BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **9**, No 3

Sofia • 2009

Selbo 2 – an Environment for Creating Electronic Content in Software Engineering

Damyan Mitev[1], *Stanimir Stoyanov*[1], *Ivan Popchev*[2]

[1] *University of Plovdiv, 1000 Plovdiv*
[2] *Institute of Information Technologies, 1113 Sofia*
E-mails: damyan_mitev@mail.bg stani@uni-plovdiv.bg ipopchev@iit.bas.bg

Abstract: *This paper gives a brief overview of Selbo 2 – environment for creating SCORM compatible electronic content. Selbo 2 uses intelligent editors (combination of component and agent) to manipulate learning content and aid content developer during content creation. Ontologies provide developers with predefined resources covering specific domain that can be used directly in the content. Selbo 2 also utilizes education templates that define pedagogical goals and agents to govern them. Furthermore, the environment employs schemes for collaborating with the Learning Management System (LMS).*

Keywords: *Software architectures, e-Learning, DeLC, intelligent editors, ontologies, agents.*

1. Introduction

In the past few years a project for delivery of electronic content to the student, called Distributed e-Learning Center (DeLC), is being implemented in Faculty of Mathematics and Informatics at the University of Plovdiv. The project aims at creation of flexible, adaptive and collaborative infrastructure, effectively supporting systems for personalized e-Learning [1-4]. The personalization of the educational content raises two questions – how it can be delivered to its users (the students) and how the teachers can be aided in creating flexible and parameterized electronic content that can be personalized for the needs of a specific user. The electronic content is compatible with the e-Learning standard SCORM 2004 [5]. There are a number of specialized editors for creating SCORM conformant content, where for

example some of the most popular are Reload's SCORM Editor [6], DeltaLearn's Advanced SCORM Editor [7], and the University of Nottingham's Xerte [8].

Creating SCORM compliant content by using specialized editors is difficult for many reasons. One of them is that in order to be able to produce non-trivial materials, teachers must know the standard in great details. Another reason is that the content creators must have a lot of practical experience in using SCORM editors. In order to facilitate content creators in DeLC, we have developed our own environment, called Selbo 2, which aims at facilitating the teachers in producing SCORM compliant content. The environment allows creating of new content, editing available content, as well as generating educational units out of preexisting standardized elements. The major advantage is that the authors work in terms of the discipline, for which the materials are being created. The architecture of the environment is developed so, that adaptation for different disciplines is easily achieved. The first version is developed for the application of the discipline "Software Engineering".

We present in the paper the current results of our efforts to develop Selbo 2 environment. The paper content is as follows: Selbo 2 architecture is presented in the second section; the main usage scenarios of the environment are described in the third section and finally we discus some problems of the implementation and future advancements.

2. Selbo 2 architecture

For the organization of personalized electronic content in the DeLC project, clear distinction and explicit presentation of three types of knowledge – specialized knowledge of the studied discipline, knowledge of the used pedagogical approach and knowledge characterizing the individuality of the students and teachers is proposed. In the infrastructure of DeLC thus proposed this knowledge is represented and managed as three independent models: a user model, a pedagogical model and a domain model [9]. The models affect both the generation and the interpretation of the electronic content and the specialized supporting components are implemented in Selbo 2 architecture.

Selbo 2 is a prototype environment which can be applied in schools, universities, distance learning or lifelong learning initiatives. The project aims to ease electronic content developers, providing them with intuitive user interface and focusing on logical structure of the content, and not on technical details. Although Selbo 2 can be adapted to different domains, we are creating specialized content editors mainly for the discipline "Software Engineering".

Selbo 2 uses some ideas from its predecessor SELBO [10, 11]. The major differences between the two are their component architectures. Selbo 2 has numerous new user interface features, new ontologies and new intelligent resource editors as well. The generated learning resources are saved as HTML pages collections. HTML format has been chosen because it is the mature standard for data visualization and allows arbitrary multimedia to be embedded in a page. The output of the content editors is converted to popular multimedia objects (mainly pictures) and is subsequently embedded into the pages. The resulting HTML pages

and their multimedia content (for e.g. pictures, video or audio) can be transformed according to SCORM 2004 requirements.

The functionality of the environment is divided between three architectural layers – one client and two server layers, as shown on **Error! Reference source not found.** The first layer (the GUI) consists of client components, which provides the users with graphical user interface for accessing the services of the environment. The second layer (Middleware) consists of server (or business logic) components, which realize the functionality of the environment. The third layer (e-Learning resources), also situated in the server part, represents the electronic learning content.

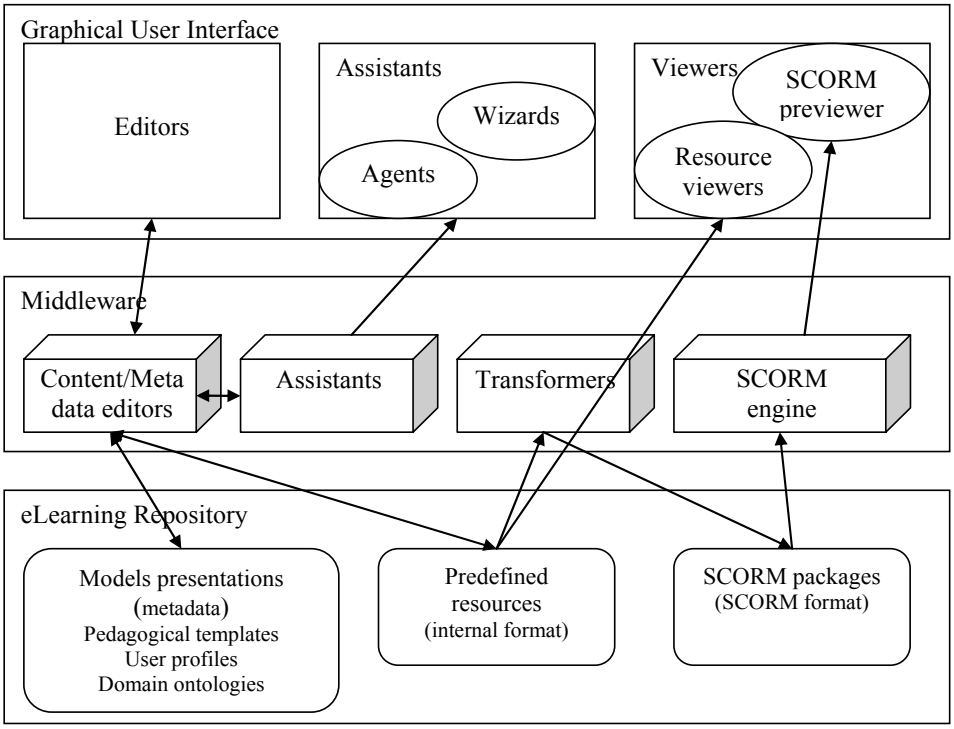

Fig. 1. Selbo 2 architecture

2.1. GUI

GUI layer is the visual representation of the components of the environment. Content creators interact with the system on this layer. In this layer three types of graphical interfaces are situated – editors, assistants and viewers, which provide the user with interactive tools to manipulate the electronic content and its resources.

We distinguish two types of editors – model editors and electronic content editors.

Model (or meta-data) editors are used by content creators to edit the meta-data for the three models. This meta-data can be objects, relations, rules, restrictions, etc. They are used to classify and aggregate knowledge, to navigate electronic content, to present user profiles and pedagogical goals, etc.

Content editors are used to generate and edit electronic educational resources that will comprise the electronic lesson. Content editors are described in more details later in this section.

Assistants are designed to help the content creator in designing the electronic lectures. Assistants are described in more details later.

Viewers provide only means to visualize resources, but not to edit them. Examples of such viewers are a viewer for predefined resources and an ontology visualizer. Content creators are not supposed to modify the predefined resources and the ontologies, because they are created by specialists in the selected domain and provide well balanced set of reusable material and metadata description of notions, terms and concepts with explicitly defined semantics.

SCORM previewer allows the teacher to preview the generated SCORM lesson. Although it can be regarded as a viewer, it is outlined as a separate component, because of its significant importance in the lesson creation process. It is an invaluable tool, which enables the teacher to achieve almost WYSWYG style of work with the environment.

2.2. Middleware

The middleware (or functional) layer consists of server components, which implement the main functionality of the environment. These server components are transparent to the user. The interaction with them is performed only through their corresponding graphical user interface component in the client part of the system.

Content editors are responsible for the creation and manipulation of the individual resources that comprise the educational unit. The resource is any multimedia object that can be presented to the learner. Examples of such objects are a text file, HTML file, or a picture. Selbo 2 recognizes some widely spread multimedia formats as resources. Content editors can be put in two distinct groups. The first group consists of editors of standard formats, which are independent of the domain of the educational unit, such as HTML editor, text editor, simple image editor. The second group consist of specialized editors, specific to a given domain, such as UML editor for Software Engineering, chemical formulae editor for Chemistry, etc. The prototype version of Selbo 2 is intended for creating materials for Software Engineering. For this reason we are focusing on standard HTML and text editors and a specialized editor for creating UML diagrams for Software Engineering. The editors of interactive content, which are used to create training materials, and most notably, tests, play an important role in this architecture. The content editors in Selbo 2 are implemented as intelligent editors – combination of a GUI component (editor) and an intelligent agent, specially created to inter-operate with this component.

The resources are the actual pieces that form a specific educational unit. The content creator may choose to create these resources from scratch, or to use as a base some predefined resources. He may choose to use the built-in content editors to edit the resources, or use external applications.

Tests are a special form of resources. They are interactive units, which provide the feedback from the learner to the learning management system. The system uses this feedback information to evaluate the sequencing and navigation rules, defined in the resource organizer. Based on that evaluation, the LMS can choose the most

suitable content for the learner. Tests are a crucial didactical instrument for control and evaluation of learner's knowledge.

The content editors, and more precisely domain specific content editors, maintain the domain model.

Resource organizer is a special type of editor. Instead of modifying the resource that will be used in the lesson, it groups separate (atomic) educational resources into a structure, representing a single educational unit. Requirements of the pedagogical model are defined over this structure, as it stores the rules of sequencing and navigation among the individual resources. In Selbo 2 the component, which is used as a resource organizer, is the content tree. The resources could be organized in different structures – from plain lists to complex graphs. We have chosen a tree, because it balances simplicity and expressiveness of possible content organizations. In addition, trees are perfect for representing the inherent hierarchical structure of classic educational resources, such as textbooks or lectures.

The resource organizer is one of the components, which maintains the pedagogical model, as it stores the didactical goals of the educational unit.

Agents and wizards, referred to collectively as *assistants*, are software entities, which automate the creation of electronic resources. In this architecture, the assistants play a very important role, because they inter-operate with all of the components in the system to aid and ease the content developer during content creation. Selbo 2 uses two kinds of assistants: wizards and agents. Wizards are passive assistants; they are called explicitly by the user, and are not active during the whole process of development. Their role is mainly to initialize other components, so they are used in the creation of new objects, like educational units or different resources. The agents, in contrast to wizards, are constantly active assistants. They work autonomously, inter-operating with other system components and agents. As a result of this inter-operation and agent's goals, they can make decisions and undertake actions, related to the developer's work. Selbo 2 uses JADE as an agent platform. For this reason Selbo's notion of an agent is restricted by JADE's implementation. For our purposes an agent is entity with independent control mechanism, capable of exhibiting different behaviors, able to communicate with other agents and GUI components in the environment. Assistants are used to ease user's (content creator's) interaction with the environment; therefore they maintain the user model.

Models govern the behaviour of other components in the system. The models are not distinct components of the system. They are rather a set of specialized knowledge of the studied discipline (domain model), knowledge of the used pedagogical approach (pedagogical model) and knowledge, characterizing the individuality of the students and teachers (user model). This knowledge is used as guidelines in the design and subsequent operation of editors and agents.

Transformers are used to transform the internal structure of the educational unit and its resources into popular standards for electronic education (and vice versa). Selbo 2 is oriented primarily to supporting SCORM 2004 standard. The author of the electronic content can use existing electronic material (possibly taken from a repository) as input for the environment, or can create this material from scratch, using the supplied editors. The generated output (through the converters) is SCORM compliant package. The package contains all the resources of the content,

along with the rules for sequencing and navigation amongst these resources, which are defined in the content tree.

SCORM engine is the backend of the SCORM previewer. It is the runtime, which parses the generated SCORM educational unit and interprets its content and educational sequencing and navigational rules.

2.3. e-Learning repository

The data structures layer is formed by the passive elements of the system. It is the data, which the editors use and produce.

Pedagogical templates are intended to ease the creation of educational unit's structure. The templates define the initial layout of the educational unit (content tree). They contain predefined pedagogical goals and a complete set of SCORM sequencing and navigational rules for satisfying these goals. The templates are used only to define the initial layout of the content tree when the educational unit is created and they are not exported as part of the resulting SCORM package. The author is free to alter the sequencing and navigation rules, or not to use templates at all. A set of templates is being developed to be included in Selbo 2 distribution, but we foresee some possibilities for templates to be additionally added to the system, for example to be downloaded from a learning management system portal. The templates, along with the Resource organizer, maintain the pedagogical model in the system, because they define the rules that the learner has to follow, in order to achieve the desired pedagogical goal.

User profiles is a collective term describing the sets of data that contain the preferences of both content creators and students. From the content creator's perspective, the user profile contains its personal preferences in working with the environment. Such preferences may include the position of editors on screen, a favorite editor for a specific resource, the most recently used resources and ontologies, and the main discipline domain. The agents may use this information in order to optimize the system to suit better the teacher's habits of work.

From the student's perspective, the user profiles are a set of data describing the intended users of the generated e-learning materials. This data may reflect students' background in the subject, level of interest, learning intensity, available time, etc. The agents can use this information to assist and advise authors during lesson creation process.

User profiles, along with component that interpret them, maintain the user model in the system.

Domain ontologies are specially created ontologies [12], containing definitions, terms, conceptions and examples, that can be used easily by content developers. Ontology is created specific to a given domain and contains meta level formal description of semantic connections between its elements. This eases the creation of intelligent agents, which can help the developer by performing more meaningful searches. The prototype ontology contains only textual information in the form of definitions from the domain of Software engineering. These definitions can be copied and pasted in textual content editor. We foresee ontologies that contain not only textual, but binary data like images or specialized formats, which can be added directly as resources in the content tree. Ontology with predefined resources will be included in Selbo 2, but others may be added subsequently, e.g.

downloaded from learning management system portal. The predefined resources, as they are domain specific in their nature, maintain the domain model.

Predefined resources contain ready-made materials, which content developers can use in their educational units. These resources may be available locally, or accessed over network. They can have different granularity: from simple definitions to whole lecture courses, and different structure: from plain text files to ontologies. Examples of predefined resources include HTML pages with explanation of terms, image files containing visual description of a process, sound files with pronunciation of words in foreign language. The resources are multimedia content, which can be added to the electronic lesson with no (or little) need of editing.

SCORM package is the output of the system. It is the electronic educational material that is to be presented to the students. It is generated by the transformers, which convert it from internal representation to selected output format. Currently Selbo 2 supports SCORM 2004 compatibility.

3. Selbo 2 usage scenario

The environment supports different usage scenarios. In most of them the environment is used in standalone mode. Some, however, foresee collaboration between the environment and the learning management system (LMS) that will make use of the materials developed. For sake of simplicity, we assume that the LMS is located on a network-accessible server. We will introduce some of the scenarios briefly in this section.

3.1. Creation of a new educational unit

During the first steps of creation of a new education material, the author uses wizard(s), which initialize the content parameters in a dialog mode. One of the main parameters is the template, which is used. Pedagogical goals and their corresponding SCORM rules are assigned to the content according to this template. The templates also define the logical structure of the unit by setting the initial structure of the content tree, and optionally, enforce restrictions on its modification. Up to date list of templates can be made available on the LMS server. Another parameter of the educational unit is its domain (field of study). It defines the list of ontologies, which the author may wish to use. The ontologies are created specially for Selbo 2, and efforts are made to incorporate already existing ontologies. Elements in the ontology are terms, definitions, and concepts from a given domain. We allow description, examples, links to other elements and external resources to be added to every element in the ontology. The content developers can copy information from these elements directly into their content. The ontologies can be deployed on the LMS server. If the author wishes to use particular ontology, which is not currently available in the environment, the system can automatically download it and make it immediately available for use.

3.2. Editing electronic content

After the wizard finishes its work, the educational unit has a fully initialized content tree. From this moment, the developer can add new branches and leaves to this tree. The branches define a logical (sub) structure of the content. Every branch represents one level of organization; there is no limit in their nesting depth. Leaves represent

actual electronic resources, which will be created/edited by the author and presented to the learner. SCORM rules are defined for these tree components. Because they are complex, the content developer has no means to manually edit them. The developer can assign simple pedagogical goals (initially inherited by the template), and content tree's agent takes care of their correct transformation into SCORM rules. SCORM does not enforce restrictions on the format of the resources used in the educational unit. Selbo 2 supports some of the most widespread multimedia formats as electronic resources: HTML pages, text files, some graphical formats, as well as a few specialized formats. These resources are manipulated using intelligent editors. Different editors can be easily integrated in the development environment, using a plug-in system.

3.3. Aiding content developers

The user manipulates the content of the resources using intelligent editors. While the user is using the editor, its agent operates in a background mode and communicates with other components and their respective agents. In addition, the environment contains personalized assistants, aiding the developer in content creation. They are also developed as intelligent agents. During content development different agents can communicate with each other and with their respective components. One of the scenarios describes the collaboration between the HTML editor and ontology visualizer: while the developer is typing terms in the HTML editor, the Ontology visualizer is searching for them in the ontology. This collaboration effectively makes Selbo 2 a multi-agent system [13]. The usage of agent-oriented approach provides the necessary flexibility and adaptability of the environment.

3.4. Content export

When the educational unit is finished, the developer can export it in SCORM package. This task is performed by another wizard, which configures the export parameters and allows the package to be deployed into various repositories, including (but not limited to) LMS server.

4. Implementation

Error! Reference source not found. shows the user interface of Selbo 2. The content tree editor is at the left side of the image, the ontology viewer is at the right side, and the HTML editor is centered, manipulating the Advanced Distributed Learning main page [14]. At the top of the picture is the mascot of the environment – the elephant Agent Selbo.

Kafenio [19] is used as a HTML editor library for HTML resources. Java Plugin Framework (JPF) [20] is used as a plug-in manager. The implementing of the intelligent components as a component of the graphical interface of the environment, coupled with a specialized agent, decreases the complexity of the application development [21, 22]. We believe that this approach is more productive than other possible solutions, i.e. creating a sole "super-agent" that collaborates with all the visual components or development of the agents themselves as visual editors [23]. It also allows us to develop Selbo 2 as a multi-agent system, featuring

several layers of agents, ranging from agents, handling specific editors to agents, governing pedagogical goals over the whole environment.

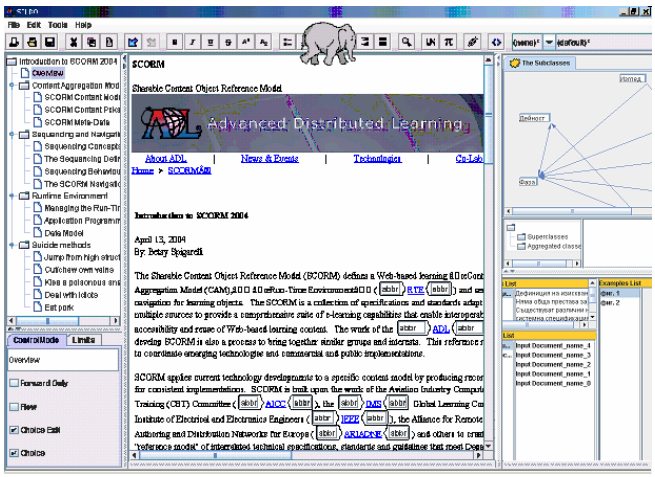

Fig. 1. Selbo 2 user interface

5. Conclusion

The future development of Selbo 2 will involve the creating of new intelligent content editors, enhancement of the current ontologies and development of new ones. Special attention will be paid to the templates, the possibilities to assign pedagogical rules to the content and their translation to SCORM rules. The behaviour of the environment during content creation will also undergo revision.

The possibilities for adaptability of the system raise particular research interests. One aspect is adapting of the environment to different domains. Another aspect is adapting the environment itself to its user. This can be achieved by using personalized agents in the system.

Acknowledgements. The authors wish to acknowledge the support of the Bulgarian Ministry of Education and Science for Research Projects Ref. No У-ТН-402/2006 and DO02-149/2008.

R e f e r e n c e s

1. S t o j a n o v, S., I. G a n c h e v, I. P o p c h e v, M. O'D r o m a, R. V e n k o v. DeLC – Distributed e-Learning Center. – In: Proc. of the 1st Balkan Conference in Informatics BCI'2003, 21-23 November, 2003. Thessaloniki, Greece. ISBN 960-287-045-1, 327-336.
2. S t o y a n o v, S., I. P o p c h e v, O. R a c h n e v a, A. R a c h n e v. DeLC – Technological Environment Supporting the Transition from CBT to e-Learning. – In: International Scientific Conference "Informatics in the Scientific Knowledge", 28-30 June, 2006, Varna Free University, 113-127.
3. S t o y a n o v, S., I. P o p c h e v. Evolutionary Development of an Infrastructure Supporting the Transition from CBT to e-Learning. – Cyberbetics and Information Technologies, Vol. **2**, 2006, No 2, 101-114.

4. G a n c h e v, I., S. S t o j a n o v, M. O'D r o m a, D. M e e r e. An InfoStation-Based University Campus System Supporting Intelligent Mobile Services. – Journal of Computers (JCP, ISSN1796-203X), Vol. **2**, No 3, May 2008, Academy Publisher, 21-33.
5. Shareable Content Object Reference Model – SCORM.
http://www.adlnet.gov/Scorm/index.aspx (to date)
6. RELOAD Content Package and Metadata editor.
http://www.reload.ac.uk/ (to date)
7. DeltaLearn Advanced SCORM Editor.
http://www.deltalearn.com/scorm_editor.php (to date)
8. University of Nottingham Xerte Editor.
http://www.nottingham.ac.uk/~cczjrt/Editor/xerte.htm (to date)
9. S t o y a n o v, S., I. G a n c h e v, I. P o p c h e v, M. O'D r o m a. From CBT to e-Learning. – Journal Information Technologies and Control, 2005, No 4, Year III, 2-10.
10. M i t e v, D. Design and implementation of SCORM Transformer. Bachelor Thesis. University of Plovdiv, 2005.
11. M i t e v, D. Design and implementation of e-Learning development environment SELBO. M.Sc. Thesis. University of Plovdiv, 2006.
12. B r a s e, J., W. N e j d l. Ontologies and Metadata for e-Learning. – In: Handbook on Ontologies. S. Staab, R. Studer (Eds.). Springer-Verlag, 2004, 555-574.
13. G r e e n w o o d, D., F. B e l l i f e m i n e, G. C a i r e. Developing Multi Agent Systems with JADE, April 2007.
14. Advanced Distributed Learning (ADL) initiative.
http://www.adlnet.gov (to date)
15. Java Agent Development platform – JADE, Telecom Italia Lab.
http://jade.tilab.com/ (to date)
16. C h m i e l, K., M. G a w i n e c k i, P. K a c z m a r e k, M. S z y m c z a k, M. P a p r z y c k i. Efficiency of JADE agent platform. – Scientific Programming, Vol. **13**, April 2005, No 2, 159-172.
17. B i g u s, J. P. Constructing Intelligent Agents with Java A Programmers Guide to Smarter Applications. John Wiley & Sons Inc., 1997, ISBN 0471191353.
18. Protégé ontology editor. Stanford University.
http://protege.stanford.edu/ (to date)
19. Kafenio WYSWYG HTML Editor component.
http://editor.kafenio.org/ (to date)
20. Java Plugin Framework (JPF).
http://jpf.sourceforge.net/ (to date)
21. M i t e v, D., I. M i n o v. Intelligent Components. – In: Jubilee International Conference "Science, Education and Time as Our Concern", 30.11-01.12.2007, Smolyan, Bulgaria.
22. B r i o t, J. P. Agents and Components: Can They be Combined? – Workshop on Software Engineering for Agent-oriented Systems, SEAS 3.10.2005, Uberlância, Brazil.
23. G r i s s, M., G. P o u r. Accelerating Development with Agent Components. – In: IEEE Computer, Vol. **34**, Issue 5, May 2001, 37-43.