DISTRIBUTED INTELLIGENT TUTORING SYSTEM FOR CONTROL CENTRE OPERATORS TRAINING

Luiz Faria, Zita Vale, Carlos Ramos
GECAD – Knowledge Engineering and Decision Support Group
Institute of Engineering – Polytechnic of Porto
Porto, Portugal
lff@dei.isep.ipp.pt, zav@dee.isep.ipp.pt, csr@dei.isep.ipp.pt
Albino Marques
REN (Transmission Network – EDP Group), Sacavém, Portugal

Abstract – The new requirements of power systems operation demand experienced and well trained operators. However, the training is not often considered a priority task, due mostly to its high costs and medium/long term results.

Usually, the training programs available in industrial environment do not consider the particular training needs of each trainee. The application of the Intelligent Tutoring technology has proved to be a good alternative to the existing training approaches in the power systems area. This paper presents a Web-based Intelligent Tutoring System (ITS) to train Control Centre operators of the Portuguese electrical transmission network.

One of the major advantages of the training based on the ITS technology is the ability to provide individualized training. To achieve this, our ITS maintains a trainee model, which models the trainee's understanding of domain concepts. In this way, the paper describes a curriculum planning module used to choose the most appropriate problem to the trainee's knowledge status. This module includes a neural network to perform a classification of each type of incident according to the trainee's current knowledge. This paper also deals with a mechanism developed to obtain the trainee's reasoning during problem solving.

Keywords: Intelligent Tutoring systems, Knowledge Based Systems, Expert Systems, SCADA Systems, Web-based Training, Multi-agent Systems.

I. INTRODUCTION

The utility industry worldwide is facing numerous changes that are characterized by:

- The growth of the power system structural and regime complexity;
- The increasing complexity of the protection, automatic and information-control systems;
- The increasing competition as a consequence of the privatization and/or deregulation processes;
- The operation of the power system closer to the safety limits.

As a consequence of the pressure for more cost-effective system operation, increased requirements on competence of all of the utility personnel are emerging, especially for the decision makers such as operators at the utility Control Centers.

Unfortunately, the utility experience shows that these requirements grow faster than the number of experienced and adequately trained operators. Thus there is an ever-increasing cost of system operator and inadequate or delayed control decisions and actions.

Furthermore, progress in the area of Intelligent Tutoring Systems (ITS) provides a new approach to power system operator training. Existing utility situation in power system operator training domain has been analyzed and reported [1, 2]. The analysis of the existing situation shows that the operator training is rarely formalized and well structured and the use of clear and measurable performance and training goals is uncommon. As a result, educational efficiency of the existing training practices is insufficient.

Actually, the training of CC operators of the Portuguese electrical transmission network, operated by "Rede Eléctrica Nacional" (REN), is based on a tool named "Dispatch Training Simulator" (DTS). The DTS is a sophisticated training simulator allowing to simulate the electrical grid components and the consequences of operators' actions. However, being this tool a training simulator, it presents a set of limitations common to this kind of systems [3]. In particular, the application of training simulators to the power systems arises a specific set of limitations [4].

In REN case, the reduced number of training sessions performed reveals the difficulties found in the use of this kind of system. The system complexity, both at hardware and software level, makes the mobility of the training system extremely difficult. This fact demands that CC operators leave their usual job places and go to the REN head quarters, where the DTS is installed, to participate in the training sessions. This problem becomes more relevant if we consider that CC operators' work in shifts, which turn out the organization of the training session with appropriate duration very difficult.

On the other hand, the increased amount of time needed to set up each training session restricts the number of sessions offered. In this context, each operator participates in a reduced number of sessions (about two sessions per year).

In this paper we propose a Web-based ITS able to overcome the drawbacks of the current training approach used by REN.

II. Incident Diagnostic Task

During the real-time control, operator's activities predominantly depend on power system operating states (i.e. normal, alert, emergency and restorative). In the presence of an emergency situation, the control center operators must perform the diagnosis of the situation from the analysis of SCADA messages arriving to the Control Center. The diagnosis of the situation involves the incidents characterization and respective localization. Based on this analysis, the operators must realize what should be the sequence of the manoeuvres needed to lead to service restoration. The REN operators have to deal with the following five incident types: DS – single tripping, DtR – three-phase tripping with re-closure, DmR – mono-phase tripping with re-closure, DtD – three-phase tripping with unsuccessful re-closure and DmD – mono-phase tripping with unsuccessful re-closure. The incident localization is defined by the power plant (i.e. substation) and panel where the incident occurred. The panel identifies the protection device activated during the disturbance.

The diagnosis performed by the operators must be correct in order to deduce the appropriate measures to isolate the incident. On the other side, the diagnosis must be performed in the shortest period in order to guarantee a fast restoration, thus minimizing the service interruption period, and to avoid the propagation of the disturbance to other plants.

Our intelligent tutor allows the operator training in the incident diagnostic task. The training of restoration procedures is not included in the scope of this work, but it will be considered in a near future.

The incident diagnostic task involves the use of a set of skills, which include [4]:

- Identification of the relevant events among the set of SCADA messages arriving to the control center;
- Knowledge about the protection devices operation, which allow to know the typical event sequences;
- To identify the correlation between relevant events, including temporal restrictions between events;
- To be able to deal with situations where the instant of SCADA messages does not correspond to the real instant of the events, which can conduce to abnormal sequence of SCADA messages;
- Use of a structured inference mechanism, which allows getting correct diagnostics.

This set of skills includes essentially declarative and procedural knowledge. The declarative knowledge is related with the knowledge about the domain concepts needed to accomplish the diagnostic task. The procedural knowledge includes the inference processes used to perform the task.

III. Intelligent System for Operator Training

Our project concerns the development of an ITS used to provide training to control centre operators of REN. The aim of the project is to utilize AI techniques in the development of tutoring systems that are more flexible and "intelligent" than previous approaches regarding the training in industrial environments.

In an earlier phase of our project we developed a standalone ITS. In order to overcome some drawbacks present in the standalone ITS version we proceed to the conversion of the existing tutor to be web-enabled. In general, web-based tutors provide two new benefits [5]: to deliver the ITS to different settings and to make it independent from the platform. With the Web version of our ITS we achieved the following goals:

- Operators do not need to abandon their usual work places to get training sessions;
- A copy of the standalone software package is unnecessary;
- Operators always use the latest version;
- Maintenance is simple since the software resides on the server;
- Logistical problems of distributing software to individuals are eliminated;
- Training supervisors have facilitated access to operators' performance/progression.

A. *System Architecture*

Our tutor is based on a distributed client-server architecture where a part of its functionality is implemented in Java and works on the client side, and another part works on the server side. The parts communicate over the Internet. In the client side a Java applet, running in a Web browser, implements a graphical interface which is responsible to send to the server all the trainee actions and to show the feedback generated by the server. The server side infrastructure is supported by LPA ProWeb Server (http://www.lpa.co.uk). The LPA ProWeb Server allows Web sites to use the powerful reasoning capacity of Prolog completely in the background, with HTML, Java and other standard tools providing the user interface. The LPA ProWeb Server hides the complexity of HTML forms and CGI programming, and by handling all the communication between the client browser and the server, it relieves the developer from the overhead of implementing a client-server infrastructure.

The tutor, running in the server, is based on a community of Prolog agents based on the LPA Agent Toolkit. Each time a trainee initiates a training session through the Web browser, the ProWeb server launch a community of agents forming an instance of the tutor. The communication between each agent is assured by a facilitator agent. KQML [6], a language and protocol for exchanging information and knowledge, is used by the agents to communicate between them. Figure 1 shows the architecture implemented.

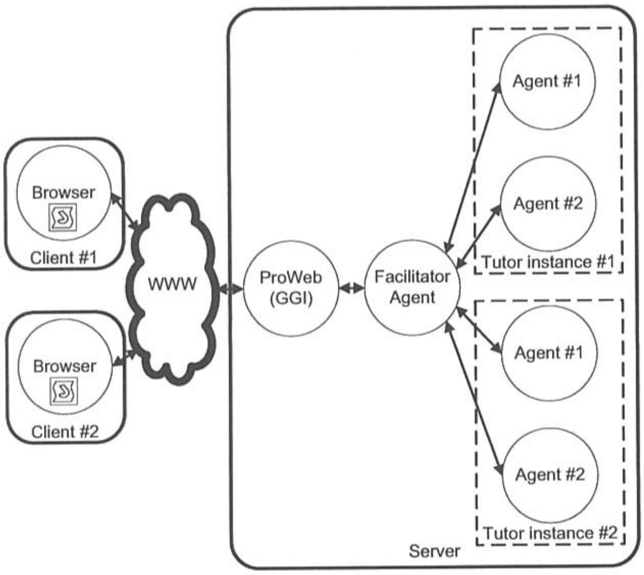

Figure 1- Tutor's Architecture

In the present version each instance of the tutor has only two agents: the tutoring agent and the trainee modeler agent. In a near future we plan to incorporate more agents as new functionalities become included. Machine learning techniques offer a solution to incorporate learning characteristics into the tutor [7]. These learning characteristics can be used to provide the tutor with capabilities to adapt its instructional methods to the trainees' individual needs. On the other hand, new agents can be included in order to simulate the actions of other operators present in the control room or in remote plants. The inclusion of other agents in the training sessions becomes more relevant when we considerer the training of tasks under a cooperative environment such as the training of restoration procedures, which requires the participation of several operators.

B. Tutor Components

Our tutor divides its components into two major classes: modules (or subsystems) and information stores. Modules are active processes that communicate and coordinate to create the required intelligent behavior for the system. As the name suggests, an information store is a store of information/knowledge. The tutoring system modules and its goals are the following:

- Planning and instruction modules – modules responsible for pedagogical decision making; the macro-adaptation module defines the decisions taken before the beginning of the training session and the micro-adaptation module is responsible by the response to the operator actions during the training session;
- Training scenario search module – search of a training scenario whose features are closer to the set of features defined by the macro-adaptation module;
- Specific situation generation module – generation of a model describing the diagnostic process of each incident included in the training scenario;
- Domain expert and operator reasoning matching module – comparing the domain solver (SPARSE expert system [8, 9, 10]) reasoning to the steps performed by the operator during problem resolution;
- Errors identification module – responsible for getting operator misconceptions comparing the operator errors with the library of error patterns;
- User interface manager.

C. An Example Incident Diagnostic Problem

In order to illustrate how a training session is conducted and the interaction between operator and tutor, this section presents a simple diagnostic problem containing a DmR incident (mono-phase tripping with reclosure), occurred in substation SEJ and panel 204. Table I shows the relevant SCADA messages related with this incident. In a real training scenario the operator is faced with a huge amount of messages, including messages received in the control center from other plants, and from both the same plant and panel but not relevant to obtain the incident diagnostic.

Table I- Relevant SCADA Messages of a Mono-phase tripping with Successful Re-closure

18-JUN-2000 04:12:25. 0	SEJ	204	CCL,2	>>>TRIPPING	0 1
18-JUN-2000 04:12:25.190	SEJ	204	CCL,2	-BK BREAKER	0 0
18-JUN-2000 04:12:25.960	SEJ	204	CCL,2	-BK BREAKER	0 1

The SCADA messages contained in the table correspond to the following events: breaker tripping, breaker moving (breaker 00) and breaker closing (breaker 01).

1) Interaction Mechanism

In the area of ITS, some commonly used ways to implement the interaction mechanism between tutor and learner rely in the use of multiple choice and natural language approaches. The effectiveness of this mechanism in a tutoring system is of major importance to the success of these kinds of application because it allows to get the learners answers and to reason about them.

In the multiple choice mechanism, answers are easily graded and the feedback to the learner can be immediate, an important consideration. The main weakness of this approach is that it discourages the development of abstract reasoning and also makes it difficult to determine how the learner generated the answer.

The approach based on natural language relies heavily on the ability to use keywords to decipher an answer. However, this technology does not overcome successfully some aspects, such as understanding textual input, yet. Therefore, the implementation of natural language interfaces becomes a hard task. On the other hand, in some environments a natural language interface does not provide a friendly interaction with the user, being the communication ambiguous and boring. In this project, it is important to ensure that the intelligent tutor is easy to use, in order to be accepted by the operators and effectively used during periods of less activity in the control centre. Moreover, being the end users of the system undergraduate operators, a natural language interface would be an obstacle to the effective use of the intelligent tutor.

The interaction between the trainee and the tutor is performed through prediction tables where the operator can select a set of premises and the respective conclusion. The premises represent events (SCADA messages), temporal restrictions between events or previous conclusions. The table conclusion can be the conclusion about an incident or an intermediate conclusion. The premises and conclusions available correspond to the concepts included in the domain model [4, 11]. The number of prediction tables used by the operator depends of his level of task automation. For instance, if an operator uses only one prediction table to obtain the diagnostic of an incident, this reveals that his level of task automation is high.

The use of prediction tables allows to obtain the operator analysis of the diagnostic problem, minimizing the tutor need to infer his reasoning. From operators' perspective, prediction tables are used as a tool allowing to maintain a record of his reasoning, acting as visual memory expansion.

2) Reasoning About Operator Answers

The contents of the prediction tables represent a model of the operator's reasoning. In order to evaluate this reasoning, the tutor will compare the prediction tables' content with the specific situation model [12, 13]. This model is obtained comparing the domain model with the inference undertaken by SPARSE expert system [4]. This matching process is based on the model tracing technique [14] and allows to identify the errors revealing operator's misconceptions. These errors are used as opportunities to correct the deficiencies in the operator's reasoning.

The operator's entries in prediction tables receive immediate response from the tutor. In case of error, the corresponding entry is assigned to red. Being the case, the operator can ask for help. The help is supplied in the form of hints. Hinting is a tactic that encourages active thinking structured within guidelines dictated by the tutor [15]. The first hints are generic, becoming more detailed if the help requests are repeated. This approach is a good compromise between immediate help and delayed answer. Errors are assigned to red immediately. More detailed help is supplied if explicitly requested by the operator. Thus, the tutor response is not supplied in the presence of the first difficulty and the operator does not need to lose time exploring incorrect steps that do not lead to the correct solution.

The use of the model tracing technique, however, does not require the operator's reasoning to follow a pre-defined set of steps, as it happens in other implementations of the model tracing technique.

The situation specific model generated by the tutoring system for the problem presented is shown in the left frame of Figure 2. This model presents a high granularity level since it includes all the elementary steps used to get the problem solution. The tutor uses this model to detect errors in the operator reasoning. The errors are detected comparing the situation specific model with the set of steps used by the operator in the problem resolution. The granularity level of this model is adequate to a novice trainee, since it decomposes the diagnostic process in several elementary phases. However, the representation level of this model is not appropriate to an expert operator. It is admissible that such an operator does not presents all the steps involved in his reasoning, either because some steps included in his reasoning were omitted or because the task execution reached such a high automation level that some steps are accomplished in a non conscious manner. Such an operator has acquired a high automation level, which allows him to omit several steps. The right frame of Figure 2 represents a model used by an expert operator, which includes only concepts representing events, temporal restrictions between events and the final conclusion. Any reasoning model between the higher and lower granularity level models is admissible since it does not include any violation to the domain model. These two levels are used as boundaries of a continuous cognitive space.

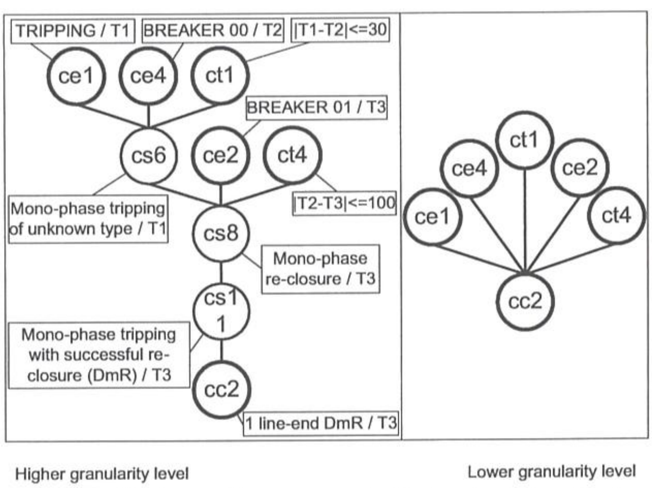

Figure 2- Higher and lower granularity levels of the situation specific model corresponding to the example presented in Table I

IV. Adapting The Curriculum To The Operator

The Curriculum Planning module has as main goal to select a problem from a library of problems. The selected problem must be the one that best fits the trainee needs. However, getting a problem from the problems library compatible with the trainee requires a huge amount of problems. This requirement is, in fact, a factor that contributes to the few cases of successful transition of ITS to the industrial environment. In particular, the preparation of the learning material to the tutoring sessions constitutes a time-consuming task. Usually, in the industrial environment there is not a staff exclusively dedicated to training tasks. This is the case of the electrical sector, where the preparation of training sessions is accomplished with co-operation of the most experienced operators. As this task requires the participation of very busy people that are daily involved in the operation of the power system, it may be difficult to accomplish [16].

In order to overcome this difficulty, we developed two tools that allow to generate learning sessions, as automatically as possible. The first application allows to generate and classify training scenarios from real cases previously stored. Nevertheless, the training scenarios thus obtained do not cover all the situations that control center operators must be prepared to deal with. Therefore an application was developed to allow to create new training scenarios or to edit already existing ones [16].

The process used by the Curriculum Planning module to define the features of the problems presented to the trainees involves two phases. First, the tutor must define the difficulty level of the problem, considering the trainee's progression. Heuristic rules are used to calculate the difficulty level. These heuristic rules relate several parameters, like the trainee's performance in previous problems and the overall trainee's level of knowledge. The difficulty level allows to establish some problem parameters, such as the number of incidents involved, number of different types of incidents, existence of chronological inversion in the SCADA messages and the number of power plants involved in the disturbance. This phase will be described in section IV.A. In the second phase, the tutor must use the contents of the user model to choose the type or types of the most suitable incidents to be included in the problem. A classification is performed in order to get the types of incidents that better match the trainee needs. This classification takes into account the domain concepts involved in each type of incident and the corresponding trainee's expertise in these concepts. The second phase will be described in section IV.B.

For example, if during the first phase the tutor defined that the problem to be presented to the trainee must contain two different types of incidents then, in the second phase, the tutor will choose the two types of incidents considered to be the most suitable ones to the trainee, which means the types with the best classification.

A. Difficulty Level Selection

In order to evaluate the difficulty level of the problems to be presented to the trainee, we proceeded with the identification of the characteristics of the cases that make them more complex or that require more expertise to solve them. Thus, we have identified the following characteristics:

- Number of incidents involved in the case;
- Number of different kinds of incidents;
- Existence of chronological inversion in SCADA messages.

The easiest cases only include an isolated incident, which must be presented to the novice trainees. The case becomes more difficult if it includes incidents of several types. The second level cases include from 2 to 3 incidents of distinct types.

In certain cases it is possible to observe the occurrence of chronological inversion in the SCADA messages, which means that the order of the messages does not follow the order the corresponding events had occurred. These alterations in the messages order are due to delays occurred in the system of acquisition, transmission and time and data setting of the SCADA messages. This chronological inversion present in the SCADA messages brings an additional difficulty to the operators. In such cases, a higher level of expertise is required since the sequence of the messages does not represent a faithful description of the real order of the events. The problems sharing the features of level 1 and presenting chronological inversion in the messages comprises the level 3. Level 4 cases are generated by selecting among the problems including more than three incidents and more than two types of incidents. Levels 5 and 6 comprise cases that share the features of levels 2 and 4 respectively and include chronological inversion in the messages. Table II presents the parameters defining the problems for each difficulty level.

Table II- Problem features used to define the difficulty level

Difficulty level	Problem parameters		
	Number of incidents	Number of incident types	Chronological inversion
1	1	1	No
2	2 or 3	2	No
3	1	1	Yes
4	>3	>2	No
5	2 or 3	2	Yes
6	>3	>2	Yes

The decision about to increase or decrease the difficulty level of the problem to present to the trainee depends on two factors: trainee's global knowledge level and global acquisition factor. Both factors are obtained from the knowledge included in the trainee's model.

The trainee's global knowledge level is a measure of the trainee knowledge level in the set of domain concepts. This factor is calculated from the mean of his knowledge level in each domain concept. The global knowledge level is categorized in three levels. The High level means that global knowledge level is good and the value exceeds the upper threshold, so the trainee is ready to solve problems at a higher difficulty level. The Medium level means that the global knowledge level is moderate, but not enough to change the problem difficulty level. This means that the global knowledge level is between the lower and the upper threshold. At last, the Low level means that the global knowledge level is poor and the value is under the lower threshold, so the trainee's next problem difficulty level should be lower.

To determine an effective adjustment, the Curriculum Planning Module needs appropriate thresholds for deciding on the next problem difficulty level. The opinion of the trainees, regarding the evolution as the problems difficulty level is changed, can be used to verify and to adjust these thresholds.

In addition to the trainee's knowledge level about the domain concepts, our Curriculum Planning module uses an acquisition factor to define the problems difficulty level. Factors such as acquisition can be beneficial to student modeling. Several studies [17] indicate that general factors are predictive of overall learning and allow for a more accurate response to the idiosyncrasies of the student.

Acquisition records how well trainees learn new concepts. When a new concept is introduced, the tutor views the trainee performance on the first few problems. If the trainee performs well on these problems, then he is acquiring skills quickly, and his acquisition factor will reflect this. However, if a trainee requires many problems on a given concept before he illustrates that he understands it, his acquisition will be lower.

The procedure used to determine the trainee's acquisition in each domain concept is based in the number of times the trainee's knowledge level about the concept increased, during the three first application of the concept. This procedure is illustrated in the Table III.

Table III- Determination of the acquisition factor

Number of times knowledge level about concept cn rises in the first 3 applications	acquisition (a_{cn})
3	1
2	0,5
1 or 0	0

The global acquisition factor (A) can take one of three levels (High, Medium and Low). This factor corresponds to the mean of all the acquisition factors of the concepts, which had already been used by the trainee. The global acquisition levels High, Medium and Low indicate that the trainee's capacity to use new concepts is good, moderate and weak respectively.

The mechanism used to define the difficulty level of the problems is based on the following rule:

- If the global knowledge level and the global acquisition factor are opposite in direction,
  - Then the problem difficulty level does not change,
  - Else the direction of the global knowledge level determines the problem difficulty level.

Table IV illustrates the application of this rule.

Table IV- Mechanism used to define the problem difficulty level

Global knowledge level	Global acquisition level	Difficulty level variation
High	High	⇧
Medium	High	=
Low	High	=
High	Medium	⇧
Medium	Medium	=
Low	Medium	⇩
High	Low	=
Medium	Low	=
Low	Low	⇩

From Table IV we can see that if the global acquisition factor is low, independently of the global knowledge level, the difficulty level never rises. To prevent this situation, whenever the trainee knowledge level rise/fall three times in a sequence of three problems solved, the difficulty level will be increased/decreased. Thus, the global acquisition factor, which characterizes the trainee performance in the application of new concepts, will not have a permanent effect in the variation of the difficulty level.

B. Problem Type Adequacy to the Trainee Cognitive Status

The mechanism used to classify each kind of incident in terms of adequacy to the trainee is based in a neural network. The nodes belonging to the input layer correspond to the concepts included in the domain's knowledge base and which the trainees must assimilate. Each node represents the application of a concept in a specific context. For instance, the nodes ce1/T1 and ce1/T5 represents two instances of the same concept. These two instances characterize the application of the concept of breaker tripping event in the situations of first tripping and tripping after an automatic re-closure, respectively. The input vector contains an estimate of the expertise level of the trainee about each concept or application of a concept. These estimated values are obtained from the user model. Therefore, this vector represents an estimate about the trainee's domain knowledge. The elements of the vector assume values in the range [0..1] (values near 0 represent low expertise and values near 1 represent high expertise).

The output layer units represent the adequacy of a type of incident to the current learner's knowledge status. The number of units corresponds to the number of incident types (DS, DtR, DmR, DtD, DmD). Between the output layer units there is no connections. Each output layer's node, representing a type of incident, is connected only to the input nodes corresponding to concepts involved with that incident type. These connections are done with links of weight w_{ij}. Figure 3 represents the map used.

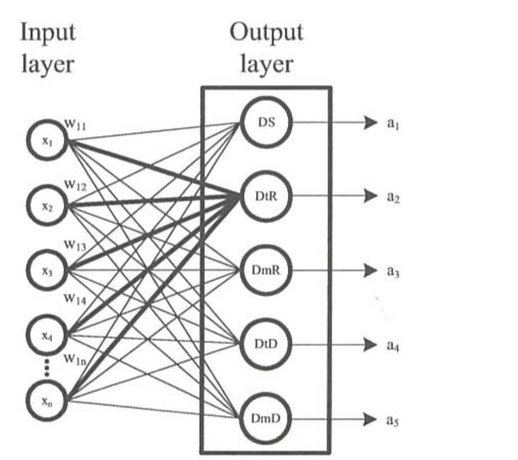

Figure 3- Classification Mechanism

The weights are set in order to get the following goals in the mapping process:

1. The mapping process must be able to establish priorities between domain concepts and, consequently, between the several types of incidents;
2. The algorithm must select the type of incident involving the concepts where the learner reveals less expertise;
3. The incident selection must be performed in order to homogenize the knowledge among the domain concepts;
4. The sequencing of incident types must be non monotonous.

The values used as weights are $w_{ij}=\{1, 0, -\}$. Value '–' is used to indicate that there is no connection between node i of the output layer and the input node j. This means that concept j is not involved in an incident of type i.

In order to illustrate how the weight vectors are set, let us consider as example the weight vector of the output layer's unit corresponding to the DS incident type. This type of problem is the easiest because it involves the shorter inference chain and includes the smaller set of domain concepts. Thus, it is this type of incident that must be presented to a novice trainee (a trainee with small expertise in the majority of domain concepts). For such a trainee, the input values of the neural network, corresponding to the user model parameters, will be near zero. The weight vector elements of the output layer's unit corresponding to the DS output class will be then set to zero in order to get the closest weight vector to the input vector. In this case, the DS neuron will be the winner because its weight vector presents the shorter distance (for instance, the Euclidean distance) to the input vector.

The incident type that follows the DS type, according to the difficulty level, is the DtR type. This incident type includes some domain concepts involved in DS incidents. A DtR incident type must be presented to the learner when he shows a medium expertise level in the subset of concepts already founded in DS incident type. Consequently, the elements of the weight vector corresponding to the DtR output class are set to one, in the case of elements representing domain concepts already known from DS incidents and zero for the elements representing new domain concepts.

Each neuron in the output layer computes its activation level using the input vector and its weight vector. The activation is defined by the Euclidean distance, given by (1).

$$a_i = \sqrt{\sum_{j=1}^{n} \left(w_{ij} - x_j\right)^2} \qquad (1)$$

We can see that a neuron that possesses a weight vector (w) similar to the activation levels vector of the input nodes (x) will have a low activation level and vice versa. The output layer's node with the lowest activation will be the winner. Where there is not connection between a node of the input layer and a node of the output layer, the corresponding element of the weight vector is "–" and the summation's parcel will be considered as zero.

The evaluation of the activation levels of the output layer units can lead to same levels for different nodes. To solve this possible conflict a priority mechanism is used. The criterion used to establish the priority between incident types consists in giving priority to the simplest types. This criterion is based on the idea that the simplest problems must be presented to the learner first.

V. Conclusions and Research Directions

Web-based ITS provides a flexible way to provide training in places such as control rooms where time and space constraints unable the use of traditional education techniques. This paper presents a case where an earlier standalone tutor was web-enabled by changing it to a server-side application and developing a relatively simple Java applet that implements interface functions at client side and communicates with an intelligent server.

The reasoning mechanism about trainee answers is based on the model tracing technique. However, our implementation of this technique does not require the operator's reasoning to follow a pre-defined set of steps. The two granularity levels of the specific situation model are used as boundaries of a continuous cognitive space. This continuous cognitive space is compatible with the nature of the gradual process of acquisition of a task mental model.

The process of learning a new concept, such as the application of a new production rule used to get a diagnosis, requires that all the relevant concepts are simultaneously present in memory. Furthermore, the working memory load must be minimized to warrant the efficacy of the learning process. This implies that one should try to provide instruction on specific components only when other components of the skill have already been relatively well mastered.

The mechanism developed to select the features of the problem to be presented to the trainee allows to get a curriculum that adapts dynamically to the trainee needs. This leads to a curriculum design in which only a few new concepts are taught at a time. Thus, we can achieve an instruction where the trainee gradually gets the skills needed to solve complex and real-world problems rather than having to acquire them all at once.

Future research goals include to provide the incident classification algorithm, based in a neural network, with a machine learning mechanism. The learning mechanism will be used to adjust the weight vectors according to the trainee's progression. We expect this learning mechanism will increase the efficiency of the training process. This will require an intensive use of the tutoring system by a large set of operators in order to collect data to conduce the neural network learning.

We also intend to explore the possibility of using new agents with different roles in the training process. This will be especially useful to train operators in tasks involving other operators when these are not available to training, being their role simulated by artificial agents.

References

[1] N. Cukalevski, and A. Johansson, "Existing Competence Requirements and Training for Control Room Personnel", *ELECTRA*, no. 171, pp. 101-115, Apr. 1997.

[2] M. D. Anderson, "Dispatcher Training Survey Results", *IEEE Trans. PAS*, 104(9), pp. 2374-2380, Sep. 1985.

[3] C. Frasson, "SAFARI, A University-Industry Cooperative Project", Montréal, Canadá, 1996.

[4] L. Faria, "Treino e Apoio a Operadores de Centros de Controlo e Condução – Uma Abordagem Baseada em Conhecimento e Tutores Inteligentes", Ph.D. dissertation, Dept. Elect. And Computer Eng., Faculty of Engineering, University of Porto, Portugal, 2002.

[5] P. Brusilovsy, "Adaptive and Intelligent Technologies for Web-based Education", in C. Rollinger, C. Peylo (eds.), *Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching*, no. 4, pp. 19-25, 1999.

[6] Y. Labrou, T. Finin, "A Proposal for a new KQML Specification", TR CS-97-03 Computer Science and Electrical Engineering Dept., University of Maryland, 1997

[7] D. Kelly, B. Tangney, "Incorporating Learning Characteristics into an Intelligent Tutor", in 2002 Proc. Intelligent Tutoring Systems Conf. (ITS'2002).

[8] Z. Vale, C. Ramos, L. Faria, J. Santos, M. Fernandes, C. Rosado, and A. Marques, "Knowledge-Based Systems for Power System Control Centers: Is Knowledge the Problem?", in Proc. 1997 IEEE Intelligent Systems Applications To Power Systems Conf. (ISAP'97), pp. 231-235.

[9] Z. Vale, A. Moura, M. Fernandes, A. Marques, A. Rosado, and C. Ramos, "SPARSE: An Intelligent Alarm Processor and Operator Assistant", *IEEE Expert- Special Track on AI Applications in the Electric Power Industry, 12*(3), pp. 86- 93, 1997.

[10]Z. Vale, M. Fernandes, C. Rosado, A. Marques, C. Ramos and L. Faria, "Better KBS for Real-time Applications in Power System Control Centers: the Experience of SPARSE Project", *Computers in Industry*, Elsevier, no. 37, pp. 97-111, 1998.

*[11]*L. Faria, Z. Vale, C. Ramos, A. Marques, "An ITS for Power System Staff Training: Matching Knowledge Representation and Learner Interaction", in Proc. 2001 IEEE Intelligent Systems Applications To Power Systems Conf. (ISAP'2001).

[12]W. J. Clancey, "Qualitative Student Models", *Annual Review of Computer Science*, no. 1, pp. 391-450, 1986.

[13]R. A. Khuwaja, "A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain", Ph.D. dissertation, Dept. of Comp. Sc. And App. Mathematics, Illinois Institute of Technology, USA, 1994.

[14]J. R. Anderson, and C. Lebiere, *The Atomic Components of Thought*, Mahwah, NJ: Erlbaum, 1998.

[15]G. Hume, "Using Student Modelling to Determine When and How to Hint in an Intelligent Tutoring System", Ph.D. dissertation, Dept. of Comp. Sc. And App. Mathematics, Illinois Institute of Technology, USA, 1995.

[16]L. Faria, Z. Vale, C. Ramos, A. Silva, and A. Marques, "Training Scenarios Generation Tools for an ITS to Control Center Operators", in 2000 Proc. Intelligent Tutoring Systems Conf. (ITS'2000).

[17]J. Beck, M. Stern, and B. Woolf, "Using the Student Model to Control Problem Difficulty", in 1997 Proc. User Modeling Conf. (UM97), pp. 277- 288.