High Level Design of Web-Based Environments for Distance Education[1]

Miguel Rodríguez-Artacho, Mª Felisa Verdejo
Departamento de Ingeniería Eléctrica, Electrónica y de control

http://www.lsi.uned.es

Universidad Nacional de Educación a Distancia

Key words: Instructional content design, Web-based learning environments, Distance Learning

Abstract:

This paper presents an approach to Web-Based learning environment authoring based in a high level description and in the use of reusable instructional components. These components have been previously categorised in a set of knowledge domains according to its instructional, didactic and pedagogical properties. Learning environment design is carried out using a description written in an SGML-derived language called PALO. PALO allows describing a variety of instructional scenarios that can be instantiated with a certain content matter using references to the domain model. This description is then turned into a Web-Based scenario by means of a compilation process.

[1] M. Ortega and J. Bravo (eds.), Computers and Education in the 21st century, 79-81.

1. INTRODUCTION

We can observe nowadays a wide development of computer supported learning environments. The growth of telecommunication technologies and an easier and more extended access to Internet, are turning the web into the new mass media. It is a matter of fact that this growth has brought an increasing demand for distance learning, most of the times using Web environments.

Current state of the art has to face problems that affect both users of learning environments and designers of learning contents. On one hand, learning scenarios do not completely fulfil the users' needs and, on the other hand, systems are complex to build and maintain because they are most of the times built from scratch or reduced to an experimental usage.

Other problems are related to student budget to afford high tech environments offered by Universities. Not all students have the same bandwidth to access the *Virtual Campus* of their Universities. From this point of view, it is not realistic to offer highly technological environments, because most of the students have only a modem connection to Internet that is slow and expensive.

At the same time, facing the problem of the designer of learning content, current approaches do not have an explicit learning content representation, and content is not separate from system structure. This causes a lack of flexibility and reusability of learning components that does not help to provide an instructional content design process based on incremental authoring and knowledge reusability.

2. AN APPROACH TO WEB-BASED LEARNING ENVIRONMENTS DESIGN

Our goal is to improve the design process of a learning environment, but also to create useful environments for a massive use, considering the increasing number of students in our University.

According to this it would be desirable that our students could be offered:

- ✓ To use the remote learning environment anytime and anywhere
- ✓ To perform interactive tasks (mainly asynchronously)
- ✓ To use the material both on-line and off-line without loss of performance
- ✓ To access the learning scenario using only a web browser

Also, content designers, like teachers and tutors, should be able to:

- ✓ Build and configure a learning scenario using a straightforward process
- ✓ Modelling and managing a knowledge domain separately from the learning environment structure
- ✓ Have interactive elements to carry out student assessment

This kind of scenarios could satisfy most of the remote learning activities in Distance Learning and is useful to carry out the activities proposed in our University, actually developed over a hardcopy of the didactic material. We propose to use the web to carry out these activities using a web environment, but facilitating the authoring process to the teacher. In this sense, what makes these environments different from other learning scenarios is that the authoring process is carried out at a higher level of abstraction by mean of:

- A separate description of the content matter
- Explicit mechanisms to select and refer knowledge components according to its didactic and instructional properties
- An explicit description of the learning environment that can be turned into a variety of different technologies (i.e. for remote and local usage)

In summary, our intentions are that the teacher could in first place organise and model a given content matter, and then describe, using this knowledge model, interactive working environments where students can carry out the work plan defined by the teacher in the description. Flexibility is provided by using the same knowledge components to describe different scenarios thus, avoiding building them from scratch, and describing them at a higher abstraction level using the mechanisms provided by the description language.

Following sections describe knowledge modelling, environment descriptions and a brief summary of technical and implementation aspects.

2.1 Instructional Objects and Instructional Knowledge

Most commonly known tools like WebCT , Toolbook, etc. are multimedia environments in which building blocks have no pedagogical information. Content and structure are not separated, and instructional content is embedded in web HTML pages and is not reusable. Even if we use the same content in another part of the environment, elements of the content are in physical formats (gif, html, mpeg, ...) and are not treated as elements at an instructional or pedagogical level of abstraction. (Murray, 1996) (Mizoguchi et al., 1997)

As we have pointed before, separate content and structure need an instructional knowledge description in knowledge components that could be referred by using instructional properties. This description is called a *conceptualisation,* that identifies the elements of a description of a content matter and creates a conceptual map of the learning content using elements (*example, concept, exercise, ...)* and instructional or didactical relationships (i.e. concept *is prerequisite of* concept, example *explains* concept, etc.)

Needless to say, structuring a content matter into learning components is not a new idea. There are some interesting developments that use an instructional components classification to provide reusability like in (Forte et al., 1997) and reusability of instructional Web resources using metadata and RDF format as in (Murray, 1998).

However, two main objectives are accomplished using the proposed learning component classification. On one hand there is an explicit description of the content matter using a domain of elements and instructional relationships. This model allows carrying out a conceptual learning supported by the conceptual relationships of the model (Adriassen and Sandberg, 1999). On the other hand an independent knowledge representation is necessary to reuse knowledge components at a certain level of granularity. These components are categorised as suggested in (Breuker et al. 1999) and (Mizoguchi et al. 1997) who pointed out the need of instructional ontologies to structure and scaffold different approaches of instructional knowledge representation.

2.2 Describing a Learning Environment

As pointed above, we have created a SGML based language to describe a learning scenario. The language provides both flexibility and expressiveness to refer to different content; for example, elements of the domain model are selected using expressions at a pedagogical level such as "i*nclude an example to illustrate concept C",* instead of using a low media level description such as an HTML page, etc.

Using properties of SGML, such as its capacity to define a variety of DTDs can lead to the definition of several scenarios, each one to be described using its own DTD. Different scenarios can reuse the same content references and can perform different instructional activities over the same content matter.

For the authoring process, the teacher has, on one hand, a domain model with a set of classified instructional elements and on the other hand a library of *instructional templates* described with a DTD. Once a given DTD is selected, instructional content can be added by filling the structure with the references to the knowledge domain either by a direct reference "*show concept C*", or using a didactic relationship like "*show prerequisites of concept C",* or a relationship filtered with an attribute like in "*show easy examples to illustrate concept C"*.

3. THE STEED SYSTEM

According to the ideas described in the previous section, we have developed a set of tools to describe and create instructional environments using a high level description.

The STEED[2] System has the following features:

- It provides an external knowledge representation of the content matter separate from the structure of the learning environment. This representation scaffolds instructional knowledge in a set of domains
- It describes learning scenarios using an explicit description in a high level language, using references to reusable instructional components

[2] STEED stands for Sistema Telemático para Enseñanza en Educación a Distancia `http://sensei.ieec.uned.es/~steed`

- It allows including structure and management information into the description.

The authoring process firstly allow to define models in which content matter is structured, and secondly, using the SGML language, an *instructional template* can be chosen and instantiated with domain content matter. Then, a PALO[3] compiler can build up a fully functional learning scenario using this description.

3.1 Content Matter Modelling

A content matter description can be represented using conceptual models based in an entity-relationship formalism. This kind of representation provides a trade-off between a robust representation and an effective implementation using a relational database.

Knowledge is structured in a set of generic levels using a taxonomy of instructional and conceptual components and a description of its pedagogical relations. Each level describes a particular aspect of the domain based in relationships or in attributes of the learning components.

3.1.1 Knowledge model of a programming subject: An example

A content matter description can be organised using different perspectives, grouping in single domains the entities and relationships that belong to a certain knowledge category. We propose as an example a Course in program verification that can be described using 3 domains:

- A structural approach to describe the taxonomy of entities using relations like *part of* and *type of*
- An instructional approach with a categorisation of instructional components such as *example*, *problem*, *exercise*, *solution* and relationships that link this elements like *illustrate, explains,* etc.
- Finally a pedagogical level combines some properties that could be interesting in order to decide which element to choose from a pedagogical point of view. Examples of this properties are *degree of difficulty* and relationships like *prerequisite*

[3] http://sensei.lsi.uned.es/palo

This generic model can be instantiated with several content matters, specially with those related to a theoretic and practical scientific domain.

3.1.2 Describing Generic Knowledge Models

Using an explicit description of a generic knowledge requires a higher level representation. The model described in the previous section has a generic structure with 3 domain models that is explicitly represented in a *metamodel*, that is, a model that describes a model.

Thanks to this *metamodel*, a tool can manage a set of models according to the knowledge structure explicitly defined in the *metamodel*. This meta level is useful to provide certain interesting operations over the models. In summary, there are two abstraction levels while describing instructional knowledge:

- A metamodel that describe the structure of a model
- A model that has the structure described in its metamodel and that has the appropriate content matter.

Both levels use the same formalism to describe themselves, the entity-relationship model. These abstraction levels provide a way to carry out interesting operations on a model. If a knowledge domain describe a certain content and a new relationship is needed, it could be added just by modifying the metamodel and re-instantiating the model according to the modification.

3.2 Describing *instructional templates*: The PALO language

We propose a way to describe a learning scenario, including content, structure and tasks model, using a mark-up language based in SGML.

SGML interest is based in its capacity to create a set of Document Type Definitions (DTDs) each one describing a certain type of document with its structure and components. If a document claims to be generated from a DTD, a program called *parser* can recognise the document with its content and elements as described in the DTD.

According to this, we have developed a language to describe learning scenarios called PALO. PALO is in fact a library of instructional templates (DTDs) that describe different learning scenarios. PALO templates can be selected, according to the kind of learning scenario that better fits the instructional needs. These templates describe different learning environments to carry out different instructional activities.

Documents written in PALO have a given structure based in the template's DTD, but also describe the content matter the instructional activity will be carried out with. This is made using a certain tags as in any SGML language.

PALO tags are classified in two different types:

- Those that describe document structure. This will become the structure of the learning scenario
- Those that refer to knowledge components both implicitly (references to the components of the knowledge model) and explicitly (embedded in the document)

The first type of tags is document dependent. Those who describe a practical activity scenario are different from other describing a didactic guide scenario. Different templates have different tags of the first type. Tags of the second type are shared by all the documents. These tags allow inserting a knowledge element from the domain models in any part of the environment. References to these elements have been designed to provide a higher level of abstraction in the authoring process, using the relations between entities described in the models, as shown in the previous section.

Other tags or attributes are related with management aspects such as directories, databases, user tracking of content elements and assessment of tasks.

In order to create the learning scenarios, a PALO compiler has been developed. This tool turns a PALO description into a Web-based interactive environment. (Rodríguez-Artacho et al. 1999)

4. EXPERIENCES USING STEED

During the second semester of course 98/99 STEED tools where evaluated within a programming subject of *algorithms design and verification* corresponding to a first course in Computer Science at UNED[4] University.

This experience served to evaluate the authoring process of the learning environment with PALO and the use of the system with a group of students. Authoring process has consisted in the modelling of the relevant knowledge plus the PALO description of the environments. The final Web environment was used to carry out a practical programming project with the help of a didactic guide to solve conceptual errors.

4.1 Authoring process

Firstly, the authoring process consists in the creation of domain knowledge. This process has the following steps:

- Choice of a content matter and creation of the cognitive structure to represent it (metamodel). Describe in the metamodel the selected conceptualisation.
- Instantiate the metamodel into an empty entity-relationship model.
- Fill the components of the model with the appropriate knowledge components: *concepts, problems, relations,...* according to the content matter.

Using this description, the teacher can now use PALO language to describe a learning scenario using references to this knowledge model.

During the development of these activities, the most time consuming task was the creation of the models and metamodels of the domain content. This took ¾ of the authoring process.

[4] http://www.uned.es

Tasks	Time
Conceptualisation	40 Hours (18%)
Creation	10 Hours (4%)
Instantiation	150 Hours (55%)
PALO description	60 Hours (23%)

Table 1

Table 1 describe the authoring process steps and the total time involved in the development of each of them. Once the knowledge structure is created, the most costly task is the instantiation of the metamodel with the content components. These tasks are carried out using a web-based tool and a relational database.

For this experience we developed three PALO instructional templates:

- ✓ A didactic guide of the subject
- ✓ A self-evaluation tool
- ✓ A small programming project

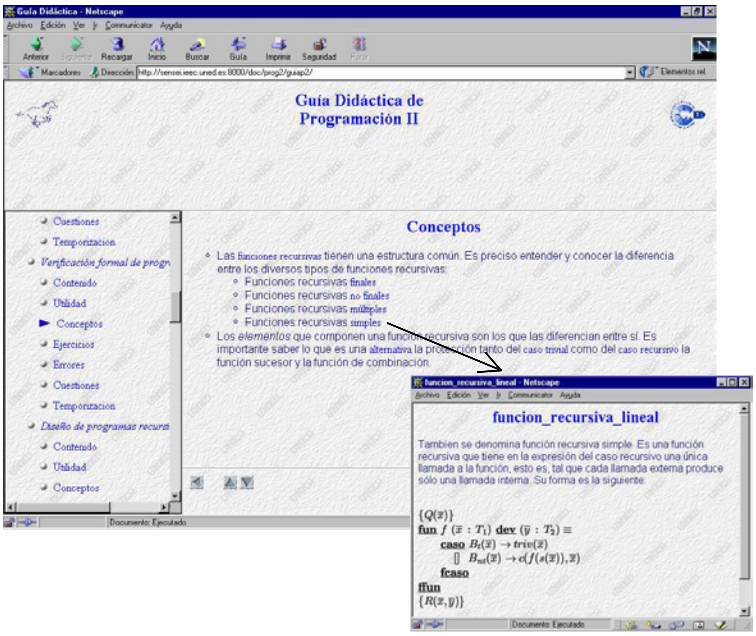

Figure 1

Figure 1 shows the final result in the creation of the didactic guide environment after the compilation process of the PALO description. The editor used to create the PALO documents was XEmacs, that can edit XML and SGML documents thanks to its ability of parsing DTDs.

4.2 Using the learning material

In June 1999 a group of students in the UNED Associated Centre at Madrid, were invited to participate in this experience. The proposed task was a small programming project to be carried out using the environment.

Two groups of 60 students were invited to join the experience. They were offered the possibility of doing it the traditional way (hardcopy) or using the internet environment. Results are shown in Table 2.

Action	Students
Offering	120
Are interested	65
Access the system	35
Work over the material (any time)	25
Finish completely	11

Table 2

Approximately half of them showed some interest in using the system and asked for a login and password. As shown, the final average of the system use is 20% of the initial population. Students were offered the possibility to quit at anytime and go back to the traditional way (in hardcopy). Thus, taking into account that the use of the system was not mandatory, the final result is acceptable.

At the end of the experience, students were posed a questionnaire in order to get their opinion about the reasons to quit. Results showed that the main reason was the cost of the Internet access. Those who continued were mainly motivated because they used free Internet access from their offices or at home.

5. FUTURE WORK AND CONCLUSIONS

Now we are working in the development of a new PALO compiler that allow creating working scenarios with the possibility of using them *off line*. In this local environment, elements have a Java applet that can save the student response or the user tracking in the local disk. To overcome the rigid security model of the Java VM, applets have been digitally signed. Digital certificates can be retrieved from the University web pages.

Our perspectives in the development of the STEED project also include instructional knowledge modelling. In this aspect, our interest fits in the research of standardisation organisms and the promotion of the use of metadata schemas in order to improve reusability of instructional components

6. ACKNOWLEDGEMENTS

STEED Project has been supported by the Comisión Interministerial de Ciencia[5] y Tecnología (CICYT TEL 97-038c0202) and Comunidad de Madrid[6]. Authors wish to thank the participation of J.I.Mayorga and M.Y. Calero in the development of this project.

7. BIBLIOGRAPHY

Andriessen, J and Sandberg, J (1999). *Where is education and how about AI?* International Journal of Artificial Intelligence in Education 10

Breuker, J. Muntjewerff, A. and Bredewej, B. (1999) . *Ontological modelling for designing educational systems*. In Proceedings of the AI-ED 99 Workshop on Ontologies for Educational Systems

Forte, E. Maria, H., Wentland, F. and Duval, E. (1997) *The ARIADNE project: Knowledge pools for computer-based and telematics-supported clasical, open and distance education (I/II)*. European Journal of Engineering Education, 22(1).

Mizoguchi, R. Ikeda, M. and Sinitsa, K. (1997). *Roles of shared ontology in AI-ED research*. In de Boulay, B and Mizogiuchi, R. editors, Artificial Intelligence in Education AI-ED 97, pages 537-544, Kobe, japan. IOS Press

[5] www.cicyt.es
[6] www.comadrid.es

Murray, T. (1996) *From story board to knowledge bases: The first paradigm shift in making CAI "intelligent"*. In Proceedings of the ED-MEDIA 96 Conference, pages 509-514, Boston, MA

Murray, T. (1999). *A model for distributed curriculum in the WWW.* Journal of Interactive Media in Education (5)

Rehak, D. (1999). *Structure guidelines for a distance educational program*. Carnegie-Mellon University Technical Report

Rodríguez-Artacho, M and Verdejo, M.F. and Mayorga, J. and M.Y. Calero (1999) *Using a High-Level language to describe and create web-based learning scenarios.* In Proceedings of IEEE Frontiers In Education Conference FIE '99 San Juan de Puerto Rico, Nov 1999 (http://sensei.ieec.uned.es/~steed)

Verdejo, M.F., Rodríguez-Artacho, M. Mayorga, J.I., and Calero, M.Y. (1999) *Creating web-based scenarios to support distance learners*. In Proceedings of the Working Group 3.3 and 3.6 IFIP. Chapman & Hall. (http://sensei.ieec.uned.es/~steed)

8. AUTHORS INFORMATION

Miguel Rodríguez Artacho (miguel@ieec.uned.es)
Mª. Felisa Verdejo (felisa@ieec.uned.es)

Universidad Nacional de Educación a Distancia, Escuela Técnica Superior de Ingenieros Industriales, Departamento de Ingeniería Eléctrica, Electrónica y Control; Ciudad Universitaria, s/n ; 28040 Madrid.