components of Ubicomp systems tightly together in order to hide the complexity of its infrastructure. It comprises both seamless integration of different components as well as seamless interactions with them through the entity they form. Adjoining GSM-cells and their infrastructure for example form a ubiquitous system that allows mobile communication. The invisible handover between different cells provides seamless interaction with this system through its different cells and components when people use their mobile phones.

Seamful design on the other hand is supported by Mark Weiser's recommendation of seamful systems with "beautiful seams" which don't sacrifice the uniqueness of its heterogeneous components for the goal of a "lowest common denominator" (see [3]). Instead different parts of a system are seamfully integrated and still provide seamless interaction with the whole system while retaining their individual features.

We want to turn our attention to such seams in mobile applications and show how to take advantage of them through seamful design. The "Bill"-game [4] already showed how to turn the seams in a mobile application supporting GPS and 802.11 into helpful features of a game, how seamful design can drive its gameplay and how players exploit them to succeed in a game. We would like to extend the ideas of seamful design to applications for mobile phones, explore which seams occur on this platform and apply the approach to seamful design to our own seamful multiplayer-game called "Tycoon". Thus we try to evaluate the benefits of exploiting seams on this platform – especially in cell-based positioning and network coverage - for better gameplay, usability, interaction and appropriation.

In the following sections we give a short introduction to our understanding of seams and seamful design, provide examples of seams in mobile applications and describe how we try to incorporate them into the design for our own seamful game for mobile phones.

2 Seams and Seamful Design

Just as seamless design refers to both the tight joints between infrastructure-components and the overall experience of smooth, seamless interaction, the idea of seams comprises the technical cracks and bumps in these joints as well as their effects on the users' experience. 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. They are mostly caused by technical limitations and constraints of the underlying infrastructure. They come to the users' attention as these interact with a supposedly seamless system and reveal themselves as uncertainties, ambiguities or inconsistencies. Seams are most common in applications for mobile navigation and communication where technical constraints include inaccurate positioning, sketchy internet-access, local variations in signal strength (e. g. in tunnels), delays in GSM networks or patchy network coverage. There they cause uncertainty about the current position, disturbance or even temporary loss of mobile communication signals.

Contrary to seamless design, which tries to hide these constraints in Ubicomp systems with costly investments in better and more reliable infrastructure-technology, seamful design embraces these inevitable limitations, reveals them, increases the awareness for them and exploits its otherwise neglected yet useful information for better interaction-design and user experience. In order to accomplish the goal of "seamless interaction but seamful technology" Oulasvirta outlines seamful design as “understanding which seams are important”, “presenting seams to users” and “designing interactions with seams” [5].

People often accept seams, adapt to them and even exploit them. They might use their knowledge about varying signal strength or sketchy network access in certain areas as an excuse for interrupting or dismissing a phone-call since such technical constraints are commonly accepted reasons for doing that. Seamful design now tries to support and model this process by revealing and presenting information about seams to users and thereby increasing their awareness for the influence technology can have on user-interaction. Giving users the opportunity and freedom to explore seams and exploit them in new ways may ultimately lead to the more general concept of designing for appropriation. This aspect of seamful design allows users to interact with seams individually, take advantage of the gaps and limitations in Ubicomp infrastructures and develop new patterns of behaviour around them in ways, that were not considered during the initial design of the system.

3 Seams in Mobile Phone Applications

We want to exploit three seams in mobile phone applications which we consider to be characteristic for this platform: dynamic cell-coverage, expensive internet-access and data-inconsistencies. These seams will guide the seamful design for Tycoon - our mobile seamful game.

Mobile phone users are usually unaware of the cell or location area their phone is using, since the handover between different cells is handled seamlessly. Fig. 1 shows coverage and propagation of GSM-cells in an area of London based on samples of cell-ids and their GPS-positions. Apart from not being visible, cells neither have fixed boundaries or propagation nor share exact borders with adjoining cells. The coverage of a cell is depending on many factors and their invisible boundaries and propagation are rather dynamic and fluctuating. As Fig. 1 shows, cell-coverage has irregular shapes and adjoining cells often overlap. These aspects can be a problem for location-based mobile applications relying on cell-id positioning where users or the application’s behaviour are dependent on knowing about the current location.

Mobile internet-access for cell phones via GPRS is still rather expensive. Yet it is indispensable for synchronising data between mobile clients and a central server. Despite the above definition of seams, costly internet-connections are neither rooted in technical limitations nor show themselves as uncertainties or ambiguities and may therefore be considered as some kind of made-up artificial seam. Still they disrupt the uniformity and continuity of a mobile application and we think that’s why they can be treated as a considerable seam in mobile applications.

An immediate consequence of insufficient synchronisation between a server and its clients are data-inconsistencies. They occur when individual, independent clients have to synchronise data with a central server that maintains data which is globally shared with several other clients. When one of these clients synchronises its local data with the shared server and thereby updates globally shared data, local copies of the same data on other clients become inconsistent with the updated, shared data.

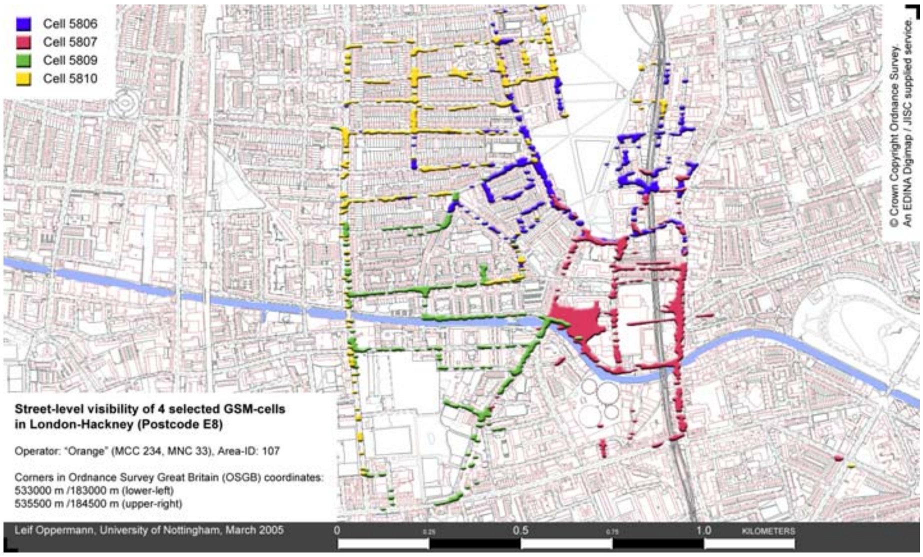

Fig. 1. Samples of cell-ids and their GPS-positions in London

4 How to play Tycoon

Tycoon is our approach to seamful design for a multiplayer-game for mobile phones which tries to incorporate the seams mentioned above and use them to enhance its gameplay. Tycoon is a location-based trading game with a simple producer-consumer-cycle that uses the different GSM-cells of a service provider network within a designated gaming-area, e. g. the centre of a city (see Fig. 1). Each of these cells in the physical area is virtually mapped to either a producer or a consumer in the game. The game uses the metaphor of a wild west scenario to communicate its central mechanisms of collecting resources from producers which are called “mines” and using them to buy objects from consumers which are called “brokers” and are named after cities or counties in California.

During the game players are travelling between the cells in the gaming-area, collect local resources in mines by staying in them for some time, use them to buy global objects from brokers and get credits for claiming them. Each mine produces an unlimited amount of one of the three local resources in the game – gold, silver or copper. They are called local resources because players can collect them independently of each other and don’t compete for them. Each broker has a list of global objects e. g. different buildings or estates in towns and counties that players can buy with their local resources. There is only a limited number of global objects in the game, each of them is unique (e. g. there is only one saloon in the city of Sacramento) and players compete against each other for claiming them, since every global object can only be claimed once by one player. Each global object has a combined price of two local resources and a value in credits. Players have to enter a broker’s cell and pay an object’s price in order to claim it and earn its value in credits in case that price in order to claim it and earn its value in credits in case that object is still available. The objective of the game is to gather as many credits as possible. It ends when all objects are sold and the player with the most credits wins.

5 Tycoon's Approach to combine Gameplay and Seamful Design

Tycoon uses the unique cell-id of the current GSM-cell to tell players where they are, help them navigate through the gaming-area and supervise their collecting of local resources. 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.

Whenever a player changes from one cell into another, an alert is triggered and he gets a notification about him entering a new cell. Afterwards Tycoon displays the name of the new cell (Fig. 2) in its GUI. This alert-mechanism provides interaction with the seam of dynamic cell-coverage and is more flexible than a rather static map of the gaming-area. It also improves the visualisation of cells' boundaries and propagation and decreases the players' uncertainty about their location without revealing too much information about it. Players can use their spatial knowledge about the gaming-area to adopt their own strategies of how to move between cells and how to find the most efficient tactics of which resources are needed to buy which available objects and where to find them in nearby mines. They can also exploit ambiguities caused by dynamic cell propagation and boundaries more effectively when they find a area where adjoining cells overlap and flip after some time without moving.

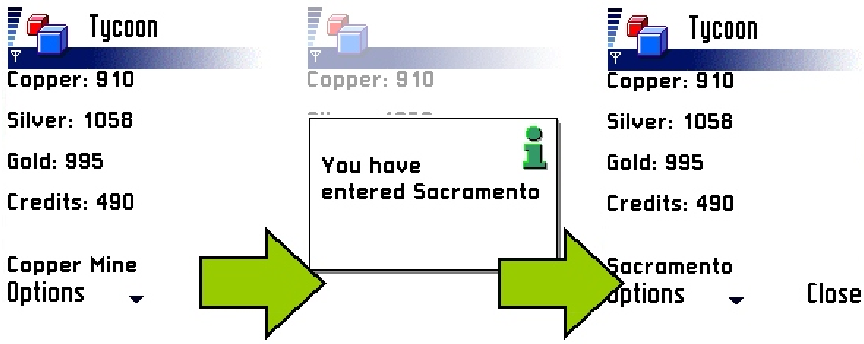

Fig. 2. The main screen of Tycoon's GUI which is updated whenever a player crosses boundaries between adjoining cells.

In order to cope with the seams of expensive internet-connections and data inconsistencies, Tycoon encourages players to spend more time offline and synchronize their local game-state (individual clients' knowledge about available global objects) less often with the game-server's global game-state (the overall availability of global objects). Therefore Tycoon's trading-mechanism features a simple economic model that gives players an incentive to engage in extended offline-play and spend more time on collecting a greater number of (more valuable) resources.

As mentioned before, global objects have a price in local resources and earn a player a certain amount of credits for claiming them. Both values are related to each other as the credit-values of objects rise with their prices in local resources. The local resources have different values themselves which are indirectly expressed by the time it takes to collect one unit of each one. Collecting a gold-nugget takes longer than collecting a silver-nugget which again takes longer than collecting a copper-nugget. That way the credit-value of a global object is also related to the time it takes to collect the resources that are necessary to buy the object. But the values of resources and objects as well as their prices don't rise proportionally. The economic model is tuned in a way so that the more time a player spends offline to collect more and more valuable resources in order to afford more and more valuable objects the more credits per second could he possibly earn than by collecting less resources for smaller, less valuable objects during the same time. Additionally a player gets a discount for successfully claiming several objects from a broker with the same request to the game-server. In order to afford buying several objects, players have to spend even more time offline to collect more and more valuable resources. The goal of this economic model is to give players an incentive to spend more time offline between connections to the game-server and collect local resources which is more profitable than spending less time offline collecting less and less valuable resources for claiming less valuable objects.

But extended offline-play increases the danger of inconsistencies between the game-server's global game-state and a player's local game-state which are different views on the overall availability of global objects. Players can claim objects which become no longer globally available but are still shown as available on other players' clients. Since players can't rely on getting constant updates of the global game-state which would require regular expensive connections to the game-server, they have to consider the growing probability of inconsistencies between local and global game-state when deciding how much time they spend offline for collecting local resources.

This is where the gambling-feature of Tycoon comes into play: It recognizes these inconsistencies in the gameplay and awards players not only for spending more time offline but also for taking the risk of inconsistencies. Players are free to collect as many local resources as they want but can only turn them into credits when they successfully buy and claim global objects with them. The more time they spend offline, the more profit is possible but the greater becomes the probability of inconsistencies and missed claims. Players have to adopt their own individual strategies and consider their chances of earning more credits during the same time against the risk of data inconsistencies which is also dependent on their knowledge of where to find the required local resources to claim a desired object.

In order to soften the possible inconsistencies in the availability of global objects between the game-server and the mobile clients, players can also ask the game-server for an update on the overall availability of global objects anytime and anywhere, but have to pay a certain charge of local resources for doing so since the update gives them an advantage in the game.

6 Conclusion

As Ubicomp systems are still affected by seams we see seamful design as a rewarding amendment to seamless design. Revealing and exploiting seams provide new views on the design of Ubicomp systems other than seamlessness. Different ideas and approaches to the design of Ubicomp systems will be even more important as future mobile devices and applications will offer more sophisticated and demanding services e. g. for multimedia and entertainment, where seams in both technology and interaction-design are even more likely to come to the users' attention and affect their interactions and experiences.

7 Acknowledgements

This project was supported by EQUATOR, an Interdisciplinary Research Collaboration (IRC), funded by the UK Engineering and Physical Sciences Research Council.

We would like to thank Mauricio Capra (MRL, University of Nottingham) for collecting the sample data used in Fig. 1. We would further like to thank Albrecht Schmidt (Institute for Informatics, Embedded Interaction Research Group, Munich) and Holger Schnädelbach (MRL, Nottingham) for their support and making this project possible.

References

1. Weiser, M.: The world is not a desktop. ACM Interactions 1(1) (1994) 7-8.
2. Chalmers, M., MacColl, I.: Seamful and Seamless Design in Ubiquitous Computing. Technical Report Equator-03-005, Equator [Technical Reports] (2003)
3. Weiser, M.: Creating the invisible interface (invited talk). ACM Conf on User Interface Software and Technology (UIST94) (1994), 1
4. Chalmers, M., Bell, M., Hall, M., Sherwood, S., Tennent, P.: Seamful Games. http://www.ubicomp.org/ubicomp2004/ adjunct/demos/chalmers.pdf (2004)
5. Oulasvirta, A.: Notes on Seams, Seamfulness and Seamlessness. http://www.hiit.fi/u/oulasvir/Haninge (2004)