Proceedings of the IASTED International Conference
HUMAN-COMPUTER INTERACTION
November 14-16, 2005, Phoenix, AZ, USA

PLASTIC HCI GENERATION FROM ITS ABSTRACT MODEL

Anas HARIRI, Dimitri TABARY, Christophe KOLSKI
LAMIH – RAIHM Group
University of Valenciennes and Hainaut-Cambresis
Le Mont Houy
F-59313 Valenciennes cedex 9
France

{anas.hariri, dimitri.tabary, christophe.kolski}@univ-valenciennes.fr

ABSTRACT
The plastic HCI (Human-Computer Interface) is one of the greatest important elements in a dynamic heterogeneous environment such as context-aware computing including a multitude of devices with diverse platforms, different user knowledge levels, characteristics and expectations, and various work environments. Therefore, the Human-Computer Interaction research concentrates on supporting and generating the plastic HCI. We propose a novel approach which supports the plastic HCI generation basing on its abstract model specified in an interactive system design method.

KEY WORDS
Human-Computer Interaction, Context-Aware Computing, and HCI Generation

1. Introduction

The world is becoming more interconnected; new devices are always appearing and becoming a principle part of our daily life. Mark Weiser [23] first described what we call now pervasive computing [11] in 1991; he noted that the intelligent information devices and a variety of technologies would surround us and that we would can use them at anytime and in anywhere [14]. On the other hand, the multitude of devices including diverse platforms, the different user knowledge levels, characteristics and expectations, and the various work environments, have created new considerations and stakes to be satisfied [12]; we are interested here in the Human-Computer interaction, and more specially the HCI (Human-Computer Interface); the HCI of the pervasive computing must be plastic; in other words it must have the ability to detect information relevant to the platform, the environment and the user for using them during its adaptation and preserving its utility and usability [22]. In this paper, firstly, we will provide a discussion of context-aware computing and how the contextual information can be captured and used later in the different levels of the plastic HCI adaptation. Next, we will propose an approach which supports and generates the plastic HCI. Finally we will evaluate our contribution by applying it to an example of a tourist guidance system with a view of validation.

2. Context-awareness

Schilit and Theimer [19] were the first who described an interactive context-sensitive system (context-aware computing): this system has used the contextual information, such as localization, neighbors' locations and changes in the context. Next work [3], in this research field, has used this preceding information and other information like time, physical environment characteristics, user knowledge and skill level, etc. Therefore the context is defined as information that can be used to characterize the situation of an entity (platform, environment or user) [1]. Moreover, the context-aware computing is the use of the context in order to provide information and services to the user. Figure 1 illustrates an example of a context-sensitive application. According to [6], the context-aware computing has three significant behaviors:

– Contextual information capture;
– Representation of information and services provided to the user;
– Automatic execution of a service.

We can distinguish three important contextual information categories [5] in the Human-Computer Interaction field:

– Information relating to platform, such as the characteristics of the target device (processor, memory and screen size), the connectivity networks and the available peripheral devices of interaction.
– Those relating to user, such as his or her profile, knowledge level, physical localization, current activity (leisure, work) and the proximity of his or her neighbors.
– And those relating to environment like noise level, luminosity, temperature and temporal information.

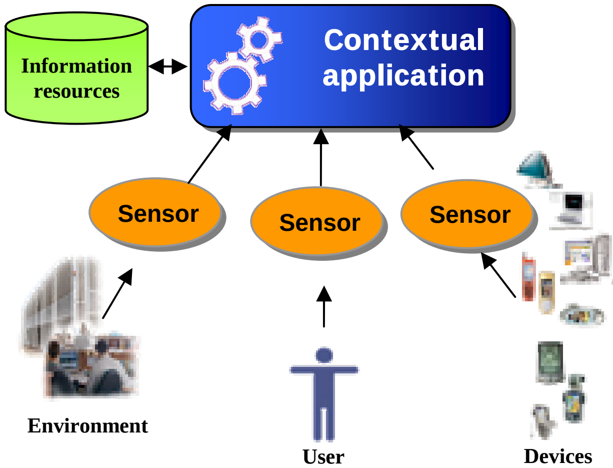


Figure 1: Contextual application

3. From context-awareness to HCI plasticity

In the context-aware computing, we are interested in the context-aware user interface (in other words the plastic HCI); it is generally an interface which can receive contextual information concerning the target platform, the physical environment, and the final user; such as target platform properties, possible connectivity networks, environmental noise level, user localization, the proximity of other users and devices, etc. Then the HCI can use these information during the adaptation to the usage context (target platform, execution environment and final user), and during the execution the plastic interface has the ability to react dynamically to contextual changes, but it is necessary to preserve the interface utility and usability [17]. Thus a target, in the plastic HCI generation, is defined by the triple (platform, environment and user) where [4]:

– The platform is the hardware and software structure; e.g. personal assistant or cellular phone. Underlying the Human-Computer Interaction, we are interested in the platform characteristics such as screen size, processing power, and interaction and communication media.
– The environment refers to the physical environment of interaction. It is described by information about luminosity, noise level, geographical localization, etc.
– The user is generally described by his or her physical and cognitive capacities and competences [13].

Plasticity thus seems as an adaptation form of "Action - Reaction" type [22] where:

– Action refers to the changes which occur in the usage context (platform, environment and user);
– Reaction indicates the procedures which are carried out automatically by the system and/or the user to realize the HCI adaptation with preserving its utility and usability.

4. Plasticity levels

During the adaptation, plasticity is generally based on these three important levels:

- **Adaptation mechanism level:** At this level, the plasticity can be composed of two capacities: adaptability and adaptivity [4]:

– The adaptability is the HCI capacity to be modified by the user;
– The adaptivity is the HCI capacity to be adapted automatically without any intervention from the user.

- **Temporal level:** The adaptation can be carried out at design time (primary adaptation) or at run time (alive adaptation) (see Figure 2). In the first case, the adaptation is carried out, during the system design and before the distribution of the executable code, by choosing HCI components suitable to the target platform, and the adaptation processes finish with considering the user preferences and physical environment characteristics. But in the second case, the adaptation is carried out at run time; thanks to UIML (User Interface Markup Language) [10] which has a markup structure easy to modify, the HCI generator can return to the interface code in UIML and can modify it according to the nature of the action or the change in the usage context; e.g. during the investment of a system supported with a plastic interface, suddenly the environmental noise level increased, the system captures this change in the environment context, then a re-adaptation order is sent to the HCI generator which replies to this change by rebuilding partially the HCI through adding a sound component to the *<Structure>* part in the interface code; the new code is transferred later to the UIML server which distributes it to the target platform; of course this sound component will allow sending sound messages to the user for helping him/her to continue working with this plastic interface in spite of this environmental noise.

In certain cases, the interface is reproduced completely by the HCI generator; e.g. a new user opens his/her session on the same machine, consequently, the user context is definitively or temporarily lost and the HCI is reinitialized for this new user with his/her preferences and characteristics. Therefore the alive adaptation is very significant and necessary for remaining the interface interactive and suitable to the triplet (Platform, environment and user) [20] with preserving its utility and usability.

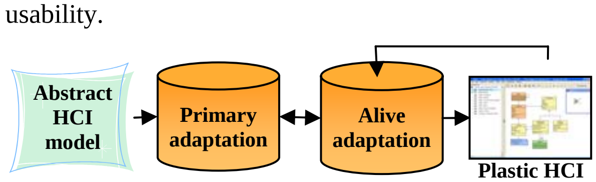


Figure 2: Temporal plasticity level

- **Generation level:** During the plastic HCI generation, we note that the target platform may be known or not to the HCI generator. If it is known, so the interface will be able to be adapted to the particular characteristics of this platform after checking it; e.g. the target platform is a PC but without mouse, so that during the adaptation to this platform, it is necessary to concentrate on other interactive devices which can replace the role of the unconnected mouse. On the other hand, if the target platform is unknown, the HCI generator asks this latter to identify itself, to send its special characteristics and to indicate its interaction devices; then, an adaptation is carried out if this target platform was well recognized, else the HCI generator chooses general and common components for establishing a general plastic HCI version without any specification, as it is presented in Figure 3.

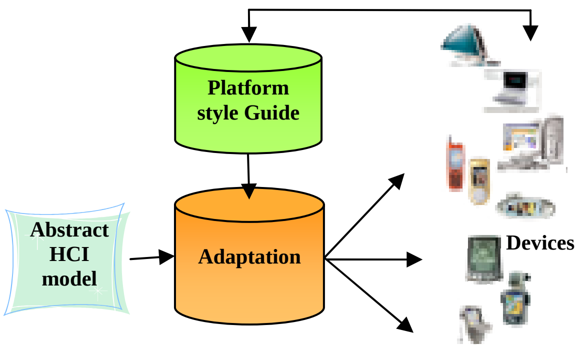


Figure 3: Detection of the Platform characteristics

5. Plastic HCI Generation from its abstract model

Our contribution consists in generating the plastic HCI from its abstract model established in an interactive system design method like TOOD [21]. This interface must adapt dynamically to the target platform, the execution physical environment and/or the end-user. We use a markup user interface description language to define the plastic HCI generated by our approach. Therefore, we have chosen UIML since it allows generating several heterogeneous graphic interfaces for the purpose of solving the multi-platform HCI problem. Currently, UIML has become a standard considering its genericity as well as its adaptability to the graphic interfaces of the most known platforms (Java, HTML, WML, and Voice).

Our approach aims, initially, to generate a primary version of the HCI adapted to the platform, after the identification and the recognition of this latter, in order to choose the adequate HCI components (see Figure 4). So if the platform is unknown, the HCI generator asks it to send its identification, its special characteristics and information about its own interaction devices; consequently, if this platform is well recognized, an adaptation is carried out. On the other hand if the platform is not known and can not be recognized, the HCI generator continues to generate the HCI by choosing general and common components, without any specification for a device. However, the plastic HCI generator is supported by a style guide [2] whose a data-base containing the most known platforms characteristics and specifications, in addition to the HCI component classifications (common or special for a platform). We have proposed, in [8][9], the last stage of the TOOD method basing on rules which ensure generating the executable code, using UIML, of the HCI adapted to the target platform (note that several papers in existing literature propose other notation, such as CTT [16][18]). After choosing HCI components suitable to the target platform, the properties of these components must be adjusted according to the user preferences and characteristics, and the work environment parameters. Finally this HCI code in UIML is transferred to the UIML server which distributes this concrete plastic HCI to the target platform.

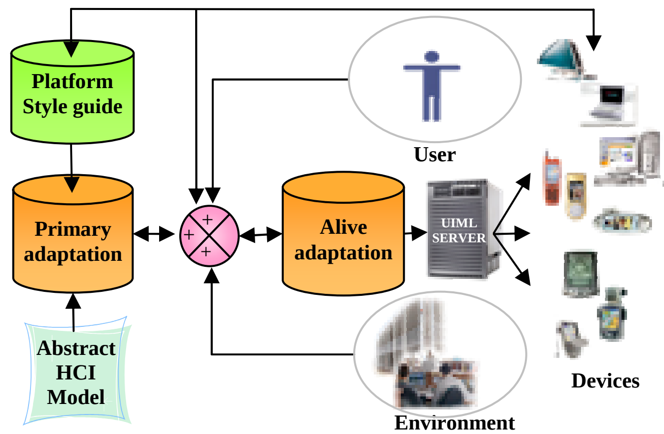


Figure 4: Plastic HCI generation Infrastructure

During the execution, if a change takes place on the level of the platform, the environment and/or the user, then a new adaptation will be necessary to realized; during the adaptation processes the HCI may be reproduced completely or partially, consequently the interface stays always interactive, usable and suitable to the triplet (platform, environment and user).

6. Application: Simulation of a tourist guidance

We will now demonstrate the use of our approach through an example of a tourist guidance system. We present this application using a scenario which we have developed in our laboratory. Then we discuss the different usage context levels. Finally we apply our approach to this simulated tourist guidance system.

6.1 Scenario

The town council of a pretty touristy city decided to place at the disposal of the visitors a virtual 3D guidance system with the possibility of choosing the visit type. During the visit, this guidance system shows the way to be followed with the important information about each place beside the visitor. Also the tourists, in this city, can use this system to find a place such as restaurant or hotel close to them, to obtain information about a place, to know places advised to visit, etc. The HCI of this guidance must thus be able to adapt to the platform on which it is manipulated (PC, PDA, cellular phone…). Moreover this HCI must consider the visitors needs, by taking into account the work environment characteristics (noise, light…) and those of the user (age, knowledge level…) [15]. In other words the HCI must be plastic or sensible and adaptable to the usage context (platform, environment and user).

6.2 Usage context

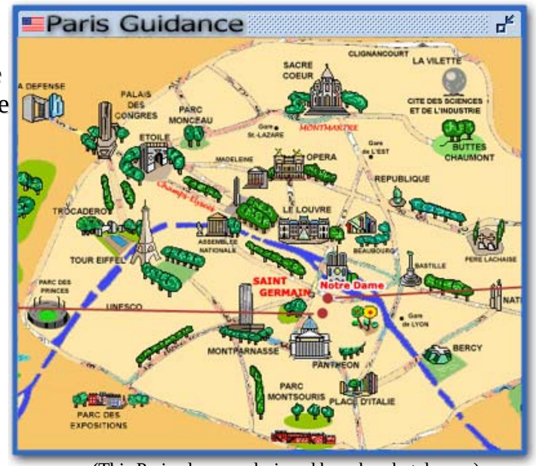
We discuss here the different usage context levels which must be considered by the plastic HCI of this guidance system during its generation and execution:

- **User context:** the visitor who uses this guidance system may be a child or an adult. In the first case, the interface must be simple and very easy to understand, with animated drawings (if the target platform supports running it); of course the guidance interface must show more clearly the places that interest the children such as resort places. In the second case, for the adult, the HCI must contain more details about the way, the places and the resorts, and must be supported by choice possibilities (see Figure 5). Moreover this HCI considers the visitor's profession for concentrating on information which interest the user during his/her visit of a place, e.g. during the visit of an ancient restaurant, the HCI provides information about the architecture arts of this restaurant to an architect visitor. Furthermore the HCI must change its language according to the user language since the visitor comes from the all world. On the other hand, the HCI, for the handicapped visitor, must use special colors, sound messages and/or vibrating movements according to the handicap state.

- **Environment context:** since the guidance system works in everywhere of the city, so that it must well adapt to changeable city environment characteristics. Moreover it needs detecting the user location therefore the guidance has to use a location detection system (such as GPS system or Ad hoc network [7]). After the detection of the environment parameters, the HCI can adapt to the later and can provide the visitor with useful information; e.g. during the visit of a place congested, the HCI must use much more sound messages but in a quiet place such as a restaurant, the vibrator must be used (if the platform has this feature).

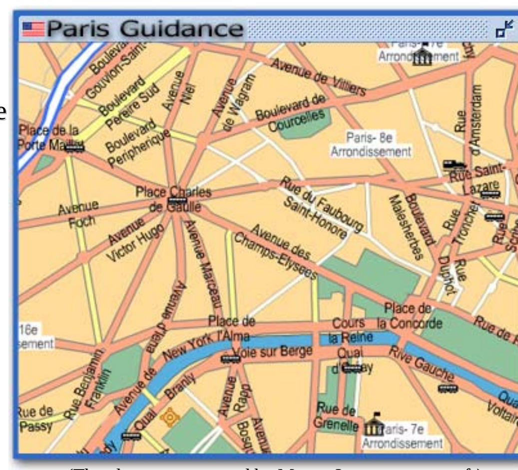
Moreover the guidance can detect the neighboring visitors' locations and contact them if it is necessary, by basing on a location detection system; e.g. if a child with his family visits the city, suddenly he loses his companions, in this case the HCI will use its location detection system for determining the parents' current place and will contact them for helping their child. The HCI adaptation to the environment depends on the alive adaptation (at run time) more than the primary adaptation (at design time), since the environment characteristics are not stable and change quickly.

Possible guidance HCI for a child



(This Paris plan was designed by sahuc-hotels.com)

Possible guidance HCI for an adult



(The plan was generated by Mappy Inc www.mappy.fr)

Figure 5: The guidance HCI adapted to the visitor age

- **Platform context:** our application must work with the most known platforms such as PC, PDA, cellular phone, etc. Thus the HCI must well adapt to the target platform even if possibly it is unknown, our approach ensures this adaptation by using UIML as a multi-platform HCI description language. Firstly the target platform is identified and then the plastic HCI generator does the adaptation (see Figure 6); once the platform characteristics change, the HCI sends a re-adaptation order to the HCI generator for preserving the interface utility and usability.

6.3 Using our approach

The abstract model of the tourist guidance HCI was described last in an interactive system design method; this abstract model contains interaction and user tasks models.

Possible main guidance window in HTML platform

Possible main guidance window in Java platform

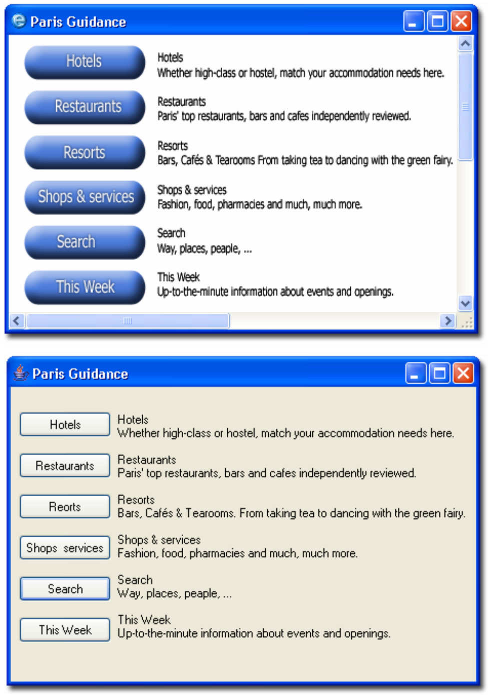


Figure 6: Screen-shots of the main tourist guidance window in Java and HTML platforms

Firstly, the HCI generator identifies the target platform (e.g. the platform is PDA); in this time the style guide provides the characteristics of this known platform after verifying them (e.g. Screen size is 120x180 pixels, the color depth is 265 colors and so on), now the generator can select the adequate components in order to start building the HCI adapted to this target platform. Figure 7 illustrates the HCI code of the main window, in UIML, containing selected components adequate to the PDA.

```
<uiml>
 <interface>
 <structure>
<!-- Defining the main window -->
< part id="main_w" class="TWindow">
<part id="hotels_b" class="TButton"/>
<part id="hotels_l" class="TLabel"/>
<part id="restaurants_b" class=" TButton"/>
<part id="restaurants_l" class=" TLabel"/>
<part id="resorts_b" class=" TButton"/>
<part id="resorts_l" class=" TLabel"/>
<part id="services_b" class=" TButton"/>
<part id="services_l" class=" TLabel"/>
<part id="search_b" class=" TButton"/>
<part id="search_l" class=" TLabel"/>
<part id="thisweek_b" class=" TButton"/>
<part id="thisweek_l" class=" TLabel"/>
</part>
</structure>
<style>
<!-- Defining the target platform-->
<property part-class=" TWindow" name="rendering">Frame</property>
<property part-class=" TButton" name="rendering">Button</property>
<property part-class=" TLabel" name="rendering">Label</property>
<!-- Defining the main window's size and color-->
<property part-name="main_w" name="size">120,160</property>
<property part-name="main_w" name="color">FF2100</property>
<style>
 </interface>
</uiml>
```

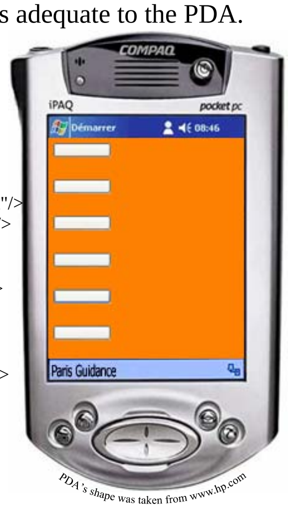


Figure 7: The tourist guidance's main window after the primary adaptation to the PDA platform

Next, the generator displays a dialog box allowing the user to indicate his/her age, knowledge level and language, then the generator modifies some component properties, or adds or removes some components for adapting the HCI to this user. Figure 8 shows the main window HCI code after it has adapted to the user's language and characteristics.

```
:
<!-- Defining the components' captions
according to the user's language -->
<property part-name="main_w"
name="text">Paris Guidance</property>
<property part-name="hotels_b"
name="label">Hotels</property>
<property part-name="hotels_l"
name="text">Whether high-class or hostel,
match your accommodation needs here.
</property>
<property part-name="Restaurants_b"
name="label">Restaurants</property>
<property part-name="Restaurants_l"
name="text">Paris' top restaurants, bars and
cafes independently reviewed. </property>
:
:
```

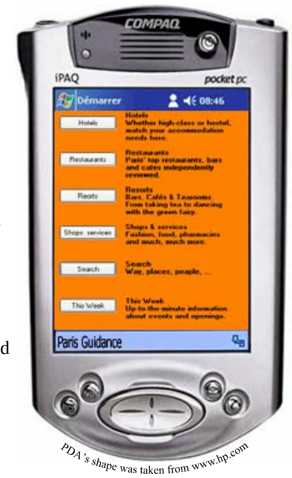


Figure 8: The tourist guidance's main window after the adaptation to the user

Now, the sensors begin sending information about the actually environment, so that the HCI component properties are modified for being suitable to these environment parameters. Figure 9 illustrates the modifications in the font and color properties, according to high environmental luminosity. Finally the UIML server sends this executable HCI code to the target platform for beginning investing this tourist guidance system by this end-user. At anytime a new adaptation is carried out, when the generator detects usage context change or when the HCI sends a re-adaptation order.

```
:
<!-- Changing the components' font size and
color according to the environment -->
<property part-name="main_w"
name="color">FFFFFF</property>
<property part-name="hotels_l"
name="font">ProportionalSpaced-Bold-14
</property>
<property part-name="restaurants_l"
name="font">ProportionalSpaced-Bold-14
</property>
<property part-name="resorts_l"
name="font">ProportionalSpaced-Bold-14
</property>
<property part-name="services_l"
name="font">ProportionalSpaced-Bold-14
</property>
<property part-name="search_l"
name="font">ProportionalSpaced-Bold-14
</property>
<property part-name="thisweek_l"
name="font">ProportionalSpaced-Bold-14
</property>
:
:
```

Figure 9: The tourist guidance's main window after the adaptation to environment parameters

7. Conclusion

In this paper, we have proposed an approach for supporting and generating the plastic HCI which is one of the greatest important elements in context-aware computing; such interface must have the capacity to use contextual information to adapt to the target platform parameters, the work environment characteristics and the end-user expectations. We designed and built this approach to make the plastic HCI capable easily to answer to changes in the usage context during its execution. Then we demonstrated the use of our approach through a simulated tourist guidance system. Our research intends to enrich this approach by considering a large variety of contexts and to develop software tools supporting this approach and the building of more complex applications.

Acknowledgements

We would like to thank the FEDER and the Nord-Pas-de Calais Region for supporting this research.

References

[1] G. Abowd & E.D. Mynatt, Charting Past, Present, and Future Research in Ubiquitous Computing, *ACM Trans. Computer Human Interaction, 7*(1), 2000, 29–58.

[2] F. Bodart & J. Vanderdonckt, Expressing Guidelines into an Ergonomical Style Guide for Highly Interactive Applications, *in Adjunct Proc. ACM Conf. on Human Factors in Computing Systems INTERCHI'93*, ACM Press, New York, 1993, 35-36.

[3] P. Brown, J. Bovey & X. Chen, Context-aware applications: from the laboratory to the marketplace, *IEEE Personal Communications, 2*(1), 1997, 1–9.

[4] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon & J. Vanderdonckt, A unifying reference framework for multi-target user interfaces, *Interacting With Computer, 15*(3), 2003, 289-308.

[5] Anind K. Dey, Providing Architectural Support for Building Context-Aware Applications, *PhD thesis*, College of Computing, Georgia Institute of Technology, December 2000.

[6] Anind K. Dey, Understanding and Using Context, *Personal and Ubiquitous Computing Journal, 5*(1), 2001, 4-7.

[7] S. Desilva & S.R. Das, Experimental Evaluation of a Wireless Ad Hoc Network, *Proc. 9th International Conf. On Computer Communications and Networks (IC3N),* Las Vegas, 2000.

[8] A. Hariri, Multi-Platform HCI Specification and Design, *AISIH master thesis*, LAMIH, university of Valenciennes, France, September 2004.

[9] A. Hariri, D. Tabary & C. Kolski, Model Derivation Principles for Specification and Design of Multi-Platform HCI, *Proc. International ACIDCA-ICMI'2005 Conf. on Machine Intelligent*, Tozeur, Tunisia, November 5-7 2005.

[10] Harmonia Inc & Virginia Tech *are the responsible of the UIML language (www.uiml.org).*

[11] K. Henricksen & J. Indulska, a Software Engineering Framework for Context-Aware Pervasive Computing, *Proc. International Conf. Pervasive Computing and communications,* Florida, USA, 2004.

[12] T. Kindberg & A. Fox, System Software for Ubiquitous Computing, *IEEE Pervasive Computing, 1*(1), 2002, 70–81.

[13] C. Kolski, C. Petit-Roze, A. Anli, M. Abed, E. Grislin-Le Strugeon, H. Ezzedine & A. Trabelsi, La plasticité vue sous l'angle de la personnalisation ou selon les besoins vis-à-vis de l'information transport, *Journées thématiques de l'AS Plasticité du RTP 16 IHM*, Namur, Belgium, August 2004.

[14] W.E. Mackay, G. Velay, K. Carter C. Ma & D. Pagani, Augmenting Reality: Adding Computational Dimensions to Paper, *Communications of the ACM, 36*(7), 1993, 96-97.

[15] T. Moran & P. Dourish, Context-Aware Computing, *a Special Issue of Human Computer Interaction, 16*(2-4), 2001.

[16] G. Mori, F. Paternò & C. Santoro, Design and Development of Multi-device User Interfaces through Multiple Logical Descriptions, *IEEE Transactions on Software Engineering*, August 2004, 507-520.

[17] J. Nielsen, *Usability engineering* (Boston, Academic Press, 1993).

[18] F. Paternò & C. Santoro, One Model, Many Interfaces, *Proc. 4th International Conf. on Computer-Aided Design of User Interfaces*, Valenciennes, France, May 2002.

[19] B. Schilit & M. Theimer, Disseminating active map information to mobile hosts, *IEEE Network, 8*(5), 1994, 22–32.

[20] S. Sire, Point de vue sur la plasticité, *Journées thématiques de l'AS Plasticité du RTP 16 IHM*, Namur, Belgium, August 2004.

[21] D. Tabary, Contribution à TOOD, a Model-Based method for interactive systems specification and design, *PhD Thesis*, University of Valenciennes, France, December 2001.

[22] D. Thevenin, J. Coutaz, Plasticity of User Interfaces: Framework and Research Agenda, *Proc. of Interact' 99 7th IFIP International Conf. on Human-Computer Interaction*, Edinburgh, Scotland, 1999, 110-117.

[23] M. Weiser, The computer for the 21st century, *Scientific American, 265* (3), 1991, 94-104.