# Embedding Plasticity in the Development Process of Interactive Systems

*Gaëlle Calvary, Joëlle Coutaz, David Thevenin*

CLIPS-IMAG
BP 53, 38041 Grenoble Cedex 9, France
Tel : (+33) (0)476514854, Fax : (+33) (0)476446675
{Gaelle.Calvary, Joelle.Coutaz, David.Thevenin}@imag.fr, http://iihm.imag.fr

**Abstract.** This article briefly presents the notion of plasticity. Plasticity refers to a particular sort of user interface adaptation. A new process model that supports the structured development of plastic user interfaces is described and illustrated with a test case.

## 1. INTRODUCTION

Information technology is an increasingly essential part of the fabric and activity of our lives. Jini-enabled information appliances, XML approaches to information modeling and rendering, and gateways between Internet and wireless network protocols, are being developed to cope with the pregnancy of the technical push. This exploratory development of novel devices and techniques is valuable in the short run. The approach, however, is not replicable and provides poor guidance to sustain future development of usable interactive technologies. As a result, there is a risk of a shortfall between technical promise and effective interaction. *Theories and principles developed so far in HCI must not be lost in the evolution!*

Although principles of user-centered design methods and modeling techniques offer a sound substrate, pervasive computing opens the way to new challenging requirements. In particular, *people want to have the choice*. They want to be able to choose among a wide range of software platforms and hardware devices to accommodate multiple needs depending on places and spaces across time. Providing different interfaces specially crafted for each type of device and modality combination is extremely costly and could result in users having many different versions of interfaces on different devices. The impact includes massive under-use of interfaces potential and excessive development costs to maintain versions consistent across multiple platforms. In [Thevenin 99], we introduce the notion of *plasticity* to cope with these problems.

## 2. PLASTICITY

The term "plasticity" is inspired from the property of materials that expand and contract under natural constraints without breaking, thus preserving continuous usage. Applied to HCI, plasticity is the capacity of an interactive system to withstand *variations of context of use* while *preserving usability.*

A *context of use for a plastic system* covers two classes of attributes:

- The attributes of the physical and software platform(s) used for interacting with the system. Typically, screen size and network bandwidth have an impact on the amount and modality of information to be rendered and transferred;
- The environmental attributes that describe the physical surroundings of the interaction. These include the set of objects, persons and events that are peripheral to the current task(s) but that may have an impact on the system and/or the user's behavior, either now or in the future. Typically, light conditions may influence the robustness of a computer vision-based tracking system, noisy environments may eliminate sonic feedback, etc. At the task level, location in space provides context for information relevance; tasks that are central in the office (e.g., writing a paper) may become secondary in a train, etc.

A *plastic user interface preserves usability* if the properties selected at design time to measure its usability are kept within a range of values as adaptation occurs to contextual changes. Although the properties developed so far in HCI [Gram 96] provide a sound basis, they do not cover all aspects of plasticity. For example, they do not express the need for continuity [Graham 2000] when migration occurs between contexts of use. Thus, we *need to extend and refine our apparatus of properties* to cope with the new situations offered by the technology.

Activity theory takes into account the situation of action early in the design process. Unfortunately, situation-dependent information is lost in the development process due to the lack of appropriate notations of the design and development tools. As a result, current tools implicitly assume that users are working with a desktop computer located at a specific place. A notable exception is the context toolkit [Salber 99] developed for encapsulating sensors at the right level of abstraction and the "literate development" [Cockton 95]. Although within the scope of plasticity, context toolkits cover low-level technical concerns only, and the literate development is not precise enough to address our problem. Therefore *a process model that supports the development of plastic user interfaces is required*. This is the topic of a subsequent section illustrated with a real life example.

## 3. AN EXAMPLE : A HOME HEATING CONTROL SYSTEM

The heating control system envisioned by EDF (The French Electricity Company) will be controlled by users situated in diverse contexts of use. These include:

- At home, through a dedicated wall-mounted device or through a Palm-like device connected to a wireless home-net,
- In the office, through the Web, using a standard work station,
- Anywhere using a WAP-enabled mobile phone.

A typical user's task consists of consulting and modifying the temperature of a particular room. Figures 1 and 2 show versions of the same system for different interaction platforms.

- In 1 a), the system displays the current temperature for each of the rooms of the house. The screen size is comfortable enough to make observable the entire system state.
- In 1 b), the system shows the temperature of a single room at a time. A thumbnail allows users to switch between rooms. In contrast with 1a), the system state is browsable due to limited screen size. As a result, additional navigational tasks have been introduced in the task model to give access to the desired information.

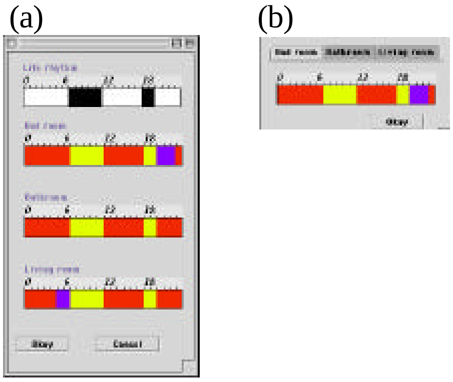
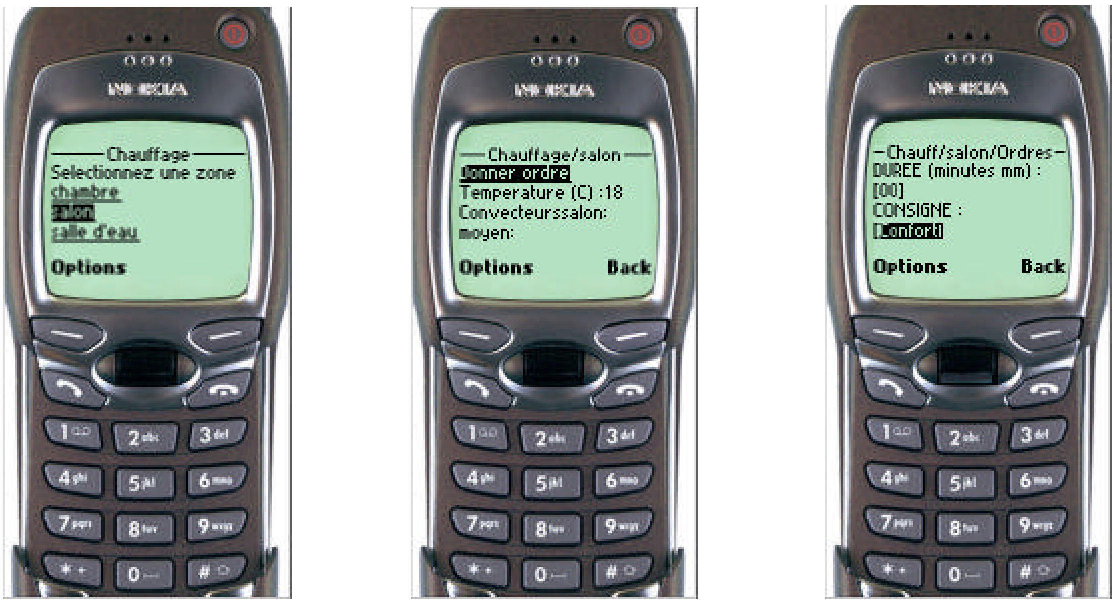


**Figure 1.** a) Large screen. Temperature of the rooms are available at a glance. b) Small screen. Temperature of one room is displayed at a time.

Figure 2 shows the interaction trajectory for setting the temperature of a room with a WAP mobile phone. In 2a), the user selects the room (e.g., le salon – the living room). In 2b), the system shows the current temperature of the living room. By selecting the editing function ("donner ordre"), one can modify the temperature (2c). When comparing with the situation depicted in Figure 1, not only navigation tasks have been introduced, but a title for every deck (i.e., WML page) has been added to recall the user with the current location within the interaction space.



Chauffage
Selectionnez une zone
chambre
salon
salle d'eau
Options

Chauffage/salon
Donner ordre
Temperature (C) :18
Convecteurssalon:
moyen:
Options Back

Chauff/salon/Ordres
DUREE (minutes mm) :
[00]
CONSIGNE :
[confort]
Options Back

**Figure 2.** Modifying the temperature using a WAP-enabled mobile phone.

All of these alternatives have been produced using the following framework.

## 4. A NEW DEVELOPMENT PROCESS MODEL

Our framework is intended to serve as a reference instrument to help designers and developers to structure the development process of plastic interactive systems. To this end, we adopt a model-based approach [Paterno 99]:

- we build upon models used in current practice,
- we improve existing models to accommodate variations of context of use,

- we explicitly introduce new models and heuristics that have been overlooked or ignored so far to convey the context of use.

Figure 3 shows the models involved in the process. The *Platform Model* and the *Environment Model* define the contexts of use intended by the designers. The *Evolution model* specifies the change of state within a context as well as the conditions for entering and leaving a particular context. The *Interactors Model* describes "resource sensitive multimodal widgets" available for producing the concrete interface.

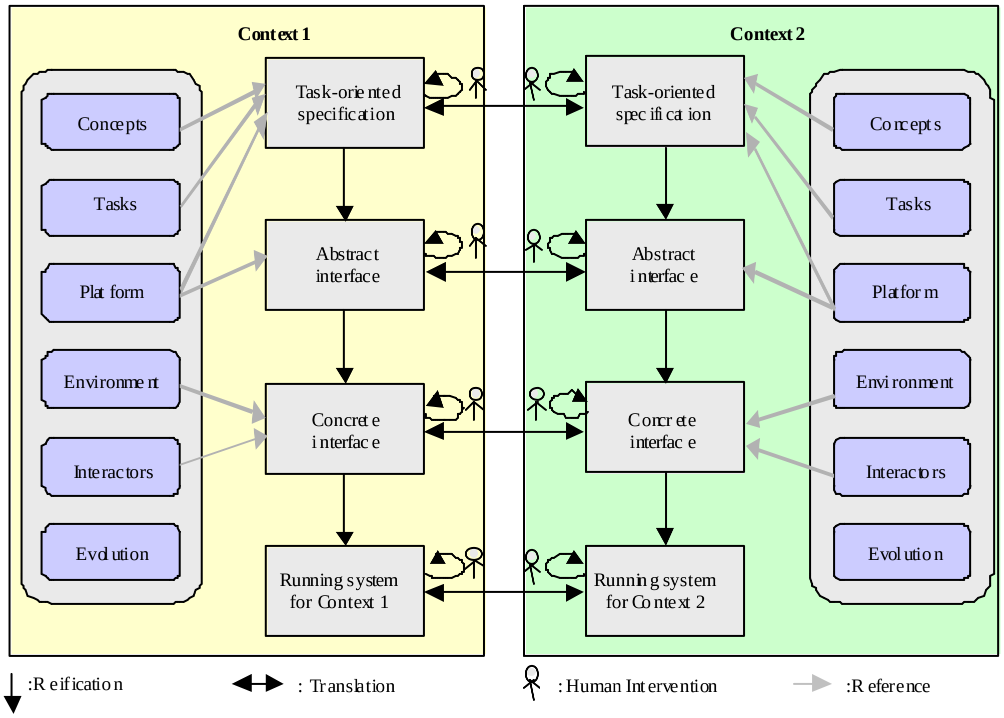


**Figure 3.** The reference development process for supporting plastic interactive systems.

Each of the above models are referenced along the development process from the task specification to the running interactive system. The process is a combination of vertical reification and horizontal translation. Vertical reification covers the derivation process, from top level abstract models to run time implementation. Horizontal derivations, such as those performed between HTML and WML content descriptions, correspond to translations between models at the same level of reification. Reification and translation may be performed automatically from specifications, or manually by human expert designers, depending on the tools available.

As shown in Figure 4, the reference framework can be instantiated in many ways:

- In 4a), two running systems are reified in parallel using input models each one being specified for a particular context of use. This situation corresponds to the current practice. It forces to maintain consistency between multiple versions.
- In 4b), the ideal situation: reification is applied until the very last step. Consistency maintenance is here minimal. This approach has been used for the Heating Control System shown in Figure 1 for Java-enabled target platforms. The user interfaces shown in Figure 1 have been derived automatically using ARTStudio (Adaptation by Reification and Translation), a tool under development in our lab.

- In 4c), the task-oriented specification is translated to fit another context. From there, reifications are performed in parallel. This approach has been adopted for the Heating Control System using a WAP mobile phone (Figure 2). Sub trees that correspond to infrequent tasks have been pruned from the original task tree developed for the Java-enabled platforms. Because ARTStudio does not yet support Web-based techniques, the reification steps have been performed manually by a human expert.
- 4d) shows a mix of interleaving between reification and translation.

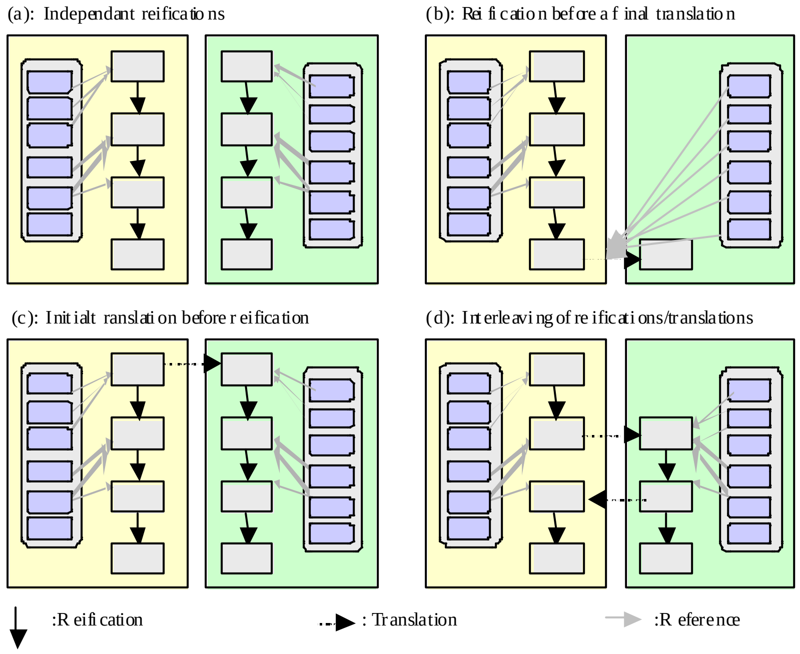

(a): Independant reifications
(b): Reification before a final translation
(c): Initial translation before reification
(d): Interleaving of reifications/translations
:Reification
: Translation
:Reference

**Figure 4.** Instantiations of the reference model.

## 5. CONCLUSION

In summary, the technology push provides opportunities for new forms of interaction and triggers new social requirements.

*New forms of interaction.* Although prospective development may be fun and valuable in the short run, we must not put aside the principles and theories developed for the desktop computer to design new artefacts. Instead, we propose to use current knowledge as a sound basis, question current results, improve them, and invent new principles if necessary. This is the approach we have adopted for supporting plasticity by considering model-based techniques from the start. Because automatic generation of user interfaces has not found wide acceptance in the past [Myers 00], reification and translation may be done manually by human experts when tools are inappropriate.

*New user's requirements.* People desire a large variety of choices (anything, anywhere, any time). Our concept of plasticity addresses one aspect of these new requirements while attempting to minimize the cost of developing and maintaining such systems. Although incomplete, our framework provides a reference structure for coping with this complex problem.

## 6. ACKNOWLEDGEMENT

This work is being supported by EDF, France.

## REFERENCES

[Cockton 95] G. Cockton, S. Clarke, P. Gray and C. Johnson, Literate Development: Weaving Human Context into Design Specifications, *Critical Issues in User Interface Engineering*, P. Palanque & D. Benyon (eds), Springer-Verlag: London Publ., ISBN 3-540-19964-0, 1995.

[Graham 2000] T.C. N. Graham, L. Watts, G. Calvary, J. Coutaz, E. Dubois, L. Nigay. A Dimension Space for the Design of Interactive Systems within their Physical Environments, *DIS2000*, 17-19 August 2000, ACM Publ. New York, pp 406-416.

[Gram 96] C. Gram, G. Cockton (eds), *Design Principles for Interactive Software*, Chapman & Hall, 1996.

[Myers 00] B. Myers, S. Hudson, R. Pausch. Past, Present, and Future of User Interface Software Tools, *Transactions on Computer-Human Interaction*, ACM, 7(1), March 2000, pp 3-28.

[Paterno 99] F. Paternò, *Model-based Design and Evaluation of Interactive Applications*, Springer Verlag, November 1999.

[Salber 99] D. Salber, A. K. Dey and G. D. Abowd. The Context Toolkit: Aiding the Development of Context-Enabled Applications, in the *Proceedings of the 1999 Conference on Human Factors in Computing Systems* (*CHI'99*), Pittsburgh, PA, May 15-20, 1999, pp 434-441.

[Thevenin 99] D. Thevenin, J. Coutaz. Plasticity of User Interfaces: Framework and Research Agenda, in Proc. *Interact99*, Edinburgh, A. Sasse & C. Johnson (eds), IFIP IOS Press Publ., 1999, pp 110-117.