БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **60**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **60**

София • 2009 • Sofia

Three View Model of e-Portfolio Assessment System

Milen Petrov, Adelina Aleksieva-Petrova

Sofia University,1164 Sofia
E-mails: milenp@fmi.uni-sofia.bg adelina@fmi.uni-sofia.bg

I. Introduction

Some new forms of assessment which divert from the existing traditional forms (such as examinations and tests) have emerged with the development in the area of human abilities evaluation. These new forms have showed that they are helpful to evaluate the competence of a given human by assessing a number of feature competences. Such examples are the e-portfolio, 360 degrees feedback, peers assessment, etc.

The present work uses one of the new non-traditional forms of assessment (e-Portfolio), which is difficult to achieve with existing software means. So analysis, design and implementation of such a tool or system will add value of the research of the assessment area. As a result such assessment type can be integrated in the entire process of assessment. Also, assessment integration must comply with the existing standards and specifications in the area – for example IMS specifications [1].

There are many definitions of e-Portfolio, and one of them is that an electronic portfolio is a collection of electronic evidences assembled and managed by the user, generally published on the Web. An e-Portfolio provides proof for the users competence, and is a way of self-expression [2].

According to the IMS definition assessment e-Portfolios are used to demonstrate achievement to some authority by comparing these evidences of the portfolio to the standards defined by that authority [3].

The paper begins by giving a general picture of design research: the methodology; then the architecture design description of the system is presented. The architecture design is divided into three views, which are described in more details following the Unified Modeling Language (UML) diagrams [4] – use case view, logical view and implementation view. The paper also makes notes on the implementation and evaluation of the tool; conclusions, benefits from work and plans for future.

2. Methodology

Fig. 1 presents the classical design of the research [5].

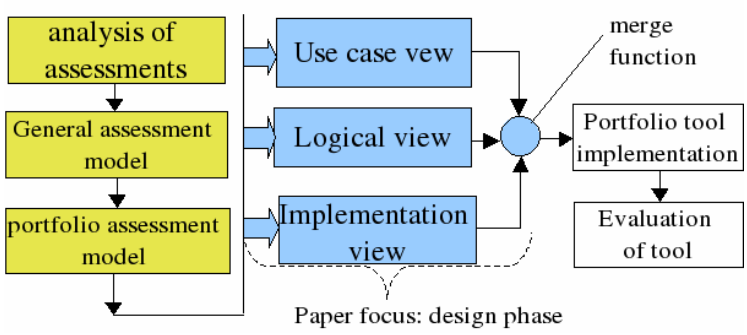

Fig. 1. Research methodology of e-Portfolio system

The first three phases of the research are shown on the left of the figure. First is an analysis of the problem of the integration of the non-traditional forms of assessment into the classical ones. During the second phase the goal was to design an assessment model for general use to be applied to the assessment process. The third phase is verification: the model was evaluated. These three phases are not an issue of the paper. The fourth phase, which the current paper is focused on, is the design of the software architecture for sys-tem/tool which supports such kind of assessments. The system design description is presented via:

- Use case view: shows the main assessment process, defined by UML use case diagrams. These diagrams classify the primary cases that will be implemented and describe the sequence of the processes in the system for each user case.
- Logical view: presents the main classes and their interaction within the system, using an-other type of UML diagram – an analysis class diagram. The interaction between classes and objects is given from user's point of view: presented through descriptions, pictures and diagrams of the functionality instead of concrete classes.
- Implementation view: shows the system main components from the developer's point of view: through framework descriptions, and descriptions of significant classes. This is done through package diagram and physical package structure of system.

On the right (Fig. 1) the last two phases of the research are shown – implementation and evaluation of the system. These phases are also not a point of the discussion here.

3. e-Portfolio assessment system architecture

The subsections below discuss each of the views of the system design description.

3.1. Use case view of the system

Fig. 2 presents the activities of the users and what they have to do to assess the portfolio by the designed software tool. The main user has access to the candidate's portfolio. The system supports assessment of the selected portfolio, and exports it to the external system or software.

The main scenario diagram to be discussed is "Perform assessment", which includes the score each assessment item has gained.

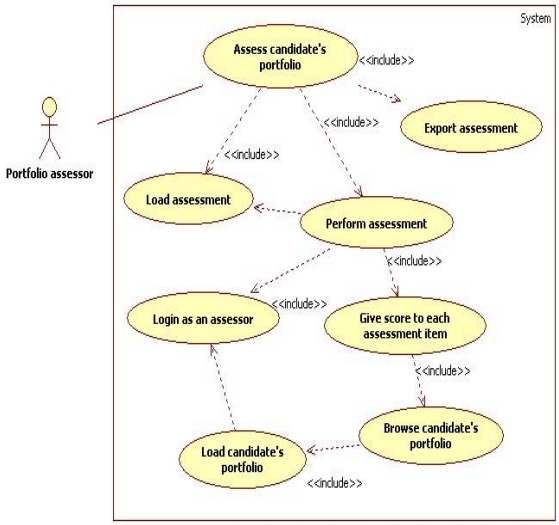

Fig. 2. Use cases of the process of e-Portfolio

It includes eight use cases such as: assess candidate portfolio, perform assessment (main use case), give score to each assessment item, browse candidate portfolio, load candidate portfolio, login as assessor, load assessment, and save assessment. The names of the use cases are more or less self-explanatory.

3.2. Logical view of the system

The logical view consists of diagrams, containing classes and illustrating the system functions. Fig. 3 shows a translation from user concepts to system concepts on the one hand. On the other it can be seen the relation between the basic concepts, which are helpful to both developers and software architects.

The analysis class diagram represents the business logic (Fig. 3), which is based on the main scenario of the system. The analysis classes give concept view of the parts of the tool and help to understand its architecture.

There are two kinds of activities, related to the tool user – portfolio assessor. The main activity is the Perform Assessment, and the secondary – Browse Portfolio. The stages of these activities are presented on Fig. 3: the logical layers (or also flows) are horizontal, and the physical layers (or packages) are vertical.

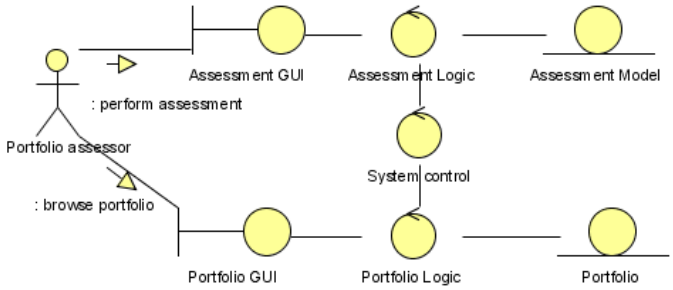

Fig. 3. Analysis class diagram of the functionality of the system

Both are described by boundary classes (the user interface), controller classes (business logic), and entity classes (data storage). Physical layers classes are for user interface – namely analysis classes AssessmentGUI from the main flow and PortfolioGUI from the secondary flow. Within these two flows we have controller logic analysis classes – respectively AssessmentLogic analysis class and PortfolioLogic analysis class. They are synchronized by SystemControl. Finally, the persistency layer is within the physical layer. AssessmentModel and Portfolio analysis classes are responsible for data storage.

The class analysis shows, that the system must contain at least two tools, which to be responsible for the different flows. Talking about a system, the tools can be viewed as separate modules from one integral system for portfolio assessment support. Despite the logical separation, we talk about a single system architecture at the end.

Fig. 4 shows the UML package diagram of the system. Its main aim is to describe the functionality of the application on the basis of a logical grouping. It organizes system elements into related groups to minimize dependencies between the packages.

The main packages represent: the assessment model, GUI, and the system business logic. The latter holds utility classes used in development.

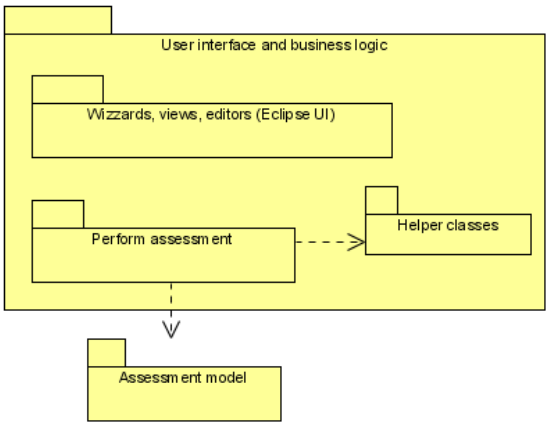

Fig. 4. Package diagram of the system's functionality

For example, eclipse environment [6] is used for development of the graphic interface, stored classes are needed for creation of wizards, views and editors and API for SWT and JFace components.

3.3. Implementation view of the system

This application contains several packages [7], grouped in a project, which contain the model implementation in the form of java classes. The names and structure of these classes are defined in the xml schema of the model.

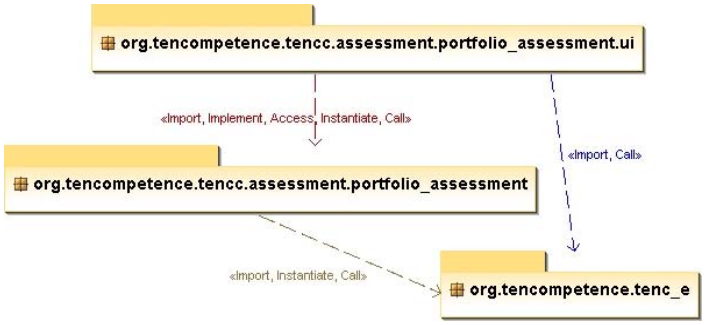

Fig. 5. Package structure of the application

There are two types of classes – on the one hand there are the classes, which are part of the business-logic of the application, and on the other hand – there are the purely technical classes for environment (framework) of the Eclipse Rich Client Platform (RCP). The technical classes are:

- Activator (extends AbstractUIPlugin) – Eclipse plug-in has a class connecting the project to the eclipse environment. The class provides callback methods, which can redefine the application to have access to specific events and resources of the environment. Typical methods are called by loading and unloading of software in plug-in by the Eclipse environment.
- Application – the Java class which controls the life cycle of the application. There is a similar function in the Activator; but the latter guarantee access to different type resources.
- The following three classes: ApplicationActionBarAdvisor, ApplicationWorkbenchAdvisor and ApplicationWorkbenchWindowAdvisor, are used to access resources and events in the Eclipse workbench. These classes manage the different parts of the environment, such as events, contextual menus, toolbars, etc.

The classes, which are part of the frame of the application, are divided into three groups:

- Comprehensive data models of the system – they are implemented through the class Model, which represent the logical organization of the system within the template design Model-View-Controller [8].
- Model Portfolio – implemented through classes PortfolioSection and PortfolioSubsection, which describe the structure of the portfolio as arranged set of files. The class PortfolioDAO, is used to extract portfolio from a database.
- Model of state of assessment – the different states which the application goes through during the run-time. It is the realization of the design patterns State and Observer [9]. It includes the classes PerformAssessmentSession, ISessionListener, and an enumeration type SessionState.

IV. Conclusions, limitations, benefits and future work

Based on these three views a system was designed and implemented. Fig. 6 shows a screenshot of the system.

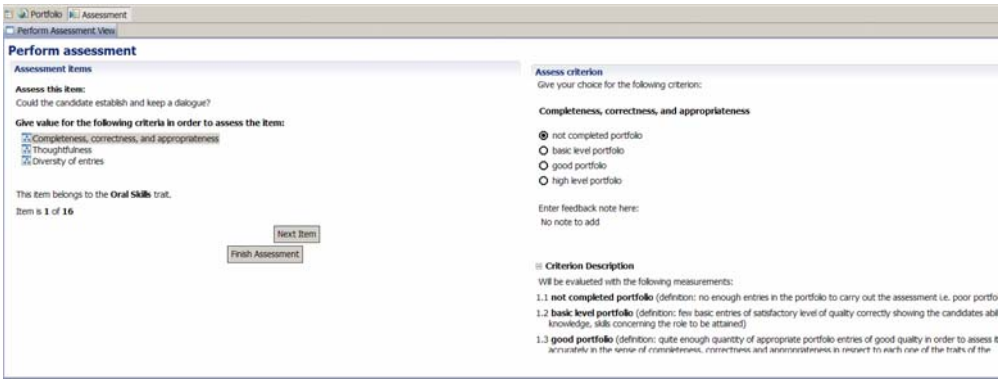

Fig. 6. A screenshot from the proof-of-concept assessment player tool

It has the capability to import and parse an xml file constructed according to the previously developed assessment specification [10]. It offers to its users to perform assessment activities according to assigned user roles. After the performance, it offers to the user to store the results using the same specification. This tool can also load the already performed assessment activities, and preview or evaluate results from the activities performed in the previous assessment run.

The evaluation of the system is carried out by black box testing on the basis of specially pre-pared evaluation samples. There are also unit tests to verify and evaluate model.

The system was evaluated and the results were satisfactory. The quality of the system could be improved, on the overall, however, it provides the necessary functionality to carry out the evaluation based on the TENCompetence Assessment Model.

In the process of system development the following key activities were performed:

- Research and analysis of problem areas.
- Definition of the system requirements.
- A module for developing and storing copies of the model was created. It includes the scheme itself, the source code generated by the scheme through JAXB technology [11], unit tests of the instance. The xml scheme developed presents instances of the model. The items in this scheme satisfy the constraints of the model.
- Choice of technology for development – Eclipse RCP was selected as a target platform for development, JAXB technology for automatic code generation, and MySQL as a supporting database.
- Design, development, and test of the system – individual components were identified and developed.

As a conclusion it could be said that the designed and developed system, which is based only on one type of non-traditional assessment is limited. Nevertheless, the current research is a very important step of assessment modeling

and assessment specification validation activities. The future work will include a wider adoption of other different methods of non-traditional assessment approaches.

Aknowledgements. The work reported in this paper is partially supported by the SISTER Project, funded by the European Commission in FP7-SP4 Capacities via agreement No 205030.

References

1. IMS Global Consortium, IMS Question & Test Interoperability Specification. 2006.
2. Hebert, Elizabeth. The Power of Portfolios – What Children Can Teach Us about Learning and Assessment. San Francisco, Jossey-Bass, 2001.
3. IMS Global Learning Consortium. IMS e-Portfolio Specification.
4. OMG, Unified Modeling Language.
   http://www.uml.org/, last visited 2008-10
5. Hevner, A., S. March, J. Park. Design Science in Information System Research. – MIS Quarterly, Vol. **28**, March 2004, No 1, p. 78.
6. Eclipse RCP.
   http://eclipse.org, last visited on: 2008-10
7. Borovska, P., M. Lazarova. Methodology for Parallel Application Development. – 5th International Conference "Challenges in Higher Education and Research in the 21st Century", Sozopol, Bulgaria, 5-9 June 2007.
8. Holub, A. Building User Interfaces for Object-Oriented Systems. JavaWorld, 1999.
9. Sun Microsystems. Customizing JAXB Bindings. 2005.
10. Gamma, E., R. Helm, R. Johnson, J. M. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0201633612, 1995.
11. Petrov, M., A. Aleksieva, H. Chipilov, S. Petrov. Ten Competence Assessment Specification, Version 1.0, 2007.
    http://hdl.handle.net/1820/1107

Модель системы для оценки, использующей e-Portfolio

Милен Петров, Аделина Алексиева-Петрова

Софийский университет, 1164 София
E-mails: milenp@fmi.uni-sofia.bg adelina@fmi.uni-sofia.bg

(Резюме)

В работе обсуждается софтверная архитектура системы, которая поддерживает нетрадиционную форму оценки, так названное e-portfolio. Архитектурная модель описана при помощи трех разновидностей: сценарий использования, логическая диаграмма, диаграмма осуществления. На основе этих диаграмм проектирована и разработана софтверная система. Обсуждаются некоторые проблемы применения.