

INSTRUCTIONAL DESIGN FOR JAVA ENTERPRISE COMPONENT TECHNOLOGY

Marco Marcellis, Ella Roubtsova and Bert Hoogveld
Open University of the Netherlands, PO 2960, 6401 DL, Heerlen, The Netherlands

Keywords: Component technology, Enterprise applications, Requirements, Instruction design, Teaching.

Abstract: We present a method for development of instructions for teaching component based development of enterprise application. The method considers development of an enterprise application as a complex task that has to be taught as a whole. The requirements to the user access and to the back-end systems serve as a natural means for the choice of the leaning tasks. We have fixed the back-end system and separated the task classes based on the requirements to the user access: a web-browser and an application client. Inside of each of those task classes the requirements are decomposed into "create", "retrieve", "update" and "remove" groups of functionality. Each of these functionalities can be seen as a simple enterprise application. Each of functionalities can be of different level of complexity and may be implemented with local and remote clients and different types of components. The Java enterprise component technology is used for implementation of learning tasks.

1 INTRODUCTION

Nowadays, almost any business role needs an enterprise application and the growing request for creating of such applications is supported by component technologies. Those technologies provide typical services used by enterprise applications and prefabricated component types for application design. For example, the Java EE technology offers several related component models, like the EJB 3.0 (Sriganesh et al., 2006) model for the server side components with the corresponding application servers, and the servlet technology (J2EE, 2009) for the server side components. There are also Integrated Development Environments (IDE) like NetBeans (NetBeans IDE, 2009), providing the infrastructure and wizards for custom deployment of components from given component types.

The process of development of an enterprise application consists of steps for creating three closely interrelated layers of an enterprise application, namely, a layer connecting the system with the database, a layer that presents the business logic of an enterprise and the client layer (Figure 1). Filling in those layers with the components and the glue code, so that the components communicate and their composition as a whole has the required functionality, is the task of the developer. Although IDEs offer automation for design of different layers, the choice of necessary com-

ponents and the glue code is to be done by the designer who should take into account the requirements to the complete enterprise system, use his knowledge of the component model and interface based component paradigm. This process of choices may become difficult because the available tutorials in this area usually focus on specific types of components, train isolated parts of the whole process and do not help to reach integrated objectives. It is usually called the "transfer paradox" (Merrill, 2002). Therefore, from the teaching perspective the component based development of an enterprise application is considered as a complex task that has to be taught as a whole.

In this paper we present a method for development of instructions for teaching component based development of enterprise application. The method applies the approach for development of effective learning material for complex tasks called Four Component Instructional Design model (van Merriënboer and Kirchner, 2007). The requirements to the user access and to the back-end systems serve as a natural means for the choice the leaning tasks. The exercises inside of task classes are decomposed into "create", "retrieve", "update" and "remove" groups of functionality. The Java enterprise component technology is used for implementation of learning tasks.

The structure of the paper is the following. Section 2 presents the approach for development of effective learning material for complex tasks. Section 3

describes how this method is applied to design of instructions for teaching of component based development of enterprise applications. Section 4 draws conclusions.

2 DESIGN OF INSTRUCTIONS FOR LEARNING OF COMPLEX TASKS

One of the most successful approaches for development of effective learning material for complex tasks is the approach called Four Component Instructional Design model (for short, 4C/ID-model) (van Merriënboer and Kirchner, 2007). The 4D/ID model is a holistic design approach opposite to an atomistic one. Atomistic approaches reduce a complex task to simpler elements that can be presented to students and trained. These approaches work good if the learning tasks have few interaction. The atomistic approaches do not work for teaching if the elements of the learning task are closely interrelated. The 4D/ID model deals with complexity of the learning tasks without losing of separate elements and the interconnections between those elements. A task given to students always presents a complete complex task but the complex tasks may be of different level of complexity. The 4D/ID approach proposes to choose the criteria and classify a set of complex learning tasks of different complexity. Each learning task can have several exercises. Some of the exercises have to be used as demonstrations, other - as partial demonstrations and should be finished by the students. The final exercises should be fulfilled by the students independently. This classification forms a training blue print that can be filled in with different learning tasks.

Applying holistic approaches in education allows solving the problem of fragmentation of knowledge. In the case of teaching of development of enterprise application the holistic approach prevents the situation when a student knows how to build one layer but cannot integrate it with the interrelated layers of the designed enterprise application. This approach also prevents the situation when a student cannot extend an enterprise application to meet new requirements. Using an integrated list of learning objectives for each of instructions allows students to overcome the "transfer paradox".

3 DESIGN OF INSTRUCTIONS FOR LEARNING OF COMPONENT BASED DEVELOPMENT OF ENTERPRISE APPLICATIONS

The criteria of separation of learning tasks depend on the domain of complex tasks. For the development of enterprise applications the requirements to the user access and to the back-end systems serve as a natural means for the choice the learning tasks. We have fixed the back-end system and separated the task classes based on the requirements to the user access: a web-browser and an application client (Figure 1). Inside of each of those task classes the requirements are decomposed into "create", "retrieve", "update" and "remove" groups of functionality. Each of these functionalities can be seen as a simple enterprise application. Each of functionalities can be of different level of complexity and may be implemented with local and remote clients and different types of Enterprise Java Beans (stateless and stateful session beans or message beans). The complexity of the business logic component and the number of middleware services and classes used for connection with the database grows if the implementation goes from the "retrieve" to "update", "create" and "delete" functionality. So, the implementations of standard functionality are the natural candidates for the exercises within one task class.

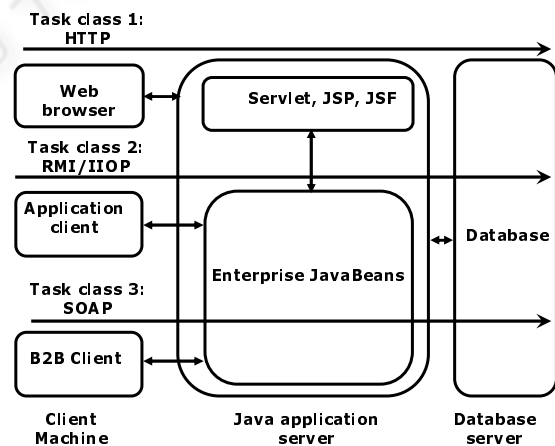


Figure 1: Layers of enterprise applications and learning task classes.

The exercise for implementation of the "retrieve" functionality is used as demonstration. It is shown as a grey box in Figure 2. The creative processes of relating requirements to the underlying business model and databases and modelling an application in terms of the component model are demonstrated to

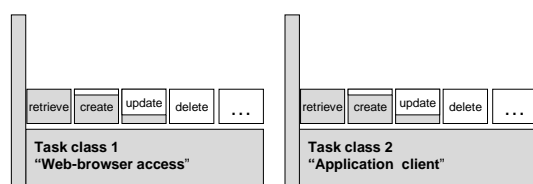


Figure 2: Learning task classes and exercises

the students as supportive material. The implementation of the "update" and "create" is partially supported and have to be finished by students. These exercises are shown as partially grey boxes in Figure 2. The "delete" functionality is fulfilled by the students independently. The box representing this exercise in Figure 2 is not coloured. From the outside, the complexity of a learning task is measured by the enterprise as demanded. From the inside, the complexity is defined by the number of services and techniques used to implement this functionality. So, the learning of component paradigm is built into the exercises of different complexity in each of the tasks of development of an enterprise application.

4 CONCLUSIONS

It is known that the holistic design of leaning tasks may simulate creating by students general cognitive schemes about the domain. When the students face with the similar tasks containing unfamiliar aspects they easily find solutions of those aspects (van Merriënboer and Kirchner, 2007). So, design of effective instructions in the area of development of enterprise applications should give students more skills and confidence.

We have applied the 4C/ID method of instructional design to the course Component based development. The application has shown that 70% of the students achieve the necessary skills fulfilling the three mentioned exercises and can independently implement the fourth exercise. After the training those students implement another enterprise application independently. For the 30% of students who need more training, an open electronic course may improve their training. As future work we are going to extend the set of learning tasks including the applications with business-to-business communication (Task 3, Figure 1).

REFERENCES

- J2EE (2009). <http://java.sun.com/products/servlet>.
- Merrill, M. (2002). First principles of instruction. *Educational Technology, Research and Development*, 50(3):43–59.
- NetBeans IDE (2009). <http://www.netbeans.org/>.
- Sriganesh, R., G.Brose, and M.Silverman (2006). Mastering enterprise javabeans 3.0. Wiley.
- van Merriënboer, J. and Kirchner, P. (2007). *Ten Steps to Complex Learning*. Lawrence Erlbaum Associates, Publishers, London.