A Framework for Mobile Interactions with the Physical World

Enrico Rukzio[1], Sergej Wetzstein[1], Albrecht Schmidt[2]

[1] Media Informatics Group, [2] Embedded Interaction Group, University of Munich

{Enrico.Rukzio, Sergej.Wetzstein, Albrecht.Schmidt}@ifi.lmu.de

***Abstract*—Mobile interactions with the physical world, meaning a person uses her mobile device as mediator for the interaction with a physical object, get more and more popular in industry and academia. Typical technologies supporting this kind of interactions are Radio Frequency Identification (RFID), visual marker recognition, Near Field Communication (NFC), or Bluetooth. Currently, there only exists very little tool support for building systems that consider this kind of interactions. But this is necessary because of the complexity, variety and distribution of such systems. A framework would also support the development and the dissemination of physical mobile interactions in our everyday live. Therefore we present in this paper the requirements for such tool support, the architecture of the Physical Mobile Interaction Framework (PMIF) and a first version of the implementation.**

Key words: Physical mobile interaction framework, physical mobile interaction, mobile device.

1. INTRODUCTION AND MOTIVATION

In the last years we notices a raising interest in physical mobile interactions in research and academia which are mostly based on innovative and promising technologies like RFID, Near Field Communication (NFC), Bluetooth or visual marker recognition. With *physical mobile interaction* we mean any communication between the entities user, mobile device, and physical objects in the physical world whereby every entity can exist one or more times. Hereby the physical objects can include a computer or not. A possible scenario is for instance like this: a person is pointing with mobile device with an embedded camera at a marker on an advertisement in a fashion journal. Another scenario could be a Bluetooth-based interaction between a mobile device and a public display. Both of these scenarios can be described as physical mobile interactions.

This research field deals for instance with *mobile interaction with enhanced physical objects* [1, 2, 3], *sensing the environment to get awareness of the context of the user* [4, 5], *mobile interaction with public and semi-public displays* [6, 7], *mobile interactions in smart environments* [13] *mobile annotations* [8] or *using the mobile device as a universal remote control* [9].

Figure 1 illustrates typical examples of physical mobile interactions in which the user interacts via the mobile device with things, peoples and places [1] in the physical world. Herby we particularly focus on the interaction between mobile devices and objects in the physical world.

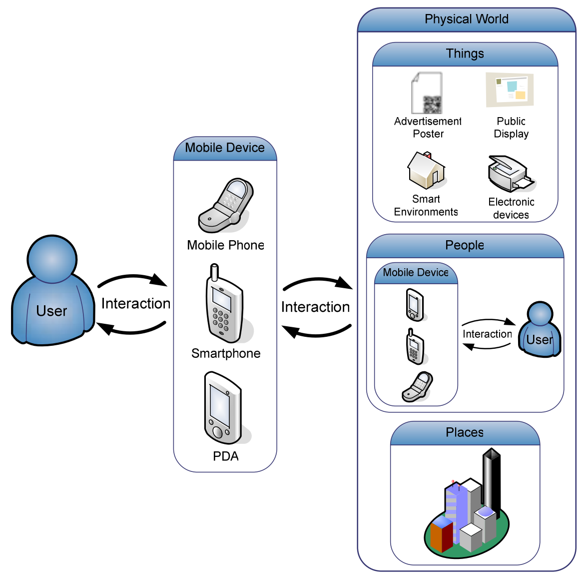

Figure 1: Physical Mobile Interactions

Currently there exists only marginal tool support for building systems supporting this kind of interactions. But this is necessary because of the complexity, variety and distribution of such systems. A framework would also support the development and the dissemination of physical mobile interactions in our everyday live. Therefore we define in this paper the requirements, architecture and a first version of the implementation of a corresponding framework.

Before working on the architecture, we defined the following goals:

- **Support for the development and implementation** of systems that takes physical mobile interactions into account.
- **Support for all relevant interaction techniques** based on the different communication technologies between the device and object.
- **Provision of abstractions** for the programmer that hides the details of the communication technologies used between the mobile device and the physical object.

- **Orientation on existing and evolving standards** in this field like the Java 2 Micro Edition (J2ME) and the Contactless Communication API (JSR 257) [10].
- **Provision of the interfaces for the integration of existing systems** for the provision of services and for future communication technologies between physical objects and the mobile device.
- **Provision of lightweight components on the mobile device** which allow the easy development of the software and which take the memory and processing constraints into account.

The paper is structured as follows. The next section describes the architecture of our framework where the focus also lies on the elements on the mobile device, the physical object and the server. Based on this we present the current state of the implementation and illustrate the usage of the framework on a physical mobile interaction technique. The paper is completed by a discussion of our work.

2. ARCHITECTURE OF THE PHYSICAL MOBILE INTERACTION FRAMEWORK (PMIF)

2.1. Overall Architecture

Figure 2 illustrates the overall architecture of the Physical Mobile Interaction Framework (PMIF). As mentioned, mobile devices, enhanced physical objects and corresponding services are involved in these types of interactions.

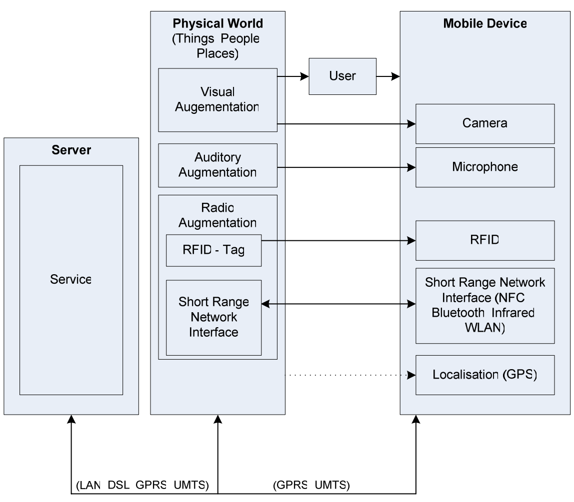

Figure 2: Generic Architecture

The most important aspect which must be supported by the framework is the communication between the enhanced physical object and the mobile device. In generally the augmentation can be visual, auditory and based on radio communication. Besides this direct interaction there can also be an indirect interaction in which the user acts as mediator between the physical object and the mobile device. An example for this is that there is a number visible on the object (visual augmentation) and the user types the number in the mobile device to establish the interaction.

Visual augmentation is mostly done by markers on the objects. The camera of the mobile device takes a picture of it and extracts the identifier which is represented by the marker. Auditory augmentation which is sensed by the microphone of the mobile devices is technically also possible, but its feasibility is currently still questionable and there are only few projects which address this approach. Currently we see a huge interest in the radio augmentation of physical objects with technologies like Near Field Communication (NFC, [15]), Bluetooth, Infrared and WLAN. We distinguish between unidirectional (e.g. read only RFID tag on the physical object) and bidirectional communication (e.g. peer-to-peer communication based on NFC).

Beside this direct or user mediated interactions, there are also other indirect interactions. E.g., the location information is used to reason about the proximity of the physical object and the mobile device.

Based in these aspects of the generic architecture we defined components of the PMIF which run on the mobile device and on the server which provides the services.

2.2. Mobile Device

Figure 3 shows the components of the PMIF on the mobile device. It provides the *PhysicalWorldConnector* and the *ServiceConnector* that are used by the application (*Application*) which takes physical mobile interactions into account.

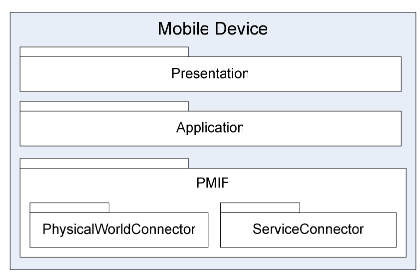

Figure 3: PMIF on the mobile device

The *PhysicalWorldConnector* provides an abstraction of the concrete connection technology (e.g. NFC or marker based) which can be used by the programmer to develop the application itself. This is achieved by a method used by the application for asking for the connection technologies that are provided by the concrete mobile device. After the decision for a connection technology the application gets a corresponding object which uses the *Stream* metaphor. This means that the application sees the connection between the mobile device and the physical object as a

Stream on which the application can read or write. Writing on such a *Stream* is only possible if the connection technology allows this. Figure 4 shows the abstractions provided by the *PhysicalWorldConnector* and the different specialisations of it.

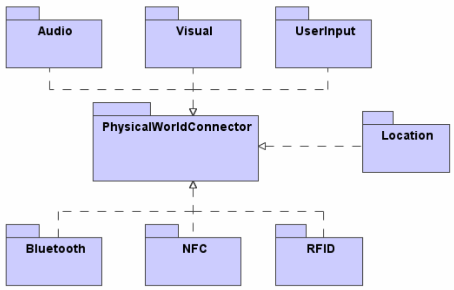

Figure 4: PMIF on the mobile device

Furthermore, the *PhysicalWorldConnector* provides an abstract user interface implementation for every connection technology and corresponding interaction technique. This can be used and adapted by the programmer during the development of the presentation (*Presentation)* of the application.

The *ServiceConnector* depicted in Figure 3 provides an abstraction for the usage of a service in an application (*Application*) on the mobile device. Such a service could be realized through using technologies like for instance (X)HTML, i-mode, WAP, Web Services, OWL-S or UPnP. The *ServiceConnector* handles the communication between the *PhyscialWorldConnector* and the service hosted on the server. It is for instance often required that the services on the server get informed about the communication between the mobile device and the physical object. Furthermore, the *ServiceConnector* can be directly used for the presentation of the service if the service is realized with a direct renderable (e.g. HTML) technology. If this is not possible, the information taken from the *ServiceConnector* (e.g. SOAP messages) must be processed by the application before the presentation can be generated.

2.3. Physical Objects

Figure 5 depicts the augmentation of the physical objects. Typical examples are

- advertisement posters which are augmented by visual markers,
- machines which are augmented by RFID tags to support up-to-date service information (e.g. when the item was last serviced and by whom)
- or a public display which is augmented by a Bluetooth-based service through which the user can interact via her mobile device with it.

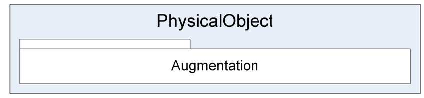

Figure 5: RealWorldObject

This augmentation is only indirectly a part of the PMIF because the provided information is either static (e.g. visual marker, non writeable RFID tags) or provides a service via a network interface (Bluetooth, NFC). In the case the provided information that can be sensed by the mobile device is changeable (writeable RFID tags, mobile device with a printer) than this functionality is provided by the PMIF on the mobile device. In the case that the augmentation provides a service, this functionality is provided by the PMIF. This is explained in the following subsection. This means that physical objects could also provide services which run on a server. An example for this is a public display whose services are provided by a local or remote server.

2.4. Server

Figure 6 shows the components of the PMIF on the server. It provides the *PhysicalWorldObjectsDatabase* and the *ServiceConnector* for the communication between the services on the server and the *ServiceConnector* on the mobile device. The server could be located in a remote destination or could even be an element of the physical object.

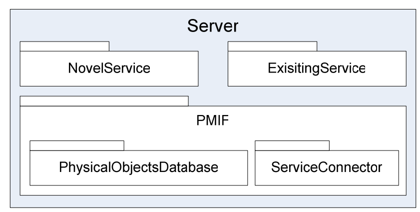

Figure 6: PMIF on the server

The *PhysicalObjectDatabase* is able to manage for every physical object the information regarding its identifier, position, properties and related services (e.g. a link to an URL to a mobile service hosted on the server). This database is used by the *ServiceConnector* on the server to provide this information to the mobile device or to the services on the server.

The *ServiceConnector* is on the one hand responsible for the communication between the mobile device and the server and on the other hand it manages the relationships between the physical object and the service.

3. IMPLEMENTATION

In this section we will discuss the technologies we used for the implementation and we will show an example for the implementation of a physical mobile interaction with the PMIF.

3.1. Used Technologies

When thinking about the best execution environment for our implementation of the PMIF, platform independence and widespread were the most important factors. Therefore we decided to use Java 2 Micro Edition (J2ME) with 1.7 billion enabled devices [1]. J2ME is platform independent and nearly all operation systems on mobile devices such as Symbian, Palm OS, Windows Mobile and most mobile phone vendor specific operation systems support J2ME.

3.2. Example: Visual Physical Hyperlink

In this interaction technique, everyday physical objects like advertisement posters, printers or sights are enhanced by visual markers. They represent services which are related to object. The user can *click* on these markers with her mobile device; the handheld then starts the corresponding service.

Figure 7. Taking a picture of the marker in the visual physical hyperlink interaction technique

Figure 7 shows the moment in which the focus of the camera of the mobile phone is focused on the visual marker on an advertisement poster. An overview of such systems can be found in [2].

For the implementation we used the specialization of the *PhysicalWorldConnector* for the handling of visual markers which is depicted in the previous Figure 4 by the package *Visual*. This is shown in detail in the following Figure 8. The *PhysicalObjectDiscovery-Manager* is the most important class which is directly used by the application on the mobile device. The source code example in Figure 9 shows how the classes shown in Figure 8 are used. As the first step, the application needs to know which communication technologies for the interaction with physical objects the mobile device supports (*getSupportedPhysical-ObjectTypes()*, line 7). After the selection of a communication technology the application can define a listener for it. In our visual physical hyperlink example we use the communication technology visual marker (*PhysicalObjectTypes.VISUAL*, line 19).

The *PhyiscalWorldObjectConnector* also provides an abstract definition of the user interface for the interaction. This is the *PhyscialInteractionController* depicted in Figure 7. When using the specialization *Visual*, there is special interaction control for visual tags (*VisualTagInteractionController*) available. This is automatically started after the corresponding listener (line 19) was created. After the user has *clicked* on the visual physical hyperlink (as depicted in Figure 7) the application can access the identifier (e.g. a number) provided by the visual marker (line 25). For this, we used the visual code software developed by Rohs and Gfeller [2]. This identifier is then used to find out which web page relates to the advertisement poster. These web pages (service) are located on the server. The *ServiceConnector* on the mobile device transmits the identifier to the *ServiceConnector* on the server. Here the *PhysicalObjectsDataBase* is used to find the related web page. The hyperlink is then sent back to the mobile phone and the application there presents with the help of the browser this web page.

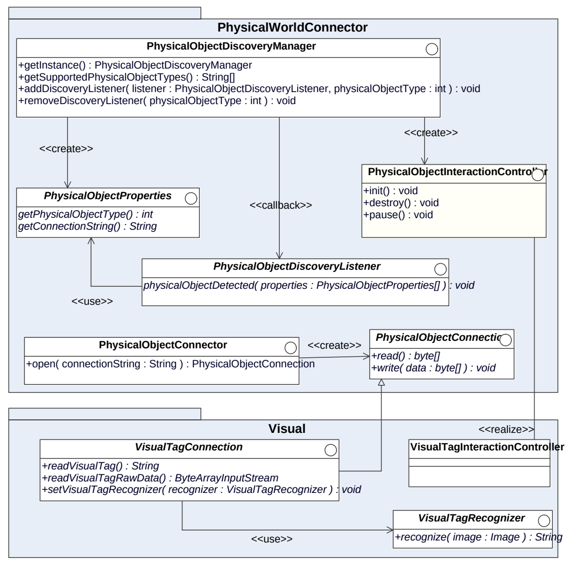

Figure 8. Specialization Visual of the PhysicalWorldConnector.

This discussed example illustrates three important advantages of the PMIF. First, it is very simple to use different communication technologies for the interaction between the mobile device and the physical object. The application can request which are supported and than it can easily select one. Second, the programmer has not to handle the tiny details of the user interface for a special communication technology. This is done by the interaction controller which can be adapted to the needs of the specific application. Third, the framework provides several abstractions like the *Stream* metaphor which hides the complexity of the concrete communication technology. As you can see in Figure 8 only few lines of code are needed when using physical mobile interactions in an application.

```
private  discoveryManager PhysicalObjectDiscoveryManager;

public void startApp() {
   //get instance of PhysicalObjectDiscoveryManager
   discoveryManager = PhysicalObjectDiscoveryManager.getInstance();
   //get supported PhysicalWorld Connectors,  initiate a list and show it
   String[] supportedTypes = discoveryManager.getSupportedPhysicalObjectTypes();
   supportedTypesList = new List("PhysicalWorld Connectors", List.IMPLICIT);
   for(int i = 0; i < supportedTypes.length; i++) {
      supportedTypesList.append(supportedTypes[i], null);
   }
   supportedTypesList.setCommandListener(this);
}

//implements commandListenr
public void commandAction(Command c, Displayable d){
   if (c == CMD_SELECT_CONNECTOR) {
      String objectType = supportedTypesList.getString(supportedTypesList.getSelectedIndex);
      discoveryManager.addDiscoveryListener(new VisualObjectDiscoveryListener(),
 PhysicalObjectTypes.VISUAL );
   }
}

private class VisualObjectDiscoveryListener implements PhysicalObjectDiscoveryListener {
   public void physicalObjectDetected(PhysicalObjectProperties[] properties){
      display.setCurrent(new Alert("VisualObject ID:", properties[0].getID(), null, AlertType.INFO));
   }
}
```

Figure 9. Usage of the implementation of PMIF for the interaction physical visible hyperlink.

4. SUMMARY

In this paper we presented a framework for the development of systems which takes physical mobile interactions into account. We call it Physical Mobile Interaction Framework (PMIF). At the beginning, we discussed the field of physical mobile interactions and we defined the requirements for a corresponding framework. Based on this we developed the overall architecture of PMIF and discussed how it is used on the mobile device, on the physical object and on the server. Then we discussed the implementation PMIF and showed based on a physical interaction technique advantages when using it.

ACKNOWLEDGMENT

This work was performed in the context of the IST Project Simplicity [11] funded by the EU and the DFG funded research group “Embedded Interaction” [12].

REFERENCES

[1] Kindberg, T. et al. 2002. People, Places, Things: Web Presence for the Real World. In: Mobile Networks and Applications, 7(5):365–376, 2002.

[2] Rukzio, E., Schmidt, A., Hussmann, H. 2004. Physical Posters as Gateways to Context-aware Services for Mobile Devices. In: *Proceedings of WMCSA 2004*, English Lake District, UK, 2-3 December 2004.

[3] Rohs, M.; Gfeller, B. 2004. Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects. In: Alois Ferscha, Horst Hoertner, Gabriele Kotsis (Eds.): Advances in Pervasive Computing, Austrian Computer Society (OCG), ISBN 3-85403-176-9, pp. 265-271, Vienna, Austria, April 2004.

[4] Hinckley, Ken; Pierce, Jeff; Sinclair, Mike; Horvitz, Eric. 2000. Sensing techniques for mobile interaction. In: UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology. 2000.

[5] Gellersen, H.; Schmidt, A.; Beigl, M. 2002. Multi-sensor context-awareness in mobile devices and smart artifacts. In: Journal Mobile Networks and Applications 7, pp. 341–351, Kluwer Academic Publishers, 2002.

[6] Ferscha, A.; Kathan, G.; Vogl, S. WebWall - An Architecture for Public Display WWW Services. In: Proc. of Eleventh International WWW Conference (2002).

[7] Greenberg, S., Boyle, M., Laberge, J. 1999. PDAs and Shared Public Displays. In: Making Personal Information Public, and Public Information Personal. Personal and Ubiquitous Computing 3 (1999).

[8] Smith, M. et al. 2003. The annotated planet: A mobile platform for object and location annotation. In 1st Int. Workshop on Ubiquitous Systems for Supporting Social Interaction and Face-to-Face Communication in Public Spaces at UbiComp 2003, October 2003.

[9] Myers, B. A. 2002. Mobile Devices for Control. In: Mobile Human-Computer Interaction: 4th International Symposium, Mobile HCI 2002, Pisa, Italy.

[10] Contactless Communication API (JSR 257), http://www.jcp.org/en/jsr/detail?id=257

[11] Simplicity Project, http://www.ist-simplicity.org.

[12] Embedded Interaction Group http://www.hcilab.org.

[13] Shahi, A.; Callaghan, V.; Gardner, M. 2005. Introducing Personal Operating Spaces for Ubiquitous Computing Environments. In: *Proceedings of Pervasive Mobile Interaction Devices (PERMID 2005)*, Munich, Germany.

[14] Rosca, J.; Sudarsky, S.; Balan, R.; Comaniciu, D. Mobile Interaction with Remote Worlds: The Acoustic Periscope. In: Proceedings of IJCAI 2001.

[15] NFC Forum, http://www.nfc-forum.org

[16] Report on Heise.de from the 25/10/2004 http://www.heise.de/newsticker/meldung/52477