Modeling Multi-Level Context Influence on the User Interface

Tim Clerckx, Jan Van den Bergh, and Karin Coninx
Hasselt University
Expertise Centre for Digital Media
and transnationale Universiteit Limburg
Wetenschapspark 2
BE-3590 Diepenbeek, Belgium
{Tim.Clerckx,Jan.VandenBergh,Karin.Coninx}@uhasselt.be

Abstract

With the increasing availability of mobile devices and their usage by many different kinds of users, the context used by an application on this kind of device, in no longer static. The changes in context can have different kinds of impact on the user interface. In this paper, we refine our earlier work on the specification of context information and its effects within the task model, a model that is used in the earliest stages in the design of interactive applications. We define the different levels on which context can influence the user interface and how the influence on several levels can be represented in the task model. The details are discussed using a case study.

1 Introduction

Emerging technologies like mobile and wearable computing, embedded systems and wireless networks has increased research efforts to investigate how a mobile user can be supported to perform her tasks in a more convenient way. As a result, mobile users and their devices can be influenced by several kinds of context, like location, user, other computing devices in the vicinity...When user interfaces need to be developed for context-aware applications, the notion of context needs to be taken into account in the development process of the interface. We believe context needs to be taken into account at early design stages of the user interface. This is why we choose a model-based approach where abstract models describing the interaction part of a context-aware system need to be gradually transformed into more concrete models in order to obtain a concrete user interface. When we want to develop context-aware user interfaces, where context can have influence on several levels of the user interface, traditional models need to be revised in order to fully support the modeling of context influence on the user interface.

In this paper we discuss how context information can influence the user interface at several distinct levels. We categorize these types of context, and we process a collection of extensions to an existing model used in the design of user interfaces in order to model context influences onto the user interface at several levels.

2 Levels of Context in the User Interface

In this section we will define several levels in user interface design where context information has a different influence on the context-aware interactive system.

We make a distinction between five levels of the user interface where context can have an influence:

Task Level Context can influence the tasks that can be performed by the user [7, 10]. For example, when we consider the experience level of a user as context information, other tasks may be available to this kind of user in comparison to the situation where a novice user is interacting with the system.

Data Level Context can influence the data to be presented to the user. For example, in a tourist guide a map of the visited area can be shown to the user, where the user's current location is indicated with a triangle. The user's location is a constantly changing context and requires updates of the data (coordinate on the displayed map) shown in the user interface.

Dialog Level Context can invoke a transition to a different state of the user interface. For example, when an employee enters his company while wearing his badge, access to the intranet might be enabled without any additional authentication dialog. Another example can

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

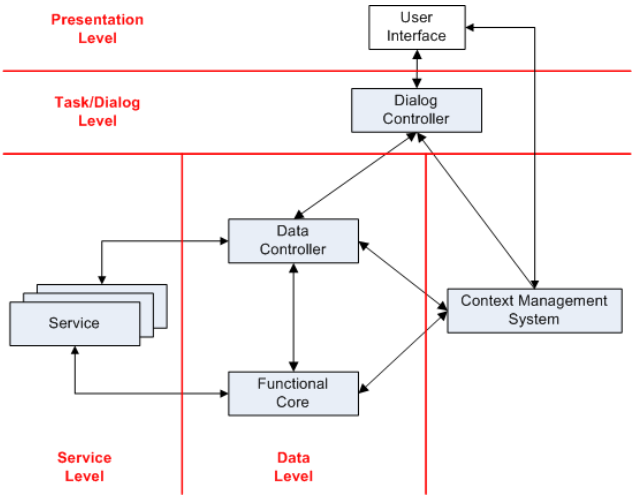


Figure 1. Context levels in a user interface architecture

be that specific information about a statue in a museum is shown when a visitor approaches it within a certain range.

Service Level Services in the vicinity of the user and the user's system can influence the user interface. When a mobile user walks around with her mobile device, services from other systems (e.g. embedded systems providing a communication interface with the system embedding the CPU) may require user interaction (e.g. remote controlling a system with the mobile device). When this is the case, a user interface needs to be presented to the user without disturbing her current task.

Presentation Level Context can influence the way the user interface is presented to the user. When a user is using a desktop computer, keyboard and mouse input is possible, but when a mobile user wants to enter text on a small portable device, speech interaction may be more useful. Another type of context influence at the presentation level is distribution of user interfaces.

The proposed levels indicate how context can have an influence on the user interface in various ways. The recognition of these levels is nothing more than a problem statement at each of these five levels. In traditional user interface design, like the model-based approach, the integration of context in user interface design is a hot topic (section 3). Therefore it is important to acknowledge the distinct levels where the influence of context information on the user interface needs to be considered.

To illustrate the necessity of defining these distinct levels, figure 1 shows an overview of a generic runtime architecture we have implemented describing the different levels. This illustration demonstrates how context, directly and indirectly, has an influence on the user interface. In order to avoid a lot of application specific code to create context-aware user interfaces, we choose to construct a model-driven development process and runtime architecture, where models used in the model-based development of user interfaces are used (e.g. task model, dialog model, data model...). The abovementioned levels of the user interface where context may have an influence implies the requirement of a revision of the models that are currently used in the model-based design of user interfaces. We choose to revise the task model in particular as discussed in section 4. The initial version of the runtime architecture was discussed in [4].

3 Related Work

The design and realization of context-sensitive user interfaces is an active research field within the human-computer interaction community. Different approaches exist to integrate the gathered context information into the user interface.

Sendín et al. [11] use the concept of Aspect Oriented Programming (AOP) as a layer between a context acquisition layer and logical layer in the development of plastic user interfaces.

Calvary et al. [2] define user interface adaptation to the context of use as a three-step process. First the current context needs to be detected. Afterwards a reaction to the current context of use needs to be calculated. Finally, the computed reaction needs to be executed. In this work, context is defined as the aggregate of platform and environmental context. They also defined a development process [2] and a reference framework [1] using a model-based approach where context information (platform + environment) influences several levels of the user interface. Platform context exerts influence on the task model and the abstract user interface description, whereas the broader concept of environmental context only exerts influence on the concrete user interface models.

For a discussion of the the suitability of different model-based approaches and notations for the design of context-sensitve interactive applications, we refer to earlier work [12]. It shows most approaches do not support the specification of environmental influences. One of the approaches that does support environmental influences was introduced by Limbourg et al [8].

They use a model-driven runtime architecture driven by the models used in model-based design of user interfaces to support the development of context-aware user interfaces. The process fits the earlier mentioned reference framework. To incorporate the notion of context in the design process,

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

an extra model, the context model, is added to describe context influence on the user interface. The context model contains a rigidly defined set of possible information about the user, the environment and the platform. The latter contains the same kind of information as the CC/PP and can be linked to the other models. Although all other models have graphical representations in one or more tools, there is no information published on how the related context information is being visualized as part of the other models, nor as a standalone model.

Pribeanu et al. [10] describe how a task model can be used in the development of context-sensitive user interfaces. They acknowledged a revision of existing task notations is needed to incorporate the notion of context in model-based development of context-sensitive user interfaces. In addition they proposed three extensions to hierarchical task notations to describe context-aware user interfaces at the task level. Clerckx et al. [3] built further on this work, introducing a separate decision node, while the Contextual ConcurTaskTrees notation [6] introduced special annotations for tasks with context influences. In this paper, we build on this work to create a more expressive task model covering three levels of context influences discussed in the introduction.

4 Reconsidering the Task Model

The task model is an important model when it comes to model-based development of user interfaces [5]. Because context information is considered to be relevant to the user when it has an influence on her current task [7], we choose to involve the consideration of context at the specification of the task model which is an early stage in the user interface design cycle. In this section, we first discuss how we extended an existing task notation to incorporate the notion of context. To clarify the changes, we describe a case study using these extensions.

4.1 Extending the ConcurTaskTree notation

In earlier work, we discussed some notations regarding some aspects of the problems described in the previous section [3, 6]. In this work we want to describe a way to model the different types of context influences on different levels in the task model. To achieve our goal of modeling context influence at different levels of the user interface, we extend an existing notation, the ConcurTaskTrees notation [9], which is frequently used in model-based design of user interfaces. In this hierarchical notation, tasks are divided into four categories, namely application tasks (tasks entirely performed by the application), interaction tasks (tasks where interaction takes place between the user and the application), user tasks (cognitive tasks entirely performed by the user), and abstract tasks (tasks at a higher level of abstraction). Furthermore, temporal relations between sibling tasks describe how the tasks are executed in time: choice ([ ]), independent concurrency (| | |), concurrency with information exchange (| [ ] |), disabling ([>) , enabling (»), enabling with information exchange ([ ] »), suspend/resume (| >) and order independency (| = |).

We extend this notation with the notion of context at several levels by introducing three new tasks types:

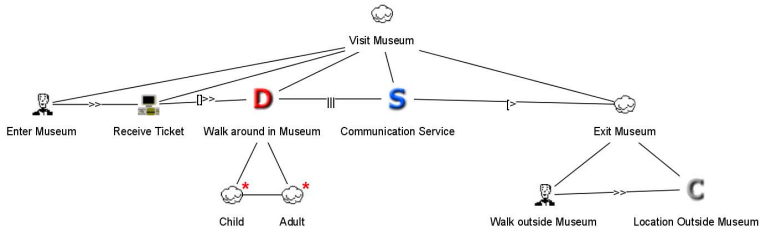
- **Decision Task:** the decision task denotes a junction in the task model where the subtrees describe the subtasks in different contexts of use. A decision task typically is a parent task collecting several subtrees. The decision task takes care of selecting the appropriate subtrees of tasks.
- **Context Task:** the context task is a basic task (a leaf in the task model) where the activity to perform the task is initiated neither by the application nor the user, but by external context influence.
- **Service Task:** the service task is a basic task describing when the tasks provided by an external service may interfere with the current interaction. When an external service becomes available to the application and the service requires user interaction, the way the current user interface has to react on the interference of the service is modeled with this task.

The decision task supports modeling context influence on the interaction at the task level, the context task supports the dialog level, and the service task supports the service level.

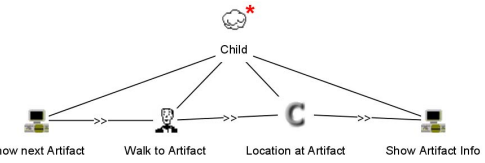
4.2 Case Study

To illustrate the introduction of the three additional tasks in the ConcurTaskTrees notation, we discuss a case study. Because the only purpose of the case is to illustrate the extended task model, we deliberately chose a simple scenario. Imagine the following scenario: in order to assist users in visiting a museum, a user can take a personalized guided tour using her personal mobile device (e.g. a PDA or a smart phone). First of all the user has to enter the museum. When the user byes a ticket at the box office, she receives a virtual ticket on her smart device and the guiding application is downloaded. Afterwards, the guided tour can commence. A distinction is made whether the visitor is a child or an adult. If the visitor is a child, the guide will be a strict sequence of the visitable artifacts in order to make sure that the child has visited the entire exhibition (e.g. useful for school visits). If the visitor is an adult, the visitor will have

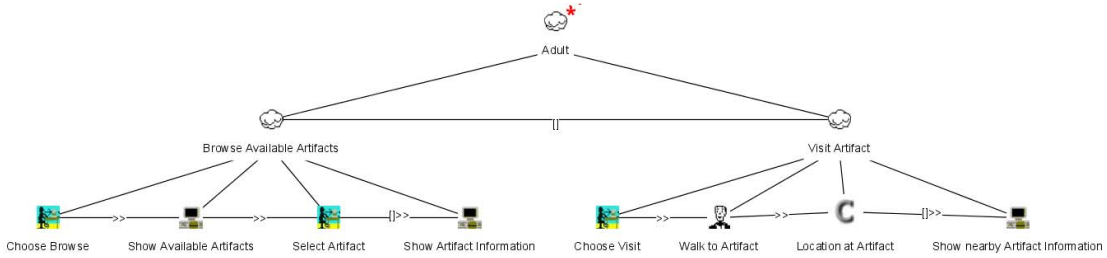
Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE



(a) Main task model



(b) Child user subtask



(c) Adult user subtask

Figure 2. The Museum Visit Task Model

a freer role. A strictly sequential visit is still possible but the adult visitor can also browse through the artifacts in the museum and walk around while detailed information about the nearby artifact is presented to the visitor with the smart device. While the visitor walks around, it is also possible to exchange information with nearby visitors, using the smart device. The visit finishes when the visitor exits the museum.

The task model in figure 2 describes the tasks elicited from the above scenario. Figure 2(a) shows the main tasks. The decision task (D) describes different tasks have to be performed whether the user is a child or an adult. The service task (S) describes a communication service can be performed concurrently (| | |) with the decision task "Walk around in Museum". Figures 2(b) and 2(c) describe the detailed tasks in case of a child user or an adult user respectively. The context tasks (C) in the different models describe context actions regarding location information. For example the context task "Location at Artifact" in figure 2(c) will be implicitly executed when a change in location is detected and information about the artifact near the detected location will be presented through the smart device.

Figure 3 shows an example of the tasks of a communication service. This model describes the tasks that become available to the user when another user walks in the vicinity of the mobile device. The task model in figure 2(a) denotes these extra tasks can be performed concurrently (| | |) with

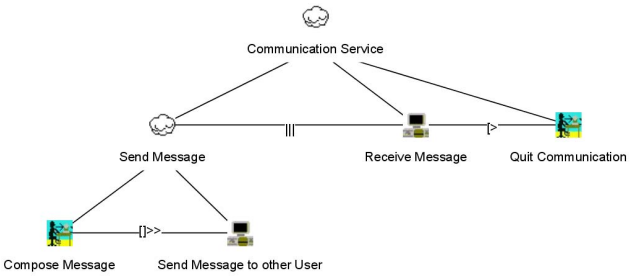


Figure 3. Example of a Communication Service

the user's main task of visiting the museum.

5 Conclusion and Future Work

In this work, we presented a refined task model for the specification of dynamic context-sensitive interactive applications. Five levels were defined where context can influence the user interface. Developing context-aware user interfaces using the model-based approach should therefore take these levels into account. To accomplish this, existing models and notations need to be reassessed. We proposed extensions of the task model to model context influence at

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

three of the defined levels of context influence.

In future work we will extend our approach to support the realization of the two remaining levels of context influence of the user interface: the data level and presentation level. This will require the integration of context influences in other models. The data level uses fine-grained context information in order to provide the user with up to date context information (e.g. user's location on a map). Data level context influences can be specified for instance as meta-data in UML class diagrams [13]. Context influences on the presentation level (e.g. dependent on the current user) should be handeled in a less abstract way than in a task specification. This can be tackled by associating stylesheets to a specific context.

6 Acknowledgements

The authors would like to thank Chris Vandervelpen for his contributions to the implementation of the architecture. Part of the research at EDM is funded by EFRO (European Fund for Regional Deverlopment), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT). The CoDAMoS (Context-Driven Adaptation of Mobile Services) project IWT 030320 is directly funded by the IWT (Flemish subsidy organization).

References

[1] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, and J. Vanderdonckt. Plasticity of user interfaces: A revised reference framework. In *Task Models and Diagrams for User Interface Design*, pages 127–134, Bucharest, Romania, July 18-19 2002. TAMODIA 2002.

[2] Gaëlle Calvary, Joëlle Coutaz, and David Thevenin. Supporting context changes for plastic user interfaces: Process and mechanism. In *Joint Proceedings of HCI 2001 and IHM 2001*, pages 349–364, Lille, France, 2001.

[3] Tim Clerckx, Kris Luyten, and Karin Coninx. Generating context-sensitive multiple device interfaces from design. In Robert J. K. Jacob, Quentin Limbourg, and Jean Vanderdonckt, editors, *CADUI 2004*, pages 281–294. Kluwer, 2004.

[4] Tim Clerckx, Kris Luyten, and Karin Coninx. Designing interactive systems in context: From prototype to deployment. In *People and Computers XIX - The Bigger Picture, Proceedings of HCI 2005: The 19th British HCI Group Annual Conference, September 5-9 2005, Napier University, Edinburgh, UK*, pages 85–100. Springer, 2005.

[5] Paulo Pinheiro da Silva. User Interface Declarative Models and Development Environments: A Survey. In Philippe Palanque and Fabio Paternò, editors, *DSV-IS*, volume 1946 of *Lecture Notes in Computer Science*, pages 207–226. Springer, 2000.

[6] Jan Van den Bergh and Karin Coninx. Contextual concurtasktrees: Integrating dynamic contexts in task based design. In *PerCom Workshops*, pages 13–17. IEEE Computer Society, 2004.

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

[8] Quentin Limbourg and Jean Vanderdonckt. *Engineering Advanced Web Applications*, chapter UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence. Rinton Press, December 2004.

[9] Fabio Paternò. *Model-Based Design and Evaluation of Interactive Applications*. Springer Verlag, ISBN: 1-85233-155-0, 1999.

[10] Costin Pribeanu, Quentin Limbourg, and Jean Vanderdonckt. Task Modelling for Context-Sensitive User Interfaces. In Chris Johnson, editor, *Interactive Systems: Design, Specification, and Verification*, volume 2220 of *Lecture Notes in Computer Science*, pages 60–76. Springer, 2001.

[11] Montserrat Sendín and Jesús Lorés. Towards the Design of a Client-Side Framework for Plastic UIs Using Aspects. In *Proceedings of International Workshop on Plastic Services for Mobile Devices (PSMD)*, Rome, Italy, September 2005. Interact 2005.

[12] Jan Van den Bergh and Karin Coninx. Model-based Design of Context-Sensitive Interactive Applications: a Discussion of Notations. In *International Workshop on Task Models and Diagrams for user interface design 2004 (TAMODIA 2004)*, pages 43–50, Prague, Czech Republic, November 15–16 2004.

[13] Jan Van den Bergh and Karin Coninx. Towards Modeling Context-Sensitive Interactive Applications: the Context-Sensitive User Interface Profile (CUP). In *SoftVis '05: Proceedings of the 2005 ACM symposium on Software visualization*, pages 87–94, New York, NY, USA, 2005. ACM Press.

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE