

Towards a Conceptual Model for UIs Context-Aware Adaptation

Nesrine Mezhoudi, Jorge Luis Perez Medina, Iyad Khaddam and Jean Vanderdonckt

Université catholique de Louvain – Louvain School of Management – Lilab. Louvain-la-Neuve, Belgium

E-mail: {[nesrine.mezhoudi](mailto:nesrine.mezhoudi@uclouvain.be), [jorge.perezmedina](mailto:jorge.perezmedina@uclouvain.be), [iyad.khaddam](mailto:iyad.khaddam@uclouvain.be), [jean.vanderdonckt](mailto:jean.vanderdonckt@uclouvain.be)}@uclouvain.be

Abstract—Today, an effective adaptation of UI is the main requirement to meet end-users needs and improve the UI usability. A significant variety of interaction requirements, given mainly the heterogeneity of context, should be accommodated via an adapted UI meeting users expectations. However current adaptation approaches still considering established rules and guidelines preventing user involvement and decreasing UIs context-awareness. Thus UIs definition and contextualization requires deeper study of how to adapt efficiently with regards to ambient contexts and end-users expectations. This paper proposes a Context-awareness Model (CAM) that considers UI models and context dimensions besides supporting the adaptation specification and the users involvement. The CAM is aimed to support stakeholders for developing and conceptualizing system that accommodate current context-awareness requirement.

Keywords: *framework; runtime adaptation; context-awareness; user involvement.*

I. INTRODUCTION

Current advances in the technological landscape and their rapid growth are creating competitive challenges, as well as new opportunities for HCI communities. Such progresses seem promising to enable the UI to offer tailored interfaces and interaction scenario that corresponds to end-users specific expectations and preferences. Accordingly, adaptation approaches are evolving with technological innovations their purposes are still to increase user satisfaction and result in successful interactions.

Up to date HCI studies are advancing and conveying new adaptation strategies to increase the UI efficiency. By attempting to cut with earlier interfaces that often needed recompilation for upgrades, which incurred increased cost, delay, and risk, UIs shift to a runtime paradigm. User interfaces turn out to be adaptive rather than being user-centered and carry out adaptation in accordance with the end-user preferences as well as the context of use.

Model based user interface benefits were widely discussed in the literature [1, 2, 3, 4, 5]. The advantageous cost's reduction and facility of interchange challenged the HCI community. The aim was to develop system with higher usability and better interaction. However conceptually such solution still lacks a runtime context-awareness. Most of existent approaches were conform to [1] and considers that ones the abstract specification is defined, several instantiation could be derived. In the same ways contexts were defined

through predefined abstract specification. However such solution should be enhanced to support runtime context-awareness and user involvement in order to improve their usability levels and meet present-day requirement.

The purpose of the proposed research is to support runtime adaptation and user involvement by means of models based UI reification at runtime. A state transition execution that is conceptualized in the model resulting context-aware runtime adaptation.

This paper is structured as follows section 1 presents a review of existing works on adaptation and system context-awareness. Section 2 describes CAM a conceptual model supporting UI runtime context awareness and end-users involvement. Section 3 shows two implementations for a car rental case study. The first implementation regards a Flippable UI for internationalization developed in accordance with UsiXML project specifications. The second implementation consists on an adaptive UI. Finally, conclusions and future works are presented.

II. RELATED WORKS AND KEY CHALLENGES

Different theoretical frameworks and models that support systematic context-awareness and that inspired the design decisions and requirements for the computational circumstances were conveyed. The Cameleon Reference Framework (CRF) [1] was introduced to structure only model-based UI approaches according to several levels of abstraction. Knutov [2] suggests a general-purpose adaptive hypermedia AH framework providing reference architecture and defining system criteria to distinguish between these elements. It provides a modular structure to enhance adaptation of web-based systems capabilities.

PersonisAD, [3] conveys an architectural framework to model and to use context, however this framework was limited in terms of domain and context-awareness. Later [4] proposes TriPlet, a computational framework that covers a broad view to support the implementation of multi-dimensional CAA. Three conceptual methods (CADS, CARF and CAMM) have been integrated within a general computational framework that considers in a structured way both context information and adaptation concepts [5]. Despite the broad scope of the framework, its extensible facets can lead to an incoherent instantiation besides the confused synchronization of different supported aspects.

Table 1: This table represents the analysis of related works based on Adaptation concepts and concerns. The dimensions were ranked regarding their support “●” fully explicit support, “◐” implicit support, and empty “ ” when it is not supported or there is no information about.

Related works	Adaptation concepts								Adaptation concerns		
	UI models				Context model			User Involvement (feedbacks)	Adaptation model	Adaptation autonomy	Adaptation technique
	Task	AUI	CUI	Transformation	User	Platform	Environment				
Cameleon[1]	●	●	●	●	●	●	●				◐
PersonisAD[3]			●		●	●	●				●
GAF [2,7]	●	◐			●	●	●		●	◐	◐
Triplet[4]	●	●	●		●	●	●		●		●
Karen [6]	●		◐				●		●	◐	
Sottet[15]	●	●	●	●	●	●	●		●		
Ganneau[14]					◐	◐	◐		●	◐	●
CAM	●	●	●	●	●	●	●	●	●	●	●

Several analyses and studies targeted adaptive systems from a different point of view, most of them focused on the dimensions of adaptation in systems and are specific for distinctive domains such as: medical [6, 7, 8] (medical, hypermedia). For instance, [2] proposed a classification for adaptive hypermedia methods and techniques by highlighting the adaptation process. We noted that on this classification the process is initiated by users and the system is adapted in function of the adaptations goals and technologies, the user features, the context, the application area and then the system methods and techniques. Likewise [6] proposed a framework for categorizing UI adaptation based on two technical descriptions of two AUI key elements: the taxonomy of adaptation describing the ways in which a system's interface can be adapted and the taxonomy of triggers used to decide when and how to change the UI or the system's behavior [6].

The lack of user-centeredness during adaptations entails the user dissatisfaction and degrades the UI quality.

The most commonly cited issues with adaptive UI are the lack of predictability, control, and privacy [9, 10], mainly because those UI adaptations consider prior interaction knowledge (explicit context, domain models) [11, 12].

To our knowledge, there are no frameworks that match with agile principles (such as: incremental, iterativity, user-centeredness) for adaptation. Most of them were focused mainly on the conventional adaptation mode or consider just some fragments for instance to adapt a UI to a user model without being user centered [13, 1, 14, 15]. Moreover supporting recent and usual adaptation strategies from different perspectives allowing the full understanding and the comparison of techniques still a requirement.

An iterative progressive adaptation enhanced by intelligent techniques can meet this shortcoming. The intent is to advance the adaptations and provide systems with the ability to learn and build novel knowledge in an incremental way in view of context changes.

III. A CONCEPTUAL MODEL FOR AGILE ADAPTATION

So far, we focused on two paths to discuss the context of the study and to outline the need for advancing adaptation topics.

Several models addressing the adaptation process, most of them were limited to conceptualize the adaptation rules in a specific way [14] as well in the similar conceptual model proposal [15]. A broader conceptual model was presented by [5], it covers context, adapter, model and adaptation rule.

The reviewed literature allowed an analysis of involved concepts and their characterization. Along with above detailed works, we propose a conceptual model for adaptation aimed to cover main involved features. The model (figure 2) is aimed at supporting an explicit, comprehensible and complete configuration of adaptation concerns and allowing advances and improvements. It is intended to cover the whole involved concepts and determines their relationship and dependency.

Three main packages were identified to distinguish involved classes belonging to different adaptation dimensions (Figure 1).

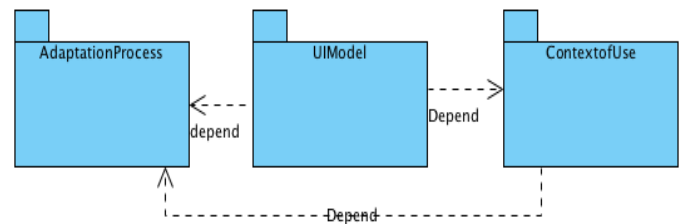


Figure 1 Adaptation main concepts

The Adaptation package is the heart and engine of contextualization, linking all key elements involved for UI adaptation. The UIModel package defines the user interface independently of both adaptation and the context of use. The context of use package corresponds to adaptation triggers and all

contextual factors. In what follow a detailed description of involved element will be presented

A. The Adaptation Package

The model of the adaptation package (figure 2) establishes the adaptation as a model separate from context and interface definitions. This dimension includes all classes related to the adaptation itself. It is intended to give an abstract conceptualization for the adaptation process in term of UI states and transitions.

A “UIState” remains a characterization of a UI model consistent with a context assessment. The state terms values of UI attributes with consideration of the context of use. For instance at a concrete abstraction level, for a phone device context the choice interaction unit for a values number up to 30 is assessed to a Drop-down list.

UI adapted features (for instance, Interactors, Task, AbstractUnit, Widgets, etc.) depend of the considered

abstraction levels from defined UI Models and the values depend on the current context. The “UIState” changes during an adaptation process through a set of “Transitions” recapitalizing the interface changes. A transition presents a set of adaptation rules targeting a set of UI attributes and accommodating a context change.

The “AdaptationRule” is a part of the “Transformation Model” which consists on different mapping models such as reification, translation, reflexion [1]. It consists of one or more “TriggerEvent” initiating an adaptation and a set of action performed to change the “UIState”. For example, we can imagine that an end-user could have an explicit control on the UI definition via his feedback.

The adaptation could be also triggered automatically based on an autonomous decision making process regarding a context assessments.

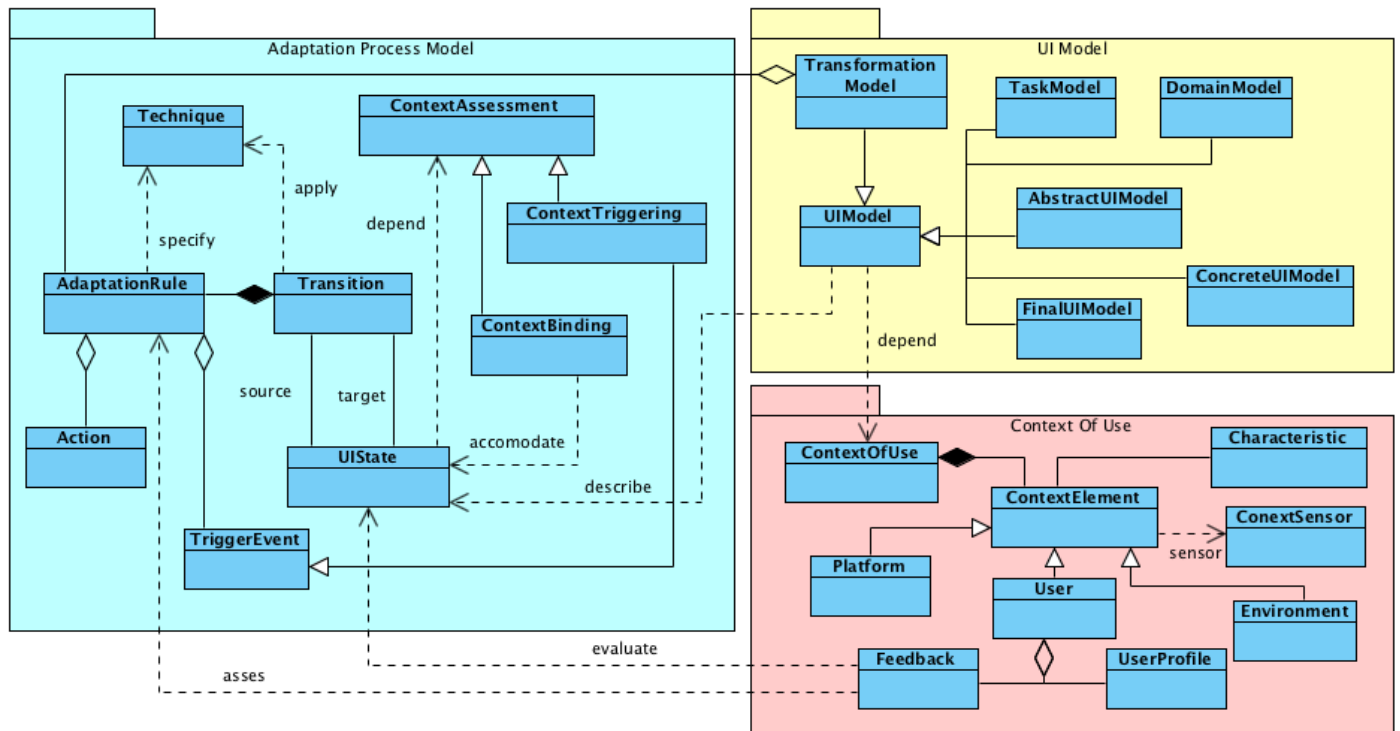


Figure 2. An Unified Adaptation model

B. The Context Of Use package

The Context of Use has been modeled as a specialization of User, Platform and Environment. The figure 2 gives an overview of the main entities modeled by the Context of Use. These entities are intended to identify attributes and proprieties influencing the adaptation process and providing a trigger event for an adaptation.

The “ContextElement” class determines the set of descriptors that can be considered to define context dimensions; in some cases of adaptive UI, features values are determined via “ContextSensors”. As the context is a

composition of information gathered regarding different dimensions, it contributes to the definition of adaptation rules conditions. The “ContextElement” defines the context of use as well they present the trigger for all “AdaptationRules”.

The “UserModel” class is expanded with the “Feedback” class and the “UserProfile” class. The “Feedback” class defines the evaluated behaviors of the user during interaction. It is aimed to enhance the user involvement during the adaptation.

The “UserProfile” class (figure 3) has been modeled as a composition of *Language*, *Knowledge*, *Country* and *PreferredRepresentationStyle*.

The “Language” class consists of the base language used by the user. The “knowledge” class defines expertise level of user. It class can be used to organize the information on the interface. For instance, and advanced user might require less guidance to accomplish the tasks. Instead a novice user will require a friendlier interface that will support and guide them to the accomplishment of tasks.

The “Preferred representation style” can be video, text and/or audio. Theses preferences help to the system to determinate the best adaptation of the information.

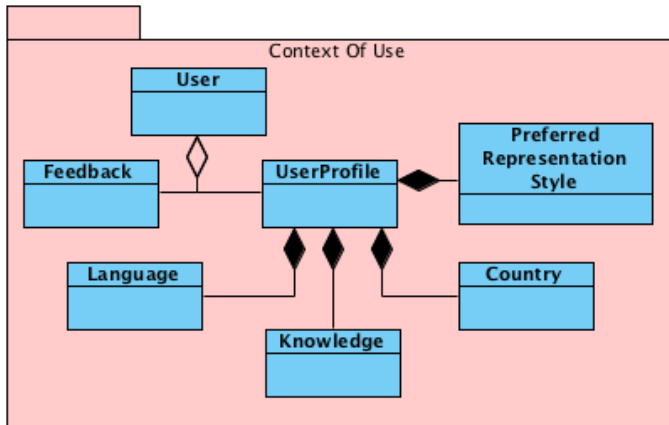


Figure 3. The User Model

The “Feedback” class as a “ContextElement” allows handling adaptations priorities that must be assigned to prevent conflicts, for instance user feedbacks could be considered to evaluate an adaptation rule and promote or demote it. The “Feedback” is involved for different purposes, for instance control trigger and/or evaluate adaptation decisions. This class is an aggregation of the user model; it is a specialization of the “TriggerEvent” class as well as the “ContextTriggering”. The “Feedbacks” class is intended to assess the “UIState” that depends on the current context of use defined by “ContextBinding” class. An adaptation is triggered by a change in this context surrounding the interaction.

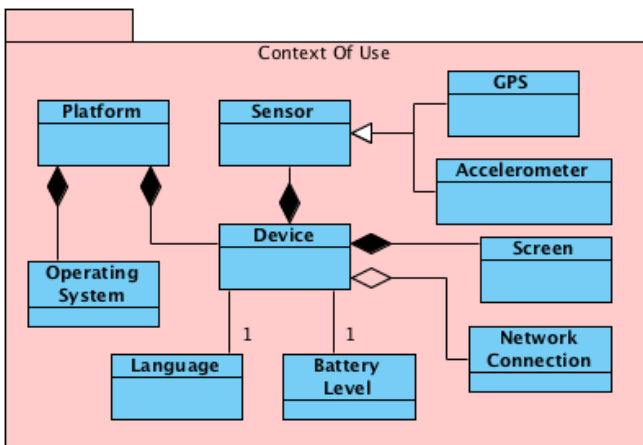


Figure 4. The Platform Model

The “Platform” class determines the set of information that can be considered to define the hardware used by the user. Figure 4 gives an overview of the main entities modeled by the Platform. The root entity is the “Platform” class with is linked to the Operating System and Device.

The information considered includes the characteristics of the Device and the operating system used to access the application. The “Device” considers integrate sensors, the screen size, the battery level, the language and the network providing the connection. For instance, a GPS that permits recuperates the coordinates of the user. In case of low level of the battery the adaptation can’t be considerate multimedia elements.

The environment model (figure 5) provides the characteristics of the environment in which user interact with the device. The environment can be represented as different aspects (Time, Date, Noise Level, Movement Status, Language, Weather, Direction and Location) considered by [16].

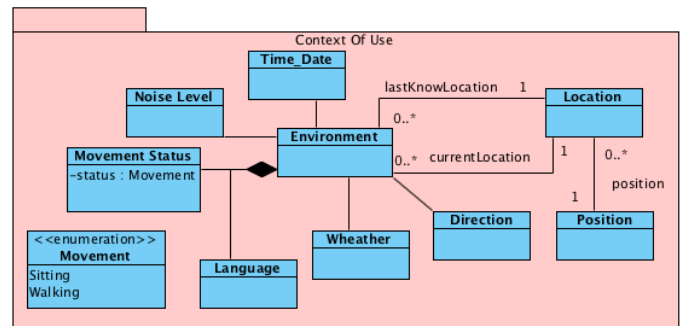


Figure 5. The Environment Model

The climate conditions like the “Weather” can determinate how the information can be presented on the screen. The “Location” of the user and the “Noise Level” can determinates the type of interaction more adequate to the user.

C. The UI Model Package

The proposal of the “UIModel” package can be related to any UI approach, such as Model-based approach showing a combination of UI models defined in the Cameleon reference frameworks [1] and PIM model which could be considered a model-based approach considering only the Final UI Model.

The “UIModel” is decomposed mainly into four step organizing four abstraction levels:

- “TaskModel” providing a goal-oriented description of interactive systems suitable for reviewing temporal relationships between tasks, and their decomposition into subtasks.
- “AbstractUIModel” outlining an expression of a UI in terms of interaction units without making any reference to implementation.
- “ConcreteUIModel” presenting the UI in term of concrete interaction object that are modality-dependent, but implementation technology independent,

- “FinalUIModel” that represents the final implementation realized in a programming language.

The “UIModel” support the execution of model throughout transformation. A “TransformationModel” links involved models during generation process.

I. IMPLEMENTATION

In this section, we account for the convenience and applicability of the above detailed model. We outline two implementations of the model presented on previous section. The first implementation regards a Flippable UI for internationalization developed in accordance with UsiXML project specifications [17]. The second implementation consists on an adaptive UI.

```
<contextofuse id="CarRental-context-fr_FR_14" name="CarRental-context-fr_FR">
  <UserModel id="CarRental-stfr_FR_14" language="fr_FR" stereotypeName="CarRental-stfr_FR">
    <Feedback id="CarRental-stfr_FR_14" Type="expliciControl" >
    <UserProfile id="CarRental-stfr_FR_14" language="fr_FR" >
  </UserModel>
  <PlatformModel id="CarRental-platform_14" name="CarRental-platform" dir="RT"/>
  <EnvironmentModel id="CarRental-env_14" name="CarRental-env"/>
</ contextofuse >
< contextofuse id="CarRental-context-en_US_14" name="CarRental-context-en_US">
  <UserModel id="CarRental-sten_US_14" language="en_US" stereotypeName="CarRental-sten_US">
    <Feedback id="CarRental-stfr_FR_14" Type="expliciControl" >
    <UserProfile id="CarRental-stfr_FR_14" language="ar_AR" >
  </UserModel>
  <PlatformModel id="CarRental-platform_14" name="CarRental-platform"/>
  <EnvironmentModel id="CarRental-env_14" name="CarRental-env"/>
</ contextofuse >
```

Figure 6. An XMI Context Model instantiation

Both implementations show a car rental case study. It serves as a preliminary guideline to work on a common scenario. A set of key functional requirements must be considered for implementing the car rental example.

The users must be able to:

- Select the city of interest to pick up the car;
- Specify the period for the car rental;
- Access a set of possible cars and select one;
- See details about selected car;
- Access and select additional car features (e.g. GPS);
- Provide personal information before renting the car;
- Access details about the car rental before submitting the request;

- Change the car rental parameters anytime before confirming the rental.

At a first time, the adaptation focus on user’s related contextual facts, specifically the culture. Figure 6 shows an XMI instantiating the contexts of use. Each context instantiation consider the triplet user, platform and the environment.

Adaptations are outlined via a set of transformations that consists on a models transformation at the Concrete UI levels. The implementation of adaptation rules was based on the Java Expert System Shell (Jess) [18]. Jess is an open rule-based engine integrated in the Java platform. An adaptation rules consisted on two main parts a condition denoting the trigger event and an action. Figure 7 outlines an example of adaptation rule that define different facts related to cultures adaptations.

```
(defrule adaptGraphicalCioThatHasResources_toCulture
  "For each fact of type JessGraphicalCioType that has a resource, adapt to culture"
  (declare (no-loop TRUE))
  ?jessCio<- (JessGraphicalCioType (id ?gCioId) (graphicCio ?gCio) (dir ?gCioDir))
  ?jessResource<- (JessResource{cioId == ?gCioId} (resource ?res))
  ?ctx <- (Context)
  =>
  (call ?gCio setContent ?res.content)

  (if (= ?gCioDir nil) then
    (bind ?platform ?ctx.platform)

    (call ?gCio setDir ?platform.dir)
  )

  (modify ?jessCio (isAdapted TRUE))
  (update)

; (printout t "Adapt a CioType " ?gCio.id "." crlf)
)
```

Figure 7. A Jess Adaptation Rule example

The figure 8 shows a visualization of the execution of adaptation. Adaptations are triggered explicitly by end-users via a control panel. The control panel consist on an implementation of users feedbacks aimed at adapting the interface regarding their evaluation. In the picture at le left area, we present the control panel allowing the manipulation of adaptation.

By moving the handles horizontally, the end user is able to manipulate the geometry in order to adapt the UI.

In this case, the adaptation consists on reversing the UI, and accordingly to change the UI language and orientation regarding cogitated culture facts. In the right area the picture show how the adaptation are visualized in the graphical UI.

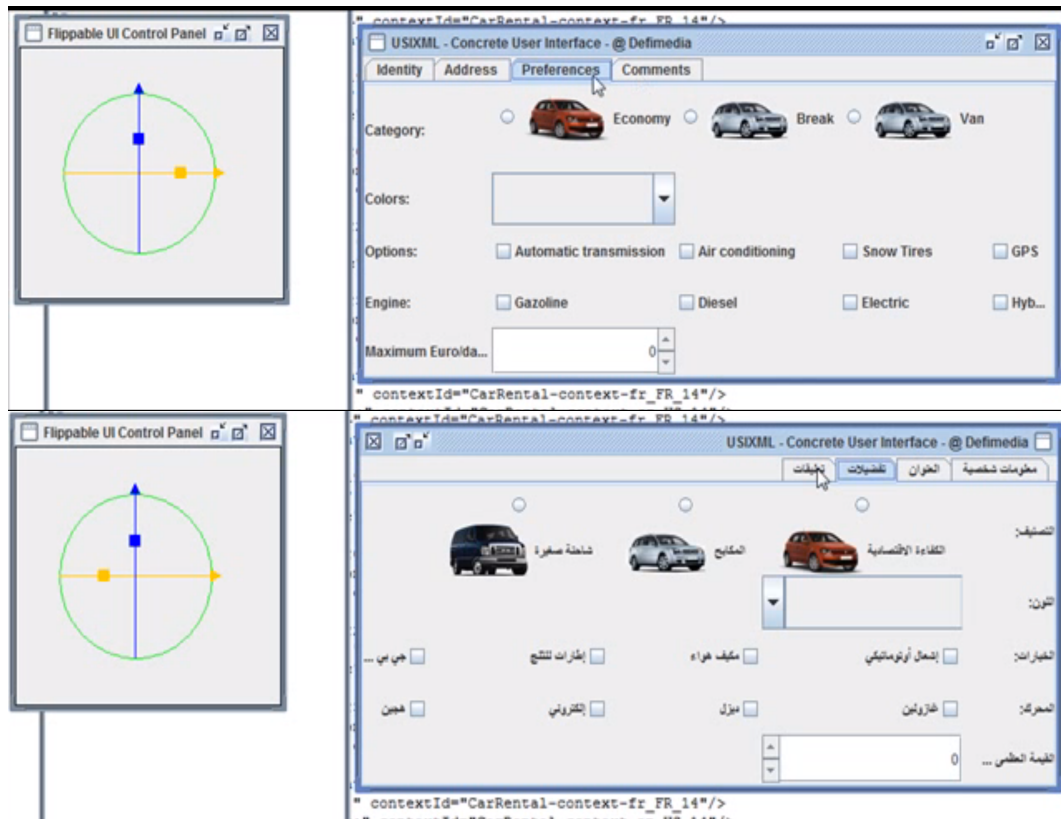


Figure 8. The Flippable Car Rental example

A further illustrative mock-up was implemented considering specific contexts, aspects and sharing the same theoretical models. The prototype is based on the same case study presented in the above illustration. It shows another situation for adaptation that considers the platform of interaction. Two platforms were considered: desktop and smartphone.

User can interact with both applications. The screen of the device is the main aspect that permits activate the adaptation of the application. The figure 9 shows the desktop version.

To shift from one platform to another adaptation rules are defined to meet different platform requirement. When a consistent set of requirements are identified, the adaptation process should carry out suitable actions to meet these requirements.

Two levels are considered to accomplish adaptation:

(1) At the interactive level, the interaction workload has been restructured by selecting interaction objects with higher guidance and accessibility. For instance the select color task (illustrate by the green box) is represented by a dropdown list to replace the selection list. This choice is justified by the adequacy with the screen size, the dropdown list allow more visibility and avoid the scrolling to visualize next tasks.

(2) At the presentation level, the formatting instructions have been suitably rewritten for the device. The arrangement and position of interface's elements is defined with regards to the propriety of each device. Such as, for the category specification task, and the disposition of tabs (Horizontal-Vertical).

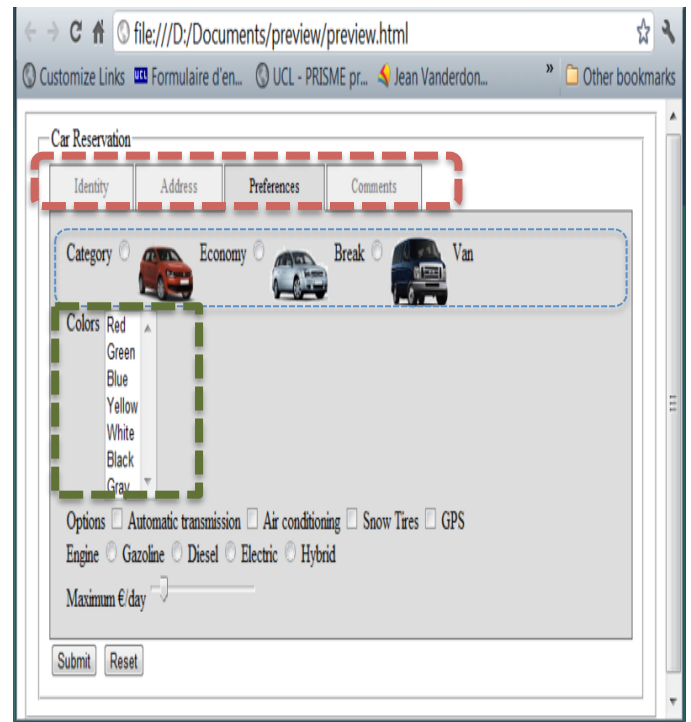


Figure 9. Car Rental case study for desktop

Figure 10 shows the screen layout after the adaptation process. It illustrates how the elements were adapted. For instance, the menu identified on the figure 9 by the red box

has been changed by a scrollable menu (see figure 10). Also the select color task represented by the select list (identified by the green box) has been changed by a dropdown list. Both adaptations are realized at runtime according to the screen size of device used by the user.

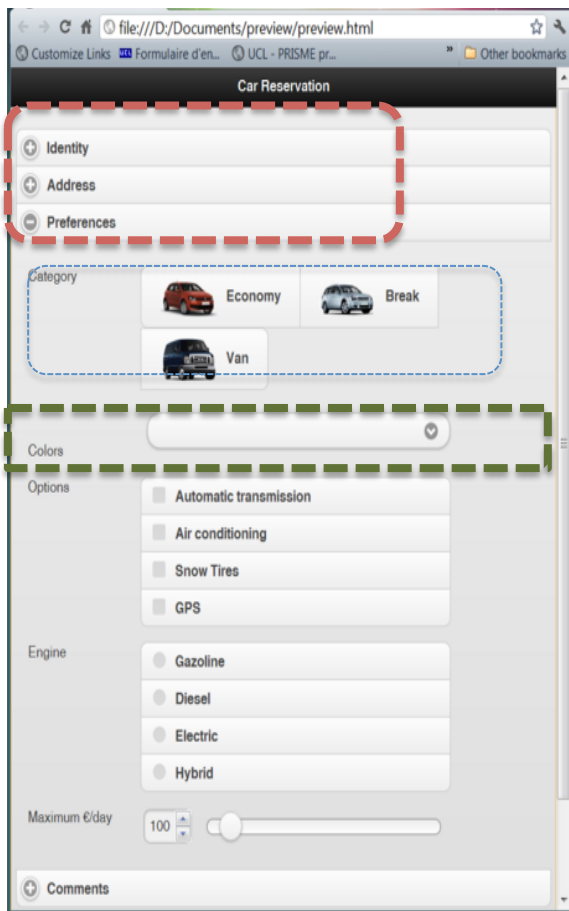


Figure 10. Car Rental case study for smartphone

CONCLUSION

This article presents a Conceptual Model for Agile Adaptation designed to supporting an explicit, comprehensible and complete configuration of adaptation concerns at runtime. It permits developing adaptive and adaptable user interfaces supporting end-user involvement.

The proposed model involves users and end-users for adaptation triggering. It is instantiated via flappable UI allowing users to adapt the UI at runtime by a control panel.

We will consider realize a methodological framework that considers structural and procedural views. As well we will take into account the study of different solutions to analyze and evaluate the information capture by the model to produce and present the UI adaptation.

A platform prototype for the implementation of runtime context-aware adaptation is foreseen to validate the model. With this prototype, we will be able to easier evaluate the interest and the usability of our proposal by conducting user experiments.

ACKNOWLEDGEMENTS

We are grateful to the “Jounum Project”, the Louvain School Management and the Catholic University Of Louvain for their financial support.

REFERENCES

- [1] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers* 15, 3 (June 2003), pp. 289-308.
- [2] Knutov, E., De Bra, P., Pechenizkiy, M. Generic Adaptation framework: a process-Oriented perspective. *Journal of Digital Information*, (2012).
- [3] Assad, M., Carmichael, D. J., Kay, J. and Kummerfeld, B. PersonisAD: distributed, active, scrutable model framework for context-aware services. *international conference on Pervasive computing (PERVASIVE'07)*. Springer-Verlag, Berlin, Heidelberg, (2007), pp 55-72.
- [4] Motti, V.. A computational framework for multi-dimensional context-aware adaptation. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems* (2011, June) pp. 315-318.
- [5] Motti, V., Mezhoudi, N, Vanderdonckt, J (Machine Learning in the Support of Context-Aware Adaptation. *Proceedings of the Workshop on Context-Aware Adaptation of Service Front-Ends* (2012).
- [6] Karen M. Feigh, Michael C. Dorneich, Caroline C. Hayes,. Toward a Characterization of Adaptive Systems Framework for Researchers and System Designer In. *Human Factors: The Journal of the Human Factors and Ergonomics Society* (2012) 1008-1024.
- [7] Knutov, E., De Bra, P., Pechenizkiy, M. AH—12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Rev. Hyperme'd. Multime'd.* 15(2009), 5–38.
- [8] Lim, B. Y., Dey, A. K. & Avrahami, D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. *CHI* (2009), 2119-2128.
- [9] Froehlich, J. et al. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In *CHI* (2009), 1043-1052.
- [10] Lavie, Talia and Meyer, Joachim. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, 2010, vol. 68, no 8, p. 508-524.
- [11] Findlater, L., and McGrenere, J. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. *Proc. SIGCHI Conference on Human Factors in Computing Systems CHI*, ACM Press (2008), 1247-1256.
- [12] Chang, K., Hightower, J., and Kveton, B. Inferring Identity Using Accelerometers in Television Remote Controls. In *Pervasive* (2009), 151-167.
- [13] Fox, D., Sillito, J., & Maurer, F. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In *Agile, 2008. AGILE'08. Conference(2008, August)*, pp. 63-72.
- [14] Ganneau, V., Calvary, G., & Demumieux, R. (2007, November). Métamodèle de règles d'adaptation pour la plasticité des interfaces homme-machine. In *Proceedings of the 19th International Conference of the Association Francophone d'Interaction Homme-Machine* (pp. 91-98). ACM.
- [15] Sottet, J. S., Ganneau, V., Calvary, G., Coutaz, J., Demeure, A., Favre, J. M., & Demumieux, R. Model-driven adaptation for plastic user interfaces. In *Human-Computer Interaction-INTERACT* (2007). pp. 397-410. Springer Berlin Heidelberg.
- [16] Sotsenko A., Jansen M., Milrad M. Discussion About Contextualization of Learning Objects. In: *Proceedings of the 12th World Conference on Mobile and Contextual Learning*, 2013.
- [17] Khaddam, I., & Vanderdonckt, J. Flippable User Interfaces for Internationalization. *Third international conference on effective interactive computing systems (EICS)* 2011, Pisa, Italy.
- [18] Hill, Ernest Friedman. *Jess in Action: Java Rule-Based Systems*. Manning Publications Co., Greenwich, CT, USA, 2003.

