

Education Portal for Reactive and Proactive Service Provision

Stanimir Stoyanov

Department of Computer Systems
University of Plovdiv "Paisii Hilendarski",
bul. "Bulgaria" № 236, Plovdiv, Bulgaria
e-mail: stani@uni-plovdiv.bg

Georgi Cholakov

Department of Computer Systems
University of Plovdiv "Paisii Hilendarski",
bul. "Bulgaria" № 236, Plovdiv, Bulgaria
e-mail: gcholakov@uni-plovdiv.bg

Veselina Valkanova

Department of Computer Systems
University of Plovdiv "Paisii Hilendarski",
bul. "Bulgaria" № 236, Plovdiv, Bulgaria
e-mail: veselina_viva@abv.bg

Mincho Sandalski

Department of Computer Systems
University of Plovdiv "Paisii Hilendarski",
bul. "Bulgaria" № 236, Plovdiv, Bulgaria
e-mail: sandim@uni-plovdiv.bg

Abstract. The DeLC (Distributed eLearning Center) project aims to develop a reactive, proactive and personalized e-Learning environment. In the paper the agent- and service-oriented portal architecture is presented, ensuring reactivity and proactivity. Furthermore, reactive and proactive behavior of the architecture has been demonstrated by help of the Evaluator Assistant (EA). EA provides expert assistance to the lecturer in assessment of the electronic tests. The results of experiments with the assistant have been discussed. The application of flexible software architecture with reactive and proactive behavior is demonstrated in the paper.

Keywords - eLearning; education portals; agent- and service-oriented architectures; intelligent assistants; reactive and proactive behavior.

I. INTRODUCTION

In recent years, the interest towards electronic education has been growing stronger. As a result of that many universities have developed and implemented their own systems for electronic and long-distance education. Alternatively, many of the large IT corporations (e.g., Microsoft (Microsoft Class Server [1]), IBM (The IBM Learner Portal [2]), and HP (HP Learning Center [3]) have developed commercial systems. On the other hand, there are different open-source systems available on the market (the best known is Moodle [4]). A number of standards for electronic and life-long learning are also emerging. There are quite a few organizations working to develop specifications and standards such as IMS [11], ADL [5], ARIADNE [12], IEEE [13], ISO [14], etc. to provide a framework for e-Learning architectures, to facilitate interoperability, content packaging, content management, Learning Object Meta data, course sequencing, and many more. A significant role is played by the Sharable Content Object Reference Model 2004 (SCORM) standard [5].

DeLC (Distributed eLearning Center) is one of the projects aiming to develop an environment that supports electronic and long-distance forms of education. The center's

main ideas are laid out in a number of publications [6]. Why, despite the presence of so many systems, do we find it necessary to dwell on this subject? DeLC is mainly a scientific research project for developing new context-oriented and adaptive architectures. An important goal of this project is the development and experimentation with prototypes of such architectures in a certain application area - in our case e-Learning. To achieve this goal we develop service- and agent-oriented environment for the delivery of teaching materials and educational electronic services.

Furthermore, in many of the existing e-Learning systems, the interaction with the teachers is somewhat static – this is achieved mainly through pre-defined templates for choosing information resources. The information resources are the electronic equivalent of the traditional textbooks. Some of the existing systems use visualization and animation for improving the means of presenting the teaching materials. In our project we would like to research how such architectures can promote the development of electronic education environments, which support an interactive, proactive and personalized process of education and stimulate the students' creative and innovative thinking and performance.

In this publication, we present the DeLC education portal, which supports a reactive and proactive service provision. The rest of the paper is organized as follows. The second section presents the portal architecture. In the third section the Evaluator Assistant is described and experiment results are summarized. Finally, the fifth section concludes the paper and considers the state of the current development.

II. REACTIVE AND PROACTIVE PORTAL ARCHITECTURE

The proactivity improves the usability and friendliness of the system to the users. Proactivity means that the software can operate „on behalf of the user” and „activate itself” when it „estimates” that its intervention is necessary. In the portal, proactivity is ensured through „reinforcement” of the service-oriented architecture with intelligent components,

which demonstrate proactive behavior. Two approaches are available:

- Direct integration of intelligent components in the service-oriented architecture.
- Building a two component architecture where the front-end delivers services to the users and the back-end implements the reactive and proactive behavior of the portal.

For technological reasons (difficulties in the integration of two environments with different characteristics – portal frame and agent-oriented environment) we chose the second approach. So the intelligent components (agents), called „assistants”, will „live in” the back-end agent-oriented server (Figure 1).

The connection of the front-end and the AV is made through the middle layer of the portal architecture, where the electronic services are located. Depending on the direction of the asked assistance we distinguish reactive and proactive behavior of the architecture. In the reactive behavior the interaction between the two components is initiated by the portal. This is necessary in the cases when a user request is processed and a service needs an “expert” assistance. The service addresses the corresponding agent, located in the AV. The problem is that, in their nature, the services are passive and static software modules, intended mainly for the convenient realization and integration of some business functionality. Therefore they must „transfer” the responsibility for the activation and support of the connection to an active component of the architecture, as agents do. To do this, the service sends a concrete message to the agent’s environment, which, on its behalf, identifies the change of the environment and reacts by interpreting the message. Depending on the identified need of assistance the agent activates the necessary actions. The reactive behavior of the architecture could be implemented using a:

- Synchronous model – this model is analogous to calling subroutines in programming languages. In this model the service sends a message to AV and waits for the result from the corresponding agent before continuing its execution.
- Asynchronous model – in the asynchronous model the interaction is accomplished through some kind of a mechanism for sending and receiving messages.

In the proactive behavior (agents work „on behalf of the user”), an agent from the AV can determine that in its environment “something is happening”, that would be interesting for the user, who is assisted by that agent. The agent activates and it can perform certain actions to satisfy the preferences (wishes) of the user. The agent can inform the user of its actions through the educational portal.

The difficulties, associated with the management of the proactivity of our architecture, result from the fact that the portal is designed for reaction of the user’s requests. Therefore the proactivity can be managed only asynchronously and for this purpose we provide

development of a specialized service, which is to check a “mailbox” periodically for incoming messages from AV.

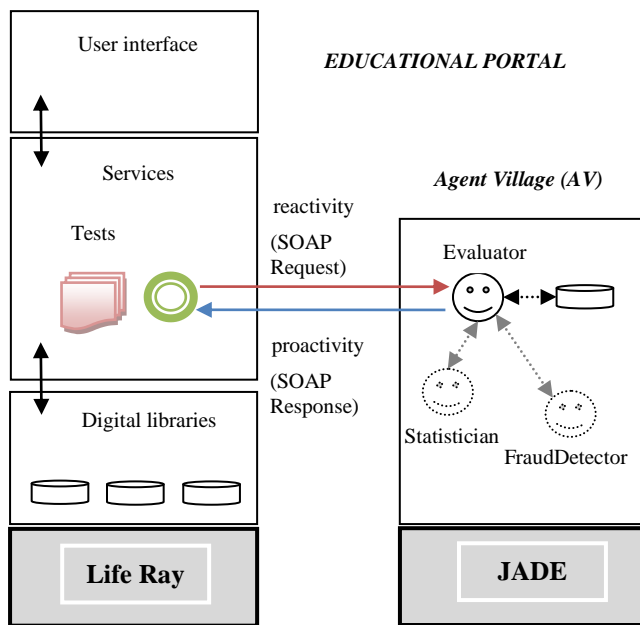


Figure 1. Extended architecture of the portal.

III. EVALUATOR ASSISTANT

The Evaluator Assistant (EA) provides expert assistance to the teacher in assessment of the electronic tests. In the Test Engine a system service is built for automated assessment of “choice like” questions. In the standard version of the architecture questions of the “free text” type are assessed by the teacher and the ratings are entered manually in the service to prepare the final assessment of the test. In the new architecture the Test Engine calls the assistant (an intelligent agent), which makes an “external” assessment of the “free text” type questions.

In case of need of an “external” assessment, where the Test Engine initiates a „request” for expert assistance, the reactive behavior of the EA is exploited. In order to be identified, the EA has a wrapper (the environment of the agent), which “masks” it as a web service for the portal. The Test Engine is extended with a new system service EstimationProcessor, which takes care of forming the request and processing the answer. When a request for assistance arises, this service generates a SOAP Request message and sends it to the Agent Village. When a SOAP Response is received, it parses the answer and extracts the estimated rating by the EA.

In the surrounding environment of the EA, the received SOAP Request messages are transformed into Agent Communication Language [10] messages, understandable for the agent. Some of the basic parameters of the messages are:

- Text, which is an answer of a “free text” type question.
- Parameters for the used estimation method.
- Maximum number of points for this answer.

The EA plans the processing of the request. In the current version of the assistant two methods are available for estimation. For these methods we have used some ideas from the existing algorithms for string matching [9]. The **Word-Matching (WM)** method counts „exact hits” of the keywords in the answer. The experiments show, that in short words (up to 5 symbols) this method gives relatively good results (over 50% matches). The minimum threshold of percentage match (i.e., a keyword to be considered as „guessed”), which is laid in the experiments, is between 70% and 80%. Intentionally, the method does not look for 100% match, in order to give a chance to words with some minor typos also to be recognized. To calculate the points, offered by this method, a coefficient is formed in the following way: the number of hits is divided by the number of keywords (RATIO1). The actual number of points for the answer is calculated as the maximum number of points is multiplied by this coefficient. The **Optimistic-Percentage (OP)** method makes an optimistic estimation of the points for the answer. Its essence is to iterate over the keywords list and summarize their percentage matches. Thus, the calculated amount of rates for each keyword, divided by the maximum possible match (in %), gives the reduction coefficient (RATIO2). The actual number of points for the answer is calculated by

multiplying the maximum number of points by the coefficient of the reduction. This method is more „tolerant” to allowing spelling mistakes in the answers, because low percentage matches are not ignored (unlike the first method) and are included in the formation of the final amount of points.

When the calculations finish, the EA generates an answer as an ACL message, which then is transformed by the environment into a SOAP Response message (a result from a web service call). In the answer there is a parameter, representing the calculated amount of points, extracted afterwards by the EstimationProcessor.

During the tests of the system there were used questions from the subjects “Introduction to Databases” (IDB) and “Enterprise Application Integration” (EAI). Here we will discuss some results from the tests of the EAI subject. The experiments show that while using the described methods above, the main challenge is the proper selection of a keywords set for each question. The exactness of the results depends on the length of the answer and on the length of the keywords list. The longer these lengths are, the greater is the probability of deviation of the results from the real ones. In Figure 2 is represented the raw data of the implemented tests in EAI using the described two methods, which is selected from the database containing all statistical data.

```

select substr(a.hits, 1, 4) hits, a.percentage, a.ratio1, a.ratio2, a.points1, a.points2,
       a.est_points est_points1, round(a.points2) est_points2, a.points_given,
       q.points max_points, a.answer_comment
from answers a, questions q
where a.question_id = q.question_id
and a.question_id in (151, 152)
order by a.question_id, order_answers(est_points1, est_points2, points_given) desc
    
```

	HITS	PERCENTAGE	RATIO1	RATIO2	POINTS1	POINTS2	EST_POINTS1	EST_POINTS2	POINTS_GIVEN	MAX_POINTS
13	5/12	773.0/1200.0	0.417	0.644	2.083	3.221	2	3	5	5
14	7/12	860.0/1200.0	0.583	0.717	2.917	3.583	3	4	5	5
15	9/12	972.0/1200.0	0.750	0.810	3.750	4.050	4	4	5	5
16	8/12	949.0/1200.0	0.667	0.791	3.333	3.954	3	4	1	5
17	5/12	665.0/1200.0	0.417	0.554	2.083	2.771	2	3	5	5
18	4/12	635.0/1200.0	0.333	0.529	1.667	2.646	2	3	5	5
19	6/12	749.0/1200.0	0.500	0.624	2.500	3.121	3	3	5	5
20	4/12	665.0/1200.0	0.333	0.554	1.667	2.771	2	3	4	5
21	7/12	823.0/1200.0	0.583	0.686	2.917	3.429	3	3	5	5
22	5/12	682.0/1200.0	0.417	0.568	2.083	2.842	2	3	4	5
23	8/12	924.0/1200.0	0.667	0.770	3.333	3.850	3	4	5	5
24	5/12	720.0/1200.0	0.417	0.600	2.083	3.000	2	3	5	5
25	7/12	866.0/1200.0	0.583	0.722	2.917	3.608	3	4	4	5
26	2/12	502.0/1200.0	0.167	0.418	0.833	2.092	1	2	5	5
27	5/12	728.0/1200.0	0.417	0.607	2.083	3.033	2	3	0	5
28	2/12	536.0/1200.0	0.167	0.447	0.833	2.233	1	2	5	5
29	5/12	745.0/1200.0	0.417	0.621	2.083	3.104	2	3	4	5

Figure 2. Raw results from processing the EAI subject answers.

The columns of the result set are HITS (keyword hits of WM), PERCENTAGE (summarized percents of OP), RATIO1 (a coefficient for calculating the points in WP), RATIO2 (a coefficient of reduction in OP), POINTS1 (points, calculated by WM, EST_POINTS1 – rounded), POINTS2 (points, calculated by OP, EST_POINTS2 – rounded), POINTS_GIVEN (points given by the assessing teacher), MAX_POINTS (maximum number of points for this answer).

Let us pay more attention to answers 16 and 27. The both methods calculated significantly higher ratings than that of the teacher, whose comment about these answers is:

“Copy/Paste does not bring points!”. And these cases are not isolated. Here is where the functionality of FraudDetector agent would be particularly useful, and this inspired the idea for its appearance in the Agent Village.

In Figure 3, comparisons of the final scores given by the two methods, and by the assessing teacher are presented. It is a clear trend that the scores given by the teacher are the “most generous”, the optimistic method (OP) is in the second place, and the most exacting method is the one for matching the words (WM).

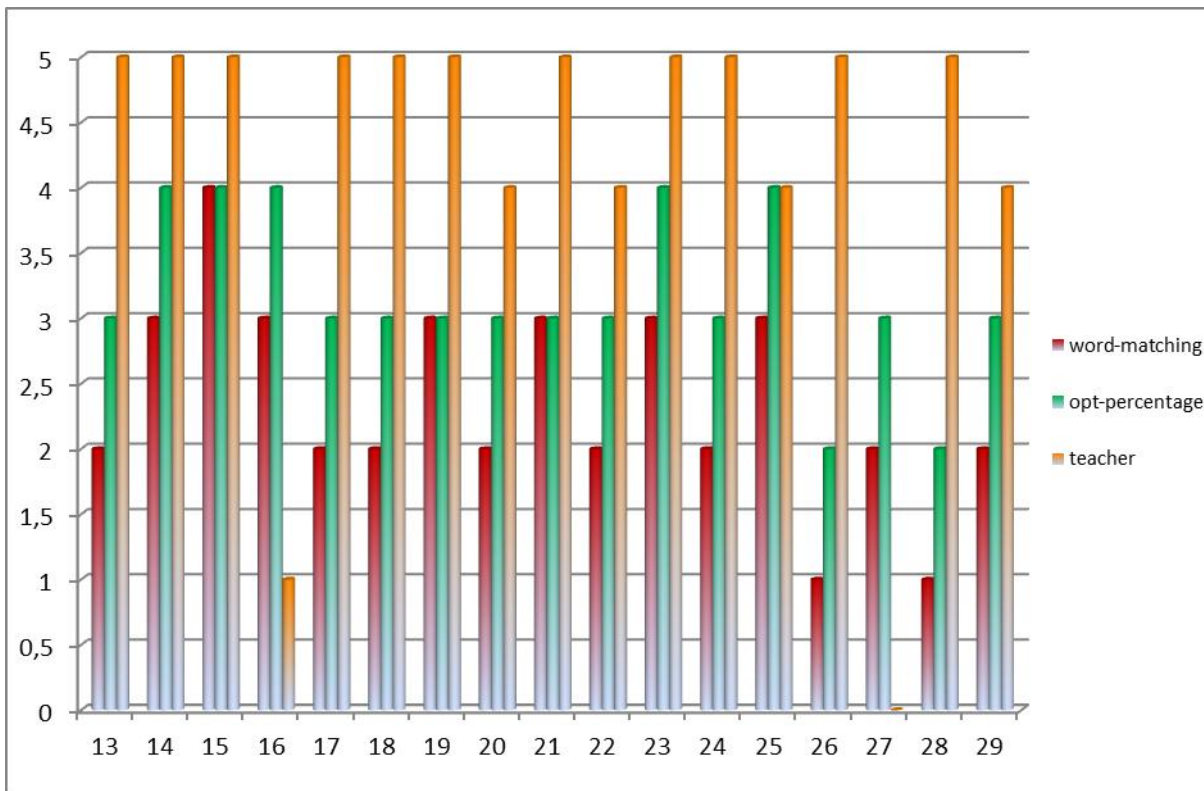


Figure 3. Comparison of the points given by the between algorithms and the teacher (EAI).

IV. CONCLUSION AND FUTURE WORKS

The portal is testing in the University of Plovdiv and a secondary school in Plovdiv for a year and 360 students had sat for examination in two subjects – „Introduction to Databases” and “Enterprise Application Integration”. Two new assistants are in a process of development. The FraudDetector will try to recognize any attempts to cheat in the answers given by the students. Such attempts would be to guess the keywords or copy/paste results from Internet search engines. This assistant will cooperate with the Evaluator agent and if its receptors detect a probability of a cheating attempt, it will inform the Evaluator agent, which will inform the assessing teacher that this answer requires a special attention, because it is a suspicious one. The

Statistician will store information about all processed answers with a full history of the details from all calculating methods used by the Evaluator agent. This assistant will need a feedback how many points are finally given by the teacher for each answer. Thus it will accumulate a knowledge base for each teacher and will be able to decide which of the methods best suits the assessment style of the current assessing teacher. Upon returning the results of the Evaluator assistant, information by this agent will determine which results from each method will be presented to the teacher as a main result, and the results of the other methods will be presented as an alternative. Another feature of this agent will be also to provide actual statistics on the performance of each of the calculating methods, as the “weakest” of them will go out of service until new and better performing methods are added to the Evaluator agent. This

monitoring of the methods' behavior becomes really significant when the so-called genetic algorithms are added, which we are still working on – as it is known, they can be “trained” and thus their effectiveness can change. In this process a knowledge base will be developed for each specific subject, which will support the methods in their work. These knowledge bases later will be transformed into ontologies for the Evaluator agent.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the National Science Fund (Research Project Ref. No. DO02-149/2008 and IO-01-01/2006).

REFERENCES

- [1] Microsoft Class Server, www.microsoft.com/education/products, (17.05.2011)
- [2] The IBM Learner Portal, <https://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=a0006064>, (29.03.2011)
- [3] HP Learning Center, <http://h30187.www3.hp.com/>, (21.04.2011)
- [4] Moodle, www.moodle.org, (11.05.2011)
- [5] SCORM 2004, www.adlnet.gov/Technologies/scorm/SCORMSDocuments/, (16.05.2011)
- [6] Stoyanov, S., Ganchev, I., Popchev, I., and O'Droma, M., From CBT to e-Learning, Journal Information Technologies and Control, No. 4/2005, Bulgarian Academy of Sciences, pp. 2-10
- [7] Stoyanov, S. and Popchev, I. (2006). Evolutionary Development of an Infrastructure Supporting the Transition from CBT to e-Learning, Cybernetics and Information Technologies (CIT), 2/2006, Bulgarian Academy of Sciences, pp. 101-114
- [8] Stoyanov, S., Popchev, I., Doychev, E., Mitev, D., Valkanov, V., Stoyanova-Doycheva, A., Valkanova, V., and Minov, I., DeLC Educational Portal, Cybernetics and Information Technologies (CIT), Vol.10, No 3., Bulgarian Academy of Sciences, pp. 49-69
- [9] Knuth, D. E. , The Art of Computer Programming: Volume 3, Sorting and Searching, Addison-Wesley, 1973
- [10] ACL. Agent Communication Language Specifications, www.fipa.org/repository/aclspecs.html, (28.03.2011)
- [11] IMS Global Learning Consortium, www.imsglobal.org, (23.04.2011)
- [12] ARIADNE, www.ariadne-eu.org/, (28.03.2011)
- [13] IEEE, www.ieee.org, (20.03.2011)
- [14] ISO, www.iso.org, (29.03.2011)