

A WEB-CASE Tool Prototype for Hybrid Software Development

F.J. Orellana, F. Guil, I.M. del Aguila, and S. Túnez

Departamento de Lenguajes y Computación,
Universidad de Almería, 04120 Almería
{fjorella, fguil, imaguila, stunez}@ual.es

Abstract. In this paper we present a Web-based CASE tool for hybrid software projects that supports first stages of InSCo, an extended methodology based on CommonKADS and RUP. The tool InSCo Requisite will guide the development of a sort of software where knowledge-based components are integrated with traditional information systems. Furthermore, this tool will allow to manage several development projects at the same time, and determine the users which will take part in each development team.

1 Introduction

When we are developing software, based or not based on knowledge, developers choose a methodology depending on the problem, the prior knowledge, the available tools or other factors. These methodologies lead us to the production process determining the documents and artifacts we must generate, the activities to do and the order that we must follow.

The main artifact generated during the software development process, is a group of models [1]. The models of first stages in the process are often specified using natural language. However, this technique has problems like ambiguity, uncertainty or imprecision. Several alternatives have proposed a solution by means of the use of a structured natural language, description language, description languages of design, graphic notation or formal specifications, that are similar to the successfully used in the final stages of software development [5].

We propose to use these techniques in the organization modelling and requirements level scope, combining the forms and diagrams to model complex systems. In most cases, the models and documentation that we generate and maintain is very complex. For this reason, we need software tools that guide us in the generation of these artifacts, its maintenance, and the control of the integrity of changes. Nowadays, we can find tools which help us in the development of traditional software. However, our main objective is the extension of this kind of tools.

2 InSCo Methodology

In many application domains, there are problems that we need to solve by building a software system that uses knowledge engineering methods. These systems are known as hybrid systems. [2]

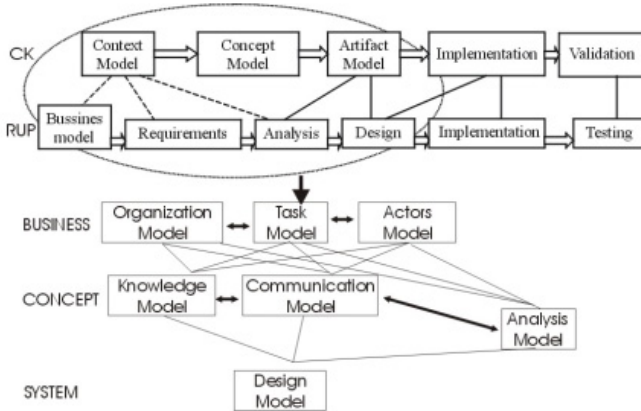


Fig. 1. Models of InSCo Methodology

Hybrid systems are needed in domains and organizations that must solve, on one hand, less structured problems with subjective requirements, uncertainty and imprecision. To deal with these problems, the incorporation of a Knowledge Based System is essential. On the other hand, it is necessary to add not-based on knowledge functionality (a traditional information system [2]) in order to reach the success of the organization and the KBS. For this reason, the methodologies that expect to support the development of this kind of software must create an unique solution, integrating both classes of software systems [4] [5].

The InSCo methodology [3] was designed from a requirements engineering perspective to deal with the hybrid software development . As we can see in Figure 1, InSCo proposes a total of seven main activities for developing hybrid software: Organization, Requirements , Knowledge, Analysis, Design, Implementation and Maintenance or Evolution. It is necessary to emphasize that the prototype we have developed covers the two first stages: Organization and Requirements.

The results of these two activities constitute the business and services models. The business model covers the organizational environment, the roles that take part in the project, the objectives of the organization, the information resources and finally, a list of items that allow us to make the decisions about the feasibility of the project. On the other hand, the services model represents a more detailed specification of the information that we obtain from the business model. This specification is related to functional, non-functional and informa-

tion requirements of the project. The InSCo methodology proposes the use of templates and graphic notation in order to represent all these models.

3 The InSCo Requisite Tool

A competitive methodology must have a set of associated tools for making the development of software based on it easier. In this paper we present InSCo Requisite, a tool designed for guiding us in the whole development process of the hybrid software. This tool helps the users to carry out the different tasks involved in the development, the forms construction or the administration of the diagrams associated.

One of the strong points of the application is the possibility of accessing the application via the Internet. Most of CASE tools designed for developing traditional or hybrid software need the installation of client programs to work with the application. InSCo Requisite is Web-based, so the only requirements we need will be an Internet connection and a browser. Nowadays, the use of Internet is very extended so, it is easy to find a fast and cheap Internet Connection. On the other hand, getting a browser is very easy because browsers are incorporated by default in most operative systems. Furthermore, this feature will allow the cooperation of users situated in different geographic areas, this way, the distance among the members of a project will not be a problem in order to carry out the development.

It is necessary to emphasize that InSCo Requisite can manage several hybrid software projects with independent development teams working at the same time. All the information related to these projects and the users that take part in them must be managed. For this reason, InSCo Requisite provides an administration area (Figure 2) which is integrated in the application, and that will allow the system administrator to control all these aspects of the application.

The screenshot shows the administration interface of the InSCo Requisite tool. At the top, the logo 'InSCo requisite' is visible. Below it, a status bar shows 'Estado: conectado' and 'Usuario: admin'. The main navigation area includes 'Opciones de Administración' with sub-options for 'Usuarios' and 'Proyectos'. The central part of the screen is titled 'Selección del administrador del proyecto' and contains a table of project data.

Nombre	Administrador	Acciones
Producción integrada en la agricultura	cmartin [Cambiar]	Ver + info Editar Eliminar
Proyecto de arrendamientos	fjorella [Cambiar]	Ver + info Editar Eliminar

Fig. 2. Administration Area

The main options we can find in the administration area are:

- Management of the projects. The administrator can insert new hybrid software projects, modify the data of the existing projects or delete them.
- Management of the users of the system. The main tasks that can be performed by the administrator are: registration of new users, the establishment of their access data (username and password) and update the information (personal details, access data)
- Appointment of responsible persons for each project. Each project must be managed individually by a person who will configure the main aspects of the project.

Getting more details about the management of the requirements in a hybrid software project, as we commented before, each project is formed by a group of users which will work with the whole group of templates that belong to the first stages of InSCo methodology. Their main task is the management of the documents and artifacts associated with the project they belong to. Furthermore, each project has a responsible person or administrator defined by the system administrator. This person can access to a group of options related to the project, which allow him to decide what users are going to participate in the project and configure some other aspects.

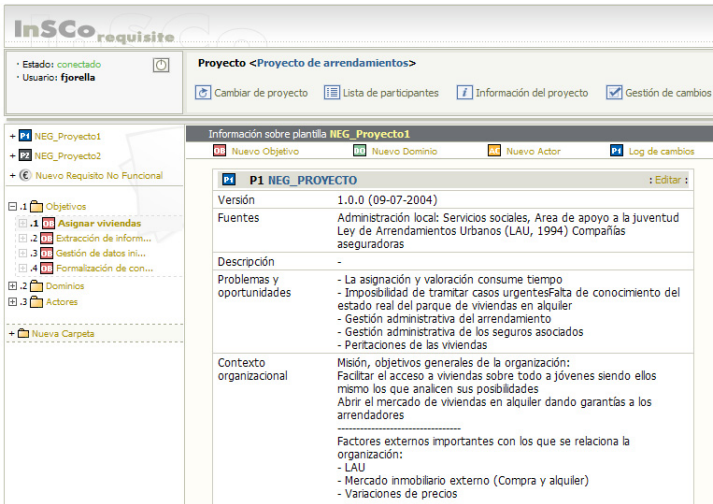


Fig. 3. Screenshot of InSCo Requisite

Each member can participate in a project interacting with the different templates proposed by the InSCo methodology. The possible operations can be carried using web-based forms (Figure 4) and hyperlinks. This way, each user can access to the whole group of templates, updating their information or establishing connections with other templates by using an intuitive and easy web interface.

Fig. 4. Forms

One of the most interesting features of the application is a tree-based menu (Figure 5), which represents the hierarchy of the templates that belong to a specified project. This tree allows the users to have a general view of the project and provides a easier method to classify and access to each element of the project. Due to the huge quantity of templates and information that take part in a project, one of the main objectives of InSCo Requisite has been to make the work lighter. For this reason, the application features this tree-based menu, icons associated to each template that allow us to identify them or texts that help us to perform the different processes.

We have proposed several improvements in order to increase the functionality of the application. As we commented, it is possible to represent the models of first stages using graphic notation like diagrams. Up to date, we can include this kind of notation in our project attaching a graphic file to a particular template. We propose the incorporation of a graphic editor integrated with the web application that allow us to create the diagrams without using an external application. Another improvement we have considered is the incorporation of a glossary of terms related to a specified project. The documentation of each hybrid software



Fig. 5. Tree-based Menu

project contains a lot of words and technical terms which are related to the project domain. Sometimes, the people who participate in the project are not familiarized with that domain terms. It would be a good idea that the members of a project could access to a glossary to look up the meaning of those words.

References

1. Object management group. In *MDA Guide Version 1.0.1*. OMG Document: omg/2003-06-01, 2003.
2. J. J. Cañadas, A. Bosch I. M. del Águila, and S. Túnez. An intelligent system for therapy control in a distributed organization. In *Proceedings of Information and Communication Technology (EurAsia-ICT 2002)*, Springer-Verlag. *Lecture Notes in Computer Science*, 2510:19–26, 2002.
3. I. M. del Águila, J. J. Cañadas, J. T. Palma, and S. Túnez. Integration of development methodologies for the building of knowledge intensive multiagent systems. In *In Proc. of the First Int. Conf. on Integration of Knowledge Intensive Multi-Agents Systems, KIMAS'03*, pages 203–208. IEEE Press, 2003.
4. I. Jacobson, G. Booch, and J. Rumbaugh. *El Proceso Unificado de Desarrollo del Software*. Addison Wesley, 2000.
5. G. Schreiber, H. Akkermans, A. Anjewierden, R. deHoog, N. Shadbolt, W. Van-deVelde, and B. Wielinga. *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press, 1999.