# Models at Run-time for Sustaining User Interface Plasticity

Jean-Sébastien Sottet[1] Gaëlle Calvary[1] Jean-Marie Favre[2]
University of Grenoble
CLIPS-IMAG[1], LSR-IMAG[2]
BP 53, 38041 Grenoble Cedex9, France
+33(0)4 76 51 48 54
jean-sebastien.sottet@imag.fr gaelle.calvary@imag.fr jean-marie.favre@imag.fr

## ABSTRACT
In the vision of ubiquitous computing, users are imagined as evolving in various, changing and not always foreseeable environments, in which platforms may arrive and disappear in an opportunistic manner. As a result, there is a need for User Interfaces (UI) to adapt to their context of use (<User, Platform, Environment>) while preserving usability. This capacity of UIs is called *Plasticity*. In a forward engineering approach, UIs are designed step by step starting from domain-based descriptions (user's tasks and concepts) to code. It is now well understood that plasticity may impact UIs at any level of abstraction. This calls for keeping the UIs design rationale alive at run-time. As models are practiced since a long time in Human-Computer Interaction (HCI), this paper investigates to which extent Model-Driven Engineering (MDE) is relevant for plasticity. UIs are designed as a net of models that define different perspectives on a same UI (user's task, domain, concepts, widgets, etc.). The net is alive at run-time and transformed when the context of use changes. Transformations are performed with respect to usability. This paper sketches the vision on a small running case study. It highlights a set of strengths and doubts that give rise to many perspectives.

### Keywords
Plasticity of User Interfaces, adaptation, context of use, models at run-time, mapping, transformation, metamodel.

## 1. INTRODUCTION
With ubiquitous computing, the Human-Computer Interaction (HCI) community has now to cope with *various*, *variable* and *unforeseeable* contexts of use. By context of use, we mean the triple <User, Platform, Environment>. This evolution is problematic as the know-how in HCI relies on an explicit description of the targeted context of use that is supposed to be fixed. From now on, UIs have to be *plastic*, i.e. capable of adapting to their context of use while preserving usability. This paper explores Model Driven Engineering (MDE) for sustaining plasticity. The challenge is all the most crucial that it would save a long and strong expertise in model-based design in HCI. As sketched in section 3, the key of our approach is a unification between design-time and run-time making the design-time models alive at run-time. UIs are envisioned as *octopuses* made of a net of models [1] and mappings, whose transformations *plastify* the UI. The paper is illustrated on a small running example that is described in section 2. Section 4 opens the paper on a research agenda.

## 2. RUNNING EXAMPLE
"Sedan-Bouillon" is a web site that aims at promoting tourism in the regions of Sedan and Bouillon in France and Belgium (http://www.bouillon-sedan.com/). Initially, the web site has been designed for PC screens only. A plastic light weight version (called LSB for Light Sedan-Bouillon) has been developed in the European CAMELEON project for exploring the distribution of a web site across a PC screen and a PDA. LSB is limited to French speaking users and covers the hotel browsing task only. LSB promotes a user-controlled distribution of the UI among the web browsers with which the user is connected to LSB.

Let us consider Lionel and Alice who are planning a trip to Sedan. Lionel turns on his PC, and connects to the LSB web site. He logs in for accessing the web site. LSB is composed of three workspaces (Figure 1): a title, a navigation bar and an informational content. The navigation bar is augmented with a "Meta-UI" link that allows Lionel to control the distribution of the UI across the available resources (here, the PC and the PDA).

In order to comfortably browse the web site while sitting in the sofa, Lionel turns on his PDA and connects to the web site with the same identifier. The meta UI is being displayed, telling Lionel that he is currently using two browsers (log_Lionel_0 on the PC, and log_Lionel_1 on the PDA) on a same web site (Figure 2), and that it is possible to redistribute the UI among these resources.


*Figure 1:* The light "Sedan-Bouillon" centralized version. It is made of three workspaces: a title, a navigation bar and a content.

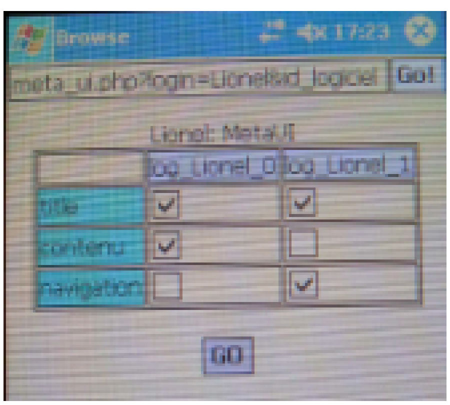


*Figure 2:* The Meta-UI allows the user to specify the platforms on which he/she would like to see the workspaces composing the UI. The Meta-UI appears when the user either clicks on the Meta-UI link (see the navigation bar in Figure 1), or connects to the same web site using another platform.

Lionel asks for the title ("titre") on both the PC and the PDA (see the first row of the table on Figure 2), the content ("contenu") on the PC only (see the second row), and the navigation bar on the PDA only (see the third row). Both the PC and the PDA screens are updated accordingly (Figure 3): the navigation bar leaves the PC and arrives on the PDA; the title is replicated on the PDA. When migrating from the PC to the PDA, the navigation bar is remolded switching from a vertical (Figure 1) to a horizontal (Figure 3) layout.


*Figure 3:* A distributed version of the light "Sedan-Bouillon" web site, according to the user's preferences (Figure 2).

Next section sketches our approach unifying design -time and run-time around the MDE key notions of models, mappings and transformations. All of them are compliant to metamodels [2].

## 3. APPROACH: TOWARDS OCTOPUSES, NETS OF MODELS ALIVE AT RUN-TIME

Models are not new in HCI. UIs are traditionally designed step by step transforming a task model into workspaces (also called abstract UI), then interactors (concrete UI) and finally code. Transformations lean on the usability properties that have been elicited during the requirements phase. At each step of the transformation process, hypotheses can be made on the context of use. As a result, if the context of use changes, then the design has to be revised from this branching point. Given that, our idea is to keep benefit from the past making the models and transformations alive at run-time so that the UI is able to reason about its own adaptation when the context of use changes:

- The UI is a net of interconnected models, those who traditionally supply the design process. They are enhanced with both the corresponding transformations and new models [3] specially crafted for plasticity: requirements, context of use, usability, and transformations;
- Plasticity is driven by an evolution model that specifies the way the net of models has to be transformed when the context of use changes.

Figure 4 illustrates the set of models corresponding to the case study. For legibility, the models are given in a concrete syntax and are limited to the user's task, workspaces and platform models (i.e., a subpart of the context of use). The mappings between workspaces and platform specify the way the UI is distributed among the available display surfaces (Figure 4): the title workspace (on the top of the Figure) is replicated on both the PC and the PDA (see the two mappings connecting the corresponding workspace to both the PC and the PDA); the navigation bar (the workspace on the left) and the content (the workspace on the right) are respectively assigned to the PDA and the PC (see the unique mapping linking each of them to one unique platform). In the same way, Figure 1 (the centralized web site) would simply be described by associating the root workspace to the PC screen.

The case study shows how the two leverages of plasticity (remolding and redistribution) are *uniformly* addressed by transforming models and/or mappings. To make the adaptation performable at run-time, models and mappings (M1) have to be compliant to explicit metamodels (M2). Figure 5 emphasizes these two levels M1 and M2.

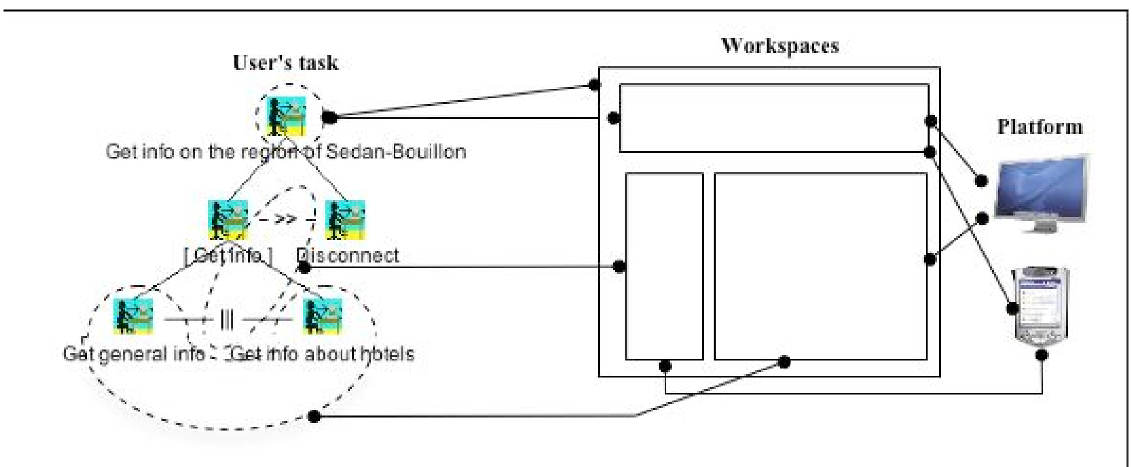


*Figure 4:* UIs are modeled as a net of models and mappings. The mappings between the workspaces and the platform models specify the way the UI is distributed among the available interaction resources.

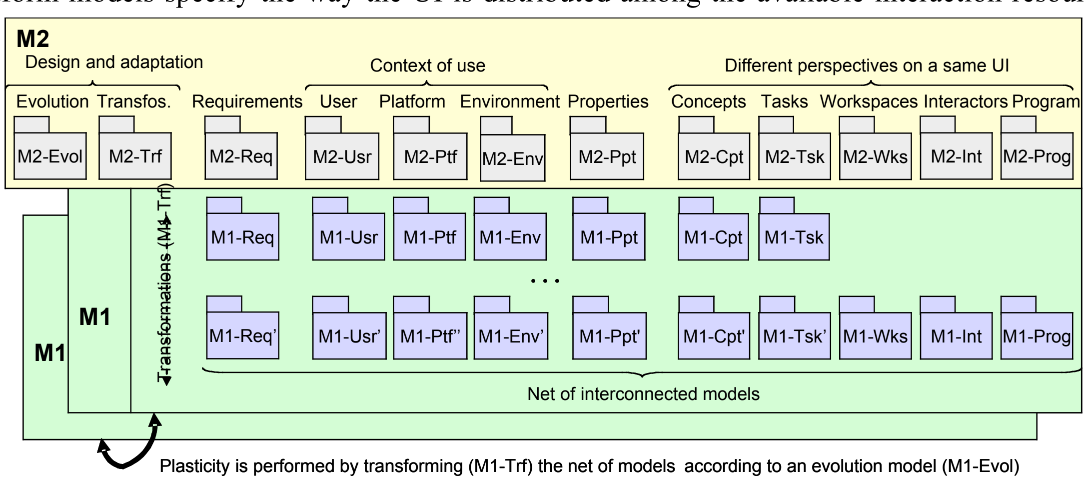


*Figure 5:* UIs are envisioned as a net of models both telling their design rationale and sustaining plasticity.

Actually, the Sedan-Bouillon prototype does not embed its net of models at run-time because the vision was not set when the web site has been developed. Now, we are mature enough to redevelop the application based on our metamodels and MDE technologies. We have defined metamodels for the traditional models (user's tasks, domain concepts, workspaces, interactors) and more innovative ones for plasticity (platform, mapping and transformation). Usability properties are elicited on the mappings.

We have selected the Eclipse Modeling Framework (EMF) for supporting our metamodels and models definitions. We use the editor of the Topcased project [4].

For the interactors and program models (Figure 5), we are defining additional transformations to target other technical spaces than EMF: C# and XAML [6], XUL [7], HTML, etc. This is necessary for being able to distribute UIs among a set of heterogeneous platforms. The mappings on the platform act as a deployment model. Transformations are ATL-based [5] and performed at run-time. One example is provided in [2].

## 4. OPEN ISSUES AND CONCLUSION

In the past, design-time and run-time have always been managed in an isolated way. This paper shows how MDE bridges the gap between design-time and run-time thanks to models, mappings and transformations that are compliant to explicit metamodels. This unification is all the most relevant that there is a strong know-how in model-based design in HCI. Solving plasticity by enhancing this know-how is a nice perspective.

Now, we have to refine our metamodels (especially the mapping metamodel for conveying the usability properties), and check whether general tools from MDE are convenient for HCI (in particular, with regard to the latency property).

In a farther future, we envision a unification with evaluation. As requirements are alive at run-time, the UI should be able to check the extent to which it is compliant with these requirements.

## 5. ACKNOWLEDGMENTS

The work has been supported by the European SIMILAR network of excellence. Authors would like to thank Joëlle Coutaz and Alexandre Demeure for their contribution to the work.

## 6. REFERENCES

[1] Demeure, A., Calvary, G., Sottet, J.S., Vanderdonkt, J., A Reference Model for Distributed User Interfaces, *TAMODIA'2005*, pp 79-87.

[2] Sottet, J.S., Calvary, G., Favre, J.M. Towards Model Driven Engineering of Plastic User Interfaces, *International workshop on Model Driven Development of Advanced User Interfaces*, MDDAUI, Jamaica, 2005.

[3] Sottet, J.S., Calvary, G., Favre, J.M., Demeure, A., J Coutaz, J. Mapping and Model Transformation For Consistency of Plastic User Interfaces, *Workshop on The Many Faces of Consistency in Cross-platform Design*, *ACM conf. on Computer Human Interaction, CHI 2006*, Montréal, 22-23 Avril 2006.

[4] http://www.topcased.org.

[5] http://www.sciences.univ-nantes.fr/lina/atl/.

[6] XML User Interface Language http://www.mozzila.org/projects/xul .

[7] Petzold, C. Code Name Avalon: Create Real Apps Using New Code and Markup Model. *Microsoft MSDN Magazine Volume 19* January 2004.