Artificial Intelligence and Neural Network Based Tools for Cooperative Learning

Paul Dan Cristea[1] & Adina Magda Florea[2]

[1] "Politehinca" University of Bucharest, Department of Engineering Sciences
Splaiul Independentei No. 313, Sector 2, Bucharest 77207, Romania, http://www.dsp.pub.ro/info/staff/pcristea.htm
[2] "Politehinca" University of Bucharest, Department of Computer Science
Splaiul Independentei No. 313, Sector 2, Bucharest 77207, Romania, http://apolo.cs.pub.ro/~adina

Abstract

The paper presents a cooperative distance learning system based on the emerging paradigm of intelligent human-computer interaction in which the group of learners are assisted by artificial agents with active role in the learning process. The tutor in the system may be a human or an artificial agent. The system offers several learning modalities that combine the traditional style of tutorial learning with the "problem-based" approach. Cooperative learning is achieved either by interaction between the student and the tutor or inside the group of learners. The underlying paradigm to support learning activities is computer-supported cooperative work.

Key words: Distance education, Agent architecture, Neural networks

Introduction

The challenges that face today and tomorrow's Information Society, make the professional qualification a variable and not a life-long constant for most individuals. To maintain the aptitude of the labour force in a competitive and ever changing environment, complex skills have to be transmitted and acquired fast. Efficiency becomes the basic request of the learning process. One way of achieving it is by combining the traditional style of tutorial learning with the "learner-centred", "problem-based" or "creative learning" approach [1]. The New Information Technologies offer the possibility to develop novel applications and learning environments that support such an enriched learning style. Computer networks allow learners to collaborate in problem solving and in developing group reasoning skills, by new means of communication and access to information.

The paper presents a cooperative distance learning system based on the emerging paradigm of intelligent human-computer interaction in which the group of learners are assisted by artificial agents [2, 3] with active role in the learning process. The system comprises a set of tools to assist the learner at several levels of the knowledge acquisition process. Providing these tools with machine intelligence can bring a new dimension to the flexibility and adaptability required to actively support the learner. It becomes possible to develop a personalised model of the trainee, to help understand the connection between theory and practice, to overcome the barrier between knowing and using, and to relieve the learner of cumbersome and routine tasks.

Cooperative learning is achieved by interaction either between the student and the tutor/expert or inside the group of learners. The tutor may be a human or an artificial agent. The learning environment is a virtual one, in which participants may be located in the same place or geographically far apart, communicating over the network. The underlying paradigm is the computer-supported cooperative work (CSCW) that refers to people working together on a product, research area, or scholarly endeavour with help of computers [4].

Learning modalities

The system offers several learning modalities that try to combine the traditional style of teaching with the "problem-based" one, namely: learning by being told, problem solving demonstration, problem solution analysis, problem solving, and creative learning (Figure 1). The *learning by being told* modality corresponds to the on-line presentation of "classical" learning materials, such as course notes or slide-show, but augmented with student-system interactions and feed-back, to enhance the pedagogical value of the environment [5]. The material is organised along two conceptual paths: the content path, which can be explored at compulsory, optional, or advanced presentation level, and the concept path in which the student is following the presentation and the use of a concept throughout the entire learning material. Roadmaps are associated to each of the conceptual paths the student may follow, to facilitate the process of knowledge organisation and concept formation. When following the content path, the student is presented with some questions or simple exercises, at the end of each chapter or section, to test his/her progress in understanding the presented knowledge. According to the quality of the answers, the student is advised to go again through parts of the material or is provided with more examples to clarify the subject.

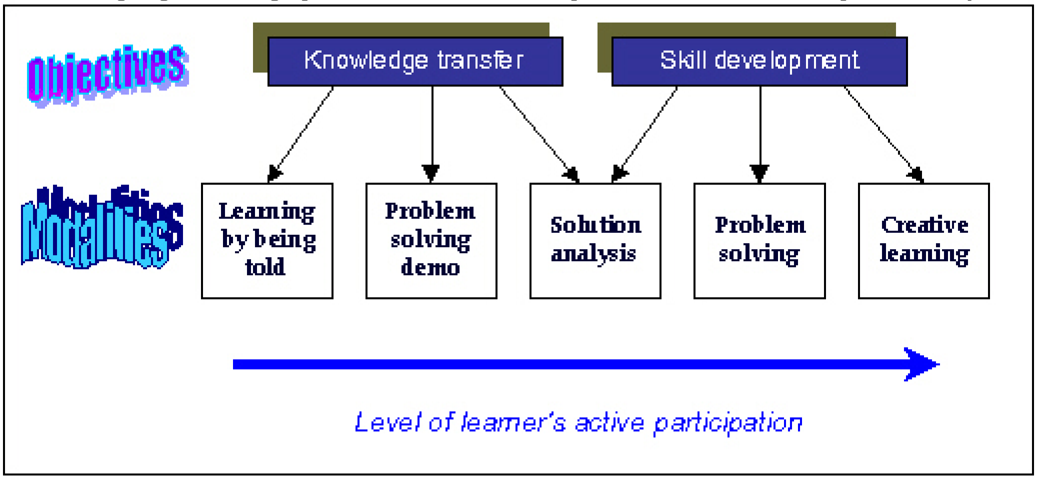

Figure 1. Learning modalities in the system

The *problem solving demonstration* modality implies showing how one given problem is solved individually or collaboratively. The demonstration may be performed by the tutor (human or artificial), by a student, or by a group of students assisted or not by the tutor. This mode corresponds to taking a pre-existing solution for a given type of problem, tuning the solution by specifying parameters or features for the problem instance, applying the solution to solve the problem, and visualising the results. Result visualisation may be followed by the interpretation of results. The interpretation of results is performed through the chat facility of the system. The artificial tutor may also give only a limited interpretation, depending on the problem. The students and the tutor collaboratively participate in setting the parameters and interpreting the results. Shared visualisation of initial parameter settings and of the results which are obtained facilitates collaboration in learning.

The *problem solution analysis* modality corresponds to the analysis of a solution proposed either by the tutor or by a student. This involves the critical analysis of the decisions that were taken and explaining the rationale behind these decisions. Currently, the proposed solutions may be selected from an existing repertoire of generic problem solutions tailored according to tutor or student options. This third modality comprises problem centred demonstration followed by the collaborative analysis and explanation of the proposed solution and of the obtained results. The problem is presented in terms of: problem specification, main steps that lead to the solution, selected input values, results, the set of underlying principles used for solving the problem, and the set of concepts useful for understanding the problem solution. The students have to select the principles that are actually relevant for solving the problem and the concepts that support the problem solving. The human or artificial tutor may be addressed in case of uncertainty by the hint or explanation facilities of this learning modality. To support the analysis, the tutor or the learners in the system may invoke a search facility to perform search of relevant information.

The *problem solving* modality implies the student being in control and solving the problem either individually, or in cooperation with the tutor, and/or with other students. The student actually builds the solution of the problem by selecting and instantiating the problem solving steps from an existing repertoire of solution chunks associated to each problem prototype in the system. The *creative problem solving* is a learning modality in which the student builds the solution from scratch, discovers new ways to solve a problem, new explanations for a problem solution, and/or new causal links between concepts in the system and problem solution. The first three modalities are mainly aimed to achieve knowledge transfer, while the last two are conceived primarily for skill development. The level of learner's active participation to problem solving increases gradually from the first to the last modality, as depicted in Figure 1.

The human tutor may select demonstrations, analyses, problem solving or creative learning experiences built by the students or by himself with the help of the students and record them in the system. In this way, a learner may be individually engaged in a learning modality of his choice or may investigate previous learning experiences performed by the tutor with peer learners, at a time of his convenience.

System architecture

The architecture of the system is a multi-agent one, human and artificial agents collaborating with each other to achieve the learning task. There are several agents in the system, as shown in Figure 2. A learner in the environment is endowed with his own digital Personal Agent (*PerA*). The agent is permanently present in the system to assist the learner, to monitor learner actions, and to assure coordination and communication with the other agents in the system. At the same time, *PerA* is responsible for developing the learner profile. Each *PerA* has an associated KB to store specific knowledge on the learner he serves. The tutor has also an associated digital assistant, namely the Tutor Assistant Agent (*TAssistA*). The *TAssistA* performs mainly the same actions as a *PerA*, but with different aims, as will be detailed in the next section. It is also responsible for eliciting knowledge based on tutor actions while interacting with the system.

The system comprises an artificial tutor, namely the Tutor Agent (*TutA*) that tries to partially replace the human one by providing help and (limited) assistance to learners while they are not assisted by the human tutor. The *TutA* has its own KB in which it stores access knowledge to learning experiences, pedagogical knowledge to guide the learner and knowledge on how to adjust the guiding to the learner profile.

Another agent in the system is the Information Agent (*InfoA*) which is responsible of retrieving and filtering information from specified sources that can range from the learning materials available in the system to the entire Web. The *InfoA* is called either by the *PerA* of a student or by the *TAssistA* with a filtering criteria specified by user. The *InfoA* is using term frequency times inverse document frequency (TFIDF) to perform information retrieving. Extra details on how the *InfoA* is built are outside the scope of this paper.

The Problem Solving Knowledge Base (PSKB) contains the learning materials and experiences necessary to support the learning modalities described in the previous section. Except for the tutorial material used in the learning by being told modality, all knowledge is organised around problem prototypes with associated specifications, generic solutions, and problem solution chunks. A set of underlying principles and supporting concepts is maintained in the PSKB so as to index problem solution analysis. The concepts in the KB are the same with those used to browse the tutorial material along the concept path, in the learning by being told modality.

The interaction among the users (learners and human tutor) of the system may be achieved either by direct communication through the network (chat-like facility) or by means of the associated Personal Agents or the Assistant Agent. The users are communicating with their own agents by means of a menu-driven approach or by a standard formal language for specific requests. The learners have access to the PSKB only to retrieve learning materials and only by means of their *PerA*s. The human tutor may also use its *TAssistA* to access this knowledge base, but has also direct access to the PSKB to create or update its content, or to add learning experiences produced by students while using the creative learning modality.

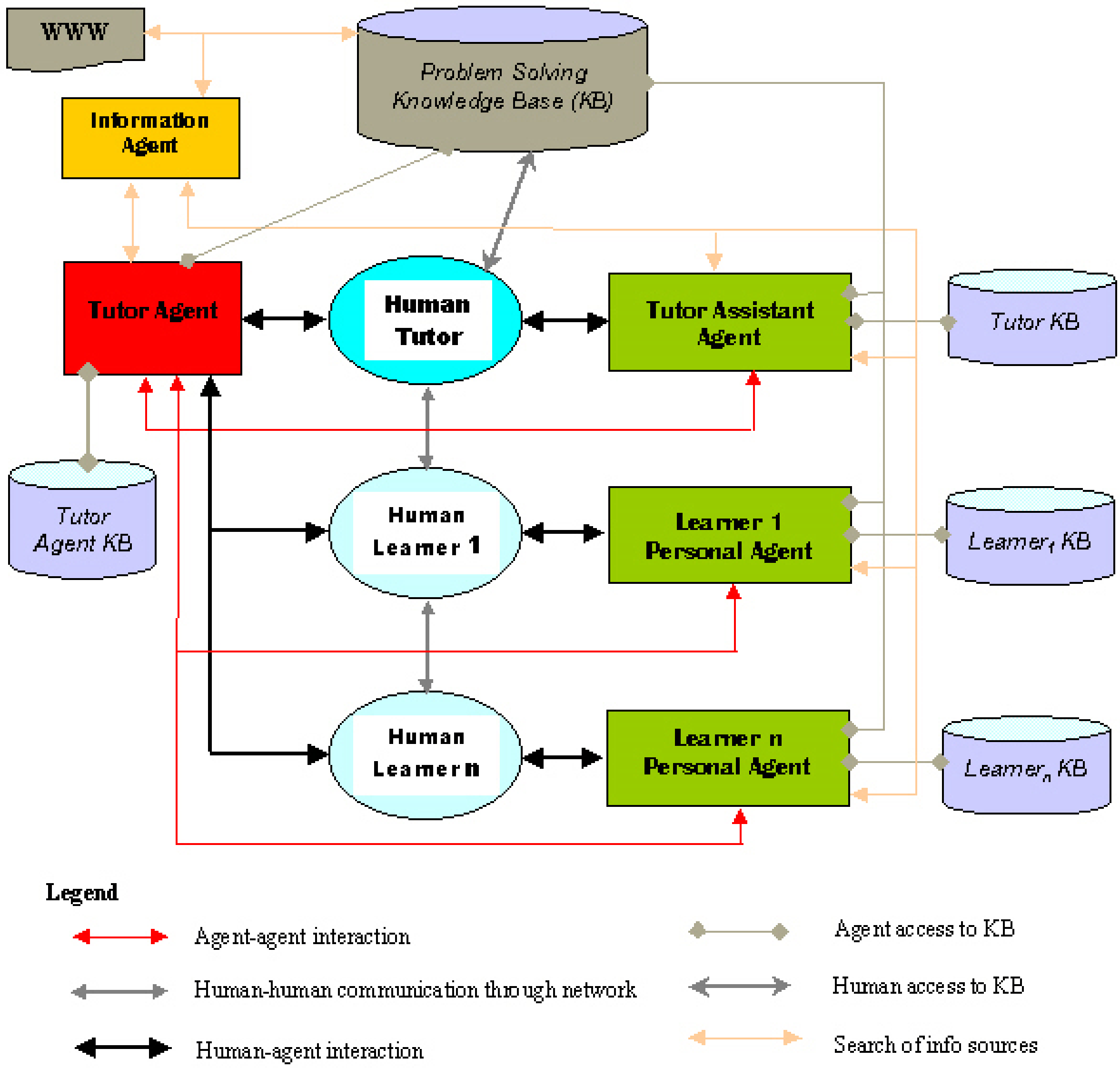

Figure 2. The architecture of the cooperative learning system

Agent specification

Tutor agent

The KB of the Tutor Agent contains: (a) knowledge to access the PSKB -- i.e., the model of the domain, the domain taxonomy and a description of problem prototypes; (b) methodological knowledge -- teaching strategies and modalities, and problem solving methods; (c) knowledge on how to adapt guiding to learner's profile ---- a standard model of the learner needs and a customised model. *TutA* has several components (Figure 3) to achieve the following actions: manage the *TutA* KB based on (a), select from PSKB relevant information to help the student based on (b), assist the learner according to the learner's profile, either standard or customised as transmitted by a *PerA*, based on (c). To perform its functions, the *TutA* interacts with the human tutor -- who can directly update the *TutA* KB, with the learners' *PerA* -- to get learners demands, and with *TAssistA* -- to get customised learners' profiles, when prompted by the tutor, or to get knowledge extracted by the *TAssistA* from the human tutor -- to update its KB.

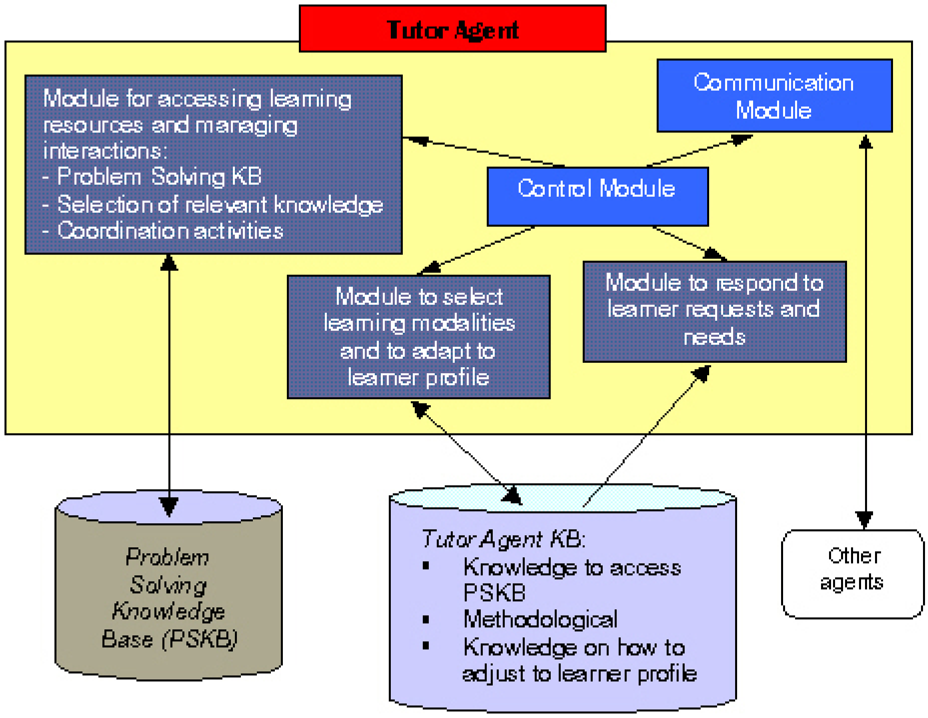

Figure 3. Tutor Agent architecture and functionality

Tutor assistant agent

The Tutor Assistant Agent (Figure 4) monitors tutor actions and helps him in managing the PSKB to extract the learning experiences related to a specified learning modality. Based on the tutor actions while interacting with the students in the system, the *TAssistA* creates a training history, extracts training and pedagogical knowledge, and stores it in its KB. Knowledge contained in large sets of examples can be first extracted in sub-symbolic form by using the examples to train an artificial neural network (ANN), then elicited in symbolic form, usually as crisp or fuzzy rules, by means of specific Knowledge Eliciting (KE) methods [6, 7, 8]. When explicitly asked for, *TAssistA* can also retrieve a learner's profile for the tutor and/or for the *TutA*, by making a request for this profile to the *PerA* of the specified learner. To this end, the *TAssistA* enters in interaction with the human to monitor him and obtain his request, with the *TutA* to send the selected learning profile, and with the *PerA*s to get assistance requests and profiles. It also accesses and updates its own KB and updates the PSKB with elicited knowledge from the tutor, if instructed accordingly.

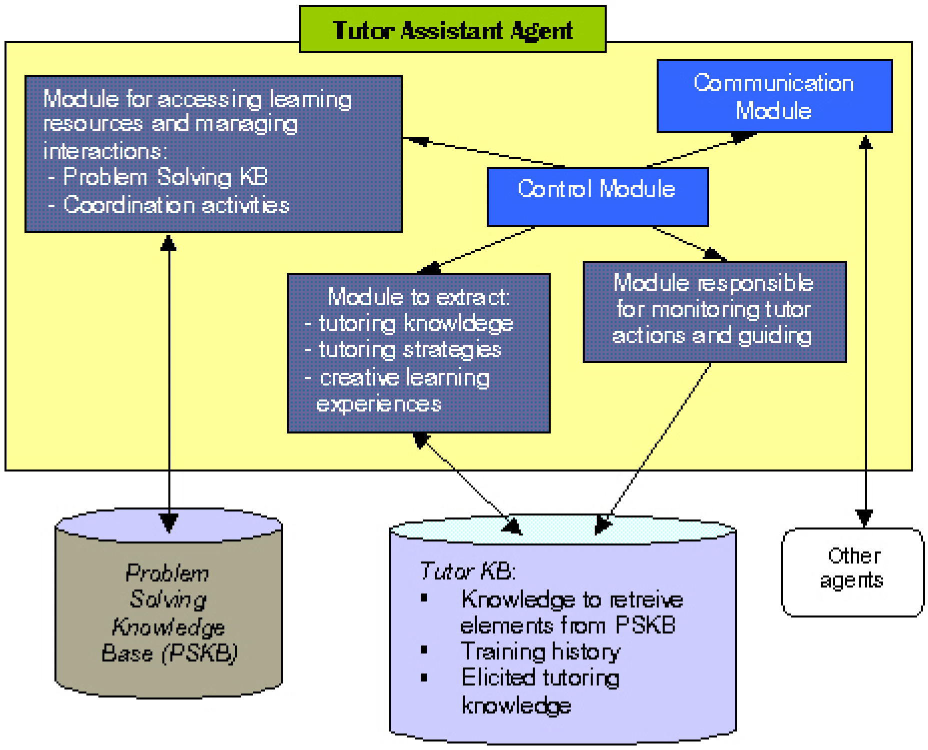

Figure 4. Tutor Assistant Agent architecture and functionality

Learner personal agent

The *Learner Personal Agent* (Figure 5) manages the access to the PSKB to retrieve for the student learning materials and experiences and monitors learner actions. Based on this monitoring, it creates the learning history and, based on this history, develops the learner profile. The learner's history includes the numbers, dates, and modes (synchronous or asynchronous) of the student sessions with the system, the level of achievement, the selected modalities of learning during each session, peer learners in the group while engaged in collaborative learning, and how often the learner asked for assistance from the artificial tutor. Many other factual details of the learning sessions can be recorded in a learner's history, if found useful to define relevant features of the learner. The learning profile of a student is a qualitative synthesis of these quantitative elements, as indicated in the next section. The *PerA* enters in interaction with the learner it monitors -- to get help requests that it passes over either to the tutor, or to the *TutA* -- and accesses and updates its KB.

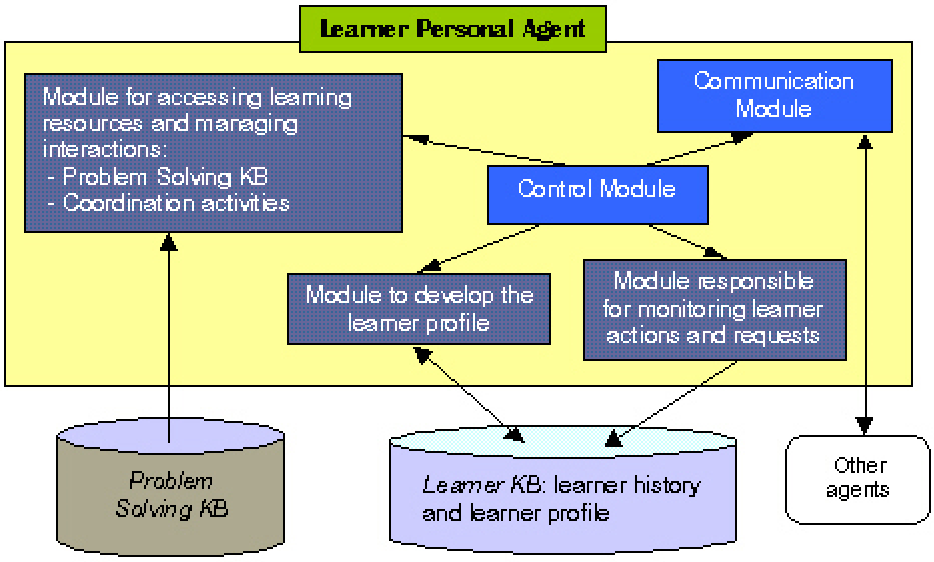

Figure 5. Learner Personal Agent architecture and functionality

Steps towards a user comprehensive model

Building any model starts with feature definition, selection and extraction. As well known, this involves an iterative refinement process, in order to achieve an efficiently working model, able to make predictions that meet reality, at least statistically.

There exists no purely empirical approach to modelling. Even the definition of attributes/features and the selection of the relevant ones in a given context are actually theory driven, explicitly or not. A prototype model of the learner can be used as a starting point that actually encodes some available general theoretical knowledge in the field of learning. Because of the large variability in human personality and in human behaviour, and because the traits that are essential in various contexts are not the same, such a prototype model can not be used directly in practice, without the penalty of being perceived as being rigid and biased. The model has to be customised by using empirical data - sets of examples collected for the given user, while interacting with the system. If the features used to parameterize the initial model do not allow capturing specific detailed behaviour, i.e., if the simple tuning of parameters can not adapt the model to properly depict the user's profile, new features have to be extracted from the empirical data and added to the model. This means, in fact, to elaborate a new refined theory that covers the specific instance of the given user. The number of features to describe a system, especially as complex as a human learner, is unbounded in principle, so that a reductionist approach is mandatory.

The available collection of examples is never large enough to cover all the possible classes in an unbiased manner, to avoid spurious correlation when elaborating a model. Small sets of exceptions may be poorly represented or even ignored. There is no systematic way to empirically identify the domains of the attribute/feature space that are not properly represented in a given example set. The underlying theory helps eliminate irrelevant features that adversely affect the adaptation of the model by parameter tuning, guides the scanning of the input space and the selection of relevant examples, and gives confidence in the solutions produced. On the other hand, a purely theoretical approach may be brittle, i.e., can yield dramatically incorrect results for special cases (exceptions), while scores of common instances that fall in the limits of the theory validity domain are treated correctly (abrupt degradation). Exhaustive theories may become intractable to use, so that the domain of validity has always to be restricted, and a compromise scope - accuracy has to be considered.

The combined use of theoretical knowledge and experimental results can offer the sought for solution, by allowing for incomplete and/or incorrect theoretic knowledge and for incomplete or noisy experimental data. Such a system has the inherent ability to recover from errors, and to keep the model in the range of an acceptable approximation. This is the reason the user model being developed uses a hybrid approach based both on artificial intelligence (AI) symbolic representation of knowledge and on neural network (NN) sub-symbolic representation. NNs have a very good ability to represent "empirical knowledge", but the information is expressed in sub-symbolic form - i.e., in the structure, weights and biases of the trained network, not directly readable for the human user. Thus, an NN behaves almost like a "black box", providing no explanation to justify its decisions taken in various instances. This forbids the direct usage of NNs in learning/teaching and makes it difficult to verify and debug software that includes NN components. On the other hand, the extraction of the knowledge contained in an NN allows the portability of the information to other systems, in both symbolic (AI) and sub-symbolic (NN) forms, as well as towards human users. AI and NN approaches are complementary in many aspects, so they can mutually offset weaknesses and alleviate inherent problems. Thus, there are good reasons to consider that a hybrid aproach, able to exploit simultaneously theoretical and empirical data, is more efficient to build a fault tolerant and adaptive model and could help discover salient features in the input data, whose importance could otherwise be over-looked.

The system is being elaborated in successive steps. In a first phase, the system operates on the basis of statistics about which buttons were selected by the lerner when using the system, in which order and about which error messages have been generated. The system is trained to use this input to offer advice in the form of access to some additional data and information, additional reading, or even to recommend or trigger an interaction with the human tutor -- if atypical behaviour recorded. In a subsequent phase, the system uses (always incomplete, but gradually evolving) error, special interest, and preference databases, structured in a more elaborate way on the basis of the input from the human tutor. The output helps identifying some profile of the user, defined roughly by the set of classes the user belongs to. This influences the future interaction of the system with the user, for instance changing the type and level of exercises on a course presented to the user.

In a next step, the system will have the possibility to include some voluntary feedback from the learners, that will be offered or used consequently for all the other learners, to help conveying original ideas and generate groups of interest. As we see it at the moment, such a system is a useful assistant and not a replacement of the human tutor. On the other hand, it can be considered that some sort of replacement takes place: the work done traditionally by two or three tutors could be accomplished in this approach by only one assisted tutor. This is actually an increase of "productivity".

Conclusions

The basic contribution of this research is twofold: the identification of several learning modalities that combine traditional teaching with “problem-centred” learning to better motivate the student and to increase the efficiency of the learning process, and the conception of a collaborative distance learning system in which human and artificial agents collaborating together to achieve the learning task. Among the agents in the system, the Tutor Agent tries to partially replace the human teacher to synchronously assist the students in the learning endeavour at the time of their convenience.

The development of the presented learning system is a collaborative effort to develop a novel intelligent virtual environment for learning at “Politehnica” University of Bucharest. The system is written in Java and is currently under development, several separate components being already functional. To test the system, we are concurrently developing learning materials on the following subjects: sorting algorithms, resolution theorem proving, neural networks, advanced digital signal processing.

The distributed solution has the advantage of creating an open and distance learning environment to be joined by any interested learner. The system is an effective response to the increased demand for cooperation and learning in today's open environments, both academic and economic, and to the necessity of developing effective learning tools that can be smoothly integrated in the professional development process but also with company work.

Care is taken to prevent such an approach to generate an "elitist" system. The system is designed to enhance the specific features of each user, without increasing the differences from one user to the other in what concerns the level of understanding or the ability to creatively use the acquired knowledge.

Acknowledgements

Part of this work has been supported by the National Agency for Research, Technology and Innovation Contract No. 2021/III/1998, additional phase 1999.

References

[1] Norman, D. A.; Spohrer, J. D. Learner-centered education. Communications of the ACM, Vol. 39, No. 4, 1996, p.24-27.
[2] Bradshaw, J.M. (ed.). Software Agents. Menlo Park, CA, AAAI Press, 1997.
[3] Wooldrige, M.J.; Jennings, N.R. Intelligent agents: Theory and practice. The Knowledge Engineering Review, Vol. 10, No. 2, 1995, p.115-152.
[4] Grudin, J. Computer-supported Cooperative work: History and focus. IEEE Computer, Vol. 27, No. 5, May 1994, p.19-26.
[5] Florea, Adina; Kayser, D.; LeCren, E. Construction et gestion d'un telecours sur les agents intelligents, In Proc. of Deuxiemes entretiens internationaux sur l'enseignement à distance, Poitier, 1-2 December, 1999 (under print).
[6] Cristea, Alexandra; Cristea, P.; Okamoto, T. Neural networks knowledge extraction. Rev. Roum.Sci.Tech. – Electrotech. et Energ., Vol. 42, No. 4, 1997, p. 477-491.
[7] Cristea, P., Neural networks as tools for knowledge eliciting, IN: 4th Seminar on Neural Network Applications in Electrical Engineering NEUREL 97 Proceedings, Belgrade, Yugoslavia: Univ. Belgrade, 1997. p. 2-9.
[8] Cristea, P., Wavelet Transform Using Neural Networks, VI International Conference on Medical Physics, Patras, Hellas, Greece, September 1-4, 1999.