Using Semantic UI Descriptions for Adaptive Mobile Games

Pascal Bihler, Holger Mügge, Mark Schmatz and Armin B. Cremers

Institute of Computer Science III, University of Bonn
Römerstr. 164, 53117 Bonn, Germany
{bihler, muegge, schmatz, abc}@cs.uni-bonn.de

Abstract: In recent years, mobile devices became more and more powerful, wireless Internet access is becoming standard. To support mobile applications adapting their User Interface to various, fast changing contexts, we proposed to use semantically described User Interfaces [BFKJ08]. In that paper, as well as in related research works, mainly form-based applications, as used for office software, have been in focus. Nevertheless, the idea of semantical UI description also comes at hand when focusing on non-form based applications like location based games, since context switches can be numerous and spontaneous here. As we show in this paper, especially when used in the context of service based reconfigurable mobile games, the fusion of semantically described interfaces can simplify the generation of complex adaptable user interfaces.

1 Introduction

“He must be somewhere around here, I can see his pointer on my device!” – “Look there, Mister X his hiding behind the tree!” – “Let’s go and catch him...”. With the rise of processing power and visualization capabilities of mobile devices, scenes like this can become ubiquitous in modern city life: a group of young people is playing a mobile, location based game. In this paper we focus on the specialties of computer-human interaction in location-based games.

In classical computer and console games, the “game reality” is completely virtual and the player gets a part of this virtual world. Location based games are different. They interweave the physical reality with the virtual game world and create a mixed reality. This fact has some important challenges for UI design as we perceived in our research.

First, the hardware of mobile computing devices is very diverse. Most mobile handsets differ a lot in processing power as well as in visualization (e. g. screen dimensions and resolution) and other interaction capabilities (multi-touch vs. digit-input only, accelerometer, headset support etc.). This is already a well-known but not yet completely resolved problem [LMMG09]. It should also be noted that handset capabilities might change even at runtime; for example when the player switches to a headset.

Second, since the player is always influenced by its current physical surroundings, different and even unexpected situations occur that might greatly influence the way she wants to interact with the game device. For example within a hide & seek game, the situation where the chasing players stand still and are planning a strategy how to catch the runaway is significantly different from the situation when the chase is culminating and all players run. It is essential to provide the players with adequate interactions for each important game situation. If the technical bridge between physical and virtual reality does not work smooth enough the players cannot immerse into the game and their game experience will diminish. Hence, a dynamic adaptation of the game UI at runtime is needed.

Thus we see two main adaptation challenges for mobile game applications and its user interfaces:

1. Adaptation to different handset capabilities.
2. Adaptation to sensor-triggered dynamic context changes during the game.

With the users' surrounding being part of the game, we experience that they identify themselves quickly with the gameplay. After playing they often imagine variations of the game and are eager to adapt the game rules and components to their needs. Therefore, a third dimension of adaptation should be added:

3. Adaptation of the game to the players' experiences and favors by modifying the game's rules and components.

To cope with these three challenges, we propose the use of semantically described user interfaces: In the next section of this paper, we demonstrate how semantic descriptions of user interfaces work and how they help with tackling the *multi-handset problem* in mobile games. In section 3, we show how *dynamic context changes* can be handled using our approach. When it comes to *game reconfiguration*, fusion of semantic UIs can be of support, as we explain in section 4. In section 5 we provide information about related work on our subject, and we summarize our approach, together with some research outlook, in section 6.

2 Adapting Games to Handsets

When developing a mobile application for a broad audience, *differences in handset capabilities* have to be taken into account. These span from adapting to different screen resolutions up to supporting different input and output mechanisms. Even if the mobile application is designed to be web-based, as proposed by [Way09] and others, different rendering mechanism of mobile browsers require visible and non-visible code adaptations.

Semantic user interface descriptions (cf. [BM07]) enable to abstract from these requirements: the developer focuses on the input and output requirements of the application, while an application-independent rendering framework manages the UI adaptation in function of the use context (used device, situation, etc.). Thus, the developer defines *what* should be displayed, and not *how*, which simplifies the application by factoring out many of the adaptation algorithms. This approach limits the developer's direct influence on the con-

Figure 1: A typical screen of *Scotland Yard to go!*, rendered for an iPhone OS based device.

crete visualization of his UI but it lifts the burden of exponential adaptations from the developer's shoulders.

Until now, most prototypical applications for semantic driven UI rendering focus on form based applications (e. g. [EVP01, CLV$^+$03, BFKJ08]). We propose to extend this technique so that it supports even the design of interfaces for location based games. We demonstrate the declarative description of a user interface with an example: the main screen of the game *Scotland Yard to go!* (see Figure 1) developed by our research group.

In the sketched situation the user currently plays with one team-mate, some location based elements are placed around him, as well as location based partner information is provided. Despite other sensor input (like GPS positioning), the user can directly interact with the mobile UI by activating one of these functions: Settings, team view, shop view and the possibility to capitulate. We concentrate on the main parts of the given interface view and identify the following semantic interaction elements (see Figure 2):

- A *group* of different location based outputs (whereas each of these location based output elements might be visualized using a map view, if the handset is capable of graphic output). These output show the *location of the player herself*, the *location of other players* (team-mates), the *location of special game elements* and *location based partner information*.
- A *group* containing four individual elements each triggering a specific action.
- A (non-visible) *input* transmitting GPS data from the client to the application.

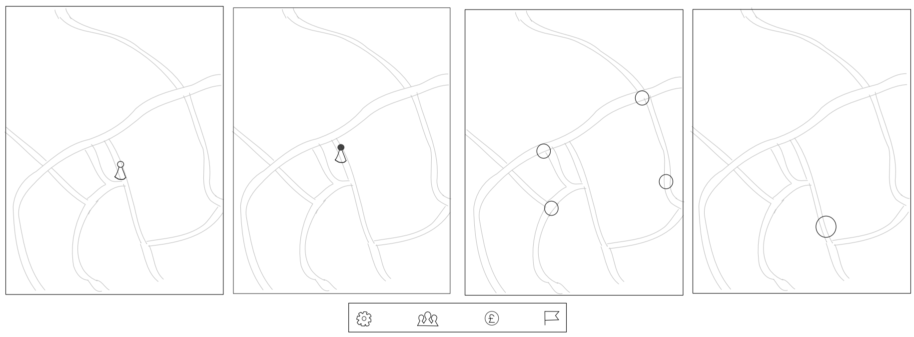

Figure 2: The main elements of the game screen: visualization of own position, team member positions, location based elements, location based partner information, and several action triggers.

To formalize the semantic UI, we use here a simplified version[1] of the XML representation of LAIM as introduced in [BFKJ08]:

```
<laim>
  <group id="locations">
    <output id="own_position" contentType="location">
      61.329109, -48.802986
    </output>
    <output id="mrx_position" contentType="location">
      61.328944, -48.800068
    </output>
    <output id="money_element" contentType="location">
      61.327462, -48.802814
      (...)
    </output>
    <output id="partner_info" contentType="location">
      61.327379, -48.795261
    </output>
  </group>
  <group id="controls">
    <action id="settings" />
    <action id="team_view" />
    <action id="shop_view" />
    <action id="give_up" />
  </group>
  <input id="location" contentType="currentLocation" />
</laim>
```

[1]Some elements of the LAIM syntax, e. g. as used for UI element prioritization, have been left out here to simplify the documentation.

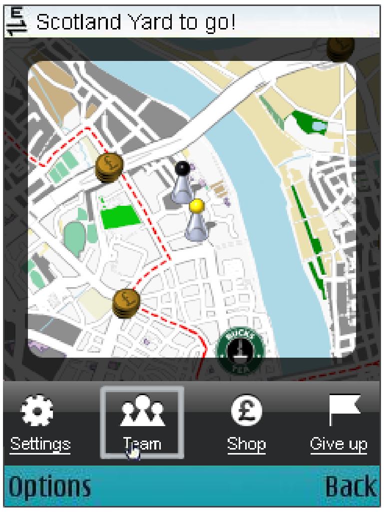

Figure 3: A web-based game screen, as it could be rendered for a Symbian device.

Based on this general description, the game UI could be adapted to other handsets requiring different implementations, as sketched in figure 3.[2] Also devices with multiple screens could be supported (e. g. a Nintendo DS), whereas in our example one screen could display position information while the other one acts as a touch sensitive control area. As an extreme case, even an non-graphical mobile handset could be used[3]: The position output needs to be translated into a textual representation (like "Mr. X: 100 meters north of you."), and the gaming experience might be different, but the underlying application does not need to perform this adaptation, it can be handled solely by the visualization layer.

3 Dynamic Context Changes

The second challenge for mobile devices are *frequent dynamic context changes*. These context changes are normally triggered by special sensors like a location sensor, an acceleration sensor or a time sensor (cf. [MBSC08]). They can incorporate a variety of factors, e. g. the user can enter a dedicated location zone (like a certain building or a playground), or the application could discriminate a moving person from one standing [MLF$^+$08].

Some of these context changes require a game logic based adaptation: It might be possible to move physically to the virtual position of a coin element to "pick it up". Other context changes, in contrast, just require a UI adaptation. Given a semantic interface description, these cases could be handled independently from the underlying game application by the UI framework itself, since the semantic of the UI does not change here.

[2] To identify the device requirements, a database like WURFL [PT08] is used during the adaptation process.
[3] Given that it's position could be determined, using network based cell-tower triangulation, for example.

Examples for such application-logic independent adaptations are:

- When running, buttons on a touch screen are harder to locate and tiny elements are more difficult to notice on the screen. This could be compensated by rendering important action elements bigger while less prioritized elements are faded out.[4] Analogously, while standing, more details of the UI could be revealed.
- Using an ambient light sensor, a weather forecast service, or information about dust and dawn time, the user interface could be switched to a “night vision” mode which is optimized for visibility in the dark.
- Some devices feature compass sensors which could be used to rotate a map correspondingly to the player’s orientation.

4 Supporting Mobile Game Kits

Experience shows that mobile location based games mainly consist of recurrent building blocks like *localization*, *visualization of position*, *active zones* etc. Many games can be build by combining these building blocks, assembled by concrete game rules. Providing the user the possibility to recombine and reconfigure these building blocks and their fitting would finally lead to some sort of a mobile game construction kit. This approach can help to master the demand for an *adaptation to the players’ experiences and favors*.[5] Each visual building block of this construction kit framework would have certain input and and output requirements. While each of the building blocks’ UIs could be visualized independently, only when merged reasonably together, they form a coherent game UI.

This situation resembles the problem sketched in [BM07] for several applications used in parallel on a mobile device: Several independent user interfaces take part in the same user task (in our case, the task is “Playing the game”). There we proposed to use semantic UI descriptions to fuse form elements for office applications on mobile devices in order to create a dynamic, task-oriented application-spanning UI. In the same way, we can apply this technique here and fuse the separate building blocks’ UI elements together to form a common game user interface.

Using UI fusion in a game construction kit has several advantages: Providing a framework for location based games, some sort of API has to be defined as common interface for the building blocks. Semantic interface descriptions can be part of such an API, so that the developer of the building block does not need not to cope with the specifics of the used handsets but can concentrate on the building block’s functionality. As the framework requires some sort of common UI generator, this seems to be a natural place to implement the *Context Aware Interface Decorator* and the *Semantic Interface Layout Engine* of [BM07].

[4] [BM07] introduced the idea of prioritizing UI elements. In our example, the *Scotland Yard to go!* application could prioritize the “Give Up”-action lower than the “Shop” action.

[5] The details of such a game construction kit are out of the scope of this paper, some ideas have been sketched in [BM08].

In addition, the complexity of these framework elements is lower in the context of location based games than it is for general office applications, and the selection of elements to combine does not need to be inferred from usage history but can be explicitly determined by the user of the game construction kit.

5 Related Work

Model based UI design for classical and mobile applications has been proposed for almost a decade (e. g. [EVP01, CLV+03, MVLC08], for an overview see also [LMMG09]). As well, UI composition based on semantic or model based descriptions has been under research [XPJK03, LHR+07, BM07]. Nevertheless, these works mainly focused on form based interfaces.

In [BHC+06], Bell et al. use the example of a "mobile game"[6] for demonstrating the adaptation of mobile applications. The gaming scenario is local, and adaptation is envisioned for predefined, device specific client applications. As well, the location based multimedia and gaming framework Mediascape [SHG+07] proposes the use of dedicated clients on the devices. The game framework Ex Machina[7] focuses on the game network and logic infrastructure without providing a dedicated API for user interaction.

6 Conclusions and Future Work

In this paper, we introduced semantic defined user interfaces as a way to cope with the three main adaptation challenges of location based games: *Adaptation to the user's device*, *adaptation to the user's situation* and *adaptation to the user's creativity*. Using an example from the *Scotland Yard to go!* game we have shown that semantic UI description seems natural and that a distinction in between game logic related adaptation and UI related adaptation can help to separate and reuse adaptation functionality. As a special case for functionality reuse, we have proposed to integrate the semantic UI rendering and fusion framework into a game construction kit, which would allow the user recombine game building blocks (and their correspondent user interfaces) where the combinations need not to be pre-planned by the building blocks' developers.

While there exist implementations for semantical UI rendering and UI fusion, as well as location based game implementations, the combination of both is still in an early phase. To support a broad variety of devices, the first implementations will concentrate on browser based games. The approach introduced in this paper will constitute a central part of a construction kit for adaptive mobile games.

[6] Mobile relates mainly to the game execution environment being mobile devices, location is only used in the sense of proximity.

[7] http://www.exmachina.nl

References

[BFKJ08] P. Bihler, M. Fotsing, G. Kniesel, and C. Joffroy. Using Conditional Transformations for Semantic User Interface Adaptation. In *Proc. of the 10th iiWAS conference*, pages 677 – 680, Linz, September 2008. ACM.

[BHC+06] M. Bell, M. Hall, M. Chalmers, P. D. Gray, and B. Brown. Domino: Exploring Mobile Collaborative Software Adaptation. In *Proc. of Pervasive 2006*, pages 153–168, Dublin, 2006.

[BM07] P. Bihler and H. Mügge. Supporting Cross-Application Contexts with Dynamic User Interface Fusion. In *Beiträge der 37. Jahrestagung der Gesellschaft für Informatik e.V.*, pages 459–464, Bremen, 2007.

[BM08] P. Bihler and H. Mügge. Scotland Yard fürs Handy – Rallyes mit neuen Medien und als Mixed Reality Spiele. *gruppe & spiel*, 03:40, 2008.

[CLV+03] K. Coninx, K. Luyten, C. Vandervelpen, J. Van den Bergh, and B. Creemers. Dygimes: Dynamically Generating Interfaces for Mobile Computing Devices and Embedded Systems. In Luca Chittaro, editor, *Mobile HCI'03*, volume 2795 of *Lecture Notes in Computer Science*, pages 256–270. Springer, 2003.

[EVP01] J. Eisenstein, J. Vanderdonckt, and A. Puerta. Applying model-based techniques to the development of UIs for mobile computers. In *Proc. of the 6th International Conference on Intelligent User Interfaces*, pages 69–76, NY, USA, 2001. ACM.

[LHR+07] S. Lepreux, A. Hariri, J. Rouillard, D. Tabary, J.-C. Tarby, and C. Kolski. Towards Multimodal User Interfaces Composition based on UsiXML and MBD principles. In *Proc. of the 12th HCI International*, volume 4552 of *Lecture Notes in Computer Science*, pages 134–143. Springer, July 2007.

[LMMG09] V. López Jaquero, F. Montero, J.P. Molina, and P. González. *Engineering the User Interface*, chapter Intelligent User Interfaces: Past, Present and Future, pages 259–270. Springer, 2009.

[MBSC08] H. Mügge, P. Bihler, M. Schmatz, and A. B. Cremers. Context and Situation as Enablers for Multi-Modal Interaction in Mobile Games. In *Workshop at HCI'08: Evaluating Player Experiences in Location Aware Games*, Liverpool, 2008.

[MLF+08] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. In *SenSys '08: Proc. of the 6th ACM conference on Embedded network sensor systems*, pages 337–350, 2008.

[MVLC08] Jan Meskens, Jo Vermeulen, Kris Luyten, and Karin Coninx. Gummy for multi-platform user interface designs: shape me, multiply me, fix me, use me. In *Proc. of AVI 2008*, pages 233–240, New York, USA, 2008. ACM.

[PT08] L. Passani and A. Trasatti. WURFL and WALL. http://wurfl.sourceforge.net, 2008.

[SHG+07] S. P. Stenton, R. Hull, P. M. Goddi, J. E. Reid, B. J.C. Clayton, T. J. Melamed, and S. Wee. Mediascapes: Context-Aware Multimedia Experiences. *IEEE MultiMedia*, 14(3):98–105, 2007.

[Way09] P. Wayner. The cross-platform option: Web apps for smartphones. *Info World*, January 20, 2009. http://www.infoworld.com/article/09/01/20/03TC-phone-web_1.html.

[XPJK03] X. Xiaoqin, X. Peng, L. Juanzi, and W. Kehong. A component model for designing dynamic GUI. In *Proc. of PDCAT'2003*, pages 136–140, Aug. 2003.