Advanced Annotation and External Representation in LKC System

Khalid Rouane, Claude Frasson, Marc Kaltenbach

Advanced Annotation and External Representation in LKC System

Khalid Rouane, Claude Frasson and Marc Kaltenbach [1]
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
C.P.6128, Succ. Centre-Ville
Montréal, Québec, Canada H3C 3J7
{rouane, frasson, kaltenba}@iro.umontreal.ca

Abstract

The main objective of the LKC approach (Learning by Knowledge Construction [Rouane, Frasson and Kaltenbach 2002]) is to provide an effective support for a student's reading activity in a learning context while promoting and facilitating his understanding and knowledge acquisition. This paper shows how active reading can be supported and sustained with the use of advance annotation approach of the LKC system.

L'approche d'apprentissage par construction de connaissance, nommée LKC (Learning by Knowledge Construction [Rouane, Frasson and Kaltenbach 2002]) vise à fournir un support effective à l'apprenant durant l'activité de lecture pour promouvoir et faciliter la compréhension et l'acquisition de connaissances. Cet article montre comment une lecture active peut être soutenue par un système avancé d'annotation utilisant cette approche.

Introduction

The domain knowledge in higher education is mainly communicated to students via textbooks and text documents. This is also the case in *intelligent tutoring systems* where concepts and definitions are still presented to the learner in the form of resources to be read. But despite the great importance of the reading activity as a means of knowledge communication, very little research in ITSs have been done to give the student an effective support at the reading stage. The propose of this paper is to present our tutoring system, named LKC for ***Learning by Knowledge Construction*** [Rouane, Frasson and Kaltenbach 2002], which aims to provide full and flexible student support in knowledge construction and acquisition based on reading activity.

Reading activity as part of a learning process aimed to knowledge construction and acquisition is a very active task. Constructivist learning theory [Bruner 1973], states that learning is an active process in which learners ***construct*** new ideas based upon their current and past knowledge. The learner ***selects information***, ***transforms it***, ***constructs hypotheses***, and ***makes decisions***. An analog process happens during reading activity in a learning context. Passive learning, as passive reading, will likely fail [BenAri 1998]: because each student brings a different knowledge framework to the classroom, and will construct new knowledge in a different manner. Learning must be active (and so must be reading): the student must construct knowledge ***assisted by guidance*** from the teacher and ***feedback from other students*** (or from the system).

Active reading is the combination of reading with critical thinking and learning new information in the current knowledge context, and is a fundamental part of education and knowledge work [Adler and Doren 1972]. For non-recreational reading, as is the case in higher education, active engagement with read materials is a key part of ***understanding*** [Phelps and Wilensky 1997]. Most of the work in ITS has been toward competence (or performance) dealing with problem solving, exercises, tasks performing and so on, but little have been done to enhance knowledge acquisition and understanding, especially at the reading stage. Understanding and competence are linked but not equivalent [Ohlsson 1996]. Someone can perform well without understanding all aspects of a given problem, and the converse is true. Understanding is the outcome of higher-order learning while competence is the result of skill acquisition [Ohlsson 1996].

Active Reading and Understanding

Active reading on conventional paper support involves not just reading, but also underlining, highlighting, placing marks or words in a margin, drawing various graphical representations and scribbling comments, either on the text itself or in a separate notebook or Post-It™ [Correia and Chambel 1999] [Levy and Marshal 1995] [Cousins, Baldonado and Paepcke 2000]. These different practices are referred to by the notion of 'annotation'.

The annotation activity is a routine part of the reader engagement with the materials [Marshall 1997]. But in digital support, as is the case in ITSs, active reading is hindered by the lack of flexible annotation tools as stated by Marshall, C. [Marchall 1998]:

> "Annotating digital materials is not a straightforward activity. We have neither the practices nor the tools for fluidly marking on digital materials in all the ways we mark on paper. Yet we often desire to do so."

Most readers of digital support, including students, for further and active engagement with reading documents,

[1] Financial support from Bishop's University is gratefully acknowledged.

prefer printing them out to annotate them easily [O'Hara and Sellen 1997]. But this practice in an ITS changes the reading activity to a ***black box*** where the system is not aware of what happening inside, and therefore cannot supervise neither help the reader student. Moreover, annotation construction cannot benefit from the help computers can provide in manipulating, organizing or sharing these notes.

The annotation process can be a valuable tool for tracking and analyzing the student's understanding, because it reflects the knowledge construction process taking place inside the student's mind. The aim of the LKC system is to take advantage of this annotation activity to ***assist*** and ***guide*** the student in an explicit process of knowledge construction. While reading a didactic document, the student will successively use the system in:

- ***Simple annotation*** tasks: we define a simple annotation task as annotation activity that doesn't involve any transformation of source information, like marking of text[2] (underlining or highlighting [Ohara 1996]) and aggregation of text sentences.
- ***Intermediate annotation*** tasks: it involves some transformation of source information like in simplification or summarization of text, where a group of sentences is replaced by one sentence.
- ***Advanced annotation*** tasks: it involves drawing various graphical representations or external representations [Cox and Brna 1995], which are used to clarify ideas and relationships among annotated elements.

The LKC system tracks the student annotation work, especially in intermediate and advanced annotation tasks, and compares it to an expert work already stored in the system knowledge base, considered as a *reference model*. Detected differences between the two works can be a sign of misunderstandings or misconceptions, which the system will try to reduce if not eliminate, by providing the student with hints and explanations using this *reference model* of the read document.

The LKC Knowledge Models

Knowledge models in the LKC system are based on the *discourse* comprehension *theory* of A. van Dijk and W. Kintsch [Dijk and Kintsch 1983]. This theory states that understanding in the reading process is essentially a 'bottom-up' process, dealing with three types of knowledge structuring:

- ***Microstructures*** of discourse are basic units of text carrying *basic understandings* or *cognitive units*.
- ***Macrostructures*** of discourse are a compilation of microstructures units by application of semantic rules such as deletion, generalization and construction.
- ***Superstructures*** of discourse are schematic forms that organize the global meaning of the read document.

Taking into account this knowledge structuring process performed by the learner in reading activity, we propose in the LKC system that each didactic document be *augmented* with the following knowledge models to give the learner specific and adapted support at each of the above structuring stages (Figure 1).

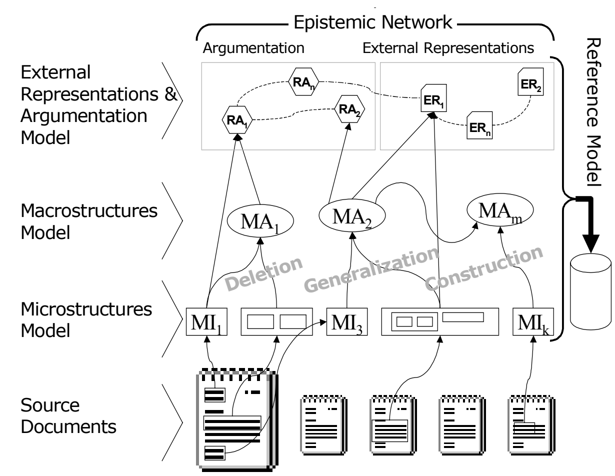


Figure 1 - Knowledge Structuring Model

The Microstructure Model (Mi). This model deals with the organization of the text at the level of microstructures (words and sentences). It aims to support simple annotation task. Microstructures are of three types:

- *Atomic microstructure (Mi-Atomic)*: it can be seen as a clause in natural language like a group of words or sentences.
- *Composed microstructure (Mi-Composed)*: is a set of atomic microstructures.
- *Complex microstructure (Mi-Complex)*: is a set of microstructures with at least one microstructure that is not atomic (is composed or complex).

The LKC system uses this model to help the student select interesting text fragments at various levels of detail, from an atomic element made with a word or a sentence, to a complex element made from a group of sentences.

$$\textbf{\textit{Mi}} = \textbf{\textit{Mi-Atomic}} \cup \textbf{\textit{Mi-Composed}} \cup \textbf{\textit{Mi-Complex}}$$
$$\textbf{\textit{Mi-Atomic}} = \{\text{Text fragments from source document}\}$$
$$\textbf{\textit{Mi-Composed}} = \{ (a_1, a_2, a_3, \ldots a_n) \mid a_i \in \textbf{\textit{Mi-Atomic}} \}$$
$$\textbf{\textit{Mi-Complex}} = \{ (c_1, c_2, c_3, \ldots c_n) \mid c_i \in \textbf{\textit{Mi-Atomic}} \cup \textbf{\textit{Mi-Composed}} \}$$

The Macrostructure Model (***Ma***). This Model aims to support intermediate annotation task. Through the action of various rules of combination and reduction, microstructures (words and sentences) are successively

[2] We talk of text as an example, but the same technique can be adapted to other media such us images, figures, …

reduced in an iterative fashion to produce higher order macrostructures that convey macropropositions. A. van Dijk and W. Kintsch [Dijk and Kintsch 1983] give three macrorules for this transformation:

- *Deletion*: propositions that are not direct or indirect condition of interpretation of another proposition can be deleted.
- *Generalization*: a sequence of propositions may be substituted by one proposition if this proposition is entailed by each of the members of this sequence.
- *Construction*: a sequence of propositions may be substituted by one proposition if this proposition is entailed by this joint set of propositions.

The application of these macrorules to produce macrostructures is a transformation process that simplifies the document read and gives it a new logical organizational structure, in addition to its original structure (chapters, subchapters, paragraphs, sentences).

$\boldsymbol{Ma} = \{ \boldsymbol{f}(m_1, m_2, m_3, \ldots m_n) \mid m_i \in \boldsymbol{Mi} \cup \boldsymbol{Ma} \}$
$\mathbf{f} \in$ { Deletion, Generalization, Construction } are transformation rules on existing annotations.

The External Representations Model (Er). This model aims to support advanced annotation task. To focus his attention on the relationships among the different ideas in the text, the reader uses various types of graphical representations to link these ideas [Ohara 1996]. External representations (ERs) is a general term used to define this kind of graphical representations [Cox and Brna 1995]. Concepts map are a well-known example of ERs [Anderson-Inman and Zeitz 1999], and all figures in this paper are kinds of ERs. The importance of ERs in education comes from their weak expressiveness (limited number of possible interpretation) that makes inference and reasoning more tractable for the student [Stenning and Oberlander 1995] [Cox and Brna 1995]. Many systems were made to deal with ERs in education, such as MindManager [Mindjet 2002] and Inspiration [Inspiration 2002]. However, as they don't integrate any conceptual knowledge about what is represented, they have no automated support to the student and can only be used as cooperative tools.

$\boldsymbol{Er} = \{\boldsymbol{N}, \boldsymbol{A}\}$ where
$\boldsymbol{N} = \{ e_1, e_2, e_3, \ldots e_n \mid e_i \in \boldsymbol{Mi} \cup \boldsymbol{Ma} \cup \boldsymbol{Er} \cup \boldsymbol{Ar} \}$
$\boldsymbol{A} = \{ (e_a, e_b) \mid e_a \in \boldsymbol{N} \text{ and } e_b \in \boldsymbol{N}\}$

Argumentation Models *(Ar)*. One special form of external representations is *argumentation diagrams*. They are special in that they deal with argumentation and have some kind of formalism. Belvedere is an example of system using argumentation diagrams [Suthers et al. 1995]. In LKC system, we use a simple but powerful definition of argumentation based on the Toulmin view of Argumentation [Toulmin 1958].

$\boldsymbol{Ar} = \{ (d, w, b, q, r, c) \mid d, w, b, q, r \text{ and } c \in \boldsymbol{Mi} \cup \boldsymbol{Ma} \cup \boldsymbol{Er} \cup \boldsymbol{Ar} \}$
d = datum is the evidence supporting the claim.
w = warrant is the principle, provision or chain of reasoning that connects the datum to the claim.
b = backing is justifications and reasons to back up the warrant.
q = quantifier is specification of limits to claim, warrant and backing.
r = refutable is exceptions to the claim; description and rebuttal of counter-examples and counter-arguments.
c = claim is the position or claim being argued for; the conclusion of the argument.

Note that the definitions of ERs and argumentations can be recursive: an ER or argumentation can be used to define another ER or argumentation. This can lead to what we call *meta-argumentations*, which are argumentations about argumentations or ERs.

Epistemic Network Model (En). External representation elements from ***Er*** model and argumentation elements from ***Ar*** model are merged in one big network, the ***epistemic network***, noted ***En***, using links such as *Similar, Uses* and *Special-case-of*. This model is constructed by the course designer as part of the reference model of the document and has a important role in pedagogical planning.

$\mathbf{En} = \{ r(m, n) \mid m \text{ and } n \in \mathbf{Er} \cup \mathbf{Ar} \}$ where
$r \in$ { Uses, Help-in, Similar, More-difficult, Special-case-of }

Implementation of the LKC system

Authoring Environment

The responsibility of the course designer in the LKC system is the creation of an *augmented document*, which is a combination of a *raw document* and a *reference model* (Figure 2). In the current version of the LKC prototype the raw document must be an HTML document, because the application uses DHTML to interact with its elements (like highlighting a sentence by changing its background color attribute to yellow or processing a mouse event when it is clicked). In the future versions, we plan to introduce more flexible and robust document formats, especially those based on an XML language, such as SVG format [Adobe 2002].

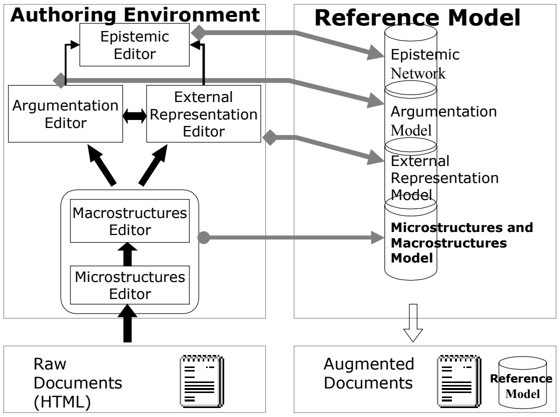


Figure 2 – Authoring Environment

The raw document can be edited inside the LKC system, but it can be received as an input in HTML format if the designer chooses to use a specialize HTML editor, like MS FrontPage or Macromedia Dreamweaver. The real work of the designer in the LKC system starts at the microstructure model edition. We will take a close look at the microstructure editor to demonstrate the basic concepts of edition in the LKC system.

The microstructure editor is a graphical editor providing two windows to the designer: one with the raw document to be augmented and the other receive created microstructures. To create atomic microstructures (Figure 3), the designer selects the desired text on the window displaying the raw document and clicks on create-atomic-microstructure button on the menu. The system automatically generates an Id for this element as its unique reference, for instance “at36260”, and adds it to the database along with other information, such as the designer Id and the date of creation. Then the HTML document is modified as follow to reflect this changes:

```
… <span id="at36260">a business use-case model</span>
…
… at36260.style.background="yellow" <!-- to highlight this element -->
```

In the window of the microstructure editor, a box symbolizing the new atomic element is added. One interesting note is that atomic elements have two modes of display:

- Text-view mode: this is applicable for text elements. Only the text is displayed without any respect to its original location, font, size or color, like in the case of element “at36259” in the (Figure 3).
- Camera-view mode: this is applicable for all element types, including images and figures. The element is displayed and highlighted in its original context. It acts as a virtual camera directed to the element in its original position on the source document. See element “at36260” representing a text, and “at36290” representing a figure.

After the creation of a set of atomic elements, the designer can link them to create composed and complex microstructures (like “cp36274” and “cx36279” in Figure 3).

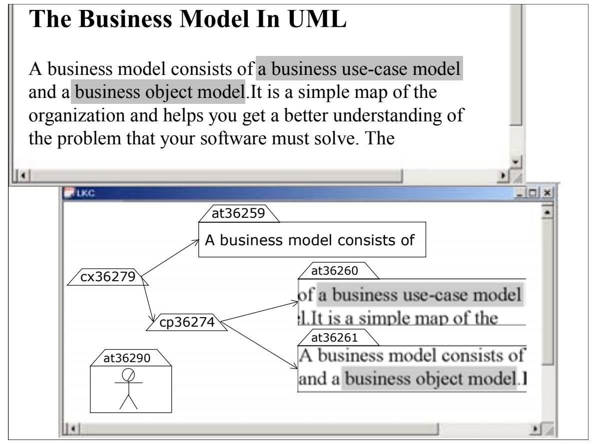
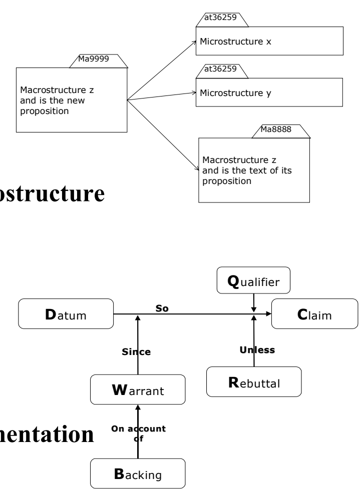


Figure 3 – Microstructure Editor

The authoring environment provides similar graphical editors for the remaining knowledge models. The macrostructures are created in a similar way to the creation of composed or complex microstructure (a)-Figure 4, except that a *new proposition* (text) is given as a replacement of all elements linked to this macrostructure, by applying one of the three macrorules of transformation: deletion, generalization and construction.



(a) Macrostructure

(b) Argumentation

Figure 4 – Sample templates of macrostructure and argumentation elements

External representations are created as a set of nodes and links. It is mandatory that each of these nodes and links be related to a microstructure, a macrostructure or to another external representation. Argumentations are created as external representations but with constraints on their form. Each element of the argumentation like a datum or a claim must respect strict positioning rule as indicated in (b)-Figure 4.

Strategies for knowledge modeling in the LKC system. The authoring process can be done in two ways: bottom-up process or top-down process.

- The bottom-up process is the authoring process we've just described. It is a document-driven process. Having at hand a raw document, the designer creates various knowledge structures (from micro, to macrostructures, to ERs, …) following the order imposed by the document. This strategy is recommended when the raw document is already available at the start of the process. It is easy to conduct but we have less control on the outcome of knowledge structuring.
- Conversely, the top-down process is an authoring process starting with the creation of high-level knowledge structures such as ERs and argumentations, followed by the creation of the raw document guided by this high-level knowledge structures and then followed by the creation of micro and macrostructures that bridge the gap.

Nevertheless, whatever the strategy the designer chooses, the result is the same: a multilevel knowledge representation of the document content, ranging from low-level and detailed knowledge at microstructure level, directly linked to the document, to high-level and concentrated knowledge at ERs and argumentation level (Figure 5). In addition, the most important thing is that any knowledge element at any level can be traced back to the original source elements at the root of its creation.

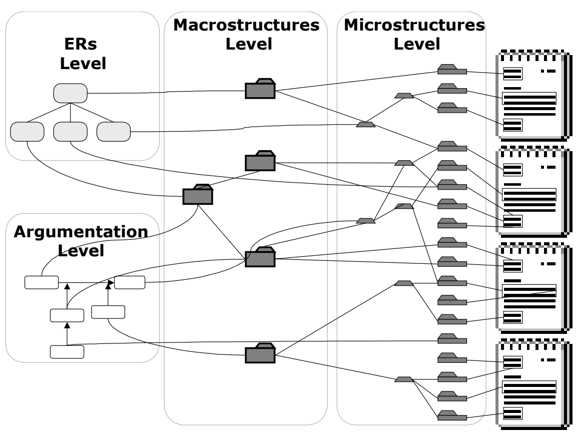


Figure 5 - Multilevel Knowledge Representation

Student Learning Environment

The student learning environment is the set of modules in the LKC system used by the student during reading activity (Figure 6). The student can use this environment in two modes: in a controlled or supervised mode and in free or unsupervised mode.

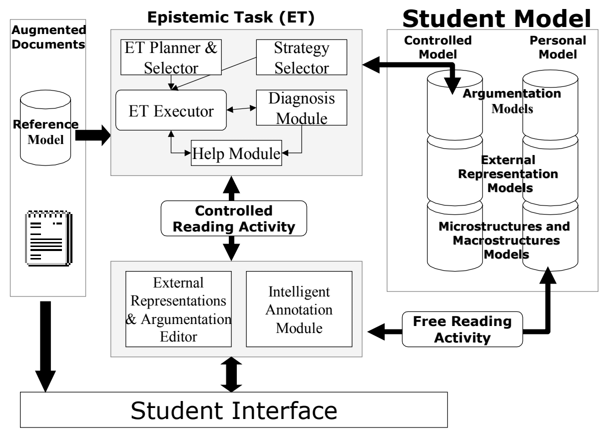


Figure 6 - Student learning environment

Controlled Reading Activity

For a student, controlled reading activity involves reading an augmented document while executing some epistemic tasks. We define an ***epistemic task***, noted ET, as the construction by the student, while he is reading an augmented document, of an external representation or an argumentation present in the *reference model* of this document and previously created by the designer. We think that a correct construction can ensure that the student understanding of read material is near or similar to the understanding of the designer.

The reason behind asking the student to reconstruct these representations instead of showing them as part of the content is that self-constructed ERs have been shown to be more effective than prefabricated ones [Grossen and Carnine 1990] and have similar pedagogical effect as self-explanations [Chi et al. 1989].

The controlled reading activity starts with a planning phase performed by the system. Based on the *epistemic network model* of the document, the ***epistemic task planer and selector*** **module** of the learning environment plans a sequence of epistemic tasks to be executed by the learner. Two constraints guide the strategy of sequencing: 1) an epistemic task must be scheduled after the schedule of all its prerequisites 2) whenever is possible, the planer schedules easy epistemic tasks first, using the number of elements (nodes and links) in this tasks as a metric.

After the planning phase, the student is invited to perform, successively, the generated sequence of epistemic tasks using the ***external representation and argumentation editor*** (Figure 7). This editor, like one used by the designer, gives the student means to construct graphical representation of text ideas requested by the current epistemic task. To accelerate the annotation process and guide the student in his reading, ***empty templates*** of required construction are presented to him. To fill these templates, the student has to read the appropriate part of the document, find the appropriate words, sentence or group of sentences and selecting appropriate microstructures and macrostructures. The ***intelligent annotation module*** is the module in charge to assist the student at this step. It takes advantage of the fact that each element in an external representation or an argumentation in an ET[3] can be traced back to macrostructure, then microstructure elements and further to source document. This way the module can guide the student to where to read on the document and what to read carefully for the current specific epistemic task. And this in turn can decrease the student cognitive load and increase his motivation and his reading speed.

Others components are also involved in providing student help. The ***Strategy Selector*** is in charge of selecting the most appropriate strategy to start an epistemic task. The simplest strategy, but the most difficult for the student, is to give him a blank screen and ask him to execute the ET[3] based only on its definition. The easiest strategy for the student is the empty template strategy discussed above and implemented in the current version of the system. An intermediate strategy would be to give him an uncompleted graph with empty and full slots.

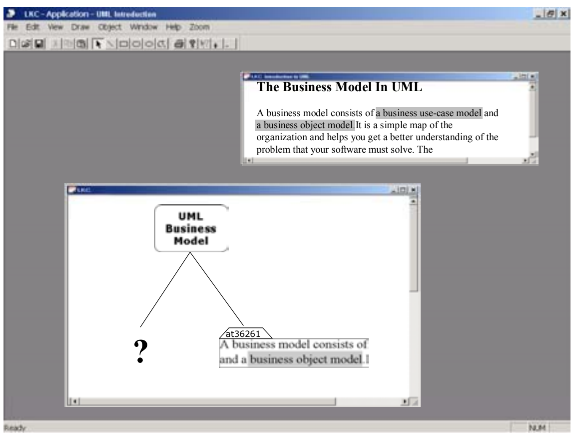


Figure 7 - Student learning interface

The ***Diagnosis Module*** is in charge of analyzing the work of the student and finding out the differences with the reference model. Based on this diagnostic and on the selected strategy, the ***Helper Module*** can give help to the student in different form. It can be text highlight of a single part in the source document to guide the attention of the student (use of microstructures). It can be the highlight of many parts of the text linked to a particular idea (use of macrostructures). Or it can be the recall of similar ET that was successfully performed by this student in the past (use of epistemic network and student model). The ***ET Executor*** is a control component in charge of managing all other components and of starting and ending all ETs.

The result of the student's modeling work is stored in the ***controlled student model***. This is an *open learner model* [Dimitrova, Self and Brna 1999] as it is build by the student. It plays the function of student cognitive profile and has a great importance in pedagogical planning performed by the planner and selector module.

Free Reading Activity

In free reading activity, the student is free to construct his own vision of the read document if he feels that the reference model made by the designer is not sufficient or that he can do better. The student uses the same editing tools as the designer but the result of his modeling activity is stored in another part of the student model, the ***personal student model*** (Figure 6).

However, even if this is a free activity, the system can still give valuable help and feedback to the student. To achieve this goal, the ***system supervision component*** of the LKC system uses his knowledge of the ***collective model*** (a compilation of all free works done by students) and the current context, to find matches between the student construction and similar constructions of other students in the same context. The student can save time by using part of already constructed element in the collective model, or even argument if he disagrees with them using standard argumentation template as those of ***Ar*** model (meta-argumentation). This argumentation and critics from the student, and maybe counter argumentation from the creators, become part of the collective model and will automatically be used by the system in the future to provide feedbacks and critics about student work. It may happen that the students' representations of knowledge be more understandable for peer students than the designer's work, so the periodic analysis of the collective model can be a valuable source of course enhancement.

Conclusion and Future Works

In this paper, we have presented how advanced annotation and external representation are supported in the LKC system. The objective is to facilitate the knowledge construction process in reading activity and enhance student understanding, especially in higher education. With a close track of student's annotation process, this approach aims to overcome the problem of misconceptions at reading stage by their early detection and correction.

The LKC system provides a framework for augmenting a raw document by several knowledge models to support the student at every stage in the reading process, from document annotation to ideas representation and argumentation. The permanent link between knowledge modeled and the raw document keep clear and visible the context in which this knowledge is defined and used. If the system facilitates the student's annotation activity, it requires, on the other hand, a lot of monetization work from the designer. Our next research task will be to enhance the assistance to the designer, especially by providing him with a library of templates in the creation of external representations.

[3] ET = Epistemic task

References

Adler, M., and van Doren, C. 1972. How to Read a Book, Simon and Schuster, New York, NY, *Communications of the ACM*, Vol. 38, N. 4, pages 77-84,

Scalable Vector Graphics, SVG. 2002. Adobe site http://www.adobe.com/svg/

Anderson-Inman, L. and Zeitz, L. 1999. Computer-Based Concept Mapping: A Tool for Negotiating Meaning. *The Computing Teacher*, 26(08).

Ben-Ari, M. 1998. Constructivism in computer science education. Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education, pages 257–261.

Bruner, J. 1973. *Going Beyond the Information Given*. New York: Norton.

Chi, M., Bassok, M., Lewis, M., Reimann, P., and Glaser, R. 1989. *Self-explanation: How students study and use examples in learing to solve problems*, volume 13. Cognitives Science. Pages 145-182.

Cousins, S., Baldonado, M. and Paepcke, A. 2000. *A Systems View of Annotations*, Xerox Palo Alto Research Center, Tech Report P9910022.

Correia, N. and Chambel, T. 1999. Active Video Watching using Annotation, *ACM Multimedia '99 (Part 2)*. Orlando. FL. USA

Cox, R. and Brna, P. 1995. Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education*, 6(2), pages 239–302.

Dijk, T. A. V. and Kintsch, W. 1983. *Strategies of discourse comprehension*. Academic Press, New York.

Dimitrova, V., Self, J., and Brna, P. 1999. The interactive maintenance of open learner models. *Proceedings of the 9th International Conference on Artificial Intelligence in Education*, pages 405–412.

Grossen, G. and Carnine, D. 1990. *Diagramming a logic strategy: Effect on difficult problem types and transfer*, volume 13, pages 168–182. Learning Disability Quarterly.

Inspiration, I. 2002. Inspiration Software. http://www.inspiration.com.

Levy, D., and Marshal, C. 1995. Going Digital: A Look at Assumptions Underlying Digital Libraries, *Communications of the ACM*, Vol. 38, N. 4, page 77-84.

Marshall, C. 1998. The Future of Annotation in a Digital (Paper) World. *The 35th Annual GSLIS Clinic: Successes and Failures of Digital Libraries*, University of Illinois at Urbana-Champaign.

Marshall, C. 1997. Annotation: from paper books to the digital library. *Proceedings of the 2nd ACM international conference on Digital libraries*, pages 131-140. Philadelphia PA, USA

Mindjet. 2002. MindManager Business and Standard Edition. http://www.mindjet.com

O'Hara, K. 1996. *Towards a Typology of Reading Goals*. Technical Report EPC-1996-107, Xerox Palo Alto Research Center.

O'Hara, Kenton and Sellen, Abigail. 1997. A Comparison of Reading Paper and On-Line Documents. In *Proceedings of CHI'97*, page 335-342. New York: ACM Press.

Ohlsson, S. 1996. *Learning to do and learning to understand. in Reimann*, P. and Spada, H. (Eds.), Learning in humans and machines, Pergamon, oxford edition.

Phelps, T. and Wilensky, R. 1997. Multivalent Annotations. In Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, Pisa, Italy.

Rouane, K., Frasson, C. and Kaltenbach, M. 2002. LKC: Learning by Knowledge Construction. *International Conference on Intelligent Tutoring Systems, ITS'2002*. Biarritz-France, San Sebastian-Spain. Springer-Verlag.

Stenning, K. and Oberlander, J. 1995. *A cognitive theory of graphical and linguistic reasoning: Logic and implementation*, volume 19. Cognitive Science. Pages 97-140.

Suthers, D., Weiner, A., Connelly, J., and Paolucci, M. 1995. Belvedere: Engaging students in critical discussion of science and public policy issues. *7th World Conference on Artificial Intelligence in Education (AI-ED 95)*. Washington, DC.

Toulmin, S. 1958. *The uses of argument*. Cambridge, MA: Cambridge University Press.