Interactive Systems Adaptation Approaches: A Survey

Imen Jaouadi
MIRACL, FSEG
Sfax, Tunisia
jaouadiiimen@gmail.com

Raoudha Ben Djemaa
MIRACL, ISIMS
Sfax, Tunisia
Raoudha.Bendjemaa@isimsf.rnu.tn

Hanene Ben Abdallah
MIRACL, FSEG
Sfax, Tunisia
Hanene.BenAbdallah@fsegs.rnu.tn

***Abstract*— Nowadays, the design and the development of user interfaces impose new requirements as a result of the diversity of platform with specifics characteristic. In this context, several approaches are proposed to develop adaptable user interfaces to context of use. This paper presents a review and classification according to different criteria of the most important research efforts in this field. In the light of this analysis, we propose to develop in the future researches a approach-based on model which adapts the functionality and user interface of an interactive system at run time.**

Keywords-user interface; adaptation; context of use; HCI.

I. INTRODUCTION

The technological progress, such as the miniaturization of microprocessors and the sensors and the success of communicating technologies, opens a wide field of possibilities in the development of Human Computer Interface (HCI) [19]. The user of application wishes to have information whenever and wherever he/she is located. Multiple types of interactive applications are necessary to adapt to the user profile (novice, expert, children, etc.) taking into account technological advances, such as the development of new platforms. The design and the development of interactive systems impose new requirements. It is necessary to design interactive systems in an abstract way, because there is adversity of different platforms with specific features. In this context, several approaches have been proposed in literature to develop adaptable HCI to the context of use [6][12].

This paper reviews major contributions made in the development of adaptable interactive systems to the context of use. The review highlights aspects that are not addressed or no existent in previous research efforts. It concerns those aspects we believe will assume a greater level of importance for future User Interfaces (UIs).

The remainder of this paper is structured as follows. We will discuss relevant related work in Section 2. In Section 3, we will characterize approaches studied on the basis of criteria. Finally, we conclude and outline our future work.

II. RELATED WORK

In this section, we present representative examples of approaches aimed at the adaptation of interactive systems to the context of use. We can distinguish three families of approaches to adaptation: approaches-based on components, approaches-based on models and approaches coupling between models and components.

A. Approaches-based on Models

Thevenin [15] provides a conceptual framework formed by several steps. In this framework, he proposed to add three more models than proposed by Szekely [14], which are: the environment, the user and the platform. In addition, in his research, Thevenin [15] offers a demonstration of his work called: ArtStudio. However, this tool only supports the diversity of platforms. In addition, it does not allow dynamic adaptation. Indeed, adaptation is assisted by the user. Thus, it does not cover all the principles of conceptual framework.

Sottet [13] is considered as a pioneer to have proposed coupling the model driven Engineering and the human computer Engineering. In his approach, Sottet [13] offers Task metamodel, Concept metamodel, User Interface Abstract (UIA) metamodel, and UI Concrete (UIC) metamodel covering Camelon reference framework [3]. In fact, Sottet [13] proposed adaptation controlled by a decision system in order to automate the generation of plastic HCI. This system allows choosing the adequate transformation to a given context. Increasingly, Sottet [13] developed a demonstrator called MARA, which is based on metamodels and transformations. The major drawback of this approach is that we have to create N transformations for N contexts and the decision system selects the most appropriate transformation to a given situation.

Hachani et al. [10] propose a new generic approach for the generation of UI adaptable. The approach is based on the variability [9] and Model Driven Engineering (MDE) [7]. The concept of variability can factorize the common parts to all contextual situations by identifying the variable parts. In this approach, Hachani et al. [10] propose to introduce the context of use at task level. This approach is distinguished by the definition of the generic rules appropriate to all contexts of use. However, Hachani et al. [10] do not specify a detailed description of each context element. Thus, the approach focuses only on the modeling task and context, and does not take the other models proposed by Camelon reference framework [3].

Bouchelligua et al. in [2] propose an approach-based on IDM for plastic HCI. This approach allows the adaptation to the context of use (platform, environment and user) based on parameterized transformation [17]. To apply the transformation, Bouchelligua et al. include in their approaches two abstraction levels: UI Abstract and UI Concrete. Thus, Bouchelligua et al. provide the context metamodels used to adapt the interface: user metamodel, platform metamodel, and environment metamodel. This approach is distinguished by modeling context elements, but it only supports adaptation at UI Abstract and UI Concrete. Other levels of abstractions defined in Camelon reference framework are not considered in this approach. Thus, Bouchelligua et al. have not developed a tool for their approach.

B. Approaches-based on Components

In [1], Balme et al. propose a conceptual framework as a functional decomposition that collects and organizes all the functions necessary for plasticity. The necessary plasticity functions are divided between the *infrastructure* dedicated to capturing the context and *adaptation manager*. Interactive systems are represented by components assemblies and connectors. The adaptation manager decomposes into three functions: *situation identifier*, *adaptation producer*, and *evolution engine*. The situation identifier can observe, match, and synthesize situations. *Evolution engine* select the most suitable components for the current state. *Adaptation producer* responsible for the implementation of adaptation plans. *Component manger* allows the dynamic discovery of components. In addition, Balme et al. [1] have developed a demonstrator for their approach called "*Ethylene*".

The proposed approach has several advantages: it is context aware and allows the user to control the system. However, in this approach we are forced to use a predefined and fixed set of components which are pre developed for specific needs.

In [11], Hariri et al. describe a design process to generate the UI from an Abstract model and/or Task model preserving ergonomic properties of HCI. This process allows the dynamic adaptation in runtime depending on the context of use. They used patterns and integrating a software architecture based on business components. A business component is composed of functional components containing the functional core and presentation components corresponding to the HCI. In addition, this approach is based on the concept of learning, that allows to continue develop the knowledge base of the system at runtime. Thus, the major advantage of this approach permits to adapt the functional core. Nevertheless, this approach has some limitations. In fact, the proposed method is not supported by a global environment; no complete development tool has been developed for the creation and management of design patterns and business. Furthermore, Hariri et al. [11] have not studied the concept of HCI migration to another totally different modality; for example, the migration of a graphical platform to voice platform.

In his work [8], Gabillon is interested in HCI dynamic composition. It is focused on the composition of the Task model from the user goal. This approach uses planning algorithms to dynamically compose the Task model. The HCI are implemented by comets [5]. Each comet is described by an operator or a method of planning. During his work, Gabillon [8] develops a demonstrator of HCI composition by planning called "Compose". In this tool, the contexts of use and ergonomics criteria are modeled manually. The major drawback of this approach it based only on the Task model and the purpose of the user. Thus, it is necessary to automate the inclusion of ergonomic criteria to dynamically compose ergonomic HCI. In addition, the preferences of users are not taken into account.

C. Approaches-based on Components and Models

Comets approach [5] provides a model for plastic interactors that are able to adapt to the context of use. Adaptation is based precisely on the description of an interactor in terms of resources. Resources refer to the terms of needs interactor functionality such as screen space. In [5], Dâassi et al. propose to encapsulate in the same component all Camelon reference framework models [3] and adaptation mechanisms. However, we believe that encapsulate in the same software component (the comet) all model specifications and adaptation mechanisms surcharge component. In addition, adaptation aspect and self-adaptation at the base component may affect the ergonomic usability of the system. Increasingly, Dâassi et al. [5] have not developed a tool to support this approach.

Criado et al. in [4] present a MDE [7] approach to development of adaptable UIs. Indeed, in this approach the UI can regenerate themselves during execution depending on user interactions and application requirements. UI components can be evolved over time through transformations of models changeable and adaptable according to the system events. However, the proposed adaptation process is not automatic; transformations and events are launched manually. Thus, the approach allows only the adaptation of interface model. In addition, Criado et al. [4], propose only six types of actions that can be performed on the component: create, delete, activate, deactivate, interact service and launch event.

III. ANALYSIS OF THE STATE OF THE ART

Table 1 provides visual summation of characteristics criteria are useful to answer respectively to the questions *Adaptation moment*, *Context element*, and *Adaptation level* Vanderdoncket et al. in [18]. The other criteria are inspired (who), (when), (with respect to what), and (what) posed by used to perform comparison between the approaches presented in the state of art (Section I). *Adaptation type*, from the analysis of works presented in Section I and our knowledge in HCI field.

TABLE 1. COMPARAISON BETWEEN APPROCHES REPRESENTED IN RELATED WORK

Approach	**Approach type**	**Adaptation type**	**Adaptation moment**	**Context element**	**Adaptation level**	**Models/functions**	**Learning mechanisms**	**Associated tool**
Thevenin [15]	based on model	Assisted generation	design	platform	UI	Domain, task, UI Abstract UI Concrete, UI Final, platform, environment	No	ARTStudio
Commet [5]	based on model + component	dynamic	design	interaction context	UI	Domain, Task, UI Abstract UI Concrete, UI Final User, Platform Environment, Transition, evolution	No	Camnote ++
Balme [1]	based on component	dynamic	runtime	interaction context	UI	situation identifier, evolution engine component manager, adaptation producer	Meta-iu	Etylene
Sottet [13]	based on model	Statique dynamic	design	platform	UI	Domain, Task, UI Abstract UI Concrete, UI Final, user, platform, environment,quality	No	MARRa
Hachani [10]	based on model	static	design	interaction context	UI	task	No	No
Hariri [11]	Based on component	dynamic	runtime	interaction context	Functional core	Task, UI Abstract, UI Concrete, UI Final	knowledge base	No
Criado [4]	based on model + component	manual	runtime	No	UI	Structural metamodel, visual metamodel, interaction metamodel	No	No
Gabillon[8]	based on component	dynamic static	runtime	interaction context	UI	Domain, task, UI Abstract UI Concrete, UI Final	No	Compose
Bouchelligua [2]	based on model	manual	design	Interaction context	UI	UI Abstract, UI Concrete	No	No

In this article, we identify three main groups of adaptation approach. These are: approach-based on model, approach-based on components and approach-based on components and model.

We can see that the component approach solves the problem of dynamic adaptation. The major drawback of these approaches is that they are based primarily on the use of fixed and predefined set of variants; whereas, the model-based approach allows representing more variants. Therefore, we propose in our future work a based model approach to adapt the HCI to different contexts of use.

We can also arrange approaches depending on the time of adaptation (*Adaptation moment*). Many researchers in the reviewed literature are proposed to adapt HCI at the design phase [2][5][10][15]. Others focused on the HCI adaptation during execution [1][4][8]. It is apparent from the review that adapting HCIs during execution is almost appropriate for the approaches-based on components. In fact, in these approaches adaptation is performed by a components dynamic assembly, while the adaptation of model-based approaches during execution is a very difficult task. It forces the system to generate code for each new adaptation.

In the literature, several approaches are proposed to develop HCI adaptable to the context of use. The major of researchers are restricted to narrow classes of context such as platform [15] or language [10]. This one is due to the difficulty of capturing and processing contextual information. There are approaches that have considered all the contexts of use; even if they did not specify a detailed description of each component of the context of use [8][10]. These lacks do not make these approaches useful in a general context and especially not for applications where personalization information is part of the essential criteria of system operation.

According to [16], an interactive system is composed of four functions: *physical presentation*, *logical presentation*, *functional core adaptor*, and *dialogue controller*. The adaptation of the HCI can take place at all these levels. An adaptation of the physical presentation preserves the nature of physical interactors, but their rendering may be different for other platforms. The adaptation of logic presentation changes the nature of their interactors but not its representational and functional capacity. The Adaptation of dialogue controller changes task scheduling but not their nature. However, adaptation at the adapter functional core result of the changing nature of concepts and exported by the functional core functions. It changes the nature of the tasks and concepts they handle. It is apparent from the review that a lot of research [1][2][4][5][8][10][15] aim to manipulate user interface without concern about the functional part. From the reviewed approaches, only Hariri et al. [11] are concerned with adapting the functional core of HCI. It is based on components. In fact, to our knowledge, there are no model based approaches proposed in the literature which are adapted functional core and the user interface of the interactive system.

In addition, each approach puts forward a number of models to adapt the system to context of use. Thus we find that some approaches focus on the Task model like central model for adaptation [10]. Many also introduced in their approaches all models defined by Szekely in [14]: Domain, Task, UI Abstract, UI Concrete, and UI Final [5][8][11][15]. These approaches have shown how to gradually build different models to define all the characteristics necessary for the construction of the final interface. This consideration favors the adaptation of the whole interactive system and is reflected in the last step in the presentation model.

Thus, we found that some reviewed approaches have proposed mechanisms of learning [11] to improve the predefined knowledge to respond to the needs of users who evolve over time. In addition, the meta-UI dimension represented in the literature [1] determines the level of control available to the user on the system. In fact, the knowledge base and Meta-UI can improve the predefined knowledge by learning the habits and preferences of users. However, most approaches have not taken these two dimensions in their work.

IV. CONCLUSION AND FUTURE WORK

In this article, we studied several approaches and classified the adaptation approach based on different criteria. The criteria are used to extract the shortcomings of the interactive systems adaptation approaches proposed in literature.

In conclusion, the adaptation is not limited to the UI component of an interactive system. The adaptation to the context of use may also impact the functional core. In the light of the criteria mentioned above, it is important to note that although there are many works in the field of research, there is no approach that is based on the model to our knowledge. There is a lack of an approach which adapts the interface and functionality of the interactive system at run time. So, in the future researches we will try to propose an approach that responds to these needs.

REFERENCES

[1] L. Balme, A. Demeure, N. Barralon, J. Coutaz, and J. Calvary, "CAMELEON-RT : a software architecture reference model for distributed, Migratable, and Plastic User Interfaces", Second European symposium on ambient intelligence, EUSAI 2004, Eindhoven, the Netherlands, November 8-11, 2004, pp. 291-302.

[2] W. Bouchelligua, A. Mahfouthi, and L. Benammar, "An MDE approach for user interface adaptation to the context of use", third international conference, HCSE 2010, Reykjavik, Iceland, October 14-15, 2010, pp. 62-78.

[3] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, "A Unifying Reference Framework for Multi-Target User Interfaces", proceeding in Elsevier journal, Interacting with Computers, vol. 15, issue 3, June 2003, pp. 289-308.

[4] J. Criado, C. V. Chicote, N. Padilla, and L. Iribarne, "A model driven approach to Graphical User Interface Runtime Adaptation", international conference on Model Driven Engineering Languages and systems, vol, 641, ISSN 1613-0073, Oslo, Norway, October 5-13, 2010, pp. 49-59.

[5] O. Dâassi, G. Calvary, J. Coutaz, and A. Demeure, " Towards a new generation of widgets for supporting software plsticity: the "comet"", In EHCI- DSVIS'04, the 9th IFIP working conference on engineering for human computer interaction and interactive systems, vol. 3425, Hamburg, Allemagne, October 2004, pp. 306-324.

[6] A. K. Dey, and G. D. Abowd, "Towards a better understanding of context and context-awareness", First international Symposium, HUC' 99, Karlsruhe, Germany, September 27-29, 1999, pp. 304- 307.

[7] J. M. Favre, "Towards a basic theory to modeldriven engineering," In 3rd Workshop in Software Model Engineering 2004, joint event with UML2004, Lisboa, France, 2004, pp. 9-17

[8] Y. Gabillon, M. Petit, G. Calvary, and H. Fiorino, "Automated planning of user interafce composition", International conference on intelligent user interfaces, Palo Alto, CA, Etats- Unis, February, 2011, pp. 3-8.

[9] J. V. Gurp, J. Bosch, and M. Svahnberg, "On the notion of variability in software Product Lines", Working IEEE/IFIP Conference on Software Architecture, August 28- 31, 2001, pp. 45-54.

[10] S. Hachani, S. Dupuy-Chessa, and A. Front, " a generic approach to HCI dynamic adaptation to context," University of Grenoble, CNRS, LIG BP72, 38402 Saint Martin Héres Cedex, France, 2009.

[11] A. Hariri, D. Tabary, S. Lepreux, and C. Kolski, "context aware business adaptation toward user interface adaptation," siwn Communication, vol 3, juin 2008. pp 46-52.

[12] S. J. Pyykkö and T. Vainio, "Framing the Context of Use for Mobile HCI", In International Journal of Mobile Human Computer Interaction, vol. 2, issue 4, October 2010, pp. 1-28.

[13] J. S. Sottet, G. Calvary, J.M. Favre, and J. Coutaz, "Megamodeling and Metamodel-Driven Engineering for Plastic User Interfaces: Mega-UI," Human-Centered Software Engineering", Human Computer Interaction Series, London, 2009, pp. 173-200.

[14] P. Szekely, "Retrospective and challenges for model based Interface Development", Design, specification and

verification of interactive systems 96, Eurographics workshop in Namir, Belgium, June 5-7, 1996, pp. 1-27.

[15] D. Thevenin, J . Coutaz, and G. Calvary, " A reference framework for the development of Plastic User Interfaces", In multi-device and multi-context user interfaces: engineering and applications frameworks, Aug 2005, pp. 29-49, doi: 10.1002/0470091703.ch3.

[16] UIMS," A metamodel for the runtime architecture of an interactive system: the UIMS tool developpers workshop", ACM SIGCHI Bulletin, vol. 24, issue 1, New York, USA, Janury 1992, pp. 32-37.

[17] S. Vale and S. Hammoudi, "Context-aware Model Driven Development by Parameterized Transformation", IEEE internationnal Conference on Web Services, 2007, pp. 302-329.

[18] J. Vanderdonckt, D. Grolaux, P. Van Roy, Q. Limbourg, B. Macq, and B. Michel, "A Design Space for Context-sensitive User Interfaces", ISCA 14th Int. Conf. on Intelligent and Adaptive Systems and Software Engineering, ISCA, Toronto, 2005, pp. 207-214.

[19] M. Weiser, "The Computer for the 21st Century", ACM SIGMobile Mobile computing and communications Review, vol. 3, issue 3, New York, USA, Jully 1999, pp. 3-11.