Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34

Annales UMCS Informatica AI 5 (2006) 113-120

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Java Server Faces and Java Bean Technologies in Expert application

Barbara Gocłowska*, Zdzisław Łojewski

Department of Applied Computer Science, Institute of Computer Science, M. Curie Sklodowska University, 20-031 Lublin, pl. M. Curie Sklodowskiej 1, Poland

Abstract

The Expert program is a part of larger project (Intelligent Tutoring System) and plays a role of an assistant and disputant with the system user. The authors of the article analyse algorithm of this module and discuss various Java technologies for preparation of application.

1. Introduction

Most part [1,2] of Intelligent Tutorial Systems is constructed as applications in two-tier architecture (client-server). As the basic idea of ITS is determination of individual learning path for each user, first of all the problem of recording current information about the students using the system should be solved. Some authors organize [3] "synchronous" sessions within which indispensable information is recorded in the form of files in suitable folders. Making use of a three-tier architecture is a more convenient solution from the user's point of view as can be seen in papers [4,5]. This enables asynchronous learning.

However, it does not allow to use ready coatings but the system must be projected from the very beginning. Instead it gives the creators full freedom of choice of technologies, tools and impetus in projecting the system.

Our works on Intelligent Tutorial System LISE began from such a point. This article is an attempt at investigating the dilemmas which appeared during projection and realization of one Tutor System LISE module – Expert. At the same time this example reflects philosophy of LISE module structure.

2. LISE – Lubelian Intelligent Tutorial System

LISE is an intelligent software for individualized learning [5] and offering advanced rules of cognivistics. Like most ITS, the platform was designed based

*Corresponding author: *e-mail address*: gocbar@tytan.umcs.lublin.pl

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34

on ideas and advantages of private classes with a tutor simulating mechanisms providing effectiveness.

Like most teachers, at the beginning it presents range and field of knowledge to get to know in order to solve a problem. Then it gives a suitable set of test questions including the introduced problems.

Based on information gathered about the student using the system (part or overall results of tests, time of solving, time of learning unit realisation etc.), it proposes the most effective form of further learning. In this way, it achieves flexibility and versatility of classes adaptating for needs, skills and perception possibilities of the user.

The application was elaborated in the three-tier technology (client, server, and database). Also Java technologies were used of which Java Server Faces [6] played the most essential role. The class structure of the system is based first of all on the components PageBean, SessionBean as well as RequestBean, ApplicationBean (Figure 1) from the packages com.sun.jsfd.app.

The above mentioned components implement the interfaces AbstractApplicationBean respectively.

The life cycle of the most important process request – response (Fig. 1) in the Java Server Faces technology [7] is begun with the restore view and the FacesContext recording. The first page in our application encourages Expert to talk in a brief one – two sentence form ending with a question.

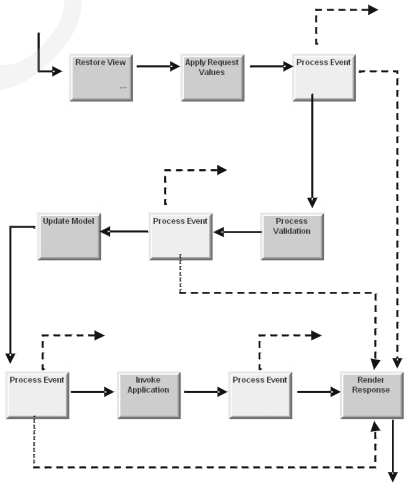

Fig. 1. The Request – Response process cycle in the Java Server Faces technology

The user writes a response using a graphic component, approves and sends response including an event (Process event), pressing button “send”. When the response is a value of another type than String, it must be converted to a suitable

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34

type (there are a few various converters – the most frequently used are: Boolean Converter, Data Time Converter, Integer Converter).

In the case of any troubles e.g. due to user's fault in the last phase of the cycle (Renderer Response) information will be generated using the component Message Group coming from the package com.sun.rave.web.ui.component. For detailed information connected with a specific graphic component e.g. text field, there is applied the component Message informing about student's error in a detailed way. The information of "Message" type is attached to each "risky" component in the process of application formation.

In the case of Expert module this plays an essential role, particularly if we predict errors connected with individual components or which the user can make. If everything proceeds smoothly, the response is sent to the server. The component Message Group can be also used to inform the student about response correctness but this way of communication is not the most convenient.

In the case of necessity of validation (Process Validations) of the value recorded by the student, it is convenient to use validators as a way directing the student to a rational response e.g. confining the range of the integer type value at the top and bottom.

There is also another reason for using a validator in Expert – limiting a number of words recorded by the student to make the semantic analysis of responses easier. Besides the standard comments of Expert drawn from the pool Expert Criticism to the component of textField type collaborating with the validator (Length Validator), the component Message informing the student about necessity of economical utterances. It case of need, the recorded value is converted to a suitable type and possibly validated (Update Model). Then the phase of response sending or invoking information in order to supplement or correct it by the user can take place (Invoke Application).

3. Expert Model

As stated above, one of the modules extending functionality of the educational platform LISE is Chatboot Expert. It performs a role of Tutor model assistant. In the case of student's erroneous response to the test question disclosing knowledge gaps or reasoning errors, thus of significant importance is that it complements the gaps in an attractive form of dialogue breaking the monotony of repeated reading of the same fragments of a handbook.

Similar to the commonly known chatboot ALICE (Artificial Intelligence Foundation) [8], using the conditional constructions, it leads a dialogue with the user. It asks questions and answers or comments on the obtained responses. It breaks in with sentences which are the most essential from the point of view of functionality of application element.

Responding to student's current responses, it creates illusion of a good tutor. Its role is not only teaching but also motivating.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34

The Expert model consists of n fragments of contents, n questions and n-1 auxiliary fragments (Fig. 2).

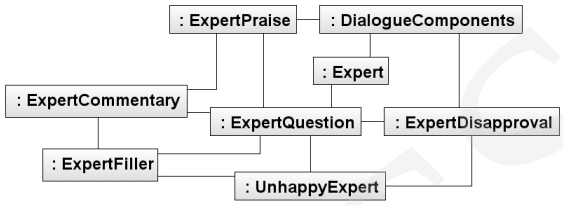

Fig. 2a. Scheme of Expert module

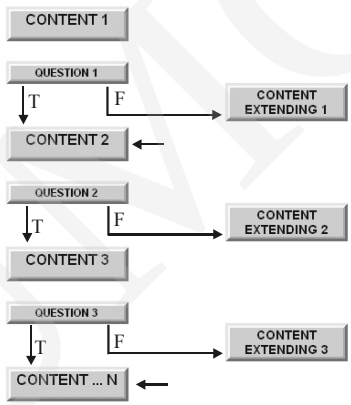

Fig. 2b. Dialogue part of Expert module

Expert activity consists in reading the user's response and in case if it is correct, in drawing a praise from the praise pool and moving to a successive fragment of the content.

When the response is incorrect, Expert draws "reproof" from the reproof poole and serves extended content for a given problem. Then not checking the acquired knowledge (the extended fragment is believed to be sufficient), it moves to another serving of knowledge fragment and to a successive question.

Business rules for the adaptative educational platform can be defined by means of commonly known conditional structures of the type (if …. Then …). For a given course, the business rules should not be frequently changed. Therefore they are defined and recorded in the Expert source code..

The fragment of code illustrating a joking dialogue in the case of participant's erroneous response is given below.

```
class Outrage extends Expert{
   String[] scandal = {" Erronous Response. ",
```

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34


```
        "  No, Not quite yes. Think!      ",
        "  Leave it, don't play the fool.  ",
        "  Eee, not exactly…               ",
        "  This is not it. Think.          ",
        "  You don't have a good day today.",
        "  You must be joking?             "};

public int list (){ return (scandal.length();
}

public String answer() {
String answ;
answ = scandal[(int) (Math.random() *
scandal.length)];
return (answ);
}
```

Store of the response drawn to chance will be extended depending on likes and dislikes as well as needs.

4. Expert Technology

In the first simplest version of Expert code fragments placed in the component ExpertBean were used in the dialogue with the user. The right side of Figure 3 shows, among others, the methods inherited from the abstract class AbstractPageBean and from the class FacesBean. The code was included into this class and connected with graphic components in a dynamic way (mainly button, staticText, textField).

In the second phase of works, due to necessity of separation of components used in various adaptative paragraphs of handbook many times which was associated with necessity of package formation containing the components of java and suitable interface classes, the method invoking proceeds through the component RequestBean or SessionBean e.g. #{RequestBean1.getStudentRequest} if the defined methods are directly in the component JSF or in #{SessionBean1.umcs.tytan.getExpertBean} in the case of additional package formation containing schematized classes and interfaces as shown in Fig. 4. In our package we based on the same project models.

Java Server Faces provide very convenient tools for guidance of information flow between the formed package and the graphic components used in application. It is even more obvious in the case of operation consisting in taking or recording information in the database where packages com.rave.web.ui.component and com.sun.data.provider.impl are applied.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34

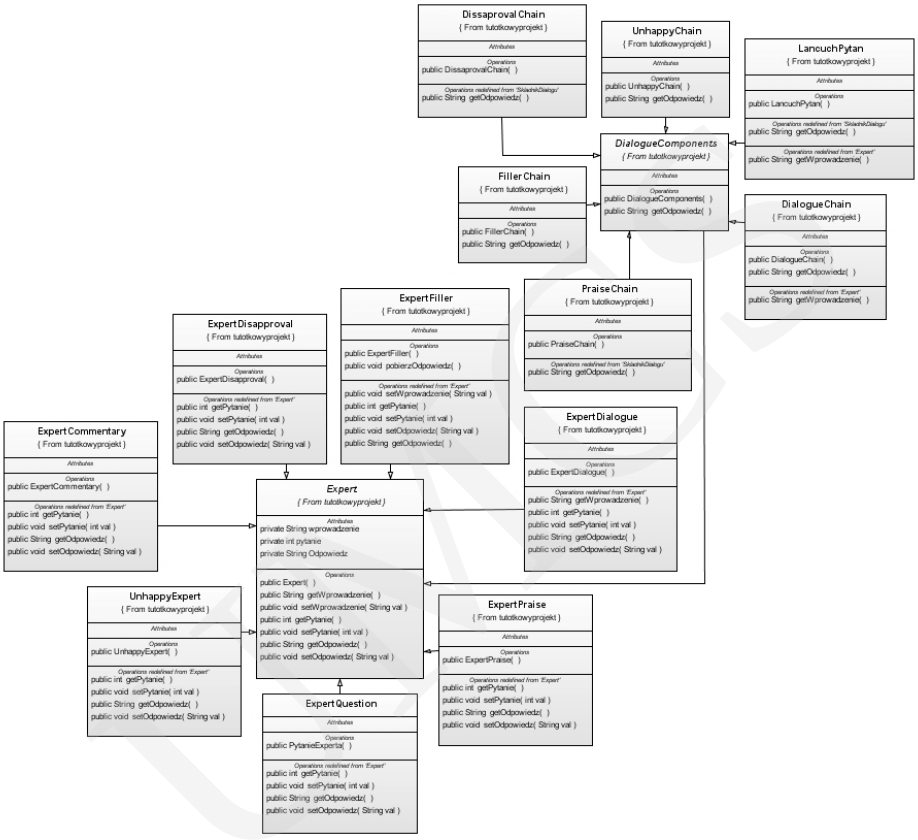

Fig. 3. Fragment of a simplified diagram of Intelligent Tutorial System LISE focusing on the chatboot Expert

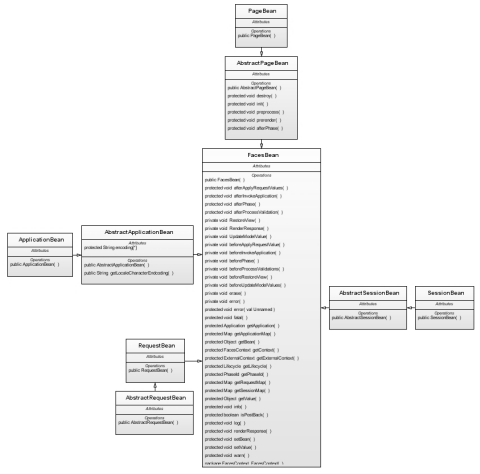

Fig. 4. Inheritance of LISE and some chatboot's classes

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34

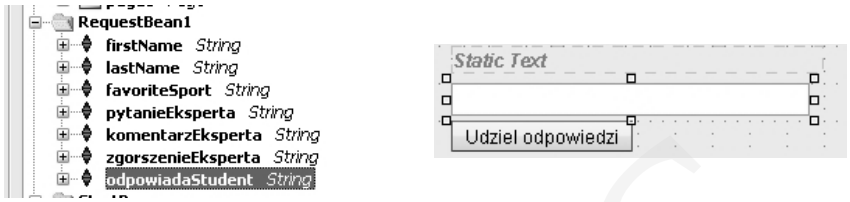

Fig. 5. Invoking methods (e.g. checking correctness of answer recorded by Student using the graphic component textField) can proceed directly from RequestBean or Sessionbean or from components recording in the package and also using the pool of session (in the case of our application) components

5. User's interface

Student records information to the text field according to the request of Expert. In response to this data Expert serves a supplementary content. It asks the student questions depending on correctness or draws reproof from the pool or sends the next page (Fig 6), praises, transfers another portion of knowledge and has a dialogue with a student. Figs. 6 and 7 present the result of the dialogue in a simple version.

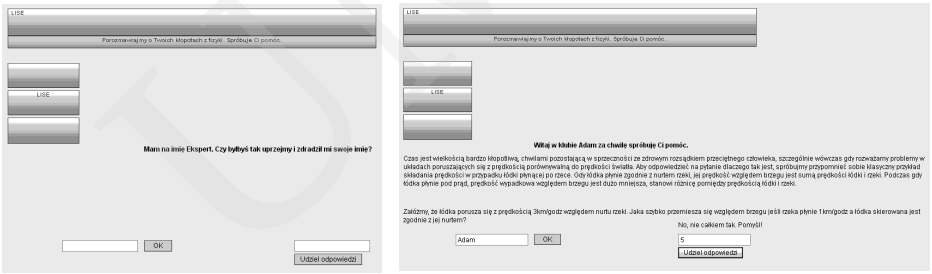

Fig. 6. User's dialogue with Expert. As shown after getting the answer of the user to the question about his name, Expert introduces time concept, presents a problem to be solved and asks a question. When a student records a wrong answer, one of reproofs is projected

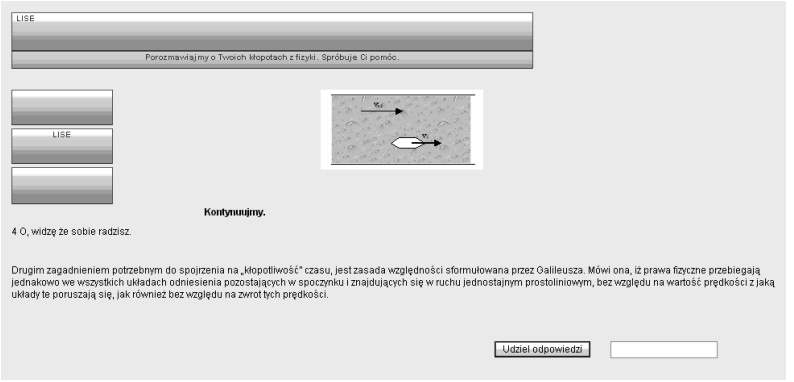

Fig. 7. Recording a correct answer by the user to Expert's question leads to a dialogue continuation

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 10/09/2025 12:14:34

Conclusions

Chatboot Expert was so programmed that it could make independent decisions, draw conclusions i.e. perform elementary actions occurring in the proposal process. In the present version there are a few such actions. Therefore we attempted to elaborate a vast set of interference rules. Application of such a module in a system definitely improves scalability of the system.

References

[1] Devedzic V., *Key Issues in Next-Generation Web-Based Education*, IDEE- Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, 33(3) (2003) IDEE-Proof 1-11.

[2] Alpert S., R. Singley M. K. Fairweather P., G. *Deploying Intelligent Tutors on the Web: An Architecture and an Example*, International Journal of Artificial Intelligence in Education, 10 (1999) 183.

[3] Cheung B., Hui L., Zhang J., Yiu S., M., *Smart tutor: An Intelligent tutoring system in web-based adult education*, Journal of Systems and Softwares, 68 (2003) 11.

[4] El-Khouly M. M., Far B. H., Koono Z., *Expert tutoring system for teaching computer programming languages*, Expert Systems with Applications, 18 (2000) 27.

[5] Zakharow K., *Feedback Micro-engeneering in EER-Tutor*, BSc. Honours Research Report, Dep. Computer Science & Software Engenireeng, University Canterbury, (2004).

[6] JavaServer Faces Component Library (JSFCL) Reference, http://developers.sun.com/prodtech/javatools/jscreator/reference/docs/apis/jsfcl/index.html, (2006).

[7] Geary D., Horstmann C., *Core Java Server Faces, Java 2Platform*, Enterprise Edition, Prentice Hall PTR, Upper Saddle River, Sun, (2004) 658.

[8] http://www.alicebot.org/