

What is teaching? Cognitive-based tutoring principles for the design of a learning environment.

Philippe Dessus, Sonia Mandin, Virginie Zampa

▶ **To cite this version:**

Philippe Dessus, Sonia Mandin, Virginie Zampa. What is teaching? Cognitive-based tutoring principles for the design of a learning environment.. International Conference in Human System Learning, May 2008, Toulouse, France. pp.49-55. hal-00308591

HAL Id: hal-00308591
https://hal.science/hal-00308591

Submitted on 31 Jul 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

In S. Tazi & K. Zreik (Eds.). (2008), *Common innovation in e-learning, machine learning and humanoid (ICHSL.6)* (pp. 49-55). Paris : Europia/IEEE.

What is Teaching?

Cognitive-Based Tutoring Principles for the Design of a Learning Environment

Philippe Dessus, Sonia Mandin
Laboratoire des sciences de l'éducation (EA 602) & IUFM
Université Pierre-Mendès-France, BP 47
38040 Grenoble CEDEX 9, France
{Philippe.Dessus, Sonia.Mandin}@upmf-grenoble.fr

Virginie Zampa
LIDILEM (EA 609), UFR des sciences du langage
Université Stendhal Grenoble-III, BP 25
38040 Grenoble CEDEX 9, France
virginie.zampa@gmail.com

***Abstract*—One of the main principles underlying the design of human–system interactions within ITSs or ILEs is that the closer the "artificial" principles are to those involved by human teachers, the more efficient the learning will be. However, the very notion of "human-likeliness" is neither very new nor very precise. We suggest here that these human-like interactions need to be grounded in the very core human social capabilities, notably those allowing mind reading. We address this problem here, first in reviewing the literature about tutoring principles, then in proposing a new classification scheme of these principles. Third, we sketch the first lines of a model and a LSA-based software architecture that attempts to comply with this "human-likeliness" by providing teacher-like advice for learning courses.**

Keywords—Tutoring Principles, Interactive Learning Environments, Computer Architecture, Latent Semantic Analysis.

I. INTRODUCTION

Teaching and learning are causally tightly bound activities, so questioning "what is learning?" might lead to have a closer look about what precisely are the components of teaching—and their underlying principles as well—that can cause efficient learning. Part of research on Intelligent Tutoring Systems (e.g., [1]), and more recently on Interactive Learning Environments (ILEs) or Pedagogical Agents (e.g., [2]) has been devoted to devise tutoring principles for the design of human-like interactions enabling teaching and/or learning. Whatever the functioning of the software to be considered (e.g., providing more or less guidance, being more or less authoritative), the underlying idea of these principles is that learners will more likely attribute mental states as well as human characteristics to a software that provides content to be learned. Likewise, human–computer interaction involved in such environments will more likely resemble interactions between humans. Finally, resulting learning performance will be better. In brief, the closer the "artificial" principles are to those used by human teachers, the more efficient the learning will be.

Since this principle is appealing and appears to be one more at humanizing learner-computer interactions in order to foster learning, it has been seldom investigated per se. First, it is not very new: previous research trends have had this goal as well (note the term "*Intelligent Tutoring* System"). Second, a problem arises with this definition: the very notion of "human-likeliness" is neither very new nor very precise. How does a human-like software actually behave? What precisely are the human features to be replicated in it? If one strictly follows this criterion, most of previous pedagogical environments, even the earlier ones, were designed to develop human-like interactions with their users. On the other side, teachers actually perform computer-like behaviors in many situations (e.g., when they present alternate answers to students or they quickly assess students' answers). We develop here the idea that these human-like interactions need to be grounded in the very core human social capabilities, notably those allowing mind reading.

The remainder of this paper is as follows. We first review the literature about tutoring principles, which will then be reframed using a new classification scheme using very basic social capabilities. Third, we propose a model and an ILE architecture that attempts to comply with this "human-likeliness".

II. TUTORING PRINCIPLES: A REVIEW OF THE LITERATURE

We begin by reporting and analyzing the several prescriptions relying on the design of tutors that have been presented in the literature. These principles are general advice given by researchers in order to design efficient tutoring student-computer interactions (mainly aimed at monitoring, scaffolding and assessing learning activities), and there are two ways to devise them. A *function-to-strategy way* by implementing core functional software modules and signaling what kind of teaching strategy they may carry out (e.g., [1]), and a *strategy-to-function way* by mimicking the different core teaching strategies and signaling how existing tutors may implement each of them (e.g., [3]–[5]). Since other approaches blending these two ways exist [6], literature seldom lies on theories stating that modules or strategies uncovered are both necessary and sufficient to cause teaching and learning.

This work was partly supported by the 7th Framework Programme (Theme FP7-ICT-2007-1-4.1) of the European Community (LTFLL-STREP Project, Grant no 212578).

We present in Table I and detail below some tutoring principles found in the literature. The first two columns detail them and their authors while the next one rephrases the principles according to the categorization we will introduce in the next section.

Ohlsson, in a seminal paper [6], divided the discussion on how to find effective tutoring principles in four parts:

- *cognitive diagnosis* (how to infer the student's cognitive state);
- *subject matter analysis* (how to represent subject matter to be delivered);
- *teaching tactics* (what is the set of instructional actions to choose from);
- *teaching strategies* (what is the most adequate teaching method, regarding the previous questions).

Ohlsson argued that the first two activities lead to generate the input of the system, while the two others produce the output. He presented six classes of teaching tactics encompassing all the teaching activities necessary in the classroom. In another important article, Anderson et al. [1] (see also [7] for an update) formulated eight principles (reframed in into six principles listed in Table I) for the design of cognitive tutors. They reviewed a decade of research and listed how cognitive-centered tutors can be implemented using the different sequels of the ACT theory (ACT* and ACT-R). Finally, Kim and Gil devised fifteen principles related to interactive knowledge acquisition that can be integrated into computerized tutors [3], [4].

The main drawbacks of these models are their lack of higher-level categorization principles, and their pedagogical or system-relatedness. Anderson and his colleagues are learning-centered and refer too precisely to the ACT-R architecture, while others refer to a precise pedagogical theory without actually considering what kind of actions the tutor would perform. In order to address this latter problem, Koedinger and Corbett [7] designed several"meta-design principles" intended to be system- and domain-independent. We do not review them here because they are very open and vague, but the very idea of "meta-design principles" can be kept in mind. We argue, first, that a categorization scheme underlying these different principles is lacking; second, that a few basic ideas are lying beyond the apparently large variety of tutoring principles; third, that the available principles are not purely "pedagogical", since they often are obscured by other variables like the pedagogical approach or the computer system used. We claim it is worth seeking cognitive-centered principles by considering features taken from real-world teaching, or between a teacher and a

TABLE I. A LIST OF INSTRUCTIONAL PRINCIPLES FOR THE DESIGN OF LEARNING ENVIRONMENTS, AS REVIEWED IN THE LITERATURE

Instructional or Learning Principle	Authors	Natural-Cognition Principle Involved
Start by introducing topics and goals	KG [3], [4]	Provide information to be learned (temporal)
Use topics of the lesson as a guide	KG [3], [4]	Provide information to be learned (level of generality)
Subsumption of existing cognitive structure	KG [3], [4]	Theory of mind
Immediate feedback	KG [3], [4]	Teacher's feedback
Generate educated guesses	KG [3], [4]	Intentions reading
Indicate lack of understanding	KG [3], [4]	Knowledge gap detection
Keep on track	KG [3], [4]	Knowledge gap detection
Detect and fix "buggy knowledge"	KG [3], [4]	Knowledge gap detection
Learn deep model	KG [3], [4]	Provide information to be learned (complexity)
Learn domain language	KG [3], [4]	Teacher/Student alignment
Keep track of correct answers	KG [3], [4]	Knowledge gap detection
Prioritize learning tasks	KG [3], [4]	Provide information to be learned (temporal)
Limit the nesting of lessons	KG [3], [4]	Provide information to be learned (complexity)
Summarize what was learned	KG [3], [4]	Feedback
Provide overall assessment of learning knowledge	KG [3], [4]	Feedback
Represent student competence as a production set	KC [7]	Theory of mind
Provide instruction in a problem-solving context	KC [7]	Provide information to be learned (context)
Communicate the goal structure underlying problem solving	KC [7]	Provide information to be learned (context and purpose)
Promote a correct and general understanding of the problem-solving knowledge	KC [7]	Causal consequence of teaching
Minimize working memory load that is extraneous to learning	KC [7]	Provide information to be learned (complexity)
Provide immediate feedback on errors relative to the model of desired performance	KC [7]	Feedback
Tactics for presenting the target	O [6]	Provide information to be learned
Tactics for presenting precursors	O [6]	Provide information to be learned (pre-requisites)
Tactics for presenting purposes	O [6]	Provide information to be learned (purposes)
Tactics for presenting justifications	O [6]	Feedback
Tactics related to errors	O [6]	Feedback
Tactics related to students' solutions	O [6]	Feedback

Note: KG (Kim & Gil); KC (Koedinger & Corbett); O (Ohlsson).

pupil. To this end, a more basic and "natural" view of the teaching/learning activity is worth considering.

III. Recategorizing Tutoring Principles in using a Natural Cognition Viewpoint

Some scholars (e.g., [8]–[10]) have performed a "back to basics" reconsideration of teaching and instructional moves. After having noticed that even children from 3.5 years or non-schooled persons can teach, they considered this activity lies to "natural cognition" ones (i.e., a universal competence acquired by young children, involving largely invisible complex skills). They detailed the natural bases of teaching, by defining them as "the intentional passing on of information from one who knows more to on who knows less" ([9], p. 371). This latter author listed some cognitive prerequisites of teaching:

- *monitoring the others' mind* (inferring emotions, beliefs, knowledge);
- *having representations of two levels of knowledge to be taught* (i.e., correct knowledge, possible incorrect student's knowledge), as well as having the ability to detect gaps between these two levels of knowledge;
- *having the ability to communicate knowledge to student* (with respect to some important characteristics like its difficulty, level of generality, temporal features);
- *having the ability to provide feedback* (i.e., assessment, corrections) to student.

These four elements would be necessary and sufficient for teaching. This short list of actions seems sufficient to rephrase and recategorize all the tutoring principles we presented, in order to make them fit with the necessary and sufficient abilities for teaching listed above. It is worth noticing that these abilities are not termed in actual cognitive processes, and a further step is necessary for that latter task. To that end, we borrow from Baron-Cohen [11] his cognitive model of mind reading, composed of four independent modules that help human beings reading others' mind, whose description follows.

- *The ID (Intentionality Detector) module*, which collects perceptual stimuli concerning self-propulsion or direction to infer representations of desire or goals.
- *The EDD (Eye Direction Detector) module*, which collects eye-direction stimuli to infer what the object considered by an agent is (e.g., the student).
- *The SAM (Shared Attention Mechanism) module*, which uses the information provided by the two previous modules in order to infer triadic representations (i.e., joint attention behaviours between two agents and an object), whereas the two previous modules infer dyadic representations (agent-to-object relations); for instance, this module could analyse how two agents are engaged in a mutually shared attention transaction about an object of knowledge.
- *The ToMM (Theory of Mind Mechanism) module*, whose role is to integrate the information of the SAM module in order to infer mental states as well as knowledge from the behaviours of others.

A fifth module, which does not pertain to Baron-Cohen's model, is necessary to manage feedback and assessment to student. To this end, we use an instructional model of feedback [12] which details the main three questions providing feedback to the learner:

- *where am I going?* ("what are the learning goals?");
- *how am I going?* ("what progress is being made toward these goals?");
- *where to next?* ("What activities need to be undertaken to make better progress?").

So we have now at hand all the components necessary to recategorize the tutoring principles found in the literature. The third column of Table I reframes all of these principles by using one of the "natural cognition" principles. It is noteworthy that these principles now fit into a minimal set of social-centered teaching capabilities, those engaged in mind reading and knowledge assessment. We have now to choose a computer-based technique for processing all this information. Latent Semantic Analysis appears to be a good candidate for the reasons exposed in the next section.

IV. Student Modelling with LSA

We detail now a typical student–teacher interaction in an instructional context (e.g., distance learning): a student submits a query (to the teacher or to the system itself) and reads some of the retrieved texts. When s/he thinks that sufficient knowledge has been acquired, s/he can write out a summary, which will be assessed (again, by the teacher or the system). This assessment can take into account both the very quality of the written production and the types of texts read. It is worth noting that these several interactions generate a lot of different "texts": the read texts, the written texts, the student's moves throughout the texts. All these texts have successively to be compared semantically, in order to process both inputs (students' intentions and goals) and outputs for the student (text selection, assessments).

Since our goal is to design a fully-automated system, we need a technique for comparing texts semantically. Latent Semantic Analysis [13], a machine-learning approach of knowledge representation and acquisition that allows the semantic comparison of texts, can be used both to assess forms of knowledge and intentions from moves. On one hand, a research [14] showed that a LSA-based technique can adequately predict student learning from written free texts by inferring their prior knowledge level. This technique can uncover students' "zones of learnability" that are just enough distant from their prior knowledge, without being too far. On the other hand, scholars [15] showed that LSA can be used to capture the goals of operators in a complex environment by comparing their moves to one other.

It is worth noting that we do not claim that LSA can read human minds, thus there is no direct functional matching between human modules presented in section III and their respective artificial equivalents. However, our goal is to

reproduce this overall human architecture whereby LSA will provide the several semantic processing described above. We will detail in the next section a computer architecture in which the underlying principles of these modules are used.

V. Description of the Architecture

Let us present a very rough description of a session where a student is connected to our ILE (see Figures 1 and 2). First, s/he performs a first query in order to get some course texts to read. At any time of the reading, the student can write out a summary of what s/he understood, and/or can ask for next texts to read. Moreover, the student can reword the query in order to get texts about a slightly different subject content. All the actions of the student (i.e., text reading, summary writing) are tracked to update its model. The documents read or to be proposed have their titles visually organized in a concept map graphically representing the different knowledge states of the student (i.e., what was read and understood, what s/he will intend to read).

We focused here on student's summary production because this activity first accounts for comprehension [16]: the better understood a text is, the better it will be summarized. Second,

TABLE II. Three Viewpoints about our Environment

Questions	Cognitive Viewpoint	User Session	Computational Processing
1. What is the initial state of knowledge of the student? What is the possible gap between this state and the course knowledge the student plans to learn?	*T* has the ability to infer *S*'s initial knowledge, whereas *S* can explain to *T* what he or she intends to learn.	*S* triggers the display of the first set of course texts to be read by performing a query which retrieves the *n* closest course texts to the query.	The processing by LSA of a large amount of general texts serves to set up the initial *S*'s knowledge model. Use the first step of *Apex* to perform the query.
2.1 Strategic moves analysis	*T* infers *S*' intentions from what *S* reads (i.e., pages accessed, or strategic information).	A graph representing all the texts read so far is displayed.	The student model is updated upon texts read and summarized. The text has a higher weight in the model if it has been summarized.
2.2 Shared attention and ToM	*T* infers what *S* knows or should know from what *S* has read (see step I) and S's summarized (epistemic information).		
3. Feedback. Some *T* strategies to improve *S* learning.	*T* involves some strategies in order to help *S*. – from where am I going? – how am I going? – where to next?	More texts are provided to *S* in order to reduce the possible gap of knowledge. The concept map is updated.	1. If the text read was not summarized or the summary covered the most important course sentences, use the ZPD-scheme to select sufficiently different course texts from the previous but not too much. 2. If the summary does not cover the most important sentences of the course, a set of closest texts from the most important sentences which were not covered in the summary is selected.

Note: *T*, Teacher; *S*, Student, *E*, Environment.

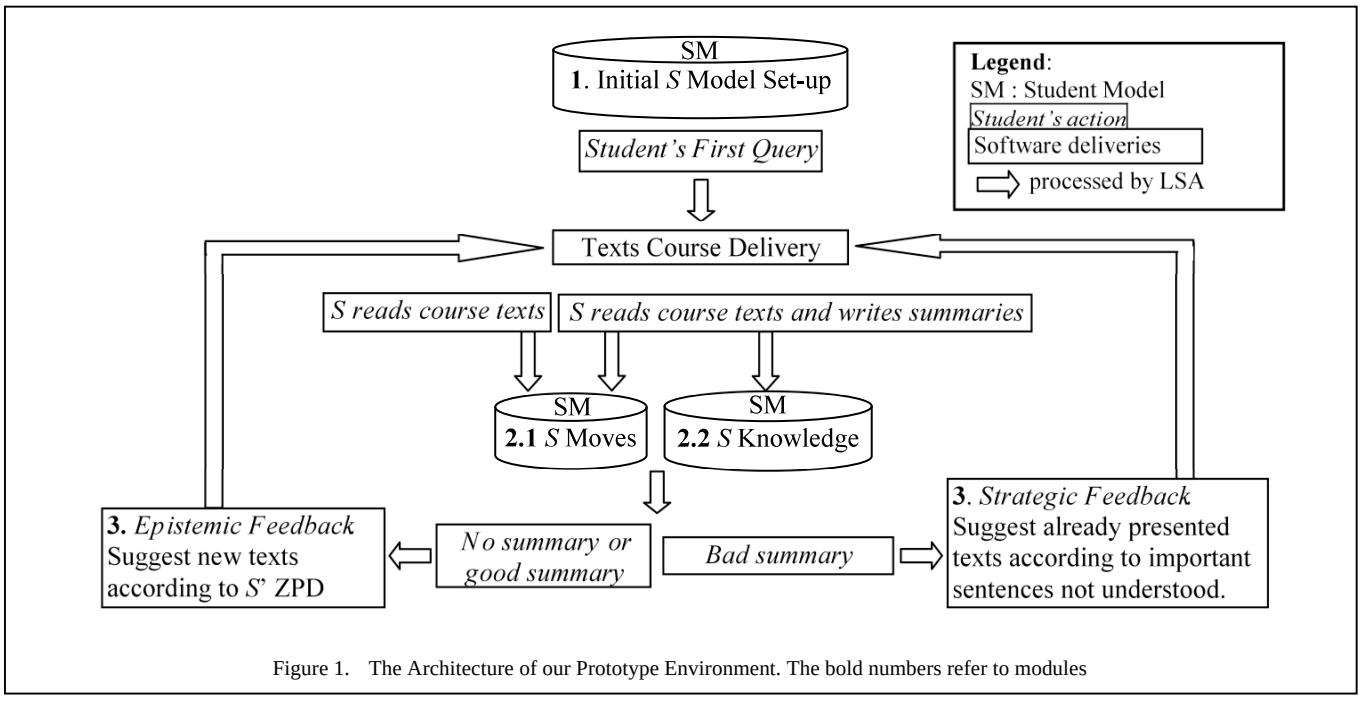


Figure 1. The Architecture of our Prototype Environment. The bold numbers refer to modules

summarizing a text fosters its understanding [17], [18]. Moreover, in an instructional context, the summaries can be used later by the student as a revising tool for exams. Analysing such summary production allows to sidestep low-level comprehension assessments like multiple choice questionnaires.

We adapted the human modules described in the section III to make them fit in a computer environment. This adaptation entails to design a model for each of the modules beforehand and to simulate them by using a computer-based technique. For instance, if we have to model the way students select the most important sentences in a text, the alternate models would be as follows: The student selects either 1. the first and last sentences of each paragraph; 2. or the key-sentence of each paragraph; 3. or, for each paragraph, the closest sentence to the others ones [19]. All these models can be implemented and run in order to keep the model which behaves the closest like humans.

The first two "human" modules can be joined into a single module that processes "perceptual" information by tracking and collecting the student's moves within the environment. The next two (SAM and ToMM) are blended again in order to manage higher-level knowledge: intentional inferences about strategic and epistemic moves. The feedback module is kept as such. We argue that these three modules are necessary and sufficient to produce human-like interaction in an instructional context. We detail in Table II the role of each of them, leading to describe the instructional interactions from three viewpoints: first, the main cognitive operations by which teachers and pupils interact and whereby teaching and learning are enabled; second, the description of a generic user session; third, the underlying computational operations. We now detail how the three modules gather and analyse data within our environment (see architecture in Figure 1 and interface in Figure 2).

A. Initial Student Knowledge Model (Module 1)

This first module allows two tasks. First, the task to set up the student model by processing a large corpus of general texts (e.g., newspaper, encyclopaedia entries). This enables a very simple student model to init, while using more specialised texts would need more sophisticated student knowledge assessment. Second, the task to perform a first natural language query (see upper left field of Figure 2) in order to retrieve the first set of texts to be read by the student, which are displayed in the Course Texts Area (see Figure 2). This latter task is performed by LSA, which successively compares the query with each of the course texts and proposes the *n* closests to the student in the lower pane [20].

B. Student's Moves and Knowledge Acquisition Modelling

Analysing student's moves within the environment (Module 2.1). This low-level module collects the student's moves (i.e., strategic information, or all the course pages that were accessed and read by the student) and presents them in the form of a concept map (see Figure 2).

Modelling knowledge acquisition by shared attention and ToM inferences (Module 2.2). This module plays the roles of both SAM and ToMM previous modules: it incorporates the information concerning perception (who sees/reads/does what). Its job is to track what knowledge is subject to attention by the student. As a result, the previous concept map is updated in adding what was understood (epistemic information) to the main topic of what was read (strategic information). This module updates the student model with data about both course texts read and summarized. The student model is updated by a

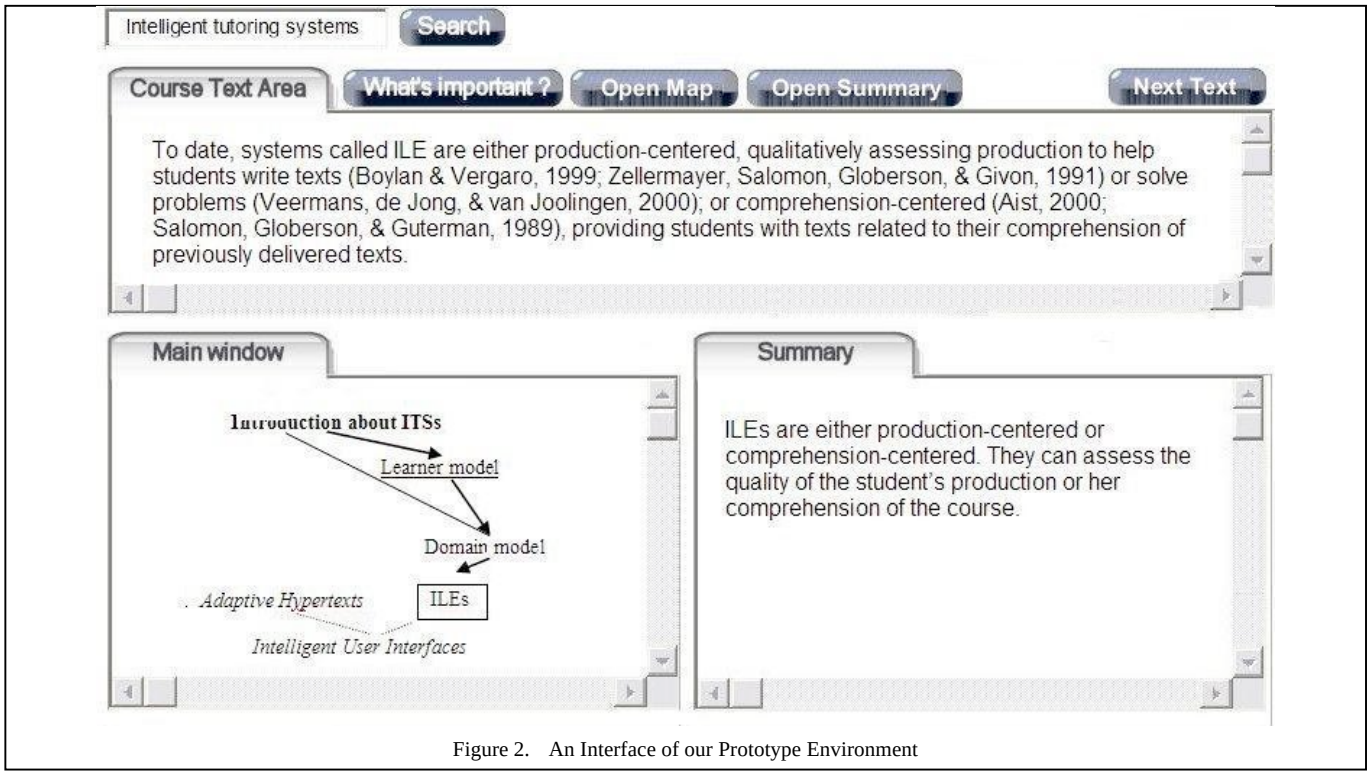


Figure 2. An Interface of our Prototype Environment

LSA-processing of texts read or summarized, among two conditions: in the read-only condition texts read are single-weight compiled while in the read-and-summary condition both texts read and summaries are double-weight compiled (i.e., simply processed two times). In so doing, our goal is to promote student's understanding (i.e., rewording) rather than merely text repeating: what is read and understood by a summarization counts as two times (arbitrary value) what is only read.

C. Providing Feedback to the Student (Module 3)

This module takes as input the two previous modules in order to provide feedback to student, like those a teacher would perform. Both strategic and epistemic feedback depend on the availability and quality of student's summaries. Two alternate cases arise.

1. If no summary exists or the student's summary covers the most important course sentences, the course is considered understood. In those cases, a ZPD scheme (Zone of Proximal Development [21], closely akin to Wolfe et al.'s "zone of learnability" [14]) is used to select the next course texts the student may read. So the problem is to know which texts must be chosen for the student to learn. Since the student could consider the closest text from the current student's model the easiest, it is probably not suited for learning because being too close to the student's knowledge. The farest text from the student's model could be in turn considered too hard and will certainly be not understood. For instance, if 10 year-old children are provided with texts made for 6 year-old children, they will probably not learn much. In the same way, they will not learn much if given a text from Freud. So, idea is to select the closest text among those that are far enough. An experiment, with four different semantic distances for selection, showed that a distance equal to a standard deviation of the closest text pulls a better learning than closest text, farest text and text chosen randomly [22].

2. If the student's summary does not cover the most important course sentences, a set of closest texts is selected from the most important sentences which were not covered in the summary, enabling a more extensive reading. These sentences are selected by successively comparing with LSA each summary sentence to the whole course text [23]: the closer the sentences are, the best they rely on student's understanding [19].

In both cases the concept map is updated accordingly, and the most important changes are highlighted (see Figure 2). For instance, well-understood topics are in bold font, topics to be rehearsed are underlined, topics not covered so far are in italics, and the ongoing topic is framed. Each node can be activated in order to access to the related topic, so the student has a view of what is to be actually read in order to go further within the course. Arrows represent the actual student's reading path while dashed lines represent the suggested texts to read. Moreover, in the course text area (Figure 2), the most important sentences can be underlined.

D. Implementation Paths

By now, since each of these modules but one has been separately implemented, their integration into a single learning environment remains to be performed. The module 1 was implemented into *Apex 2* [20], the module 2.1 remains to be implemented by the way of a concept map representation of the course documents, the module 2.2 is part of a tutor architecture for assessing summaries [19], while the module 3 borrows its ZPD management scheme from the RAFALES tutor [22], [24].

VI. Conclusion

Since most of the tutoring principles used in the literature are unintentionally and implicitly used [4], we suggested here that a more cognitive and explicit view of these principles is possible. First, we presented a new classifying scheme of tutoring principles, inspired by the literature on teaching as a natural cognition. Second, we sketched an ILE architecture aimed at applying some of these principles in a non-constraining way for the student. The text summaries are not compulsory since the student who understands the course can view new texts without involving a summarization process. Moreover, even the course texts delivered to the student do not have to be extensively read since the student can formulate new search requests. The guidance and feedback provided by the environment are thus neither too tight nor too weak: they help the student detecting the most important sentences of each course text and they diagnose what the next texts to be read are, according to an adaptive student model.

Our approach merely focuses to a back-to-basics reconsideration of what the features of teaching are: capturing strategic and epistemic cues from student's reading and writing, generating appropriate feedback. Research on Computer-Automated Essay Assessment has been frequently criticized for lacking "human-likeliness" features [25]; we here attempted to answer the difficult question of what kind of features human-like agents should have. As Balacheff [26] pointed out: "The educating function of a system is an emerging property of the interactions organised between its components, and not a functionality of one of its parts". The aim of this paper was both to define the core functionalities of a tutoring system, and to explain how its "educating function" can emerge from them.

Acknowledgment

We thank Benoît Lemaire for thoughtful comments of a previous version of this paper, as well as Carol Bevan, Ceri Bevan and Frédéric Occédat for checking the English.

References

[1] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: lessons learned", *The Journal of the Learning Sciences*, vol. 4, no 2, pp. 167-207, 1995.

[2] R. Moreno, R. E. Mayer, H. A. Spires, and J. C. Lester, "The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents?", *Cognition and Instruction*, vol. 19, no 2, pp. 177-213, 2001.

[3] J. Kim, and Y. Gil, "Deriving acquisition principles from tutoring principles", in *Proc. ITS 2002 Conf.* Berlin: Springer, 2002, pp. 661-670.

[4] J. Kim, and Y. Gil, "Incorporating tutoring principles into interactive knowledge acquisition", *International Journal of Human-Computer Studies*, vol. 65, no 10, pp. 852-872, 2007.

[5] R. C. Schank, and M. Y. Jona, "Empowering the student: New perspectives on the design of teaching systems", *The Journal of the Learning Sciences*, vol. 1, no 1, pp. 7-35, 1991.

[6] S. Ohlsson, "Some principles of intelligent tutoring", *Instructional Science*, vol. 14, pp. 293-326, 1986.

[7] K. R. Koedinger, and A. Corbett, "Cognitive tutors. Technology bringing learning sciences to the classroom", in *The Cambridge Handbook of the Learning Sciences*, R. K. Sawyer, Ed. Cambridge: Cambridge University Press, 2006, pp. 61-77.

[8] G. Csibra, and G. Gergely, "Social learning and social cognition: The case for pedagogy", in *Processes of change in brain and cognitive development*, Y. Munakata and M. H. Johnson, Eds. Oxford: Oxford University Press, 2006, pp. 249-274.

[9] S. Strauss, "Teaching as a natural cognitive ability: Implications for classroom practice and teacher education", in *Developmental psychology and social change*, D. Pillemer and S. White, Eds. New York: Cambridge University Press, 2005, pp. 368-388.

[10] S. Strauss, M. Ziv, and A. Stein, "Teaching as a natural cognition and its relations to preschoolers' developing theory of mind", *Cognitive Development*, vol. 17, pp. 1473-1487, 2002.

[11] S. Baron-Cohen, "How to build a baby that can read minds: Cognitive mechanisms in mindreading", *Current Psychology of Cognition*, vol. 13, no 5, pp. 513-552, 1994.

[12] J. Hattie, and H. Timperley, "The power of feedback", *Review of Educational Research*, vol. 77, no 1, pp. 81-112, 2007.

[13] T. K. Landauer, and S. T. Dumais, "A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge", *Psychological Review*, vol. 104, no 2, pp. 211-240, 1997.

[14] M. B. W. Wolfe, M. E. Schreiner, B. Rehder, D. Laham, P. Foltz, W. Kintsch, and T. K. Landauer, "Learning from text: Matching readers and texts by Latent Semantic Analysis", *Discourse Processes*, vol. 25, no 2/3, pp. 309-336, 1998.

[15] J. Quesada, W. Kintsch, and E. Gomez, "A computational theory of complex problem solving using the vector space model (part II)", in *Proc. Cognitive Res. with Microworlds Conf.*, Grenade, Spain, 2001.

[16] T. A. van Dijk, and W. Kintsch, *Strategies of discourse comprehension*. London: Academic Press, 1983.

[17] T. W. Bean, and F. L. Steenwyk, "The effect of three forms of summarization instruction on six graders' summary writing and comprehension", *Journal of Reading Behavior*, vol. 6, no 4, pp. 297-306, 1984.

[18] A. S. Palincsar, and A. L. Brown, "Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities", *Cognition and Instruction*, vol. 1, no 2, pp. 117-175, 1984.

[19] B. Lemaire, S. Mandin, P. Dessus, and G. Denhière, "Computational cognitive models of summarization assessment skills", in *Proc. 27th Annual Conference of the Cognitive Science Society*, B. G. Bara, L. Barsalou, and M. Bucciarelli, Eds. Mahwah: Erlbaum, 2005, pp. 1266-1271.

[20] P. Dessus, and B. Lemaire, "Using production to assess learning: An ILE that fosters Self-Regulated Learning", in *Proc. ITS 2002 Conf.* Berlin: Springer, 2002, pp. 772-781.

[21] L. S. Vygotsky, *Mind in Society*. Cambridge: Harvard University Press, 1978.

[22] V. Zampa, and B. Lemaire, "An intelligent tutor uses LSA for modeling ZDP", *The EARLI Biannual Conf.*, Nicosia, Cyprus, aug. 23-27, 2005.

[23] W. Kintsch, *Comprehension, a paradigm for cognition*. Cambridge: Cambridge University Press, 1998.

[24] V. Zampa, and B. Lemaire, "Latent Semantic Analysis for user modeling", *Journal of Intelligent Information Systems*, vol. 18, no 1, pp. 15-30, 2002.

[25] S. Dikli, "An overview of automated scoring of essays", *The Journal of Technology, Learning, and Assessment*, vol. 5, no 1, pp. 1-35, 2006.

[26] N. Balacheff, "Teaching, an emergent property of eLearning environments", in *Proc. IST'2000 Conf.*, Nice, France, 2000.