An Agent-based Intelligent Tutoring System: A Case Study in Legal Domain

Ig Bittencourt [1] Marcos Tadeu [1] Evandro Costa [1] Camila Nunes [1] Rômulo Nunes [2] Alan Silva [1]

[1] Federal University of Alagoas - Computation Institute
Tabuleiro dos Martins, Postal Code 57.072-970 Maceió - AL
Email: {ibert, mts, evandro, cpbn, alan}@tci.ufal.br

[2] Federal University of Alagoas - Campus Arapiraca
Campo da Sementeira - AL 115, KM 3 S/N Arapiraca - AL
Email: romulo@dsc.ufcg.edu.br

Abstract: Computer based learning gets more and more important in higher education. Particularly, in Legal domain, students have little chance to deal with realistic situations. One way to alleviate this problem is to provide Law students with real cases, rules, and viewpoints of which a given body of knowledge is often recognized as important to their successful learning. In this paper is proposed a novel approach to Intelligent Tutoring System (ITS) applied to Legal domain to address each of the above concerns. Then, it was defined an agent-based architecture to support multiple views of domain knowledge, improving the quality of student-ITS interactions and the learning success of the students. Each tutoring agent from the system contains a knowledge-based system that combines Case-Based Reasoning (CBR) and Rule-Based System (RBS). In addition, each agent adopts the Reinforcement Learning Algorithm aiming at identifying the best pedagogical strategy by considering the student profile. This paper focuses on both architecture and the mentioned Artificial Intelligence techniques into a Legal System.

Keywords artificial intelligence and law, intelligent tutoring systems, case-based reasoning, rule-based systems, reinforcement learning.

1. Introduction

Providing Law students with real cases, rules, and different viewpoints of a given body of knowledge is often recognized as important to their successful learning. In fact, these requirements are hot topics and seem to be more realistic to be explored in the context of Legal Intelligent Tutoring System research. However, little research has been done on these aspects.

In this paper is proposed a novel approach to ITS applied to Legal domain in order to address each of the mentioned concerns. Then, it was defined an agent-based architecture to support multiple views of domain knowledge, aiming to improve the quality of student-ITS interactions and the learning success of the students. Here, learning is considered as problem-solving, where students solve problems and are individually assisted and guided by the system during the solution process. The system may also solve problems by using CBR or RBS or combination of them. CBR has been used for checking the similarity with old cases to justify new problems and RBS for evaluating the rules of Normative Knowledge. In addition, to improve the pedagogical interaction each agent adopts the Reinforcement Learning Algorithm aiming at identifying the best pedagogical strategy by considering the student profile.

The use of a hybrid solution to the problem solving has been motivated due the structure of the juridical system [1]. Legislation is the main Legal research, where magistrates making their decision based on the code and laws, originating case solutions. One of the best ways to solve Legal problems based on legislation is using rules, due this, rules are approached in the ITS. For this reason, Case Based Reasoning and Rule Based System make necessary to the ITS.

In the approach here presented, the idea is to engage Law students into interactions with ITS based on the resolution of Legal problems and their consequences on other tutorial activities, concerning the Civil Law. The starting point of these interactions occurs when ITS submits a penal situation to Law students. Then, they will learn two fundamental but different skills of Legal problems. First, know how to identify relevant cases and Legal concepts (Normative Knowledge, for instance) of the cases. Second, know how to use them effectively as examples justifying position in a Legal argument. When using an automated information retrieval system, one needs to use a query or a set of queries that captures the issues and the intended use of the cases in an argument [2]. Therefore, Case-Based Reasoning has been used to know what kind of cases can result better solutions and Rule-Based System to find better concepts, and consequently giving support for better explanations of the problem.

This paper is organized as follows. In Section 2. are discussed related works. A brief discussion of used AI techniques is presented in Section 3.. System Architecture and Implementation are described in Section 4., including system details. Section 5. discusses the dynamic of the architecture with an illustrative scenario. Finally, conclusions and future works are presented in Section 6..

[1] Civil Law, also known as Continental Law or Roman Law has been used in the system.

2. Related Work

Some related works were developed taking into consideration legal tutoring or hybrid reasoning involving CBR and RBS.

In [1], Aleven proposes an intelligent learning environment designed to help beginning law students learn basic skills of making arguments with cases.

An ITS for Legal domain, using Rule-Based System and approaching problem-based learning as pedagogical strategy is presented in [20]. The proposal refers to a novel ITS approach applied to Legal domain, using hybrid reasoning (CBR and RBS). Besides, it describes the modeling of multiple views of domain knowledge, providing two-way interaction in the problem solving.

[19] describe an expert system that deals with (US) income tax law relating to the deduction for expenses relating to a home office.

[5] combines both the blackboard architecture and distributed AI methods for creating hybrid systems. This means that both RBS and CBR are attempted at the same time. Both inference mechanism run concurrently (the distributed AI aspect of the system) - with the ultimate report being the best of the two reports produced (on by RBS and one by CBR).

[17] describes a Dutch expert legal system, focused on the domain of landlord-tenant law, it combines knowledge groups as legislation, legal doctrine, expert knowledge, and case law.

In [25], the project uses a distributed artificial intelligence approach, operating in the area of credit law, that combines CBR and RBS independently. First, the system will make an inference in the RBS, after that, if not succeeded, use CBR.

The proposal refers a novel ITS approach applied to Legal domain, using hybrid reasoning (CBR and RBS), besides the modeling of multiple views of domain knowledge, providing two-way interaction in the problem solving. The related works [1] and [20] approach an educational system. Different from these proposals, the approach of an intelligent pedagogical mechanism (Reinforcement Learning) had not been find in such proposals.

3. Used AI Techniques

The approach of Case-Based Reasoning can be contrasted[2] with (an option of different reasoning) Rule-Based Reasoning, which is largely used in knowledge-based systems. In Rule-Based Systems, there is a base of rules that is composed by a set of production rules as follows: IF A, THEN B, where A is the condition and B is the action [18]. While in Case-Based Reasoning, the problem solving approach is through the learning with past experience [26].

A hybrid system combines more than one method of reasoning, that contributes to solve a specific problem. Typically, hybrid systems combine the two more known reasoning ways: CBR and RBR [16].

A good example of the combination of AI techniques, as Case-Based and Rule-Based Reasoning is the forensic system GREBE, according to [5], which combines architecture and distributed AI components with hybrid reasoning methods.

In subsections below, there is a detailed approach about these two forms of reasoning.

[2]Sometimes CBR and RBR complete themselves.

3.1 Rule-Based Reasoning

The Rule-Based Reasoning remains a knowledge base made by an expert. This knowledge base is composed by rules. The system needs a consistent base of rules stored and facts given by the user to get a conclusion of a problem. According to [6], a rule is a logical statement that relates two or more objects and includes two parts, the premise and the conclusion. Each of these parts consists of a logical expression with one or more object-value statements connected by the logical operators *and*, *or*, or *not*.

A simple example of a rule is:
IF sun = yes THEN it is day.

The premise of the rule is the logical expression between the words IF and THEN, and it can contain one or more object-value connected by the logical operators *and*, *or*, or *not*. The conclusion is the logical expression after the keyword THEN.

There are many types of inference and control strategies to obtain simple or mixed conclusions. To obtain as many conclusions as possible, several inference strategies are frequently used in an inference engine of a single system. As shown above, Rule-Based Reasoning is a crucial component to the system; since it has different, efficient, and simple strategies to solve problems imposed by the user.

3.2 Case-Based Reasoning

CBR is a problem solving approaching by analogy. Then, in order to solve a new problem, a similar situation already solved and kept in a memory base is aimed. After that, attempt to adapt the retrieved solution in a way to solve a new problem. Moreover, the review capacity is inherent to validate the adapted solution.

The structure of a case is composed by [13]:

(i) problem: a set of attributes is described to define a problem.

(ii) solution: a set of attributes is described to define a solution. Moreover, according to the domain, some steps can be defined to the problem solving.

As shown in Fig. 1, there are four phases in the inference process in CBR, called 4R cycle, which are:

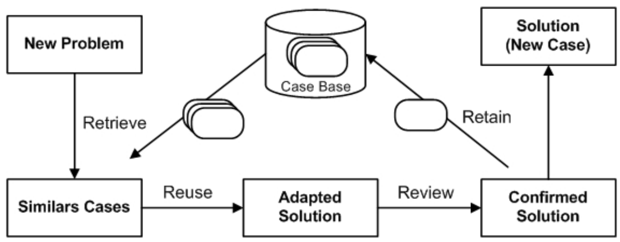

Fig. 1. Case-based reasoning cycle.

1R - Retrieve: its purpose is to find a case or a set of similar cases, providing an element to the current problem. Some ways to retrieve cases can be sequential, in two levels, beyond others. Another important aspect in this phase is the similarity, in which exist two types. The first type is the local similarity which find similar attributes; the second one, global similarity, to cases similarities.

2R - Reuse: its objective is to provide the reuse of the case to be used to a problem solution. In this phase, the case adaptation occurs when it is necessary. Some ways of adaptation are: compositional, hierarchic, structural, among others. This is a very complicated phase to foresee and generalize, because it is very specific of each domain.

3R - Revision: its purpose is to confirm a reused solution. The revision process can be done in an automatic or manual way. It is a very complicated phase and in some cases can have another inference form into this phase just for validation.

4R - Retain: in this phase occurs the Case-Based Reasoning learning, because after 3R's the new case is stored in the case base and after that is returned the solution to the user.

3.3 Reinforcement Learning

The Reinforcement Learning is used in a domain to learn about the pair state-action. Then, its objective is to assimilate the best action applied to an initial state that maximizes the expectations in order to achieve final/objective state. This approach is adopted when does not exist a definition or compacted study about the actions. In addition it takes into consideration each state of the environment. The learning process consists in the reinforcement continuous adjust, through rewards or punishment. The Reinforcement Learning is able to process according to the Markov [11, 4, 22].

3.3.1 Markov Decision Process

According to [14], "an environment satisfies the Markov's property if its state summarizes compact form state and it can say what next expected state and reward, being the actual state and action". The Reinforcement Learning technique satisfies the Markov's property is named *Markov Decision Process*, and it is defined as a set (S, A, P, R), where:

- S: a finite set of states of the system.
- A: finite set of actions.
- $P : S \times A$: state transition function that maps the pairs state-action in a probability distribution about the set of states.
- $R : S \times A$: a utility value, mapped as a reward function.

In other words, actual pair $s \times a$ tries to choose the best action to the next state s' according to the probability $P(s'|s, a)$ and the associated reward $r(s, a)$. In addition, a process is Markovian if the states transition depends only the current state of the system and it is independent of all the previous states [15].

4. System Architecture and Implementation

Basic components in the architecture represent the classical ITS approach, however, details about specific agents in these components are explained in the next subsections.

The **Interface Agent** is responsible for communicating the student, providing access to Legal information such as jurisprudence and Normative Knowledge. The kinds of screens displayed are about student credentials, student information, problem, solution, solution evaluation, and argumentation. In addition, a **Broker Agent** was inserted for mapping the information between the agents.

It was adopted an ontology that represents the base of the Legal knowledge to be used within Intelligent Tutoring System.

4.1 Interface Agent

The Interface agent is responsible for showing information about the system. Moreover, Interface Agent works in an assistant way, in other words, it is done a step-by-step for the user describes the problem. For example, consider the screens presents at the description/specification of a legal problem:

- problem specification: the steps for the problem specification are:

1. personages specification (name, age, deficiencies, ...).
2. relation among the personages (father, mother, son, brother, brother in law, friend, ...).
3. fact specification:

(i) Did the murder happen ?

(ii) What are the personages positions (victim, killer, accomplice, witness, ...)?

(iii) Which was the gun (slashing/piercing object, poison, revolver, ...) used ?

(iv) What was the crime reason (revenge, ordered)?

(v) What are the personages conditions (drunken, strong emotion, sleeping, ...)?

- problem solution.

4.2 Expert Module

The Expert Module was modeled as shown in Fig. 2. In this subsection, details of the Domain Architecture are presented.

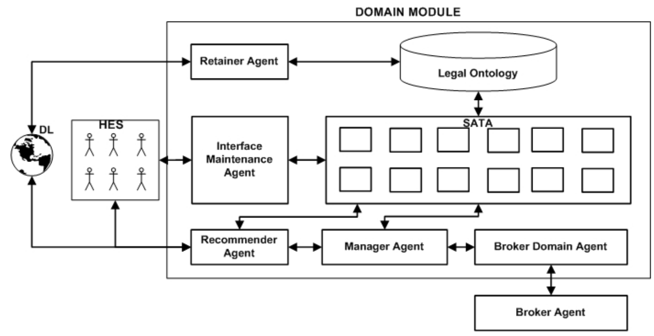

Fig. 2. Expert module architecture.

The ontology defined to the ITS was developed by using Protégé [21] tool, that provides flexibility and a lot of plug-ins.

In Fig. 3 is shown a relevant part of the legal ontology. In this figure, it is possible to see some concepts and their relationships. Ontology has been developed taking into consideration MATHEMA Model [7].

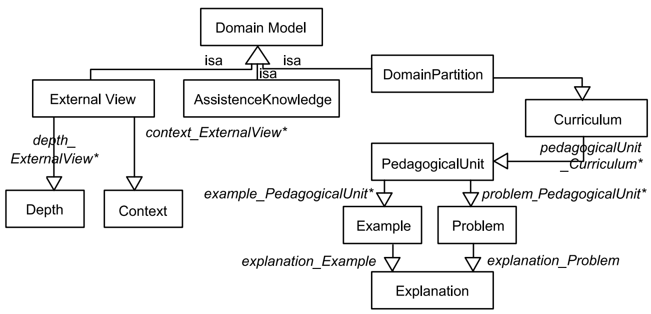

Fig. 3. Ontology description.

Based on the Legal ontology developed to the ITS, a problem is defined as a 3-Tuple $\langle P, I, F \rangle$, where:

• P: represents the real penal situation.

• I: represents a set of interpretations (solutions) to the problem P. The related interpretations are based on:

(i) Prosecutor view.

(ii) Lawyer view.

• F: P x I: represents the theoretical foundation in the relation P x I, that can be a Doctrine, Jurisprudence, or web site.

4.2.1 Agents

• **SATA:** it is a Society of Artificial Tutoring Agents that represents an agent-based ITS acting in a specific domain. Each agent in SATA plays pedagogical roles. These agents are responsible for problem solving and providing information of the student profile and pedagogical strategies. In the problem resolution, cooperation is one of the features of the SATA. Cooperation is approached by considering a multidimensional view, where the expert knowledge is distributed in each agent from SATA. The idea is the division of the domain knowledge into many sub domains, represented by agents. This division tries to guide a search in the expert knowledge distributed. In addition, each agent has a curriculum manager that chooses problems for the student, but it is optional.

SATA Reasoning: This subsection approaches the reasoning presents in SATA structure. Rule-Based System and Case-Based Reasoning are two known approaches adopted in the Expert Module Agents in the Intelligent Tutoring Systems (ITS). They are natural alternatives in knowledge representation. Rules usually represent general knowledge, whereas cases encompass knowledge accumulated from specific (specialized) situations. Each approach has advantages and disadvantages. Due to their interchangeable nature, rules and cases can be integrated and thus produce effective ITS. Below characteristics of CBR and RBS reasoning are described.

CBR (Case-Based Reasoning) Reasoning

The CBR Reasoning is responsible for evaluating the similarity between the jurisprudence (inserted in the case base) and the penal situation sent by Law student.

The knowledge representation was done as a relational representation, with n attributes $A = \{a_1, a_2, ..., a_n\}$ where each attribute has a weight $W = \{w_1, w_2, ..., w_n\}$, for details of the knowledge representation and similarity functions, see [10]. The similarity function between two cases is defined as shown in Equation 1:

$$SIM(C_1, C_2) = \sum_{i=1}^{n} \Big(w_i * sim(a_{i_{C_1}}, a_{i_{C_2}}) \Big) \quad (1)$$

The retrieval process was done in a sequential way. The reuse and revision phases were regardless, due to the impossibility for law to alter a jurisprudence.

The evaluated and considered attributes in the environment are:

(i) Co-authorship: participation of other person at the crime.

(ii) Crime Qualification: it defines the kind of qualification (if was ambuscade, nasty, or futile reason).

(iii) Kind of action.

(iv) Crime modality.

(v) Attempt: if have or not the attempt.

(vi) Crime Qualification: defines if the crime is qualified or not.

(vii) Result: if the result was favorable to the lawyer or to the promoter.

The steps for the use of the CBR component are:

Step 1, the user will choose the case base and then will define which attributes will be used.

Step 2, now the user defines the weight of each attribute. The weight default values is one, because the user can not want to distinguish the attributes.

Step 3, it is defined the local similarity functions, in other words, the similarity calculus among the attributes. Moreover, it is defined in this phase the global similarity function, in other words, the similarity calculus among the cases.

Step 4, it defines the way to retrieve cases, that can be sequential, two levels, and others.

Step 5, there is a screen to define the jar file to the phase of reuse of the case. This phase is not deployed in the environment due to big detail of this phase.

Step 6, it is a screen to define a jar file for the revision phase. It occurs in same way and reason of the step 5.

Step 7, it defines the retention threshold of the case.

In both cases, the case retrieval was divided into two levels. First, it is used the similarity function for simple attributes (numeric and boolean, for instance) and more important index. Second, it is compared the complex attributes such as strings. This division turns the case retrieval faster than the sequential retrieval method. Adaptation and retention are not necessary because the jurisprudence can not be changed.

RBS (Rule-Based System) Reasoning

RBS (Rule-Based System) Reasoning is responsible for the rules evaluation in the Legal ontology. The rules were modeled by considering the Normative Knowledge, which enables the whole validation of a penal situation. When the Law Student describes a problem, the system tries to infer about the features and map them in doctrine concepts defined in the ontology. In addition, were modeled 49 rules to infer of the domain. Follow an example of rule developed in the Jess [8] environment and integrated within protégé [21]:

```
(bind ?article new Article) (defrule concept
    ("corporalLesion")
    ?article getInstance() )
```

The interactions between the Law students and the ITS in the problem solving can happen in two ways:

(i) when the student submits a penal situation to tutoring system.

(ii) when the tutoring system submits a penal situation to the student.

In both cases, can use a hybrid mechanism of reasoning, CBR (Case-Based Reasoning) and RBS (Rule-Based System) working together with the legal ontology to solve problems submitted by the student or by the tutoring system.

The steps for using RBR component are:

Step 1, the user will register the variables and their respective values.

Step 2, now the user already can create the specifics rules of his/her domain.

Step 3, in this step, the user can choose the inference method, that are: forward chaining or backward chaining.

Step 4, the explanation module helps the non-programmer knowledge in the subject, because the explanation used by literature is impossible for general use.

The interactions between the techniques can be structured as follows:

(i) CBR reasoning and RBS reasoning return the solution, exploiting the jurisprudence and Normative Knowledge researches.

(ii) CBR reasoning returns the solution. This situation happens when the RBS reasoning is unable to infer from the given problem.

(iii) RBS reasoning returns the solution. This situation happens when the CBR reasoning is unable to infer from the given problem.

(iv) none of them. If this situation occurs, the system will send a message to the student, requesting better description from the given problem.

When the *student submits the tutoring system to a penal situation*, ITS tries to solve the penal situation integrating CBR and RBR. This interaction algorithm was implemented as follows:

Initialize $Evaluate(studentProblem)$;
Initialize $RBSInfer()$;
$rbsSolution \leftarrow$ try infer from NormativeKnowledge;
Initialize $CBRCycle()$;
$casesBase \leftarrow$ select jurisprudence from Ontology;
Execute Retrieve from CBRCycle;
Select similarCase;
Select similarityValue;
$MountSolution(rbsSolution, similarCase)$;

Algorithm 1: The evaluation student problem algorithm.

Second, when the *tutoring system submits a penal situation to the student*, the student describes the solution as far as her/his concerning. After that, ITS evaluates the student solution according to the algorithm below.

Initialize $Evaluate(studentSolution)$;
Initialize $CBRCycle()$;
$casesBase \leftarrow$ select casesSolution from Ontology;
Execute Retrieve from CBRCycle;
Select similarCase;
Select similarityValue;

Algorithm 2: The student solution evaluation algorithm.

• **HES (Human Expert Society)**: represents an external knowledge source integrated in SATA, which provides necessaries maintenance in the SATA, such as inclusion and exclusion of agents, besides give support to Law students. The support to the Law student is possible through the recommender agent.

• **Interface Maintenance Agent**: represents the communication between HES and SATA.

• **Manager Agent**: responsible for all the flow interaction ITS-student. The tutoring process can happen as: (i) guided, where the ITS gives assistance to the student about the curriculum or (ii) free, where the student chooses the curriculum he/she intends to study or search about Legal Concepts as jurisprudence, Doctrine, and Legislation. The characteristics of this agent are:

(i) choose which SATA Agent will tutoring the student or given assistance.

(ii) manage kinds of pedagogical strategy are given to the student, like hints, warnings and others.

(iii) choose which SATA Agent(s) will solve a problem sent by the student.

• **Broker Domain Agent**: responsible for delegating actions to the Recommender Agent, Explainer Agent, and SATA, besides send information to the Broker agent.

• **Recommender Agent**: responsible for providing supports to the Law student, providing additional researches aiming the improvement of learning process. The following are the kind of supports provided for the Recommender Agent:

(i) HES (Human Expert Society) can be recommended to give support to the Law student.

(ii) information in a Digital Library can be sent to support the learning process.

(iii) an agent in SATA to supervise the student is recommended to provide better resolution about the problems and to help him/her with doubts concerning the domain knowledge.

• **Retainer Agent**: one of the biggest problems found in Case-Based Reasoning systems for the Legal domain refers to the jurisprudence that updates everyday. For solving this problem, it was added a Retainer Agent at the intelligent tutoring system for evaluating new jurisprudence. This agent retrieves information through virtual libraries.

4.3 Pedagogical Module

Some researches have been doing into Intelligent Tutoring System, exploring Reinforcement Learning in pedagogical activities [3, 23, 12]. The Pedagogical Module was modeled as shown in Fig. 4.

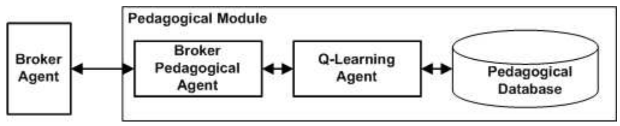

Fig. 4. Pedagogical module architecture.

In order to improve the quality of the solutions and student learning, a Reinforcement Learning Algorithm was used concerning the strategies below:

• increase the problem difficulty degree.
• decrease the problem difficulty degree.
• same difficulty degree.
• change the level.
• change the issue.
• change the problem issue to past issue.

Q-Learning algorithm is used after the definition of the student profile (features of the student profile were not focused), due this, in student-ITS interaction, the algorithm, through rewards, defines and selects the best strategy to use in a specific situation. The selection of strategy takes into consideration groups of profiles as shown in Fig. 5.

4.3.1 Q-Learning Structure

The goal of the agent in a Reinforcement Learning Problem is to learn an action policy that maximizes the expected long term sum of values of the reinforcement signal, from any starting state [4]. In the present work, the problem is defined as a Markov Decision Process (MDP) solution.

The chosen of better strategies has been modeled as a 4-tuple (S, A, T, R), where:

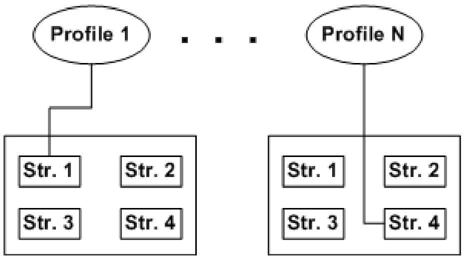

Fig. 5. Q−learning strategy.

- S: set of pair of strategy and MATHEMA Context.
- A: finite set of strategies.
- $T : S \times A \rightarrow \Pi(s)$: state transition function represented for the probability value, signalizing the betters strategy to be chosen.
- $R : S \times A$: it is described as a utility value, defined for the similarity of the attributes, mapped as a reward function.

It was used in the e-learning environment a proposal approached in [4], that implements an algorithm which is used in the action choice rule, which defines which action must be performed when the agent is in state s_t. The heuristic function (Equation 2) included was:

$$\pi(s_t) = \begin{cases} argmax_{at} & \left[\hat{Q}(s_t, a_t) + \xi H_t(s_t, a_t)\right] & ifq \leq p, \\ a_{random} & & otherwise. \end{cases} \qquad (2)$$

- $H : S \times A \rightarrow R$ is the heuristic function.
- ϵ: it is a real variable used to weight the influence of the heuristic function.
- q: it is a random uniform probability density mapped in [0, 1] and $p(0 \leq p \leq 1)$ is the parameter which defines the exploration divided for exploitation balance.
- a_{random} is a random action selected among the possible actions in state s_t.

Then, the heuristic value $H_t(s_t, a_t)$ can be defined as shown below:

$$H(s_t, a_t) = \begin{cases} max_a & \hat{Q}(s_t, a) - \hat{Q}(s_t, a_t) + \eta & if a_t = \pi^H(s_t), \\ 0 & & otherwise. \end{cases} \qquad (3)$$

Initialize $Q(s, a)$
Repeat:
  Visit the s state
  Select a strategy using the choice rule
  Receive the reinforcement $r(s, a)$ and observe next state s'.
  Update the values of $H_t(s, a)$.
  Update the values of $Q_t(s, a)$ according to:
  $\hat{Q}(s, a) = \hat{Q}(s, a) + \alpha \left[r + \gamma max \hat{Q}(s', a') - \hat{Q}(s, a)\right]$
  Update the $s \leftarrow s'$ state
Until some stop criteria is reached,
where $s = s_t, s' = s_{t+1}, a = a_t e a' = a_{t+1}$

Algorithm 3: The Heuristics Algorithm.

5. An Illustrative Example

To illustrate the functionality of the system is shown a learner working with problem-solving of a penal situation approaching multiple views of the knowledge.

Imagine that the student works the first time in the ITS. The student answers a set of question about Legal issue and then, the level knowledge of the student is defined. Below, it is exploited an example where the student submits a problem to the system.

5.1 Case

Problem: *John arrives in his home and see Maria and Joseph (John´s brother), sleeping in the bed, naked. Then John overdrew his gun and shot against Maria, which died.*

When the user specifies a problem, the system concerns to the rules and cases, evaluating the attributes [3]:

- Personages = John, Maria, and Joseph.
- relation among the personages = Brother(John, Joseph), Married(John, Maria).
- What are the personages positions: Victim(Maria), Accused(John), and Witness(Joseph).
- Personage's deficiency: it specifies if the patient has some physical deficiency that can be considered, for example, in cases the victim can not protect itself = Maria sleeping in the bed.
- Fact (attempt well successful or not): if the crime was concretized = yes.
- Gun used: the gun is very important, because it can characterize the gravity of the crime = gun.
- Reason of the crime: it specifies if the crime was perpetrated for revenge, ordered, among others = adultery.

Solution: The solution is divided into two views: The Prosecutor view, where tries to increase the punishment and the Lawyer view that tries to decrease the punishment.

Prosecutor View:

- Normative Knowledge - Qualified Homicide: Art. 121, §2º, IV;

Doctrine - Qualified Homicide can be used when happens a crime through research that makes difficult or impossible the defense or the offended person, by the fact of the victim been sleeping.

- Jurisprudence - Summary: JURI. Qualified Homicide. Research that turn defense of the offended person impossible. Victim Sleeping. [...]

Below follows the rule used to prosecutor view solution.

1. Rule
   If victim = 'impossible defense' or Fact = 'concretized', then Article = 121 and Paragraph = 4 and item = IV

When the user describes a rule, the system attempts to infer about the characteristics and mapping them in the doctrinaire concepts.

Lawyer View 1:

- Normative Knowledge - Self-Defense: Art. 23.

Doctrine - Self-Defense can be used when the author has his honor stained for the victim.

- Jurisprudence - Summary: Homicide - Self-Defense of the honor - Accused that, [...].
- Site: `http://jus2.uol.com.br/doutrina/texto.asp?id=980`;

1. Rule
   If AccusedCondition = 'self-defense', then Article = 23

[3] The context of the attributes is considered relevants in Brazilian Code

Lawyer View 2:

• Normative Knowledge - Privileged Homicide: Art. 121, §1º;
Doctrine - Privileged Homicide can be used when the author acts through strong emotion.

• Jurisprudence - Summary: JURI. Qualified Homicide. Cohabitation. Condemnation for Privileged Homicide.

1. Rule *If CrimeReason = 'adultery' then AccusedCondition = 'strong emotion' or AccusedCondition = 'depression'*
2. Rule *If AccusedCondition = 'strong emotion' and Fact = 'concretized', then Article = 121 and Paragraph = 1*

In Fig. 6 is shown the retrieved solution of the system.

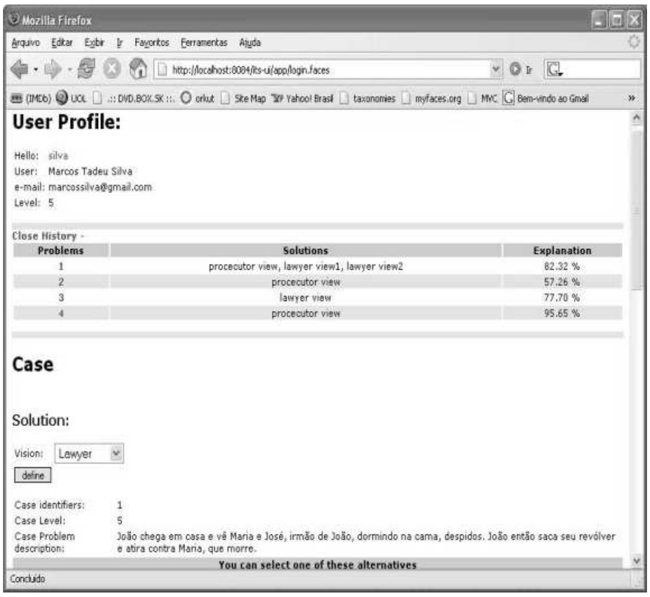

Fig. 6. Prototype resolution.

In the case, three solutions were returned to the ITS. Three SATA Agents were used to solve the case, where each solution represents one agent. The Table 1 with the agents characteristics follows below:

Table 1. Agent characteristics.

Agent	Context
SATA Agent 121_2	Crime versus live
SATA Agent 121_1	Crime versus live
SATA Agent 23	About the Crime

Other characteristic presents in the example is that all SATA Agents knew how to solve the case using the hybrid reasoning. On the other hand, only the SATA Agent 23 found a site to guide the student with others knowledge of the solution. In Fig. 7 is shown a diagram detailing the communication through messages between the agents.
In the Fig. 7, the user only sends the problem to the system to solve it. The agent responsible for receiving this message is the *broker*, that it sends the information to the *Manager Agent*, and after that to the *SATA*. *SATA* infers to solve the problem with its society of agents. After SATA solves the problem, the solution is sent to the *broker agent*, that it sends the solution to

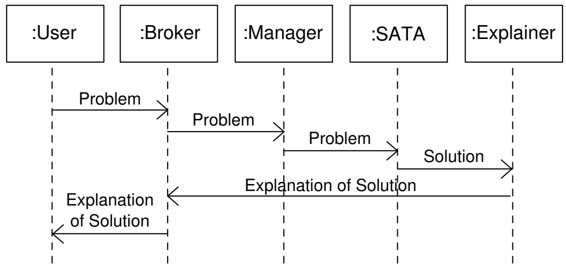

Fig. 7. Agents communication.

the *explainer agent*. The Explainer Agent shows the solution and explains the problem to the student.

6. Final Remarks and Future Work

This paper describes a hybrid ITS which provides human learning by problem posing to Law students and giving those appropriate tutorial feedbacks. The prototype has been used with three types of knowledge domain (Jurisprudence, Normative Knowledge, and doctrines). At the moment, it was described the Case-Based Reasoning model and Rule-Based System that integrate Jurisprudence, Normative Knowledge, doctrines, and the application of the corresponding Legal concepts in the problem solving process. Technologies such as JADE [24], JESS [8], Protégé [21] were used on the development of the prototype. Each one of these tools can be considered as the state of the art in its application domain.

Now, It has been developed the version 2.0 of the ITS. It is planned to this new version: (i) create the strategy structure to the pedagogical model in others parts of the tutor; (ii) create the student modeling structure to the student model, providing a holistic view of each individual student is stored, allowing the tutor to be highly personalized [9]. Finally, it is planned evaluates the current system with undergraduate students to improve the system's robustness.

References

[1] V Aleven, Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment, Artif. Intell., Vol. 150, 2003, pp. 183–237.

[2] V Aleven and K D Ashley, What law students need to know to win, in Proceedings of the 4th international conference on Artificial intelligence and law in (ICAIL '93), New York, NY, USA, 1993, ACM Press, pp. 152–161.

[3] A L Baylor and Y Kim, Simulating Instructional Roles through Pedagogical Agents, International Journal of Artificial Intelligence in Education, Vol. 15, 2005, pp. 95–115.

[4] R A C Bianchi, Uso de heurísticas para a aceleração do aprendizado por reforço, PhD thesis, Escola Politécnica, Universidade de São Paulo, 2004.

[5] L K Branting, Reasoning with portions of precedents, in Proceedings of the 3rd international conference on Artificial intelligence and law in (ICAIL '91), New York, NY, USA, 1991, ACM Press, pp. 145–154.

[6] E Castillo, J M Gutiérrez, and A S Hadi, Expert Systems and Probabilistic Network Models, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.

[7] E B Costa, Um Modelo de Ambiente Interativo de Aprendizagem Baseado numa Arquitetura Multi-Agentes, PhD thesis, Universidade Federal da Paraíba, Campina Grande, 1997.

[8] JESS the Rule Engine for the Java Platform. http://herzberg.ca.sandia.gov/jess/, 2003.

[9] V Kumar and D Brokenshire, An ontological framework to collect and disseminate user model data, in I2LOR: ELearning for the Future: from Content to Services, 2nd Annual Scientific Conference of LORNET research network, Vancouver, Canada, accepted for publication, 2005.

[10] R Lee, Pesquisa Jurisprudencial Inteligente, PhD thesis, Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, 1998.

[11] M L Littman, Markov Games as a Framework for Multi-Agent Reinforcement Learning, in 11th International Conference on Machine Learning in (ICML '94), 1994, pp. 157–163.

[12] K N Martim and I Arroyo, Agentx: Using reinforcement to improving the effectiveness of intelligent tutoring system, in Intelligent Tutoring System in (ITS '04), Maceió, Brazil, 2004, pp. 564–572.

[13] A Martins, E Ferneda, and E Chouraqui, Raciocínio baseado em casos: Fundamentos, Revista Tecnologia da Informação, Vol. 3, 1993, pp. 83–99.

[14] S T Monteiro and C H C Ribeiro, Aprendizagem da Navegação em Robôs Móveis a Partir de Mapas Obtidos Autonomamente, Vol. 1, 2003, pp. 152–152.

[15] V Narayanan and N R Jennings, An adaptive bilateral negotiation model for e-commerce settings, in Seventh IEEE International Conference on E-Commerce Technology in (CEC '05), 2005, pp. 34–41.

[16] T A O'Callaghan, J Popple, and E McCreath, Shystermycin: a hybrid legal expert system, in Proceedings of the 9th international conference on Artificial intelligence and law in (ICAIL '03), New York, NY, USA, 2003, ACM Press, pp. 103–104.

[17] A Oskamp, R F Walker, J A Schrickx, and P H v. d Berg, Prolexs divide and rule: a legal application, in Proceedings of the 2nd international conference on Artificial intelligence and law in (ICAIL '89), New York, NY, USA, 1989, ACM Press, pp. 54–62.

[18] S K Pal and S C K SHIU, Foundations of Soft Case-Based Reasoning, 2003.

[19] E L Rissland and D B Skalak, Cabaret: rule interpretation in a hybrid architecture, Int. J. Man-Mach. Stud., Vol. 34, 1991, pp. 839–887.

[20] G Span, Lites, an intelligent tutoring system for legal problem solving in the domain of dutch civil law, in Proceedings of the 4th international conference on Artificial intelligence and law in (ICAIL '93), New York, NY, USA, 1993, ACM Press, pp. 76–81.

[21] Stanford, Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu, 2000.

[22] G Tesauro, Extending Q-Learning to General Adaptive Multi-Agent Systems, in Advances in Neural Information Processing Systems 16, S Thrun, L Saul, and B Schölkopf, eds., MIT Press, Cambridge, MA, 2004, pp. 215–250.

[23] J Tetreault and D Litman, Using reinforcement learning to build a better model of dialogue state, in 11th Conference of the European Chapter of the Association for Computational Linguistics in (EACL '06), Trento, Italy, 2006.

[24] Tilab, Java Agent Development Framework. http://jade.tilab.com/, 2005.

[25] G Vossos, J Zeleznikow, T Dillon, and V Vossos, An example of Integrating Legal Case Based Reasoning with Object-Oriented Rule-Based Systems: IKBALS II, in Third International Conference on Artificial Intelligence and Law in (ICAIL '91), New York, NY, USA, 1991, ACM Press, pp. 31–41.

[26] C G V Wangenheim and A V Wangenheim, Raciocínio Baseado em Casos, 2003.

Author Bios

Ig Ibert Bittencourt Santana Pinto is an Associate Professor at the Center of Superior Studies of Maceió, Alagoas, Brazil. He received his B.Sc in informatics from Center of Superior Studies of Maceió, Alagoas, Brazil, in 2003. He is pursuing his M. Sc. in computational modeling of knowledge at the Federal University of Alagoas, Maceió, Brazil. His research interests include Artificial Intelligence and law, Case-Based Reasoning, Rule-Based Reasoning and Interactive Learning Environment.

Marcos Tadeu Silva is pursuing his B.Sc in Computer Science from Federal University of Alagoas, Maceió, Brazil. His research interests include Software Engineering, Artificial Intelligence, Game Design, Negotiation, Rule-Based Reasoning and Case-Based Reasoning.

Evandro Costa is an Associate Professor in the Computer Institute at the Federal University of Alagoas (UFAL), Brazil. His research interests include agent and multiagent systems, e-learning and e-commerce systems, groupware, computer supported collaborative learning, software composition, Intelligent Tutoring Systems. He is a member of the Brazilian Computer Society. He obtained his DSc in Electrical Engineering from Federal University of Paraiba, Brazil in december 1997.

Camila Patrícia B. Nunes is pursuing her B.Sc in Computer Science from Federal University of Alagoas, Maceió, Brazil. Her research interests include Software Engineering and Artificial Intelligence.

Rômulo Nunes de Oliveira is an Associate Professor in the Computer Institute at the Federal University of Alagoas (UFAL), Brazil. He received his M.Sc in informatics at the Cognitive and Computational Models Group at the Federal University of Campina Grande (UFCG), Brazil, in 2006. His research interests include Artificial Intelligence, Neural Networks, Machine Learning, Case-Based Reasoning, Rule-Based Reasoning, Interactive Learning Environment, and Automated Negotiation.

Alan Pedro da Silva is an Associate Professor in the Computer Institute at the Federal University of Alagoas (UFAL), Brazil. He received his M.Sc in computational modeling of knowledge at the Federal University of Alagoas (UFAL), Brazil, in 2006. His research interests include Artificial Intelligence, Case-Based Reasoning, Rule-Based Reasoning, Interactive Learning Environment, and Automated Negotiation.