# An Integrated Approach to Develop Interactive Software

Begoña Losada, Maite Urretavizcaya, Isabel Castro



# An integrated approach to develop interactive software

Begoña Losada, Maite Urretavizcaya, Isabel Fernández de Castro

Dept. of Computer Languages and Systems
Faculty of Computer Engineering
University of the Basque Country
20001 San Sebastián
{b.losada, maite.urretavizcaya, isabel.fernandez}@ehu.es

**Abstract.** In this poster we present InterMod, an approach that combines Agile Methods, Model-Driven Developments and User-Centered Design, which are widely accepted in the development of interactive software. The planning and project organizing are based on *User Objectives* (user desires). The project is organised as a series of iterations and the work is distributed in different workgroups according to some developmental and integration activities. The requirements are incrementally collected and validated with models based on user-centered design. To speed up this validation, we put forward the *SE-HCI model*, which enriches a human-computer interaction model with the semantics of the application and some basic characteristics of an abstract prototype.

**Keywords:** User-Centered Design, Agile methods, Model-Driven Development, Software Engineering.

## 1 Introduction

Agile methods (AM), model-driven developments (MDD) and user-centred design (UCD) are three approaches widely accepted by the community and share a common objective of efficiency in the resulting software. However, none of them alone achieves success without encountering problems during application development. Because of that, efforts are being made to integrate these techniques so that the advantages of one mitigate the problems of the others [1][2][3][4]. However, due to the fact that a majority of software engineering development processes focus on software architecture, satisfactory integration has not yet been achieved. Therefore, we focus our efforts on a new approach to improve software development that combines agile characteristics, MDD and user-centred techniques.

## 2 InterMod, an integrated proposal

InterMod [5] is a methodology whose aim is to help with the accurate development of interactive software. Although it is suitable for use with web design, its utility is not restricted to just that area.

Our proposal, shown in Fig 1, is to organise the project as a series of iterations, just as the agile methodologies do, and distribute the work in the iterations according to different developmental activities of the *User Objectives* (UO). A *User Objective* is a user desire e.g. "*buying a t-shirt*" or "*reserving a meeting room in a workplace*", that can be achieved by one or more user functionalities.

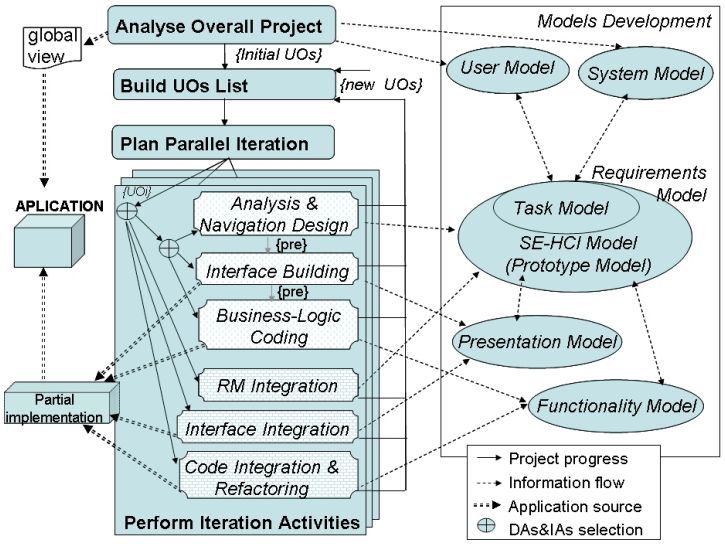


**Fig. 1** InterMod process and development activities

The new UOs are in turn objectives to be refined in subsequent iterations. Project progress is reflected in the activities done to achieve these UOs current and the resulting models.

## 2.1 InterMod Activities and Models

InterMod has two kinds of ***Activities***: *Developmental Activities* and *Integration Activities*. We represent graphically Activities as shown in Table 1.

**Table 1. InterMod Activities**

| Development Activities | | Integration Activities | |
|---|---|---|---|
| **A1.** Analysis & Navigation Design | | | **I1**. RM Integration |
| **A2.** Interface Building | | | **I2**. Interface Integration |
| **A3.** Bus.-Logic Coding | | | **I3.** Code Integration & Refactoring |

The ***Developmental Activities*** (DAs) associated with each UO are strongly related: A1.*Analysis and Navigation Design*, A2.*Interface Building* and A3.*Business-Logic Coding*. Each UO requires the three DAs to be developed but a prerequisite relation must be done A1<A2<A3 ('<' means prerequisite). Just as UCD recommends, before

coding a relevant UO, its interface must be validated. However, unlike UCD, it is not required that the complete application interface be developed before moving to the implementation of the business logic; instead this approach stays framed in the development of one or several UO groups. Furthermore, to assure a correct incremental progress of the project, some ***Integration Activities*** (IAs) are needed: *I1.Requirement Models (RM) Integration*, *I2.Interface Integration* and *I3.Code Integration & Refactoring*. A restriction is necessary for controlling the correct development of an IA. Thus, it is possible to carry out an IA $I_k$ (K=1..3) for a concrete $UO_j$ (j=0..n) *if and only if* the $UO_j$ is the fusion of two UOs belonging to the UO List *and* the DAs $A_k$ of these fused UOs are already made. To ensure consistency in the final application, evaluations of the incrementally obtained products as well as heuristic and metric evaluations are included in all activities.

All iterations are guided by the same action plan that divides the work according to the activities of different UOs, in such a way that each DA will be next driven by models and all the integration processes can lead to the revision and modification of these models. Even during final integration of the software there may be revisions of all models and new UO can be created. The activities of **analysis and navigation design** and **RM integration** deals with the *Requirements Model* (RM), which includes the *Semantically Enriched Human-Computer Interaction* (SE-HCI) *model*. The *SE-HCI Model,* that incorporates information from the *User* and *System Models,* is an abstract description constructed over the *Task Model*. It also incorporates three essential aspects: a) The description of both the actions that users and the system can carry out at the user interface level during an interactive session [7], and their possible temporal relations, b) The descriptions of the correct and incorrect interactions that represents the semantics of the application and c) The basic visual characteristics, such as colours, sections, button types, etc. We propose the evaluation of the requirements involved in the SE-HCI with an abstract prototype created automatically by transforming the *SE-HCI model*. From this point, the evaluation can be carried out jointly by the designers, customers and developers.

In the **Interface Building activity**, the *Presentation Model* is created for a UO previously designed and evaluated, and the **Interface Integration activity** fuse together the *Presentation Model* of some UOs. The *Presentation Model* of a specific UO settles the graphical elements and others characteristics gathered from the *Requirements Models*. There are several languages for modeling user interfaces widely used and tested, as XIML[7] or UIML [9], and they may be used to reflect this model. Finally, the **Business–Logic Coding** and **code Integration & Refactoring activities** deal with the *Functionality Model* that guides the implementation in a particular programming language. This model inherits the behaviour characteristics from the UO *Requirements Models* evaluated in the first activity. UML or SysML [10] are alternative languages typically used to represent this model.

### 2.2 InterMod process steps

InterMod has four main steps (see Fig.1), i.e. the initial *Analyse Overall* step, and then an iterative process with three steps follow: *Build User Objectives List*, *Plan Parallel Iteration* and *Perform Iteration Activities*.

At the beginning of the project, InterMod proposes the **Analyse Overall Project** step in order to determine: (a) what the starting UOs are, such as those most important or needed, that provide the initial global view of the application, and (b) the *System Model* and *User Model* that help to collect the defining characteristics of the system type (e.g. device type, security, window size, colour, logo, etc) and those of the user (e.g. colour preferences, font, size, some limitations as colour blindness, deafness, vision loss, etc). All developments in the project will inherit or extend these models in order to guide and to ensure coherence throughout the entire application.

The application requirements are incrementally collected during the progressive UO List construction. Each iteration begins with a revision of the UOs list. The ***Build User Objective List*** step updates the list with the new UOs derived either from the previous UOs developments or from the new needs of the project. That is, the UOs included in the list may be modified, in the sense of agile methodologies [5], through the different evaluations undertaken by developers and users, or by the continuous meeting among members of the same and different teams. In order to achieve an UO, different activities must be realized. The next step, ***Plan Parallel Iteration***, decides for the current iteration: (a) what UOs to develop, (b) what activities to make for those UOs and (c) how to distribute these different activities to the workgroups (if there is more than one). The iteration ends with the ***Perform Iteration Activities*** step. Each workgroup performs the activities established in its plan.

A snapshot of a **Project Progress State** and the Plan obtained for the **Parallel Iteration** after some iterations (iteration i) are shown in **Fig 2.** Three aspects characterize the state of the project: the UO list, the UOs fusion list and the UOs progress according to theirs ***Activities*** made. In this iteration, the ***Parallel Iteration Plan*** has been the follow: The first team takes responsibility for two activities: **A1** activity for $UO_6$ and **I1** for $UO_4$. As it is shown, $UO_4$ is the fusion of the OU 2 and 3.The team 2 must build the interface (A2) for the $UO_1$ whose prerequisite is reached ($UO_1$ is in the A1 list). Meanwhile, team 3 must integrate and refactor the code referred to $UO_{10}$ that is composed of the OU 0 and 5 that have been already coded.

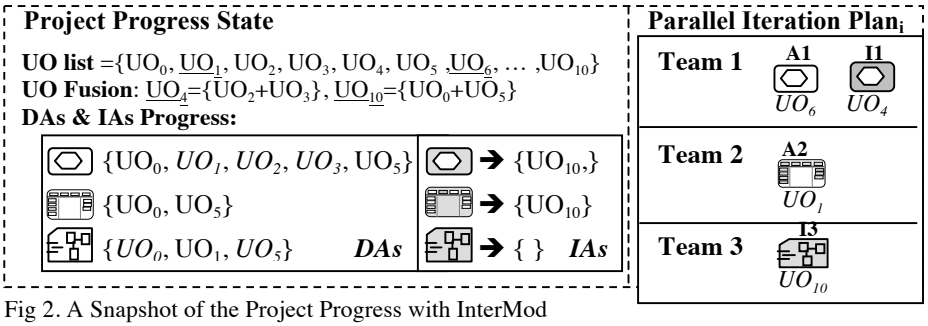

Fig 2. A Snapshot of the Project Progress with InterMod

## 3 Conclusions and Future Work

In this poster we present a new vision of the InterMod methodology, a proposal integrating three philosophies: UCD, MDD and AM. From the point of view of agile

methods, our work is organized in a series of iterations in which the user objectives to be dealt with are developed. This iterative process speeds up the development and gets results of the project progress. InterMod proposes some developmental and integration activities driven by models to achieve the UOs. The possibility to distribute the work in parallel increases the speed of resolution, although the process itself requires integration points to ensure consistency.

This process allows gather and validate the requirements incrementally. Because of this agile approach, InterMod, unlike UCD, does not require the complete development of the application interface before the implementation of the business logic, but assures usability. The SE-HCI model is the core of our proposal models architecture. It is involved in a Model Driven Process that obtains an abstract prototype created automatically by transforming the *SE-HCI model*. This prototype allows the evaluation of the requirements and facilitates the end user's participation, as recommended by UCD and AM. Early evaluations of the requirements reduce the number of the corrections further on in the process and therefore, reduce its cost.

The new InterMod methodology has been refined in parallel with the development of a demonstrator. A small initial set of UOs has evolved to a complex system. This make us think of the scalability and practicability properties of the proposed methodology. However these aspects have not been treated in this paper as a deeper work needs to be done.

**Acknowledgements.** This work is supported by TIN2009-14380 and DFG 157/2009.

## References

1. Robles, E., Grigera, J., Rossi, G.: Bridging Test and Model-Driven Approaches in Web Engineering, in: Gaedke M., Grossniklaus M, Díaz O. (eds.) ICWE 2009. LNCS, vol. 5648, pp. 136--150. Springer, Heidelberg (2009)
2. Ambler, S.W.: The object primer: agile modeling-driven development with UML 2.0. Cambridge University Press, Cambridge (2004)
3. Ferreira, J., Interaction Design and Agile Development: A Real- World.Pers.,Ph.D. (2007).
4. Propp, S., Buchholz, G., Forbrig, P.: Integration of Usability Evaluation and Model-based Software Development, Journal Advances in Engineering Software. Vol. 40 Issue 12. 1223—1230 (2009)
5. Losada,B., Urretavizcaya,M.,Fernández-Castro,I.: The InterMod Methodology: An Interface Engineering Process linked with Software Engineering Stages, In New Trends on HCI: Reseach, Develop., New Tools and Methods. Springer (2009)
6. Larman, C.: Agile & Iterative development: A manager's guide. Addison-Wesley, (2004).
7. Paternò, F. Model-Based Design and Evaluation of Interactive Applications, Springer-Verlag London, (1999)
8. eXtensible Interface Markup Language http://www.ximl.org/
9. Abrams,M., Helms, J., UIML Specification (2002)
10. Nolan, B., Brown, B., Balmelli, L., Bohn, T., Wahli, U.: Model Driven Systems Development with Rational Products. ibm.com/redbooks (2007)