Markov Decision Process Based Multi Agent Learning Style Intelligent Tutoring System For Modelling Students

Achi I. I.
PhD Student
Department of Computer Science, University of Nigeria - Nsukka, Nigeria.
bishopachai@yahoo.com

Prof. Bakpo F.S.
Professor
Department of Computer Science, University of Nigeria - Nsukka, Nigeria.

Prof. Inyiama H.C.
Professor
Department of Electronics and Computer Engineering, Nnamdi Azikiwe University – Awka, Nigeria
inyiama-drhcinyiama@gmail.com

Prof. Inyiama H.C.
Professor
Department of Electronics and Computer Engineering, Nnamdi Azikiwe University – Awka, Nigeria

Agwu C.O.
Lecturer 1
Department of Computer Science, Ebonyi State University - Abakaliki, Nigeria.
emeka2010@yahoo.com

Abstract—In this paper, we propose a decision-path based approach to Intelligent Tutoring Systems (ITSs) that seeks to alleviate the need for extensive development and hand-tuning in the design of such systems. This was borne out of the difficulties the teachers encounter in a conventional classroom where one teacher is designed to tutor several students thereby making it difficult to meet the specific needs of each of the student. Given a set of available learning styles as path to each students knowledge, our approach enables the ITS to track the students’ difficulties and provide the right path to students knowledge at the right time. We model the learning process as a Partially Observable Markov Decision Process (POMDP), where the hidden information corresponds to the student's familiarity with each of the topics to be learned. The student’s progress is monitored from his/her performance in different problem given to each students after each lecture and, depending on this performance, the ITS actively determines which type of learning style should be assigned to the student each time he/she learns. In attempt to test the workability of this new system, we deployed the new system in a typical learning environment and compare the ability of our system to model the students' self-study reactions to the different learning path available. In this new system, four learning styles were used as paths to students' knowledge; learning styles, such as Auditory Learners (Through Hearing), Visual Learners (Through seeing), Kinesthetic Learners (Through Touch or practice) and hybrid (Combination of two or more) were used while modeling the student. Our initial results show that such responses by students being modeled are not trivial to model and that our proposed system which works on POMDP approach better matched the observed student behaviors in comparison with a baseline teaching policy that corresponds to a fix set of actions hand-designed by a human expert.

Keywords—Partially Observable Markov Decision Process (POMDP), decision-path based approach (DPBA), Auditory Learners (Through Hearing), Visual Learners (Through seeing), Kinesthetic Learners (Through Touch or practice) and hybrid (Combination of two or more), Reinforcement Learning (RL)

I. INTRODUCTION
The education community has long been familiar with the use of ITSs to improve the efficiency of teaching [28]. ITSs directly addresses the problem of providing individualized teaching to students, which is one of the main challenges in education. In a traditional classroom environment, one teacher faces multiple students, making it impractical to attend to each student’s specific needs. Instead, teachers commonly opt for giving an average” class, which is known to be less than optimal [12, 21, and 23]. Solving the previous problem is far from trivial, since there are many variables that influence the student’s learning process. Despite the complexity of this area, impressive achievements have already been accomplished, showing that ITSs can improve learning significantly [2, 11, 22, and 29].

One well-known issue with ITSs, however, is the significant effort required to develop such systems [7]. Several works proposed authoring tools that seek to minimize the design effort involved in the development of ITSs, with different levels of success [17]. Recently, an alternative line of work proposed the integration of Reinforcement Learning (RL)-based approaches in ITSs [5, 6, 13, 14, 26, 27]. The main difference between the latter and the former is that, in well-defined environments, RL-based approaches seek to completely eliminate the authoring effort required, instead of just reducing it.
In spite of the achievements of the previously mentioned works, the development cost behind ITSs remains high, since there is a need to design the possible pedagogical actions (such as hints) that the ITS can choose from, depending on the current state of the student's learning process. The cost involved in defining such actions is significant, and constitutes an important bottleneck that probably prevents the massification of ITSs.

In this paper we propose an alternative perspective on the use of ITSs. We argue that it is possible to alleviate the design effort needed to come up with pedagogical actions for such systems, investigating their potential in a novel and unexplored tutoring environment. In particular, we propose that ITSs should actively intervene during the period in which students are doing their self-studies, since the latter is a key point of the student's learning process. Self-study is part of the common learning paradigm in which students complement the traditional classroom environment with periods of self-study with some learning materials. Most existing ITSs are unable to properly accommodate this aspect of the learning process and, instead, focus on repairing the student in an exercise context [10, 11, 22].

In this work we describe the first steps toward an ITS that explores a student's self-study environment. In the designed environment, students work with various learning styles as incorporated in the system so as to determine the best learning style that actually suits the student. This is achieved by presenting the student with exercise or problem after each study. The content of each learning styles will be the same. Our emphasis is really not in the content but in the path, the learning style with which the content is delivered to the students while learning. We designed an ITS based on a decision-theoretic model (POMDP) that is able to successfully track the students' needs and provide the right type of learning style at the right time, in order for the student to overcome his/her difficulties when performing the exercises. One of the benefits of using this methodology is that we transfer the burden of having to manually define help mechanism to the learning materials(content) that already exist to the new system which focuses on the path to student learning(learning styles). Another advantage is that this setup can bootstrap from past experience. For example, the ITS can use information about the students previous learning process to help guide the students while in a similar situation. This is possible with the help of the database system incorporated in the system where all the information as well as their learning processes are stored and to be used subsequently.

THE PROPOSED LEARNING ENVIRONMENT

In our approach, we argued that ITSs should actively intervene during the period in which students are doing their self-studies, since the latter is a key point of the student's learning process. As such, it is essential to analyze how students make use of learning materials through the path to knowledge (learning styles) to surpass their difficulties during the self-study periods. Such analysis provides key insights both on the situations that students have difficulties with and on the learning style they resort to in such situations with no emphasis on the content of the material as some authors have reported in previous literature. If an ITS can successfully model this type of interaction, great educational benefits can be achieved. In this context, we developed a learning environment that enables the aforementioned analysis to be carried out. The topic presented to the students was parts of speech in English course [1]. It is a topic currently taught in the first year of the Computer Science diploma at OSISATECH Polytechnic (GNS101) and one in which students generally presume to have difficulties so as to demonstrate how path to student learning is relevant to the comprehension of a subject matter in any domain. There are eight parts of speech but in attempt to implement this learning style based on learning, we demonstrated with one of them, which is NOUN. The students learn noun by interacting with a web page (Figure 1). The main study material is composed by a set of 3 slides divided in 3 different subtopics:

- Definition of a Noun
- Examples of a Noun
- Types of a Noun

On the bottom right of the slides is a help button that, when pressed, pops up a window with a list of Frequently Asked Questions (FAQs) designed by the course instructor. The different available FAQs are organized by subtopic and are only shown in the window if the student reached that subtopic. Another characteristic of the FAQs is that they can either relate to a specific slide or to some general aspect within the corresponding subtopic. The webpage also includes the possibility of performing drill exercises. Whenever a student goes over all slides within a certain subtopic, a message is presented indicating that a set of exercises will be presented for the student to solve. A total of 3 exercises is provided for each subtopic. An example of an exercise is shown in Figure 2.
All the exercises are multiple choice questions. After the student submits his answer, feedback that indicates whether it was correct or not is provided. Students have multiple attempts to solve each exercise. Additionally, when solving exercises, students have the option of using a button that leads them back to the first slide of the current subtopic. Upon successfully concluding an exercise, the student can then choose to do the next exercise or to move on to the next subtopic. This possibility was provided to allow students with the same flexibility when using the web application that they have in their normal study. The last feature in the designed learning environment is the presence of a table of contents menu on the left side of the web application. The subtopics coloured in black are the ones the student has already concluded or is currently studying. Students can click on these subtopics at any time, which leads them to the first slide of that subtopic. Also, in the current subtopic, students can click on the exercises button to immediately go to the drill exercises. The subtopics in gray are locked until the student completes at least one exercise from the previous subtopic, which means that the order of the subtopics to which the student is exposed is sequential. The path with which the student learns are sequentially arranged in such a way that the student automatically picks one out of the four learning styles at the beginning and subsequently others in a case of backtracking to choose another learning styles when it is necessary. The system works in such a way that if the student fails the exercise after a tutorial, then the next learning will be chosen. After the path selection process, the last action is the decision making process using Markov Decision Process (MDP).

RESULT FROM THE SYSTEM
Having developed the web interface for self-study, we allowed a total of 18 first year Computer Science students to use it in their individual study. The students were invited to carry out their study in a room in which access to alternative materials was controlled, so that the data from each student's study process could be monitored as closely as possible. The interactions of the students with the interface were then used to extract and model the observed study proceedings. The resulting model was used to design and implement an ITS for self-study. In the conducted study, we were particularly interested in monitoring the process by which students addressed their difficulties. Specifically, we want to detect when
a student is faced with a drill exercise that it cannot immediately solve and revisits the tutorial path for a new learning style to successfully address the exercise. Adequately modeling such student is important because if an ITS is able to understand that a student is having difficulties in a given exercise and provides the appropriate learning style, significant learning gains can be expected, specifically with students that would not try to go back to the learning path selection process. Towards the goal of integrating the study model described above in our ITS, the first step is to extract the students’ study responses and identify the one we are interested in. For the purpose of the work in this present paper, we focus on the subtopic of a noun, the first subtopic that raises some real difficulties in terms of path tracking for each student. We categorized the individual slides and video or audio depending on the learning style chosen as provided to the students using the notion of Knowledge Components (KCs), which correspond to the smallest units of knowledge that are being taught (a specific definition or concept), or a combination of knowledge units (representing a more general principle or technique) [19]. The target subtopic was divided in 4 KCs: i. While some students may want to do as many exercises as they can, others prefer to do fewer exercises if they feel that they have successfully learned a certain topic.

<table>
<thead>
<tr>
<th>KC (KNOWLEDGE CONTENT OF EACH PATH)</th>
<th>NO OF EXCERCISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC1 AUDIO</td>
<td>5</td>
</tr>
<tr>
<td>KC2 VIDEO OR SIGHT</td>
<td>5</td>
</tr>
<tr>
<td>KC3 KINESTHETICS</td>
<td>5</td>
</tr>
<tr>
<td>KC4 HYBRID</td>
<td>8</td>
</tr>
<tr>
<td>NONE</td>
<td>27</td>
</tr>
</tbody>
</table>

Table 1: Number of times students needed to review KCi to get an exercise correct.

In order to determine which KCs the student actually reviewed, an average of the time spent on each slide was computed, and those slides in which the student spent a time period inferior to that average were discarded. This filtering was performed because students seeking a specific part of the presentation will rush through the other slides until they reach the intended part. The resulting counts for each of the reviewed KCs are in Table 1, and they were extracted from 8 different students. In the table, the None reference represents the situation where students needed to perform some review, but afterward they would provide the correct answer on the first attempt. The diagram below further explains the access to the Knowledge Content (KC).

In order to determine which KCs the student actually reviewed, an average of the time spent on each slide was computed, and those slides in which the student spent a time period inferior to that average were discarded. This filtering was performed because students seeking a specific part of the presentation will rush through the other slides until they reach the intended part. The resulting counts for each of the reviewed KCs are in Table 1, and they were extracted from 8 different students. In the table, the None reference represents the situation where students needed to perform some review, but afterward they would provide the correct answer on the first attempt. The diagram below further explains the access to the Knowledge Content (KC).

**MODELING STUDY OF THE STUDENTS**

We now describe how the study were modeled using a decision-theoretic model (namely a POMDP). We provide a general overview of this model before introducing our proposed modeling.

**MDPS FOR INTELLIGENT TUTORING SYSTEMS**

[28] Formally a Markov Decision Process (MDP) is a tuple \( \{ S, A, T, r, \} \). The set \( S \) represents the state space, each element of which (a state) describes a particular configuration of the system upon which a decision
must be made. In our case, this state includes all relevant information about the student, corresponding to the traditional student model in the ITS literature. The set \( A \) is the set of possible actions which, in our case, correspond to the tutorial actions that can be taken by the ITS. \( T \) represents the state transition probabilities. \( T(s'|s,a) \) represents the probability of the system moving to a state \( s \) at time step \( t + 1 \) given that the state at time-step \( t \) was \( s \) and action \( a \) was taken. They have this probability is defined for each possible pair of states and actions and then for each possible subsequent state. In our case, this represents the "learning dynamics" of the students, i.e., the probability of a student learning each particular KC upon being presented to the material and the various learning styles by a particular tutorial action. The function \( r \) represents the reward, which encodes the goal of the decision process. In our case, this reward assigns a positive value every time a student shows learning progress. Finally, \( \gamma \) is a discount factor taking values in \([0,1]\). The defined MDP model describes a general stochastic environment in which it is possible to act, in the sense that the successive selection of actions determines (to some extent) how the state evolves. A policy \( \pi \) maps each state in \( S \) to an action in \( A \). Upon fixing a policy \( \pi \) it is possible to compute the value of each state \( s \in S \) as the expected sum of discounted rewards obtained when following policy \( \pi \) from the state \( s \) onward:

\[
V^\pi(s) = r(s) + \gamma \sum_{s' \in S} T(s'|s,\pi(s)) V^\pi(s').
\]

Solving an MDPs consists of determining the policy \( \pi^* \) that maximizes the value of each state. Such a policy is called the optimal policy and the associated value function is denoted as \( V^* \). The optimal value function for a given MDP can be computed iteratively using the recursive relation

\[
V^*(s) = \max_{a \in A} \left[ r(s) + \gamma \sum_{s' \in S} T(s'|s,a) V^*(s') \right]
\]

POMDPs [4, 8,25] extend the MDP model to those situations in which the state of the system is not unambiguously observable. Instead, the state must be tracked from noisy observations. In the process of modeling the learning process of a student, POMDPs offer a more adequate modeling framework, since the knowledge acquired by a particular student cannot be directly observed, but must be tracked by the student's performance in the drill exercises. A POMDP can be described as a tuple \((S,A,O,T,O,r,\gamma)\), where \( S, A, T, r \) and \( \gamma \) are as in a MDP. The set \( O \) includes the observations available to the decision-maker (in our case, the ITS). Represents the observation probabilities. In particular, \((o|j,s,a)\) represents the probability of making observation \( o \) when at state \( s \) and action \( a \) was taken. Although computationally harder than MDPs, POMDPs can also be solved exactly using dynamic programming, or approximately using any of a wide range of available methods [14, 16, 24]. Most such methods involve tracking of the underlying state of the system in the form of a probability distribution known as the belief, since it roughly expresses, at each time-step, the decision-maker's belief about the underlying state of the system. Using the previous framework, we obtain the following model.

i. States are defined through the assignment of 0 or 1 values to four different state variables: \( s = (sKC1 ; sKC2 ; sKC3 ; sKC4) \). Therefore, the state space \( S \) corresponds to all combinations of state variable assignments. Also, the state variables match the identified KCS and their value represents that students either know the KC or not.

ii. The set of possible actions is defined as \( A = (scKC1 ; scKC2 ; scKC3 ; scKC4 ; ex) \). The scKC actions correspond to showing content to the student regarding the corresponding KC.

Figure 4: CPTs Examples for the action that corresponds to content showing the learning style for KC4.

These actions only represent a situation where the ITS has to provide learning in order to help the student, thus, it does not provide information about a particular learning path (there are several different slides related to the same KC, four in number). The result corresponds to give the student an exercise to solve. The fact that there is a single exercise action indicates that there is no distinction between the
different possible exercises, since all of them address all KCs.

iii. The observation model is defined as \( O = \{ \text{correct}; \text{ incorrect}; \text{ none} \} \). The first two observations can only be obtained when the action \( ex \) is taken, and correspond to the student's answers. The none observation is obtained when a \( scKC_i \) action is performed. It represents the fact that it is not possible to directly observe the impact of showing content to students.

iv. The reward model \( r \) provides a positive reward of 1 when the students are in the state that corresponds to knowing all the KCs.

DEMONSTRATING POMDPs

Since the POMDP model used a featured state-space \( S \) which can be factored into four subsets, each corresponding to one of the state variables \( SC\,KC_i \), it is possible to leverage POMDP solution methods that take advantage of such factored representation [3]. In particular, the factorization of the POMDP allows expressing the state space in a more compact manner, by representing the action's effect using a two-slice temporal Bayes net [9, 15, 18, 2, 27]. This means that we have a set of nodes (one for each KC) representing the state prior to the action, another set of nodes representing the state after the action, and directed arcs that indicate the casual influence of the two sets of nodes. This influence must be defined for each different possible action. The adopted modeling approach is linear, in the sense that each state variable is only affected by itself, as it is possible to observe in Figure 3.

<table>
<thead>
<tr>
<th>Exercises</th>
<th>Correct Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise1</td>
<td>85 percent</td>
</tr>
<tr>
<td>Exercise2</td>
<td>80 percent</td>
</tr>
<tr>
<td>Exercise3</td>
<td>83 percent</td>
</tr>
<tr>
<td>Exercise4</td>
<td>78 percent</td>
</tr>
<tr>
<td>Exercise5</td>
<td>56 percent</td>
</tr>
<tr>
<td>Exercise6</td>
<td>72 percent</td>
</tr>
</tbody>
</table>

Table 2: Percentage of correct answers in Noun subtopic of English.

Linked to the expressed influence is the specification of a Conditional Probability Table (CPT) that encodes the system's dynamics. In Figure 4 there are two examples of CPTs for the \( SC\,KC_i \) action. The CPT models in a probabilistic way the change of the values of the state variables when content relative to KC4 is shown. In our model these probabilities were computed from the data obtained in our study, namely the percentage of correct answers in each exercise (Table 2). From this data it is possible to observe that Exercise 5 has a specifically low percentage of correctness. The two most likely candidate KCs that explain this situation are KC3 and KC4, since in the exercise there is a significant increase in the size of the tree to be analyzed by the students. The course instructor argues that most students will have more trouble with the application of the formula in KC4 rather than knowing the height of a tree (KC3). Therefore, it was considered that it is harder to acquire KC4 than KC3, and, thus, in the CPT of action \( scKC4 \) the probability of the variable \( SC\,KC_i \) changing to 1 is lower than the probability of a similar transition in \( SKC_3 \) for the CPT of the \( SC\,KC_i \) action (the concrete values are 0.8 and 0.85 respectively). These same values for the KC1 and KC2 were both set to 0.9 probabilities. This was due to the fact that if these path to KCs were hard to acquire it would be noticed in the percentage of correct answers in the first three exercises. In what respects the CPTs for the exercise action, if \( skKC = 0 \) the corresponding state variable will deterministically continue in that same state. For the opposite case, our main concern was to enable the system to shift its dynamics such that the \( SC\,KC_i \) action is not too much frequently tried, and allow other actions to be explored. In this context, when \( skKC = 1 \) the probability of making a transition to \( skKC = 0 \) is 0.85. This same type of transition for KC1; 2 is set to 0.9, which again is related with percentage of correct answers in the exercises. Finally, for KC4, the probability is set to 0.95. The last CPTs that need to be defined are for the observation model. If the students can use all the paths to KCs and an exercise action is made, then a correct answer is observed with 0.9 probability (the student can always make a mistake). For the case that students do not know some path to KC, an incorrect answer is observed with 0.75 probabilities, which corresponds to a random guess, since there are four possible answers.

When the action is of the type \( SC\,KC_i \), then the none observation is always obtained. Although the number of possible state is just 24 = 16, we still opted for the factored POMDP approach in order to assess its effectiveness, so that we can determine if it is an option for the full domain, in which the number of possible states is actually a problem (there are 19 KCs).

EXPERIMENTS

In the following sections, the experimental setup that defines the learning environment for an ITS to act and the corresponding obtained results are described.

EXPERIMENTAL SETUP

The environment for the first experiments uses the study result gotten from Section above-result from the
system to create tutorial situations where the ITS must perform the appropriate action. These situations correspond to the interactions students made until they provide a correct answer. If the student did not perform any review, and the ITS gives the student an exercise to solve, he/she will provide the correct answer. In case the student performed some review, he/she will only get the exercise correct after the appropriate show content actions are provided by the ITS, which corresponds to the actions that match the reviewed path to KCs. Using this setup it is possible to calculate the minimum number of actions that an ideal ITS would do.

In order to have a baseline comparison, a scripted ITS was designed to act in the defined learning environment. Such ITS always performs the same sequence of actions: it proposes an exercise to the student. If the student fails, a $KC_3$ action takes place, followed by another exercise. The order of the different actions was defined beforehand by the instructor, according to his own perception of what path to KCs usually present more difficulties of understanding and also by analyzing the data regarding the percentage of correct answers in the exercises (Table 2). This order is summarized as KC4, KC3, KC2, KC1. We then used the Symbolic Perseus software [20] to solve the POMDP obtained from the student model described in Section 5. The POMDP was initialized to a pre-defined initial belief state, in which the student is assumed to know all KCs except KC4, reflecting the fact that we are analyzing students who at some point made a review and also because a significant percentage of incorrect answers to the exercises were caused by this path to KC.

EXPERIMENTAL RESULT

In this section, the results obtained using the previous experimental setup are reported. The comparison between the percentage of actions, regarding the optimal ITS, used by the baseline and POMDP ITSs is in Figure 4. In this context, the goal of the compared ITSs is to achieve a 100% score, meaning that they would have used an optimal number of actions. The results show that the POMDP consistently obtained a better performance than the baseline with the exception of Student#8. The last column of Figure 4 shows that the baseline spent, in total, twice more actions than the optimal strategy, whereas the POMDP model only used 140% of the actions - 28 less actions than the baseline, and 27 more than optimal. Also, the non-overlapping error bars indicate that these results are statically significant. When inspecting the reviewed path to KCs it is possible to observe that the majority of them include all 4 paths to KCs. This happens since the first slide contains the formal definitions of all paths to KCs. In this situation, the POMDP model has a better teaching policy, because if the student still gives wrong answers after an action of showing content, it uses consecutive show content actions of the remaining paths to KCs before trying another exercise, which proved to be more suitable than always alternating between exercise and show content actions. Another important aspect to mention from the used data is that in order to achieve optimal performance it is necessary to take into many account different states trajectories, since students might have difficulties in distinct sets of the paths to KCs. In fact only two students showed the same behavior throughout the exercises of the analyzed subtopic. Some of these behaviors are particularly hard for scripted strategies to cope with, such as having students reviewing some paths to KC to get the current exercise right, but in the next one student still review other paths to KCs, even though all exercises require all paths to KCs in order to be solved. For these cases the POMDP approach is advantageous since it automatically computes different teaching policies to be followed according the current Belief-state of the student.

![Figure 5: Performance comparison between the baseline and the POMDP ITS.](image)

CONCLUSIONS

In this work we presented a methodology for studying students self-studies reactions, with the ultimate goal of bootstrapping an ITS that can learn what is the appropriate tutorial action to take in each particular situation. This problem was tackled from a perspective of identifying what path to KCs the student has not mastered and guiding them to the appropriate path; learning style. The preliminary evaluation in one of the subtopics of the domain showed that the POMDP approach was more efficient than the baseline version. This result is important since it shows that a fix teaching policy, even when provided by a human expert, is not optimal and also that there are other sub-optimal policies, such as the one presented in this work, that can achieve better performance. The explanation for this is that the optimal teaching paths are different from student to student, and, thus, they have deferent levels of mastery of the paths to KCs when starting to solve the exercises. This leads us to the conclusion that there are expectable performance gains to be obtained if the initial belief state of the model is not the same for all students, as it is currently. In order to calculate a more adequate initial belief state, one possibility is to use the information...
regarding the time students spent in each path to KC prior to the exercises. Another important conclusion from this work is that it is indeed possible to alleviate, to some extent, the authoring effort required to design ITSs. This was done by using a POMDP model that automatically computes a teaching policy for every possible state and also provides a mechanism to update the estimation of what state the student is in according to the gathered observations. Obtaining such automation is important since it deals with the time consumed and hard task of the system’s design. This is related with the fact that although the expert is able to provide an order of difficulty for a set of the path to KCs, specifying some concrete teaching policy to follow when certain conditions are met in a much harder task. It should also be noted that just defining the path to KC difficulty order is not a trivial task. As mentioned previously, the need of authoring work has not been eliminated completely. However, it is necessary to specify a specific path to KCs structure to back up the POMDP model. This structure can be complex and there are multiple possibilities for both the shape of the structure and what the actual path to KCs is. Even for our simple domain this type of doubt was certainly felt. In tackling this challenge, the use of Natural Language Processing might allow the system to obtain the path to KCs structure automatically. Another important improvement would be to make the CPTs estimation data driven, so that this process becomes automatic and possibly closer to optimal.

REFERENCES