Communication & Cognition
Vol. 41, Nr. 1 & 2 (2008); pp. 53 - 66

STRUCTURING DIDACTIC MATERIALS ON THE WEB (STRUCT)

Marco Alfano, Nicola Cuscino, Biagio Lenzitti

Abstract

There is a universal agreement that the structuring of didactic materials helps the student in his/her learning process when dealing with such materials. It is then important for the teacher or a student to choose among different structures so to use the one that best conveys the desired knowledge. This need is nowadays a priority also in the e-learning world since several e-learning courses exist on the web and many more are created every day.

Unfortunately, most of the existing e-learning platforms offer just a single way to organize the course contents (book structure). Whoever is interested in organizing the course contents in a different way (e.g, with a concept map) must use specific tools that are external to the e-learning environment and are often difficult to use.

This work presents the details of the tool we have created to overcome the problems and limitations presented above. We have implemented an on-line application, called STRUCT, that allows teachers and students to easily publish didactic materials on the web and to organize them by choosing the most appropriate structure among different alternatives. STRUCT has been built in a way to be easily integrated inside the most popular e-learning platforms such as Moodle or ATutor.

Categories and Subject Descriptors

J [Computer Applications]

General Terms

Design, Experimentation.

Keywords

On-line Education, Didactic Structures, On-line Materials, Concept Maps, Ontology.

1. INTRODUCTION

Scientifically based research demonstrates that a research base exists to support the use of graphic organizers of didactic materials for improving student learning and performance across grade levels, with diverse students, and in a broad range of content areas (Ausubel, 1968, Gallick-Jackson, 1997, A.r., (2007_4).

In particular, it has been experimented that the use of graphic organizers is effective in improving students' reading comprehension and achievements across content areas and grade levels. Achievement benefits have also been seen with students with learning disabilities.

Moreover, the process of developing and using a graphic organizer enhances skills such as developing and organizing ideas, seeing relationships, and categorizing concepts.

Finally, the use of graphic organizers aids students in retention and recall of information and supports implementation of cognitive learning theories such as *dual coding theory* (Paivio, 2006), *schema theory* (Brewer, William and Nakamura, 1984), and *cognitive load theory* (Sweller, 1994).

It is then important for a teacher who organizes a course or a student who wants to create his/her own didactic path, to understand the importance of materials structuring and to choose among different structures so to use the one that best conveys the desired knowledge. This need is nowadays a priority also in the e-learning world since several e-learning courses already exist on the web and many more are created every day.

The most common ways to organize didactic information inside on-line educational environments are the following:

- *Book*, which presents an index and is divided into chapters, sections, sub-sections and so on.
- *Concept Map* (Cañas et al., 2005)**,** is a diagram showing the relationships among concepts. Concepts are connected with labelled arrows.

- *Lattice*, which is similar to a concept map but with arrows without labels. A special lattice is the *mind map*, that is used to represent words, ideas , tasks or other items linked to and arranged radially around a central keyword or idea.
- *Tree*, which is a way of representing the hierarchical nature of a structure in a graphical form. It is a "tree structure" because the graph looks like an upside down tree. Information is usally placed in the most synthetic form, at the top, and the most analyical form, at the bottom. A special tree is an *ontology,* an explicit specification of a conceptualization.

As said above, the use of such structures is necessary inside an on-line education environment. In fact, those structures, especially the diagrams, help in the process of focusing and understanding information and become a fundamental tool in knowledge construction.

Unfortunately, only “book” structures are usually found in e-learning platforms whereas the “map” structures are considered as complementary structures and are usually realized through off-line tools.

We have searched the internet for tools that create the structures listed above. We have analyzed and tested some of them. The main characteristics of those tools together with their advantages and disadvantages are shown in Table 1.

Product	**Supported Structures**	**Pros**	**Cons**
CMAP (A.r., 2007_2)	Concept map	Free Easy to use Resource mngt. Node customizat. Synchr. collabor. Asynchr. collab. Multilingual	Off-line One structure Difficult integ.
INSPIRATION (A.r.,2007_3)	Concept map Tree graph Lattice graph Book	Easy to use Resource mngt. Node customiz.	Off-line Not free English only
MINDOMO (A.r., 2007_7)	Mind map	Free On-line Node customizat. Asynchr. collab.	One structure No res. mngt. English only

MIND MEISTER (A.r., 2007_6)	Mind map	Free On-line Synchr. collab. Asynchr. collab.	One structure No res. mngt. English only
KAYUDA (A.r., 2007_4)	Concept map Lattice graph	Free On-line Asynchr. collab.	One structure No res. mngt. English only

Table 1. A comparison of tools for structure creation.

The analysis of the tools presented in the table above together with the exam of the main e-learning platforms such as Moodle (A.r., 2007_8) and A-Tutor (A.r., 2007_1) has shown the lack of a complete tool that on one hand allows a teacher to easily publish on-line material using a structure of choice and, on the other hand, can be easily integrated in an e-learning platform exploiting its features. This lack has led us to the design and development of the STRUCT solution which will be presented in the next chapters.

The paper is organized as follows. Chapter 2 describes the STRUCT principles and architecture. Chapter 3 provides the implementation details of STRUCT which allows to create all the different structures listed above and publish didactic materials on the web. Chapter 4 provides some details on the use of STRUCT and finally, Chapter 5, reports some conclusions and future work.

2. Struct: a tool for structuring didactic materials on the web

The goal of the STRUCT solution is to realize an on-line tool which allows an user (either a teacher or a student) to easily create the most appropriate structure for his/her didactic materials.

STRUCT presents the following main features:

- Creation of different structures (book, concept map, lattice or tree);
- Linkage of structure nodes to any digital resource (documents, images, audio, video, …);
- publishing and sharing created structures;

- possibility of easy integration in an e-learning platform such Moodle or ATutor.

STRUCT has been thought as an autonomous tool and does not need any specific SW installed on the user PC beside, of course, a common web browser. It is mix of a Content Management System (CMS) and a graphical interface for structure creation.

STRUCT mainly focuses on the easiness of use and allows to publish and structure didactic material on the web also by whoever is not interested in the several features of e-learning platforms (which sometimes make them quite difficult to use without their deep understanding).

2.1 Basic Choices

To build STRUCT we have decided to use the same open-source components (under the GPL license) used by the main e-learning platforms, i.e., the Php and Javascript scripting languages, the MySQL database and the Apache web server. The choice of such components allows STRUCT to be independent from any operating system and allows an easier integration in the e-learning platforms developed with the same open-source software.

2.2 STRUCT Architecture

STRUCT is a three-tier web application (client, server, database) that works in the following way. A client asks for information to a server which in turns queries a database and builds the result based on the transmitted contents and the processing result of the scripting code of the web application. (Fig. 1).

It is possible to define the three levels as follows:

Presentation level – is the closest to the user, i.e., the application interface. Its task is to send the user requests to the intermediate level and present the results to the user. It is implemented in the web browser.

Intermediate level – is the meeting point between the user requests and the data in the underlying level. Its task is to process the data and make them representable from the presentation level. This is accomplished by a web server such as Apache or a web-server module such as Php, Asp, Perl or others.

Data level – is the data set that is analysed by the intermediate level. Data are stored in web-independent applications such as a database manager or an email management system.

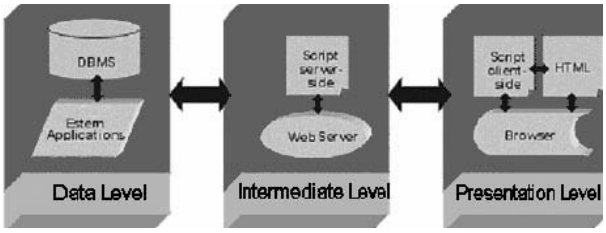

Figure 1. STRUCT three-tier architecture.

3. Struct development

STRUCT is made up of four modules, *user*, *structure*, *resources* and *database* (Fig.2). All the modules have been realized with the open-source SW indicated above.

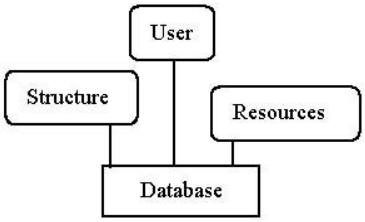

Figure 2. STRUCT modules.

The *User* module deals with the user management and is included in the presentation level as well as in the intermediate level. The presentation level contains all html forms which are concerned with users registration and authentication, whereas the intermediate level contains all Php functions which enable to control and include users data in the database.

The *Structure* module deals with the creation of the various graphical structures (book, concept map, lattice and tree). Most of the module is included in the presentation level which contains all javascript functions and allows to create the structures. Every structure consists of nodes and lines. Every node is a html (div) element which can be moved (within a concept map or lattice) by

means of the *script.aculo.us* libraries (A.r., 2007[9]). Lines too are html (div) elements which can be created through *wz_jsgraphics* libraries (A.r., 2007[10]). The section of the module, which consists of php functions and enables to include and modify all data of every structure in the database, is integrated in the intermediate level.

The *Resources* module deals with the management of the digital resources (documents, links, pictures, video, audio, …) that users can associate to all nodes of a structure. The presentation level includes all html forms which enable to create and modify a resource, whereas the intermediate level contains all php functions which allow to manage files on the web server, as well as include and modify all data connected to the resources in the database.

The *Database* module contains all the information on users, structures and resources. The module is included in the data level and consists of a MySQL database called STRUCT which is made up of five tables:

- UTENTI (Users): which contains all the information on users;
- RISORSE (resources): which contains all the information on resources;
- STRUTTURE (structures): which contains all the information on structures;
- NODI (nodes): which contains all the information on the structure nodes;
- LINEE (lines): which contains all the information on the lines which connect the structure nodes.

4. Struct usage

As explained above, STRUCT has been implemented giving priority to easiness of use and flexibility. It can be accessed through any web browser by connecting to the address http://altair.math.unipa.it/struct/struct.php.

In order to fully use STRUCT, a user must authenticate (or register when he/she accesses STRUCT for the first time). It is also possible to access STRUCT as a guest and, in this case, it will only be possible to browse already existing structures.

Once entered STRUCT, a user is presented with a management page divided in two parts, *resources* and *structures* (Fig. 3). In the *resources* part, the user can insert, modify, visualize and delete his/her own resources. Moreover he/she can visualize (but not modify or delete) resources that are shared by other users. To create a new resource, a user specifies a name and the address of

the file in his/her PC. STRUCT will automatically upload the file on the server and make available for inclusion in a one or more structures. STRUCT supports files with any kind of extensions (.doc, .htm, .mp3, .jpg, ...) and also links to web resources. It should be noted that when a .htm file is selected, a zip file with all the objects referenced by the web page must be provided.

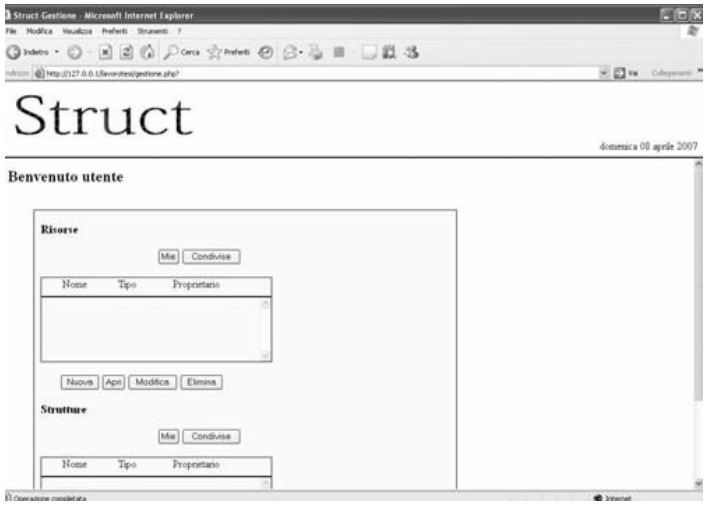

Figure 3. Resources/structures management.

The *structures* part allows to create, modify, visualize and delete own structures. Also in this case, it is possible to visualize structures shared by other users but it is not possible either to modify or delete them.

When a new structure is to be created, a specific mask will ask for the structure name, its type (concept map, lattice, book or tree) and whether it has to be shared with other users.

If a concept map is being created, the user will be presented the web page shown in Fig. 4. The page is divided in two parts. The left part is the working area and allows to draw the various elements of the concept map, i.e., nodes, lines and handles (to divide lines). The right part shows the property of the structure and of each selected element. It allows to change the property of each

element depending on its type. In particular, it allows to associate a resource to a node element or to change its aspect with a picture.

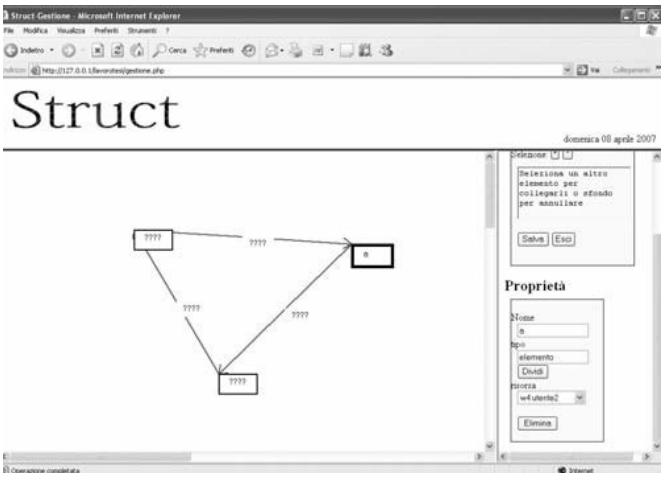

Figure 4. Concept map creation.

A node is drawn by just double-clicking on the chosen spot in the working area and a line is created by just selecting and clicking on the initial node and then selecting and clicking on the final node.

If a book structure is being created, the user will be presented the web page shown in Fig. 5. The page is divided in two parts similarly to the one of the concept map. In this case, the working area, on the left, will not be empty (as in the concept map) but will contain an element that represents the first element of the structure (it can be thought as the book title).

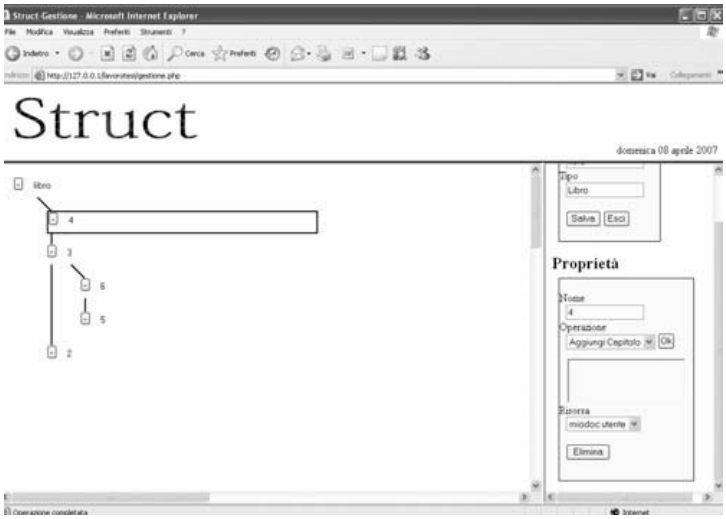

Figure 5. Book creation.

The first operation is to select this element, go in the property part on the right and select the desired operation. Four types of operations are possible:

- *add below*, allows to add an element under the selected element (so to create new chapters in the book);
- *add section*, allows to add a topic in the selected element (so to create new sections inside a chapter);
- *move below*, allows to move an element below another element;
- *exchange*, allows to exchange the positions of two selected elements.

As for the concept map, the property part, on the right, allows to associate a resource to an element.

It should be noted that the construction of a lattice structure is very similar to the one of the concept map and that of a tree is very similar to the construction of a book.

Once a structure has been completed and saved, it can be selected and opened by the user alone, if it remains private, or by anybody if it is being shared. By clicking on a node, the associated resource will open up (see Fig. 6).

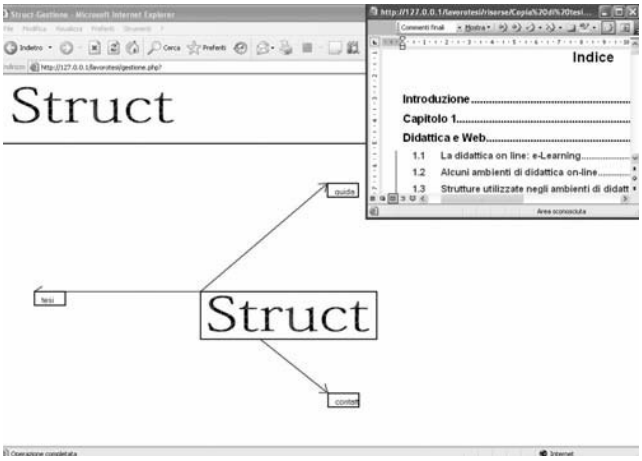

Figure 6. Opening a resource inside a lattice structure.

5. Conclusions and future work

This paper has presented STRUCT, an on-line tool which allows an user (either a teacher or a student) to easily create different types of structures for his/her didactic materials and publish them on-line.

STRUCT has already shown easiness of use and flexibility and will be used to build e-learning courses at the university and in a secondary school in Palermo.

An important feature that is being developed, is the possibility to automatically change structure after being created (e.g., changing a book in a concept map or viceversa). This will allow the teacher, for example, to organize his/her didactic materials in the most natural way to him/her and then to dynamically modify the structure so to choose the best way to present the materials to the students. Moreover, the export of the structures in html or jpeg formats is being implemented. This will allow, for example, an easy inclusion in

word processing or presentation applications. Finally, the integration of STRUCT in Moodle and ATutor e-learning platforms is being started. This should be quite straightforward because, as explained above, STRUCT has been developed using the same open software of such environments.

6. References

A.r. (2007_1). ATutor e-learning platform. URL: http://www.atutor.ca.

A.r. (2007_2) Cmap. URL: http://cmap.ihmc.us.

A.r. (2007_3). Inspiration. URL: http://www.inspiration.com.

A.r. (2007_4). Institute for the Advancement of Research in Education (IARE) at AEL. Graphic Organizers: A Review of Scientifically Based Research. July 2003. URL: http://www.inspiration.com/download/pdf/SBR_summary.pdf

A.r. (2007_5). Kayuda. URL: http://www.kayuda.com.

A.r. (2007_6). Mindmeister: URL: http://www.mindmeister.com.

A.r. (2007_7). Mindomo. URL: hhtp://www.mindomo.com.

A.r. (2007_8). Moodle e-learning platform. URL: http://moodle.org.

A.r. (2007_9). Script.aculo.us library, URL: http://script.aculo.us.

A.r., (2007_{10}). wz_jsgraphics library, URL http://www.walterzorn.de.

Ausubel, D. P., (1968). *Educational Psychology: A Cognitive View*. New York, Holt, Rinehart and Winston, 1968.

Brewer, William F., and Nakamura, (1984). "The Nature and Functions of Schemas". ed. Robert S. Wyer, Jr. and Thomas K. Srull. In Handbook of Social Cognition, Vol. 1, Hillsdale, NJ: Erlbaum, 1984.

Cañas, A.J., et al. (2005). Concept Maps: Integrating Knowledge and Information Visualization. In S.-O. Tergan, and T. Keller, Editors. *Knowledge and Information Visualization: Searching for Synergies*. 2005. Heidelberg / New York: Springer Lecture Notes in Computer Science.

Gallick-Jackson, S. A., (1997). *Improving Narrative Writing Skills, Composition Skills, and Related Attitudes Among Second Grade Students by Integrating Word Processing, Graphic Organizers, and Art Into a Process Approach to Writing*. Unpublished M.S. Practicum Project, Nova Southeastern University, 1997. (ERIC Document Reproduction Service # ED420064).

Paivio, A. (2006). *Mind and its evolution; A dual coding theoretical interpretation,* Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Sweller, J., (1994). "Cognitive Load Theory, learning difficulty, and instructional design". *Learning and Instruction* **4**: 295-312, 1994.

About the authors

Marco Alfano, Anghelos Centre on Communication Studies Via Pirandello 40 90144 Palermo, Italy Tel. +39091341791, marco.alfano@anghelos.org.

Nicola Cuscino, Computer Science University of Palermo Via Archirafi 34 90123 Palermo, Italy Tel. +390916040427, nicolacuscino@gmail.com.

Biagio Lenzitti, Computer Science, University of Palermo Via Archirafi 34 90123 Palermo, Italy Tel. +390916040427, lenzitti@math.unipa.it.