Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

24

Instructional design for Java enterprise component technology



extended abstract

Marco Marcellis - Open University of the Netherlands, Faculty of Informatics
m.marcellis@telfort.nl

Ella Roubtsova - Open University of the Netherlands, Faculty of Informatics
ella.roubtsova@ou.nl

Bert Hoogveld - Open University of the Netherlands, Faculty of Informatics
bert.hoogveld@ou.nl

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 [1] model for the server side components with the corresponding application servers, and the servlet technology [2] for the server side components. There are also Integrated Development Environments (IDE) like NetBeans [3] providing the infrastructure and wizards for custom deployment of components from given component types.

INTERRELATED LAYERS

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 components 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 parts of the whole process, train isolated skills and do not help to reach integrated objectives. The situation when partial knowledge does not help in solving the complete task is usually called the 'transfer paradox' [4].

4C/ID-MODEL

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. 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) [5]. The 4D/ID approach proposes to choose the criteria and classify a set of compete complex learning tasks of different complexity. Each learning task can have several exercises. Some of the exercises have to

be used as demonstration, other - as partial demonstration and should be finished by the students. The final exercises should be fulfilled by the students independently.

TASK CLASSES

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 2).

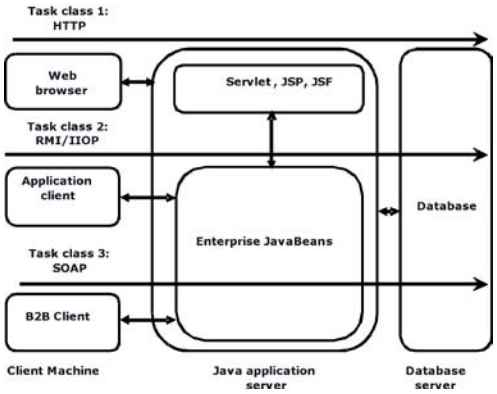

Figure 1.
Layers of enterprise applications and learning task classes

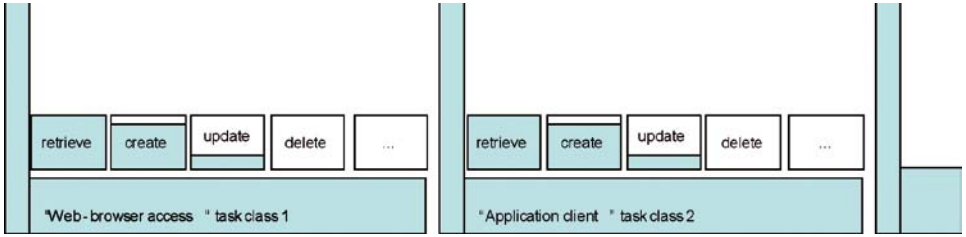

Figure 2.
Learning task classes and exercises

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 the functionalities can be of a 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.

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 modeling an application in terms of the component model are demonstrated to the students as supportive material. The implementation of the 'update' and 'create' is partially supported and has 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 colored. From the outside, the complexity of a learning task is measured by the complexity of the functionality provided 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.

RESULTS

We have applied this method of instructional design to the course Component based development taught to regular and commercial students at Open University of the Netherlands. 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 shown as Task 3 at Figure 1.

↑

REFERENCES

[1]
R.P. Sriganesh, G Brose, M. Silverman Mastering Enterprise JavaBeans 3.0. Wiley, 2006.
[2]
2EE. Java Servlet Technology, http://java.sun.com/products/servlet/.
[3]
NetBeans IDE. http://www.netbeans.org/.
[4]
Merrill, M. D. First Principles of instruction. Educational Technology, Research and Development 50(3), pages 43-59., 2002.
[5]
J.J.G.van Merrienboer, P.A. Kirchner. Ten Steps to Complex Learning. LawrenceErlbaum Associates, Publishers, London, 2007.