SOFTWARE—PRACTICE AND EXPERIENCE
Softw. Pract. Exper. 2002; **32**:833–836 (DOI: 10.1002/spe.461)

Specifying a role-based guide for learning to work with an enterprise framework

Wilhelm Hasselbring[1,*,†] and Ralph van den Houdt[2]

[1] *Software Engineering Group, Department of Computer Science, University of Oldenburg, Oldenburg, Germany*
[2] *IBS Consist B.V., Nieuwegein, Netherlands*

SUMMARY

Learning to work with enterprise frameworks requires considerable effort, because of the inherent complexity of all the knowledge that is needed. However, different roles in project teams need to learn different knowledge components according to the different tasks to be performed. A role-based guide that has been developed for new developers in the California projects at the software house International Business Systems, Inc. (IBS) Consist is presented. The California projects develop financial applications based on the IBM SanFrancisco enterprise framework. The requirement for this guide originates in the need to guide (new) project members in learning the required skills for being able to work productively after focused training for the California projects.

The starting point for this guide is the IBS development process. The individual tasks during the different phases in the process are specified by means of the Unified Modeling Language (UML) use case diagrams. An important property is that the guide is role-based to allow new project members to focus on their roles in the project, while learning to work with the framework. For each role, a UML activity diagram guides one to find the right paths for learning the required skills (which we call knowledge components) for the particular project role. The knowledge components are related to the tasks in the development process and to the roles. For all roles, at least some basic knowledge of the UML is required. Therefore, it is reasonable to specify this learning guide by means of the UML. This paper illustrates the use of the UML for specifying such a guide in a specific setting, namely the California projects of IBS Consist. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: enterprise framework; learning guide; object-oriented modeling; UML

1. INTRODUCTION

Object-oriented application frameworks are class hierarchies plus models of interactions that can be turned into complete applications through various kinds of inheritance and usage associations [1]. Design patterns often guide the construction and documentation of frameworks [2]. Frameworks offer

[*] Correspondence to: Wilhelm Hasselbring, Software Engineering Group, Department of Computer Science, University of Oldenburg, Oldenburg, Germany.
[†] E-mail: hasselbring@informatik.uni-oldenburg.de

the infrastructure for a set of compatible, reusable components. After several years of developing financial applications with the IBM SanFrancisco enterprise framework, International Business Systems, Inc. (IBS) Consist created a basic structure for their development process when using SanFrancisco. The definition of roles and tasks makes it possible to focus on these tasks to create paths for learning the required knowledge for new project members to help them with their new tasks. The goal is to create a guide that covers all the necessary knowledge components needed to contribute to the actual development of SanFrancisco applications for the California projects.

The notion of 'knowledge component' can be defined as an area of knowledge, which can be combined with other knowledge components to create learning paths. Using such knowledge components, we can create a guide that can easily be adjusted when new knowledge components or roles are introduced into the development process. Several questions are addressed as follows.

- What does the general development process look like?
- Which knowledge components are relevant in the development process and how do they relate to the roles of team members and to their tasks?
- How do the dependencies for learning the knowledge components look for each individual role?

The work presented here is an experience report in a specific setting. International Business Systems, Inc. (IBS), founded in 1969, is an international supplier of business software and professional services based on client/server solutions. IBS is a total solution vendor, offering software as well as professional services, hardware, training, and advice for implementations. IBS is specialized in business management and administration. Currently, IBS has about 2300 employees and a network of subsidiaries and business partners in more than 30 countries. As an acquisition in 1998, Consist B.V. became a Dutch subsidiary named IBS Consist B.V. The Consist expertise is mainly in financial software and human-resource management systems. California Financials is a department of IBS Consist, which since 1997 focuses on the development of new applications based on the SanFrancisco framework. Meanwhile, additional groups from other IBS subsidiaries cooperate in the California projects. The projects develop new software components based on the SanFrancisco framework, which can be combined to construct financial enterprise applications. IBM SanFrancisco is a Java-based enterprise framework, which enables the creation of server-side applications [3]. In this paper, only an coarse overview of the guide can be presented.

2. TASKS IN THE DEVELOPMENT PROCESS

The phases in the development process can be broken down into tasks. This section describes the tasks and relates them to the roles in the project teams. We specify those roles and tasks by means of the Unified Modeling Language (UML) use case diagrams [4]. In UML use case diagrams, it is shown which actors perform which use cases. Each use case is further specified informally as a transaction with the system to be developed. Here, we employ use cases to relate roles to development tasks, therefore modeling roles as actors and tasks as use cases. For example, the use case diagram in Figure 1 displays the roles for each task in the business logic design phase. The domain manager is a domain analyst too (specified in Figure 1 as specialization through inheritance [5]).

Business logic design addresses the translation of the system requirements into SanFrancisco specific models. The creation of these models is carried out by the designers. The test cases are specified together with domain analysts to create business-oriented test cases. The design manager role is the

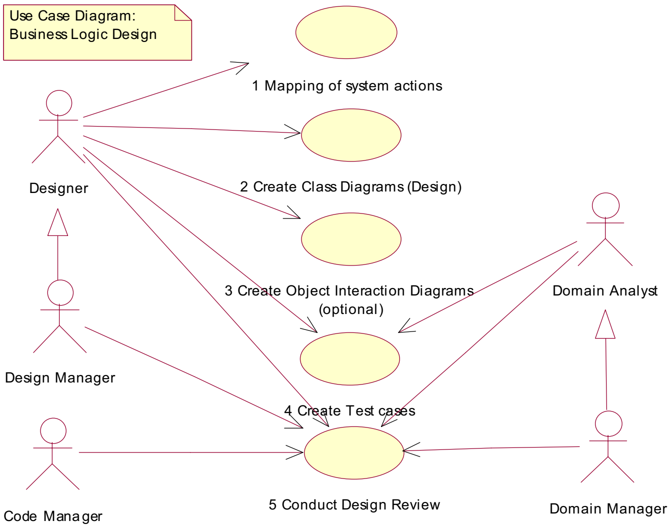

Figure 1. Use case diagram for business logic design.

driving force in this phase. The review is done by the design manager, assisted by the designers. If required, the domain manager is involved in the review.

Additional use case diagrams specify the roles and their related tasks in the requirements engineering, design, and coding phases. In software engineering, it is advised that the development tasks and responsibilities be assigned to the different roles that can be filled by the individual team members [6]. Each team member may fill several roles and each role may be filled by several team members. The roles are mapped to the tasks in the development process to describe the tasks for each role. These tasks also determine the knowledge components to be learned.

3. LEARNING ACTIVITY SPECIFICATION FOR THE ROLES

Knowledge components define learning modules, which are mapped to the tasks for which they are required. The required activity paths for learning the knowledge components by new project members in specific roles are specified. For that purpose, we use UML activity diagrams [4]. These activity diagrams visualize the path a new project member should take to study the relevant knowledge components. For example, Figure 2 displays the required learning activities for domain analysts. The domain analyst's knowledge learning path has an emphasis on UML modeling and documentation. Learning activity diagrams for designers and coders are more complex because of the high amount of tool knowledge needed.

4. CONCLUSIONS

The UML-based specification of a role-based guide for learning to work with an enterprise framework has been presented. The applied development process has been introduced as a basic structure and the

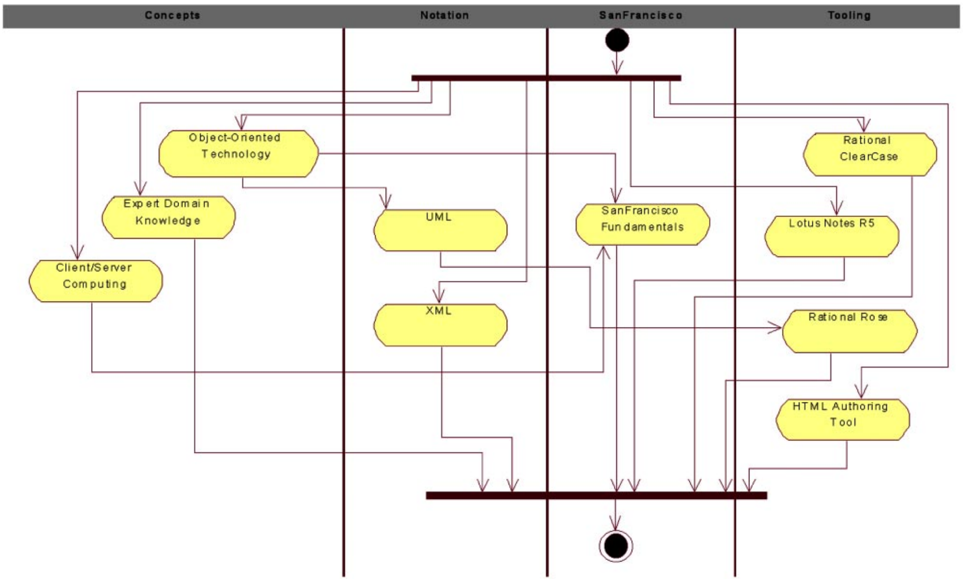

Figure 2. Learning activity diagram for domain analysts.

roles involved were identified. These roles have different responsibilities in the process. The roles are mapped to the various tasks in the different development phases. We have explained the knowledge components and ordered them into learning activity paths for the individual roles.

The identification of the required knowledge components to be included into the learning guide revealed about 20 components that can be divided into several categories. Many of the knowledge components are related to the SanFrancisco framework. The UML activity diagrams, which illustrate the dependencies for learning the individual components for each role, form the basic structure of the guide. They also illustrate a categorization of knowledge components by means of the 'swim lanes' in the learning activity diagrams (see Figure 2). Since for all roles at least some basic knowledge of the UML is required, all new project members start learning parts of the UML when studying the learning guide.

Looking at the identified components, development teams can be formed that consist of members who together have an advanced knowledge of all the required components, without the need to have one developer that has advanced knowledge of all components. Forming teams around the knowledge components should result in the leverage of the complete team's knowledge.

REFERENCES

1. Fayad ME, Schmidt DC. Object-oriented application frameworks. *Communications of the ACM* 1997; **40**(10):32–38.
2. Srinivasan S. Design patterns in object-oriented frameworks. *IEEE Computer* 1999; **32**(2):24–32.
3. *SanFrancisco Concepts and Facilities* (2nd edn). IBM: Rochester, 1998.
4. Booch G *et al*. *UML User Guide*. Addison-Wesley: Reading, MA, 1999; UML Resource Page http://www.omg.org/uml/.
5. Meyer B. *Object-Oriented Software Construction* (2nd edn). Prentice-Hall: Upper Saddle River, NJ, 1997.
6. Sommerville I. *Software Engineering* (6th edn). Addison-Wesley: Harlow, U.K., 2001.