APPLICATIONS OF A CONCEPT MAPPING TOOL

Urko Rueda, Mikel Larrañaga, Ana Arruarte, Jon A. Elorriaga

University of the Basque Country (UPV/EHU)
649 P.K. , E-20080 Donostia

E-mail: {jibrumou, jiplaolm, arruarte, elorriaga}@si.ehu.es

Abstract. In this paper a general purpose tool for editing Concept Maps (CM-ED) is presented. This tool exhibits the following functionality: templates, views and facilities for multilingual concept maps. The tool has been used in different tasks of the computer aided teaching/learning. Concretely: domain representation, exercise design and student model visualization.

Category/Categoría: Full Paper/Artículo Completo

1   Introduction

In recent years it has become apparent to many educational researchers that representing knowledge in a visual format allows one to better recognize and/or understand incoming information. It is easier for the brain to make meaning when information is presented in visual formats; visual symbols are quickly and easily recognized. Since Novak (Novak, 1977) placed concept mapping on the educational agenda, it has become an increasingly popular advanced teaching and learning tool (Bruillard & Baron, 2000).

Over the last years, and due mainly to the innovation of visual design software (Dabbagh, 2001), a number of products have emerged to support computer-based concept mapping. This kind of software enables much easier production and modification of CMs (Jonassen et al., 1997). Even more, some authors (Chang et al., 2001) argue that constructing maps with a computer might be easier that constructing a map using paper and pencil (p&p). They carried out a study comparing the effectiveness of constructing CMs by p&p or using a software product. Moreover, maintaining and revising CMs developed using p&p requires considerable amounts of time and effort (Chiu et al., 2000). (Anderson et al., 1998) argue that the practical advantages of constructing CMs electronically are similar to those of using a word processing program to write. There is an ease of construction, an ease of revision, and the ability to customise maps in ways that are not possible with p&p.

In this paper a concept map editor (CM-ED) is presented together with its applications in different tasks of the computer aided teaching/learning area. Concretely: domain representation, exercise design and student model visualization.

2   CM-ED Tool

CM-ED (Concept Map EDitor) is a general purpose tool for editing CMs. The aim of the tool, implemented in Java, is to be useful in different contexts and uses of the educational agenda, concretely inside the computer-based teaching and learning area. On the one hand, it is a common concept mapping editor in which the user can draw nodes and relations generating different CMs. On the other hand, the tool can be adapted to specific purposes. The interface of the tool (see figure 1) is composed of a working area in which the CM is shown and modified. The operations are included in the menu bar, in tool bars and in contextual menus. Thus, the user can choose his preferred way of working.

The tool allows the user to work with **templates**, which specify the kind of nodes, relations and operations that are going to be used in a set of CMs. The tool shares the same treatment for designing a CM and a template. Thus, the user defines in the template concept map the graphical characteristics of the nodes and relationships. Regarding the operations, it is also possible to restrict the operations allowed in the CM that are constructed using a concrete template. The available operations are selected also in the template concept map.

CM-ED can be specialised by means of templates. Specializing the tool means that the final tool is adapted to a template and provides less functionality but a simpler interface. This is useful, for example, when the teacher plans a concrete task for the learners. When specializing the tool is restricted to the resources (types of nodes, relations and operations) specified in the template.

In order to offer a better adaptation to the user when s/he develops the CM, the interface offers the possibility of working with different **views** of the same CM. All the views share the same structure, i.e. the same

nodes and relationships. However each view can have its own way of representing the nodes and relationships: graphical characteristics, spatial distributions, etc.

In addition, CM-ED is a **multilingual** tool (Larrañaga et al., 2003) localizable not only at user interface level but also regarding the final CMs it generates.
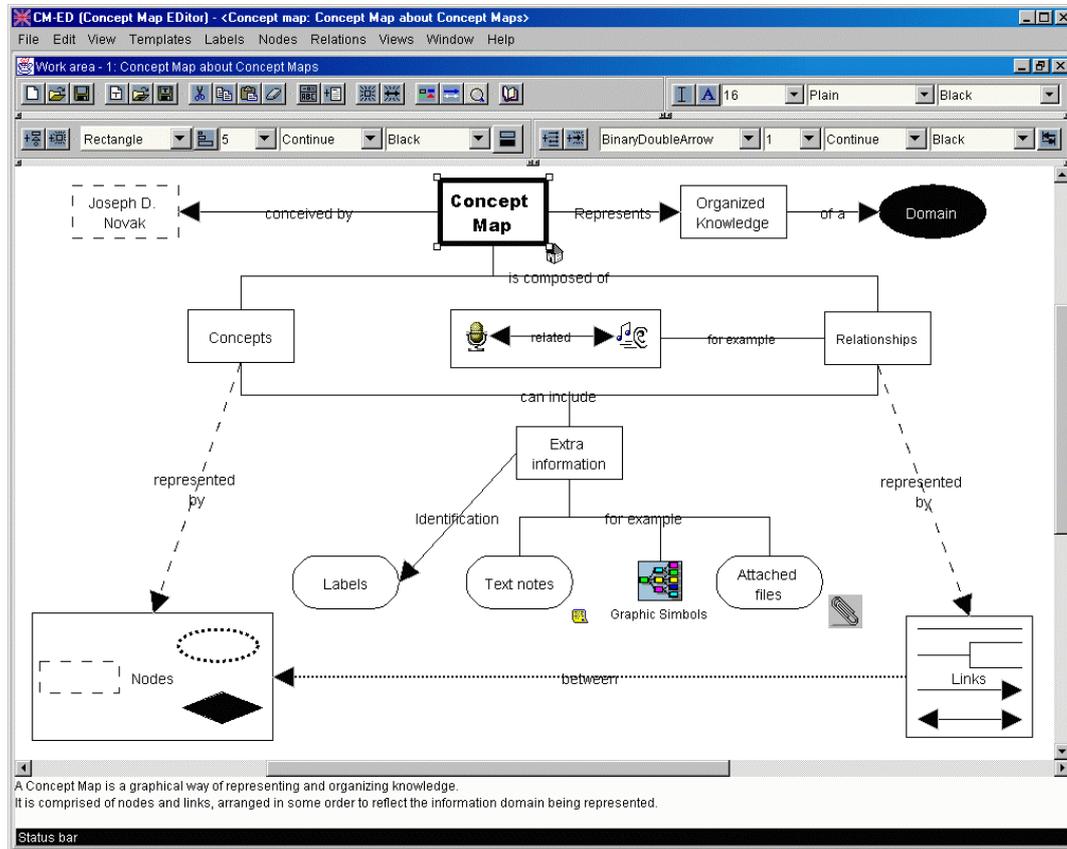

Figure 1. Interface of CM-ED.

3 Using CM-ED for acquiring the domain knowledge in IRIS

IRIS (Arruarte et al., 2003) is an authoring tool developed to help human instructors to build intelligent teaching-learning systems in a variety of domains. Four basic elements describe the domain of all tutors built by using IRIS:

- Basic Learning Units (BLUs) or kind of teaching-learning contents.
- Pedagogical Relationships between contents (PRs). In order to establish a pedagogical view for selecting and/or sequencing the BLUs, IRIS includes two types of relationships, structural relationships –is-a, part-of, and sequential relationships –prerequisite, postrequisite and next.
- Instructional Objectives (IOs) or skills to be reached. IOs refer to the application of particular skills over BLUs.
- Didactic resources (DRs). Any kind of IO needs a set of presentation resources – definitions, examples, analogies, etc.– and evaluation resources – tests, fill gaps, item sorting, etc.–.

The process of building the domain in IRIS consists of two phases. During Phase 1 the designer specifies the types of elements (BLUs), skills (IOs), relationships (PRs) and didactic resources (DRs) needed for describing the domain. During Phase 2 the concrete contents and exercises are defined. Although IRIS provides a specialised interface for acquiring each feature of the domain knowledge, this is still a difficult task for the instructional designer. Moreover, the designer can feel lost because s/he doesn't know what is done and what is still to do. This is mainly because s/he has not a complete view of the domain.

In order to alleviate the difficulties found in users while working with IRIS, a more intuitive and graphical interface is proposed (Rueda et al., 2002). Trying to help the users not getting lost this interface minimise the use of menus and the number of different windows. CMs are the basis of the interface and CM-ED the tool that supports it. The interface shows the same appearance for both phases of using the IRIS shell.

In the first phase the instructional designer must establish the characteristics of the teaching/learning domain. This is performed by selecting the type and properties of BLUs, IOs, DRs and PRs. The result of this task will be the adaptation of the initial template of IRIS for the new tutor.

The potential of this kind of interface is shown mainly in the second phase. In this phase the user represents the complete teaching domain by means of a CM. S/he specifies the instances of BLUs and their PRs together with the DRs to teach each BLU. In order to offer more flexibility to the user when s/he develops the domain, the interface offers three working areas corresponding to different views of the CM.

The three views, which are described below, share the same structure. In the left upper side of the window there are two button bars: the node types bar and the relation types bar. The set of nodes and relations of these bars are different in each view (see figure 2). In the right upper side there is a rubber button. The bottom of the window corresponds to the working area in which it is represented the view of the CM. The user manipulates the three view windows in the same way. S/he creates new nodes by dragging and dropping the generic types of the upper side to the working area and deletes them by moving the rubber to the node. The mechanism to set relations between nodes is also simple. First the user selects the relation type and then the nodes that it connects. Relations are also deleted with the rubber.

In addition, it is necessary to include some textual information attached to each node (BLU or DR). For example the file where the text is saved in a text type DR. This information is gathered by very simple forms that are open when clicking the node. Next the three views are described.
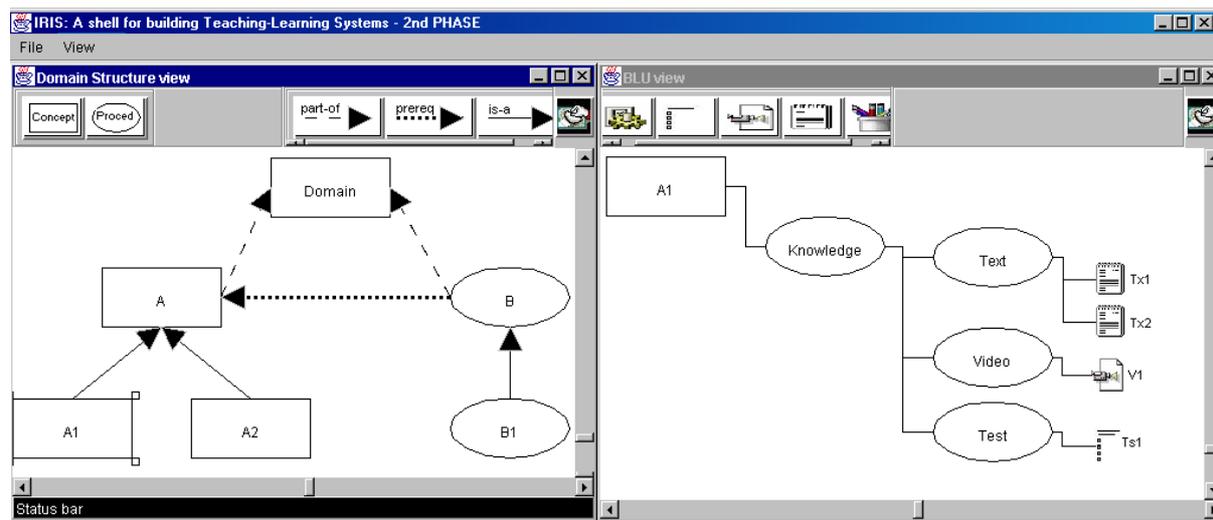

Figure 2. CM-based interface for IRIS

- The **Domain Structure view** represents the set of BLUs of the domain and the PRs between them. The components of the node bar and relations bar are those types of BLUs and PRs selected by the instructional designer in the first phase. In addition, the user can select a BLU in order to define its components in the BLU view. The left part of figure 2 represents a fragment of the domain structure of a tutor.
- The **BLU view** (right part of figure 2) shows the information related to the BLU selected in the Domain Structure view. This information is shown by a tree that represents the components of the BLU, the IOs and the DRs. The purpose of this window is to allow the user to establish the relations between BLUs and DRs.
- The **DR view** shows the complete set of resources ordered by type. In addition to the common manipulation way, the user can change the BLU related to each DR.

The recommended process to follow in the second phase starts with defining the domain at high level using mainly the Domain Structure view. Then the user is able to create the DRs and connect them to the BLUs with both the BLU view and the DR view. Nevertheless, the user can change in any moment the domain structure as well as the DRs attached to each BLU.

The system helps the process by means of a code of colours that represents the state of the nodes. The system highlights automatically nodes that are not initialised or not completely defined and also allows the user to denote those nodes that need some further attention. The system also propagates the state automatically

through the CMs according to a set of rules. For example, when a DR is not completely defined, its corresponding BLU is also highlighted. The user can also attach comments to each node.

With this interface the user defines the domain by drawing a CM with a specialised tool. The tool infers a set of operations when the user performs some action on it. For example, when the user creates a new BLU, the system creates some other nodes automatically: the corresponding IOs and DRs type according with the specification made in the first phase.

4   Using CM-ED for designing exercises: MapEx

When learning about any domain it is necessary to perform different types of exercises. The selection of the appropriate exercises depends on the kind of domain and skills the learner is trying to acquire. Objective tests are one of the most widely used evaluation mechanism not only in the traditional education but also in the computer assisted education. Even more, "testing components were the first to be implemented and currently are the most well developed interactive components also in Web-based education" (Brussilovsky & Miller, 1999). The easy development and evaluation of this kind of test is the main reason for their wide use. Different types of questions such as true/false, multiple choice and fill-in-gaps are recognized in already developed educational authoring tools (Reinhardt & Schewe, 1995; Arruarte et al., 2003; Major et al., 1997; Hsieh et al., 1999).

Although as (Cicognani, 2000) points out, concept mapping is useful in achieving some skills in the learning process, none of the revised authoring tools offer authors the possibility of creating exercises based on CMs. In other kind of learning environments, such as Internet, some concept mapping based systems have been developed. WCOMT (Tsai et al., 2001) is a clear example. It is a CM based testing system that provides an alternative indicator for exploring students´ understandings in physics that may differ somewhat from traditional standard test.

The aim of MapEx is to provide a means for designing and solving graphical exercises based on CMs (Larrañaga et al., 2002). It is based on CM-ED and it profits from the template-based specialization. The tool has two kinds of users: teachers and learners. Teachers design new exercises or types of exercises and supervise the answers of the students. The system allows the learners to solve the exercises with or without help. On the one hand, the tool can be used as an assessment tool in which the student receives no feedback. On the other hand, it can be used in a formative way allowing the learner to see correct and incorrect solutions together with the explanations that the teacher has attached to them. In both cases, the evaluation of the answers is made by the teacher and shown in the notes area. This area is used by both the teacher and the student to engage in a dialogue that supports incremental process of evaluation and development of the answer.

An exercise is an activity in which the learner answers a question drawing a CM. The task may consist of completing a CM that is provided to the student or on drawing the CM from scratch. MapEx identifies two sets of exercises: free exercises and specific purpose exercises. The degree of freedom allowed to the student when constructing the CM is different in each set. In free exercises the learner constructs CMs without any restriction in the kind of operations s/he can carry out and the types of nodes or relations s/he can use. Free exercises do not have any attached template and the student can construct her/his own template as well as the CM. Specific purpose exercises are created with a particular aim, so there are some restrictions on the resources (nodes and relations) and the operations. The restrictions of specific purpose exercises are implemented by means of a template for each kind of exercise. On the one hand, the author limits the types of nodes and the type of relations the student can use. On the other hand, it is also possible to restrict the operations on the nodes and relations allowed to the learner in the exercise. Regarding to the limitations on the resources, the following exercises have been identified: taxonomy drawing, structure drawing, sorting, relation drawing and author defined exercises.

For managing exercises MapEx identifies four main objects: exercise subtypes, exercises, predefined answers and student answers and system answer evaluations. For each exercise subtype the tool stores the following information: identification, author, associated template and date. MapEx represents instances of concrete exercises with the following: exercise identification, exercise subtype, question, an optional initial CM, the correct and incorrect predefined answers defined by the teacher and the list of the answers of the students. A predefined answer is represented by a concept map drawn by the author that represents a correct or incorrect solution associated with a comment with formative purposes. For representing student answers MapEx uses: answer identification, corresponding exercise, student, the CM that represents the student answer and a flag indicating the state of evaluation.

MapEx for the teacher: MapEx allows teachers to define exercises and to supervise the answers of the students through a Multiple Document Interface composed of the following windows:

- **Exercise management window**. A CM representing the classification of exercises types, subtypes and instances is shown. This window allows the author to create, modify or eliminate subtypes, instances and solutions of exercises. To create a new subtype of exercise the author has to specialise the default template

attached to the selected type using the template management window. The creation of exercise instances implies the definition of an optional initial CM and the correct and incorrect answers. The author will use the exercise window for these tasks.

- **Template management window**. In this window, the author specifies in a CM the type of nodes, relations and the operations allowed to the learner when working with the exercise. This window is used to create an author-defined exercise or to create a subtype.
- **Exercise window**. This window allows the author to define an exercise writing the question and drawing the initial CM when necessary. Thus CM is based on a template and, therefore, the window is a specialized version. However, the author is not restricted to the operations of the template. The teacher can define correct and incorrect answers through the answer window and use the note area to describe the exercise.
- **Answer window**. This window is also specialized to a template. It has two uses for the teacher. First it will be used to define the correct and incorrect solutions and it can include explanations in the note area that will be shown to the student with a formative aim. Second, the teacher will supervise the answers of the student through this window. The approach for the evaluation that the tool supports is based on a negotiation between teacher and learner. The note area will be used to implement the communication between them.
- **Supervision window**. In this window the tool shows the list of students with the answers each students has drawn for the exercises s/he made. Clicking in the node, the corresponding CM will be shown so that the teacher can mark it and write notes for the student. The nodes of the answers have also attached flags meaning that the exercise is new or that it has a new comment of the teacher or the student.
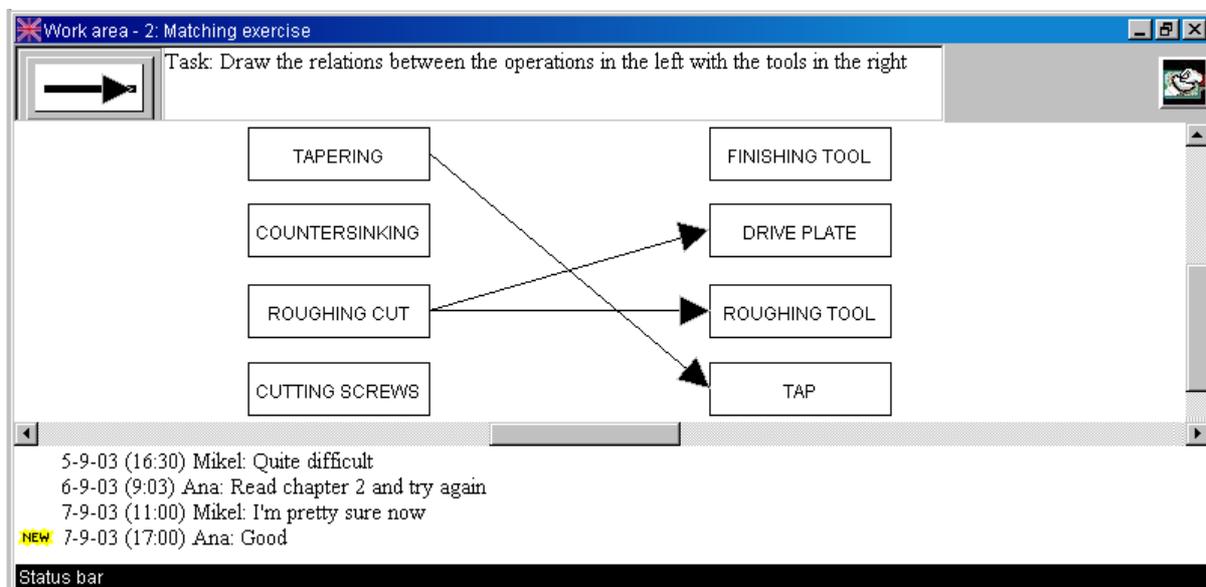

Figure 3. Interface for solving exercises

MapEx for the learner: The tool allows the learners to select the exercise to solve and to draw the corresponding answers. In addition, the student can get involved in a dialogue with the teacher on an exercise or evaluation. MapEx offers students a Multiple Document Interface composed of the following windows:

- **Exercise browser**. In this window the learner selects the exercise to work on. S/he can answer the exercise in the answer window. Exercises have attached flags meaning the following situations: new exercise, new comment from the teacher, new comment of the student and finished exercise.
- **Answer window**. This window is shared with the teachers and supports the communication between them. This window has three uses for the learner. During the resolution of the exercises, MapEx presents the exercise formulation to the learner and it allows him/her to draw the answer in the specialized CM window. When the exercise is being evaluated it is the means of communication between student and learner and will support the negotiation and also the instruction of the learner. Finally, the student can observe, using this window, the correct and incorrect answers defined by the teacher. Figure 3 corresponds to a relation drawing exercise in which the student has to complete an initial CM establishing the relations between operations and tools. The student is only allowed to create and delete relations.

5   Using CM-ED for representing the student: DynMap

Up to now, several authors have considered visualization and inspection of student models (WS, 1999). This component collects the learning characteristics of the student and his/her evolution during the whole learning process. Open Learning Modelling can be viewed as an attempt to directly involve learners in the diagnosis process and, as a result, infer their knowledge about the target teaching/learning domain (Dimitrova, 2002).

Commonly, systems built following the approach of open learner modelling allow to externalise the student model and, in some cases, provide mechanism for both teachers and learners, to change the content of the models. The efforts range from simply visualizing the model, to actively involving the student in the modelling process through negotiation or collaborative construction of the model. The model is not just a source of knowledge about the student of value to the system, but becomes an important learning resource on its own (Hartley & Mitrovic, 2002). (Kay, 2001) collects some of the reason that different authors argue for making the learner model available: users should have access to and control over personal information, correctness of the model can be assessed by the user, access to the learner model improves the learner's appreciation of the system's learning goals, programmer accountability is enhanced if the programmer knows the learner will have access to the model, there is merit in making complex systems more comprehensible, and finally, and importantly, an accessible learner model may aid learning by supporting reflection and planning.

(Cook & Kay, 1994) makes one of the first attempts for representing graphically the learner model; they use a mixture of text and conceptual trees based diagrams. (Morales et al., 1999) presents a table-like graphical representation format of the learner model for a sensory-motor task in which every row represents a rule. (Bull & Nghiem, 2002) claims that the use of simple learner models, easy to represent in different ways, e.g. graphical and tabular, allows teachers and students to better understand students' learning of a target domain. VisMod (Zapata-Rivera & Greer, 2000) provides a flexible architecture where students and teachers can create their own views of a student model by choosing nodes they want to inspect from the Bayesian network that represents the student model. In addition, (Dimitrova et al., 2002) justifies an approach of inspecting and discussing the learner model in a graphical manner using conceptual graphs. They argue that the selection of an appropriate communication medium reduces comprehension problems, facilitates reasoning about learner's behaviour, and promotes metacognitive processes. Finally, the tool CourseVis (Mazza & Dimitrova, 2003) processes student data collected by web-based course management systems and creates pictorial representations (2D and 3D) to help instructors gain understanding of social, cognitive and behavioural aspects of distance students.

The main goal of DynMap is to show graphically the knowledge of the students through the learning process to both the student and the teacher (Rueda et al., 2003). Although the main and initial purpose was to present the internal student model of an ITS it is also applicable to other kind of teaching/learning systems and even to other educational contexts. It includes specialized tools for inspecting the student model addressed to both kinds of users: teachers and students. It uses the core of CM-ED adapted by means of a template.

DynMap represents the knowledge of the student in a concept map following the overlay approach (Golstein, 1982). The main characteristic of an overlay approach is that the knowledge that a student has about a domain is represented as a subset of the whole domain. The domain module represents the curricula to be transferred to the student in a concept map and provides the basis for the graphical view of the student knowledge. Therefore, the student model is a subset of the concept map of the domain that includes those concepts that the student has learnt or is learning.

Taking into account that the student's knowledge changes along the learning process it is interesting that the student module reflects this evolution. Unlike most of the revised student models, DynMap is able not only to keep the current state of the student knowledge but also to record the evolution of the student through the learning sessions. This feature is used to show dynamically the sequence of states of the student model.

DynMap is composed of a knowledge base that maintains the set of students concept maps, the DynMap Repository, and four functional modules: Translator, DynMap Builder, Student-DynMap and Teacher-DynMap. The translator is a piece of code to be integrated into the student data source while DynMap Builder, Student-DynMap and Teacher-DynMap constitute an independent Java application.

DynMap Repository: This repository manages the storage of the dynamic evolution of all the students' concept maps. The **Student Concept Maps** (SCMs) are stored after each learning session and the changes performed on it while the session is carrying out in the **Operations Log** file. An Operation Log contains the sequence of operations performed in a session and the time when each one was done. Therefore, the **Dynamic Student Concept Map** (DynSCM) is composed of a sequence of SCM and Operation Logs.

Translator: This module is responsible for connecting the data source about the student with the inspection tools. As said before, this data source can be different, an ITS, an Open Learning System, a tool for managing marks of students in classic teaching, etc. Therefore, the translator is a piece of code integrated into the data source written in a programming language appropriate for that data source. It translates the internal learner model to an intermediate representation in XML that can be processed by DynMap.

DynMap Builder: This module is responsible for reading the XML files containing the data of the students and creating the concept maps that represent them. It uses the concept map of the domain as the skeleton of the student model and represents the individual data of the learner.

Teacher-DynMap: This component allows the teacher to select the student to observe by means of a browser. Then, the teacher can observe the DynSCM that represent the beliefs of the system about the current state and the evolution of the student's knowledge. Using a set of video-like buttons of a tool bar the user can move through the learning process of the students. The bottom part of the Teacher-DynMap has the note area. This will support communication between the teacher and the student.

Student-DynMap: The student viewer is simpler. The student's inspector allows the learner to observe the concept map that represents the current beliefs of the system about his/her knowledge.

6 Conclusions

Throughout this paper the authors have shown the potential of CM-ED in different tasks of the Computer Aided Teaching/Learning area. It has been used inside an authoring tool to specify the domain knowledge of an ITS.

MapEx allows the generation and solving of graphical exercises based on CMs. The tool has two kinds of final users: authors and students. On the one hand, teachers or curriculum designers define free exercises, exercises without any restriction, and specific purpose exercises, exercises with particular aims. On the other hand, students solve the proposed graphical exercises creating or completing CMs. MapEx can be used as an assessment tool in which the student receives no feedback or in a formative way allowing the learner to see correct and incorrect solutions together with attached explanations. In both cases, the evaluation of the answers is made engaging the teacher and the student in a dialogue that supports an incremental process of evaluation and development of the answer.

DynMap is a system for dynamically visualizing the Learner Model of an Intelligent Tutoring System (ITS) by means of Concept Maps. The main goal is to facilitate teachers and students to better understand ITS outcomes, concretely the beliefs of the ITS about the student's knowledge. DynMap uses the core of the Concept Map EDitor (CM-ED) adapted by means of a template. It has two types of potential users: teachers and students. On the one hand, teachers can observe the dynamic evolution of the student and the state of the learner model at any time of the learning process. On the other hand, students are allow to observe what the automatic tutor knows about their current knowledge.

Finally, CM-ED has been used in other areas such as Natural Language Processing. Concretely it has been integrated in Abar-Hitz, an annotation tool for the Basque Dependency Treebank (Diaz de Ilarraza et al., 2004).

7 References

Anderson-Inman, L., Ditson, L. and Ditson, M. (1998) Computer-Based Concept Mapping: Promoting meaningful learning in science for students with disabilities. Information Technologies and Disabilities Journal, Special K-12 issue, article 2.

Arruarte, A., Ferrero, B., Fernández-Castro, I., Urretavizcaya, M., Alvarez, A., Greer, G.: The IRIS Authoring Tool. In: Murray, T., Blessing, S. and Ainsworth, S. (eds.): Authoring Tools for Advanced Technology Learning Environments. Kluwer Academic Publisher (2003) 233-267.

Bruillard, E. & Baron, G.L. (2000). Computer-based concept mapping: a review of a cognitive tool for students. In Proceedings of ICEUT2000, 16th IFIF World Computer Congress (eds. D. Benzie & Passey, D.), pp. 332-338. PHEI.

Brussilovsky, P. & Miller, P. (1999) Web-based testing for distance education. In Proceeding of WebNet'99, World Conference of the WWW and Internet (ed P. De Bra and J. Leggett), pp. 149-154. AACE.

Bull, S. and Nghiem, T.: Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers and Instructors. In: Brna, P. and Dimitrova, V. (eds.): Proceedings of Workshop on Individual and Group Modelling Methods that Help Learners Understand Themselves, ITS2002 (2002) 5-13.

Chang, K.E., Sung, Y.T. and Chen, S.F. (2001) Learning through computer-based concept mapping with scaffolding aid, Journal of Computer Assisted Learning, vol. 17, pp. 21-33.

Chiu, C.H., Huang, C.C. and Chang, W.T. (2000) The evaluation and influence of interaction in network supported collaborative concept mapping. Computers & Education, vol. 34(1), pp. 17-25.

Cicognani, A. (2000) Concept Mapping as a Collaborative Tool for Enhanced Online Learning, Journal of Educational Technology and Society and IEEE Learning Technology Task Force, vol. 3(3), pp. 150-158.

Cook R., and Kay, J.: The justified user model: A viewable, explained user model. In: Fourth International Conference on User Modelling. The MITRE Corporation, Hyannis, MA. (1994) 145-150.

Dabbagh, N. (2001) Concept Mapping as a Mindtool for Critical Thinking. Journal of Computing in Teacher Education, 17(2), pp. 16-24.

Díaz de Ilarraza, A., Garmendia, A., Oronoz, M. (2004) Abar-Hitz: An Annotation Tool for the Basque Dependency Treebank. Language Resources and Evaluation Conference (LREC 2004).Lisbon, Portugal.

Dimitrova, V.: Interactive cognitive modelling agents – potential and challenges. In: Brna, P. and Dimitrova, V. (eds.): Proceedings of Workshop on Individual and Group Modelling Methods that Help Learners Understand Themselves, ITS2002, (2002) 52-62.

Dimitrova, V., Brna, P. and Self, J.: The Design and Implementation of a Graphical Communication Medium for Interactive Learner Modelling. Intelligent Tutoring Systems ITS'2002 (2002) 432-441.

Golstein, I.P.: The Genetic Graph: a representation for the evolution of procedural knowledge. In: Sleeman, D. and Brown, J.S. (eds.): Intelligent Tutoring Systems, Academic Press (1982) 51-77.

Hartley, D. and Mitrovic, A.: Supporting Learning by Opening the Student Model. In Cerri, S. Gouardères and Paraguaçu (eds.) Proceedings of International Conference on Intelligent Tutoring Systems (2002) 453-462.

Hsieh, P.Y. Halff, H.M. Redfield, C.L. (1999) Four Easy Pieces: Development Systems for Knowledge-Based Generative Instruction. International Journal of Artificial Intelligence in Education, vol. 10, pp. 1-45.

Jonassen, D.H., Reeves, T.C., Hong, M., Harvey, D., Peters, K. (1997) Concept Mapping as Cognitive Learning and Assessment Tools. Journal of Interactive Learning Research, vol. 8(3/4), pp. 289-308.

Kay, J.: Learner Control. User Modelling and User-Adapted Interaction, Vol. 11 (2001) 111-127.

Larrañaga, M., Rueda, U., Elorriaga, J.A., Arruarte A.: Using CM-ED for the Generation of Graphical Exercises Based on Concept Maps. In: Kinshuk, Lewis, R., Akahori, K., Kemp, R., Okamoto, T., Henderson, L., and Lee, C.-H. (eds.): Proceedings of ICCE'2002 (2002) 173-177.

Larrañaga, M., Rueda, U., Elorriaga, J.A., Arruarte, A.: A multilingual concept mapping tool for a diverse world. In: Hoppe, U., Verdejo, F. and Kay, J. (eds.): Proceedings of IEEE ICALT'2003 (2003) 52-56.

Major, N., Ainsworth, S. & Wood, D. (1997) REEDEM: Exploiting Symbiosis Between Psychology and Authoring Environments. International Journal of Artificial Intelligence in Education, vol. 8, pp. 317-340.

Mazza, R. and Dimitrova, V.: CourseVis: Externalising Student Information to Facilitate Instructors in Distance Learning. In: Hoppe, U., Verdejo, F. and Kay, J. (eds.): Proceedings of AIED'2003 (2003) 279-286.

Morales, R., Pain, H. and Conlon T.: From behaviour to understandable presentation of learner models: a case study. In: Proceedings of the Workshop on Open, Interactive, and other Overt Approaches to Learner Modelling, AIED'99, Le Mans, France (1999).

Novak, J.D.: A theory of education. Cornell University, Ithaca, NY (1977)

Reinhardt, B., Schewe, S. (1995) A Shell for Intelligent Tutoring Systems. In Proceedings of Artificial Intelligence in Education, (ed. J. Greer), pp. 83-90. AACE.

Rueda, U., Larrañaga, M., Arruarte, A., Elorriaga J.A.: Using a Concept Mapping Tool for Representing the Domain Knowledge. In: Petrushin, V., Kommers, P., Kinshuk and Galeer, I. (eds.): Proceedings of IEEE ICALT'2002 (2002) 387-391.

Rueda, U., Larrañaga, M., Arruarte, A., Elorriaga, J.A.: Dynamic Visualization of Student Models Using Concept Maps. In: Hoppe, U., Verdejo, F., Kay, J. (eds.): Artificial Intelligence in Education. Shaping the future of learning through intelligent technologies, IOS Press (2003) 89-96.

Tsai, C.C., Lin, S.S.J. and Yuan, S.M. (2001) Students' use of web-based concept map testing and strategies for learning. Journal of Computer Assisted Learning, vol. 17, pp. 72-84.

Workshop on Open, Interactive, and other Overt Approaches to Learner Modelling. AIED'99, Le Mans, France, July, 1999 (http://aied.inf.ed.ac.uk/members99/resources/modelling/modelling.html).

Zapata-Rivera, J. and Greer J.: Inspecting and Visualizing Distributed Bayesian Student Models. In: Gauthier, G., Frasson, C. and VanLehn, K. (eds.): Proceedings of International Conference of ITS (2000) 544-553.

8   Acknowledgements