A Theoretical Framework for Specifying and Analyzing Context-Aware Adaptation

Vivian Genaro Motti, Nesrine Mezhoudi, Jean Vanderdonckt
LILab – Louvain Interaction Laboratory - Université catholique de Louvain
Place des Doyens 1 – Louvain-la-Neuve 1348

vivian.motti@uclouvain.be

ABSTRACT
An effective adaptation of user interfaces approximates technological benefits to the actual needs, wishes and requirements of end users. Today the significant heterogeneity of contexts of use, given mainly by the diversity of devices and the easier access to technology, enables an interaction from many distinct environments and covers different application domains and users' profiles. However the majority of the applications currently available still rely in a conventional context of use, i.e. an able-bodied user, with a Desktop PC in stable environment. Thus several usability issues are often found, requiring a deeper investigation of how adaptation can be efficiently defined and analyzed regardless of application domains. This paper proposes a theoretical framework that considers dimensions of context and adaptation to support stakeholders in the specification and analysis of context-aware adaptation.

Author Keywords
Context-aware Adaptation, Adaptivity, Adaptability.

ACM Classification Keywords
D.2.2 [**Software Engineering**]: User interfaces. H.1.2 [**Information Systems**]: Human factors. H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems. H5.2 [**Information interfaces and presentation**]: User Interfaces – *User-centered design*.

General Terms
Human Factors; Design.

INTRODUCTION
The universal access in a current landscape of ubiquitous, pervasive and mobile computing, can only be achieved by means of considering the context information and effectively adapting user interfaces. By considering several distinct contexts of use and by providing adaptation in an efficient manner, higher levels of usability and accessibility can be provided. However, due to the fact that the scenarios in which the interaction takes place significantly vary and that the current applications still rely on conventional contexts of use, there exists a significant gap between the real users' needs and what technology actually offers them. In this sense context-aware adaptation aims at providing users, systems that are more suitable to their actual needs and also to the specific characteristics of their contexts of use. Due to the increasing variety of platforms, and easier access to technology, adaptation has been receiving a growing attention since the early 90's. However, even with many studies dedicated to this field, still there is no unified framework able to support stakeholders in the specification and analysis of such applications. Moreover, the results achieved so far do not have a general-purpose that covers multiple application domains.

This paper presents a theoretical framework for context-aware adaptation, composed by two modules that support the development, analysis, evaluation and comparison of adaptive and adaptable applications in an integrated and flexible manner. This framework includes: a reference framework (CARF) and a design space (CADS). To orient the development phases considering the involvement of developers of different profiles, we abstracted relevant concepts of adaptation, and categorized them in specific dimensions of varied granularity levels, aiming to provide a framework that is intuitive and also easy to use.

This paper is organized as follows: the Section 2 discusses related works; Section 3 defines the requirements and the design decisions; Section 4 presents the theoretical framework; Section 5 presents and discusses the results, provides final remarks and future works.

RELATED WORKS
Although several works have been dedicated to the domain of context-aware adaptation, they focus on a specific dimension of adaptation at a time.

Concerning Design Spaces for adaptation, a design space for context-awareness has been proposed by Vanderdonckt et al. (2005). They identified the main challenges for context-aware UIs and principles that must guide its implementation. They also proposed a design space focusing on model generation and adaptation. Our work inherited a lot from this design space, however we

separated concepts that are simply *declarative* (e.g. possible agents *who* trigger adaptation) in the CARF, from *comparative* concepts (i.e. that enable comparison among their granularity levels) in the CADS.

Concerning Frameworks for Adaptation, there are several proposals that target at its specific aspects. Dey et al (2001) proposed a framework for facilitating the development of context-aware application, defining as main requirements: the separation of concerns, context acquisition, storage and interpretation, resource discovery, and distributed communication. Our framework, although focusing on theoretical aspects, not only considers context-awareness, but also completes it, including also adaptation aspects. ROAM is a framework that assists developers in implementing applications able to run in multiple devices, and that enables users to migrate their applications across devices without much efforts. It follows as adaptation strategies: transformations, dynamic instantiation and offloading computation. Agents support the migration across devices. This framework considers as context information only the device properties, including: display size, input method and user interface library [Chu04]. PersonisAD is an architectural framework to model and to use context. Its key concern is scrutability, i.e. the users can access and understand their models by using simple operations like access, tell, and ask. Their main contribution is a generalised framework to simplify the creation of ubiquitous computing applications; they focused on modelling the environment and on the distributed and active nature of the models [Ass07]. According to Ardissono et al. (2008) to enhance the flexibility of the workflow in web service composition systems, the context information and the adaptation rules must be explicitly represented in the adaptation logic. They propose CAWE, a framework that manages context-aware applications with a hierarchical representation of the workflow, thus supporting the execution of alternative courses of actions and the context-aware invocation of web services. It considers the adaptation of the UI and the workflow execution.

It is important to investigate specific dimensions of context-aware adaptation in depth; but today there is no methodology that supports CAA broadly and flexibly, covering both its specification and evaluation. To contribute in this sense, we identified related requirements, proposed, and implemented a theoretical framework for CAA.

REQUIREMENTS FOR ADAPTATION

Context-aware adaptation involves four main concepts: the *context information*, the *adaptation process* (e.g. techniques and methods), the *inference* phase (taking optimal decisions, reasoning) and the production of the *adapted final user interface*. These concepts are defined as follows:

The *context information* needs to be gathered, processed and used in a dynamic manner by the *system*, and needs to be selected according to its relevancy by the *developer*. The *adaptation process* needs to be performed by the *system* that stores, retrieves, instantiates and applies adaptation rules, techniques, methods and strategies, based on the previous creation and selection of the *developer*. The *inference* process is based on algorithms that fill adaptation rules according to the context information gathered, and decide the best methods and strategies. While the *developer* creates, provides and selects such algorithms based on his needs and interests, the *system* provides support to perform such operations. To generate *adapted final user interfaces*, the *system* must be able to produce them based on: the results of the inference process and the application of adaptation rules according to the context gathered. Then *users* must be able to access and provide their feedback.

To handle such concepts, requirements from a *system* and *developer* perspective must be fulfilled (see Table 1).

Table 1. Requirements for Defining an Adaptation Process

The System must support:	The Developer must:
Gathering, Processing and Using Context Information	Identify and Select relevant context information, and application aspects
Storing and Querying Adaptation Rules	Identify and Select adaptation rules
Executing Inferences and Reasoning	Identify and Select Adaptation Methods, Techniques and Strategies
Generating the Adapted Final User Interface	Analyze and evaluate the adaptation levels

To meet these requirements, a theoretical framework supporting context-aware adaptation is proposed, being composed by two theoretical modules: CARF and CADS.

THEORETICAL FRAMEWORK MODULES

The theoretical modules abstract adaptation concepts in a way that developers with different profiles are able to develop applications that execute context-aware adaptation. The *Theoretical* framework proposed comprises the CARF and the CADS. The CARF is a context-aware reference framework that lists seven adaptation dimensions and their possible instances. The CADS is a design space for the analysis, evaluation and comparison of coverage levels of adaptation. While the CARF specifies dimensions and their possible instances for implementing adaptation, the CADS specifies analytical dimensions and their respective coverage levels for performing adaptation.

The **CARF** is a graphical representation of relevant concepts for performing context-aware adaptation, thus providing stakeholders an extensive list of possibilities to be considered while creating adaptive or adaptable applications. The CARF, as Figure 3 illustrates, comprises 7 dimensions defined in clockwise sense as:

- **What:** the resources subject to adaptation, i.e. the navigational flow, the contents (of any type, like audio, text, or images) or the presentation;
- **Why:** regarding the software qualities, as to improve the performance or the accessibility level;
- **How:** which are the techniques, methods and strategies applied to adapt (e.g., improve the contrast level, by changing the colors of the background and text, with a smooth transition);
- **To What:** defines the context information that is taken into account to perform adaptation, mainly regarding User, Platform and Environment;
- **Who:** the agents responsible for triggering an adaptation process, such as the end user, a third-party or a developer;
- **When:** if the adaptation occurs at run-time, design-time, compilation time;
- **Where:** adaptation can be performed at the client, server, proxy, or with a mixed approach.

The CARF can be used before the implementation phase, to inform stakeholders about possible alternatives for deciding the application requirements, but also after the implementation phase to analyze further possibilities that were not initially considered.

The **CADS** graphically represents a context-aware design space, highlighting relevant dimensions for adaptation. Orthogonal axes represent dimensions and their respective granularity levels. Adaptive and adaptable applications can be analyzed and compared by means of CADS. Its benefits include: extensibility (once additional dimensions and granularity levels can be incorporated), flexibility (once dimensions can be included or removed, aiming a more focused analysis), exploratory (once it provides a unified view of all dimensions and their coverage levels simultaneously), comparative (once multiple applications can be consistently compared), and descriptive (once all dimensions and their granularity levels are well defined) [Lafon, 2000]. Figure 4 illustrates the CADS applied to the analysis of a given application. The black axes correspond to respective coverage levels regarding each of the CADS dimensions. In clockwise sense, they are defined as follows:

- **UI Component Granularity:** defines the level of granularity that is subject to adaptation, e.g., one edit box, a window, or the complete application;
- **Modality:** refers to the modality types involved in the adaptation process, i.e., intra when the same modality type is considered, inter when the type changes, and multi when multiple types are available;
- **State Recovery Granularity:** refers to the impact in the user interaction, i.e. if the user needs to start a new session, if the user re-starts from the task level, or if the user re-starts just from his or her previous action;
- **UI Deployment:** defines whether it is dynamically executed, or statically executed;
- **User Feedback:** if users can accept or reject the adaptation after (pos) or before (pre) it is performed, or if the users can evaluate it numerically or literally;
- **Technological Space Coverage:** if the technologies involved in the adaptation are all of the same (intra), switch types (inter), or if multiple technologies are involved (multi);
- **Existence of a Meta-UI:** the Meta-UI is an abstract model able to govern the adaptation process; it can be absent, or a meta-UI without negotiation (i.e., pre-defined), with negotiation (i.e., able to evolve), or a plastic meta-UI (capable of automatically adapting across several contexts);
- **Autonomy Level:** designed systems are pre-defined by default and no adaptation is performed, adaptable ones allow users to intervene, adaptive ones are automatically adapted and mixed-approach ones combine user and systems' adaptations.

The CADS and the CARF are complementary approaches that provide a theoretical methodology to support the implementation and analysis of adaptive and adaptable

Figure 1. CARF

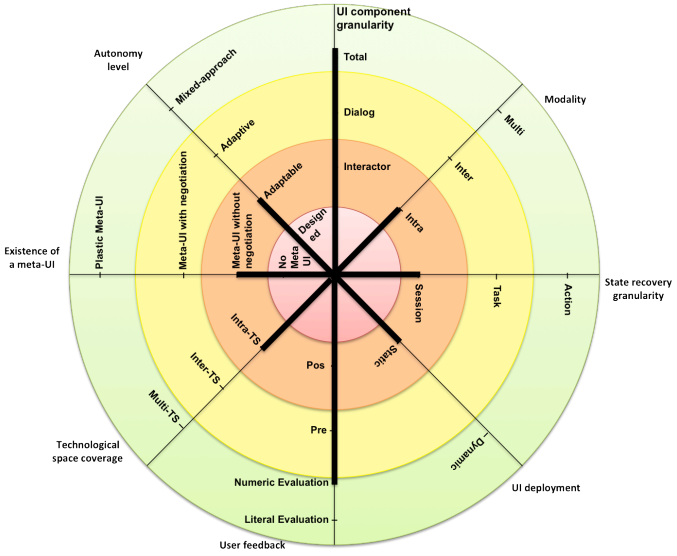

Figure 2. CADS

applications. They enable stakeholders of different technological profiles to take decisions about an adaptation process and to analyze and compare multiple applications.

Table 2. Adaptation Techniques detailed: Dyslexia.

Name	Dyslexia
Reference	http://www.studiostudio.nl/
Description	Given a text content, its font type is replaced by a specific one for dyslexic users
Rationale	The text content is defined, its font type is modified
Example	The user reads an e-book, he is dyslexic, its e-reader automatically changes the font type
Context	According to the user impairments
Advantages	Improve the accessibility levels
Disadvantages	It may affect the performance (due to the text processing and rendering)
Picture	

FINAL REMARKS

To identify potential interaction issues concerning the theoretical framework, a well-defined evaluation plan and evaluation criteria must be established. Our evaluation plan intends to obtain not only the end users' perspective about the interaction with the system, but also the experts' one.

The requirements initially identified aided to guide the definition of the application. The theoretical modules supported the requirements gathering and analysis of study cases, permitting to select the concepts of interest based on an extensive landscape of concepts (provided by the CARF). The specific results consist in the development of the sub-modules, namely: a context-aware reference framework, defining and listing dimensions and possible instances for adaptation (CARF), a context-aware design space that identify coverage levels of adaptation, permitting multiple applications to be analyzed and compared (CADS). As main results, this work defined requirements to perform context-aware adaptation of user interfaces. Besides it also proposes, implements and validates a theoretical framework to support the specification and analysis of context-aware adaptive and adaptable application. Such a framework includes two main theoretical modules. The design space enables many applications to be analyzed and compared regarding their coverage levels for adaptation.

Given the modularity of the framework and its generic approach, we believe it is sufficiently extensive and flexible to equally accommodate further application domains. In this sense we conclude that the application and re-use of the proposed framework is feasible, and that it can be in the future applied to effectively support the development of adaptive and adaptable applications for different domains. We believe that further evaluation efforts are needed to clearly identify the usage, and potential adaptations, of the framework according to specific stakeholder profiles. As future works we intend to provide online versions of the tools in order to make them publicly available.

ACKNOWLEDGMENTS

This work received funding from the European Commission's Seventh Framework Program under grant agreement number 258030 (FP7-ICT-2009-5).

REFERENCES

1. Ardissono, L., Goy, A. and Petrone, G. ''A framework for the development of distributed, context-aware adaptive hypermedia applications'', in AH'08 Proc. of the 5th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, Berlin/ Heidelberg: Springer, pp. 259-262, 2008

2. Assad, M., Carmichael, D. J., Kay, J. and Kummerfeld, B.. 2007. PersonisAD: distributed, active, scrutable model framework for context-aware services. In Proc. of the 5th int. conf. on Pervasive computing (PERVASIVE'07), Springer-Verlag, Berlin, Heidelberg, 55-72.

3. Chu, H., Song, H., Wong, C., Kurakake, S, and Katagiri, M. 2004. Roam, a seamless application framework. J. Syst. Softw. 69, 3 (January 2004), 209-226. DOI=10.1016/S0164-1212(03)00052-9 http://dx.doi.org/10.1016/S0164-1212(03)00052-9

4. A. K. Dey, G. D. Abowd, and D. Salber. 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum.-Comput. Interact. 16, 2 (December 2001), 97-166. DOI=10.1207/S15327051HCI16234_02 http://dx.doi.org/10.1207/S15327051HCI16234_02

5. Lafon, M. B. (2000) Instrumental Interaction: An Interaction Model for Designing PostWIMP User Interfaces. In Proc. of the SIGCHI conference on Human factors in computing systems (CHI 2000), ACM Press, New York, NY, 2000, 446-453.

6. J. Vanderdonckt, D. Grolaux, P. Van Roy, Q. Limbourg, B. Macq, B. Michel, A design space for context sensitive user interfaces, Proc. of 14th Int. Conf. on Intelligent and Adaptive Systems and Software Engineering IASSEi'05, Toronto, Canada, 20–22 July 2005.