Abstract

We have developed a JAVA prototype of a template-based Concept Mapping Tool, the CM-ED tool, that supports the above mentioned applications. In this paper we will show how CM-ED can be used within an Intelligent Tutoring System authoring tool, the IRIS authoring tool, to represent the domain knowledge.

2. Domain knowledge in IRIS

IRIS [1] [3] 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. Based on Merrill’s Component Display Theory [5] IRIS represents any domain in terms of the following four BLUs – concepts, procedures, principles and facts;
- 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, corequisite, postrequisite and next;
- Instructional Objectives (IOs) or skills to be reached. IOs refer to the application of particular skills over BLUs. They form a useful part both in planning the teaching-learning process and in creating didactic activities. The default IOs in IRIS are those of the widely accepted Bloom’s taxonomy [4] – knowledge, comprehension, application, analysis, synthesis and evaluation;
- Didactic resources (DRs). Any kind of IO needs a set of presentation resources or techniques which can be used to transmit the domain BLUs to the learner – definitions, examples, analogies, etc.– and evaluation resources or techniques for assessing domain contents – 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) between contents and didactic resources (DRs) needed for describing the domain at pedagogical level. During Phase 2 the concrete contents and exercises are defined.
Let’s suppose the designer wants to build a tutor for the machine tool domain, particularly for using a lathe. Once the domain is chosen, the immediate task is to determine the kind of characterisation required to describe the contents to be conveyed to the student (Phase 1):

- **BLUs** - The student must learn concepts (theoretical contents) and procedures (operations needed for manipulating the lathe).
- **IOs** - The student will be able to know all domain concepts and apply the procedures.
- **DRs** - The new tutor should use several types of presentations, such as definitions and examples, and several evaluation resources, such as multiple choice tests and fill gaps.
- **PRs** – Some order among contents must guide the new tutor performance.

IRIS provides an interface based on chained forms for specifying these requirements. Figure 1 shows the selection of characteristics for the knowledge instructional objective in the lathe domain tutor. The selected characteristics are those marked.

Concerning Phase 2, the instructional designer must identify the elements that must be conveyed to the learner together with the skills s/he has to reach for each one. Therefore, IRIS gathers using classical chained forms: the whole set of BLUs, the PRs among them, the IOs and the DRs. Figure 2 shows a fragment of the BLUs and PRs identified in the lathe domain.

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 elements.

**Figure 1. Specification of the Knowledge IO.**

**Figure 2. Fragment of the domain.**

### 3. CM-ED Tool

CM-ED (Concept Maps Editor) is a flexible tool for editing CMs. The aim of the tool is to be useful in different contexts and uses. On the one hand it is a
common concept mapping editor in which the user can draw nodes and relations generating different CMs. Consequently, it can be used as a general purpose cognitive tool for supporting learning. On the other hand, the tool can be adapted to specific purposes. Particularly, this second use is going to be presented along the paper. The specialization of the tool is implemented by means of templates (see figure 3). A template specifies the kind of nodes, relations and labels that are going to be used in a category of CMs. Thus, in order to specialise the tool the user designs an appropriate template, for example by modifying one of the templates from the library of the tool. Then, the tool adapts the interface to meet the template specification. This means that the tool is restricted to the resources specified in the template. In the end, the final user works with the adapted tool and produces the CMs with a particular purpose.

The interface of the tool is intuitive and friendly. It has been designed with the aim of sharing the same treatment for designing the template as well as for building the CM. Thus, when designing the template the user selects it from the library and the tool shows it as a CM allowing the user to make modifications on it (see figure 3). The interface 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.

In order to integrate CM-ED in the IRIS authoring tool it was necessary to design a template that includes the graphic elements corresponding to the nodes and the pedagogical relationships identified in any teaching domain. Next, CM-ED was adapted with this template. Finally, the adapted version of CM-ED is used by IRIS to help the gathering of information from the instructional designers.

4. A Concept Map-based Interface in IRIS

In order to alleviate the difficulties found in users while working with IRIS, a more intuitive and graphical interface is proposed. 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 support 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. For example, although the IRIS initial template offers four possible types of BLUs (concepts, procedures, principles and facts), the representation of the lathe domain only needs two types of BLUs, concepts and procedures, so the initial template must be adapted. The interface is organised in two areas (see figure 4). The left side shows the complete set of properties that must be established in a hierarchical tree in which each node corresponds to a subset. The user selects any node in the left area and the interface shows the corresponding properties in the right part of the window as another CM. The user can switch on properties by connecting them with the main concept of the map or switch them off by deleting the relations between them and the main concept. The interface uses a colour code (explained below) to represent the configuration state of the nodes. These simple graphical resources together with the tree structure helps the instructional designer to know what sets of properties are still undefined or should be supervised.

The second phase confirms the potential of this kind of interface. 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. Therefore, the user transmits a lot of information to the IRIS authoring tool and s/he will need support in doing that. 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 5). In the right upper side there is a rubber button. The bottom of the window corresponds to the working area in which the view of the
CM is represented. 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.

- The Domain Structure view (left side of figure 5) represents the set of BLUs of the domain and the PRs between them. The components of the node bar and relations bar are those type 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 BLU view (right side of figure 5) shows the information related to the BLU selected in the Domain Structure view. This information is shown by an horizontal 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 user creates and deletes the resources dragging and dropping the icons of the generic DRs, for example texts, tests and so on.

- 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 instructional designer uses these three views to build the domain. The user can choose between the views depending on her/his needs.

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.

In both phases the system helps the process by means of a code of colours that represents the state of the nodes. In the first phase the state is assigned to a node in the hierarchical tree. In the second phase it is assigned to each node that represents BLUs, IOs and DRs. The system highlights automatically nodes that are not initialised or not completely defined and also allow 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. In those cases the user can 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.

5. Conclusions

In this paper the authors have shown how a concept mapping tool can be used inside an authoring tool to specify the domain knowledge of an Intelligent Tutoring System. There are some advantages using CMs to design such a complex structure: it allows the designer to have all the information directly accessible, it makes it easier for the designer to see information in different ways or from different points of view, and the designer always knows the development stage of the domain, where s/he is, what s/he has already done and what remains to do.

6. Acknowledgements

This work is funded by Gipuzkoa Council, Univ. of the Basque Country (1/UPV00141.226-T-13946/2001), Basque Government (UE1999/36) and CICYT (TIC1999-0252)

7. References


