A Cognitive Theory in an Authoring Tool for Intelligent Tutoring Systems

Maria Virvou
Department of Informatics,
University of Piraeus
80 Karaoli & Dimitriou St., Piraeus, 18534, Greece
mvirvou@unipi.gr

Abstract—In this paper a knowledge-based authoring tool is described. The authoring tool is called NEITS-Author and may be used by human instructors to author their own Intelligent Tutoring System (ITS) in the domain they are interested in. NEITS-Author provides a user-friendly environment for instructors to author lessons and create tests for the students. The main feature of the resulting ITSs is their ability to model both the learners' level of factual knowledge on a particular domain and their reasoning abilities which are domain-independent. Learners are modelled based on a cognitive theory that aims at formalizing human reasoning patterns and is called Human Plausible Reasoning. A domain-independent adaptation of parts of this theory has been incorporated in the authoring tool so that each ITS may use it in its learner modelling component.

Keywords: Authoring Tools, Intelligent Tutoring Systems, student modeling.

1. INTRODUCTION

Computer Assisted Learning (CAL) is the area of software development that deals with educational software applications. This area has grown enormously during the past decades and has been enhanced by the recent advances in web-based applications, multimedia technology, intelligent systems and software engineering. A whole fast growing field of computer-based education is gaining interest among Educational Institutions that tend to incorporate Information Technology in their educational services.

However, many researchers (e.g. [1, 2]) point out that technology is effective when developers thoughtfully consider the merit and limitations of a particular application while employing effective pedagogical practices to achieve a specific objective. For example, multimedia educational products are often criticized that they use the narrative mode, or unguided discovery, neither of which supports the learner well nor exploits the capability of the medium [3, 4, 5].

In view of the above criticisms, there is a research goal that multimedia systems be more “intelligent” and adaptive to the learner’s demands, abilities and knowledge [6]. Intelligence and adaptivity to learners’ demands may be provided by Intelligent Tutoring Systems (ITSs). Indeed, ITSs are very good at providing individualized instruction and support to students, because they are designed to know what they teach, who they teach and how to teach. ITSs are computer-based learning systems, which attempt to adapt to the needs of learners and are therefore they may be considered as the only such systems, which “care” about learners in that sense [7].

It has been widely agreed that an ITS should consist of four components, namely the domain knowledge, the student modelling component, the tutoring component and the user interface [7, 8, 9]. The domain knowledge consists of a representation of the domain to be taught (e.g. Biology, Chemistry, etc.). The student modelling component involves the construction of a qualitative representation that accounts for student behaviour in terms of existing background knowledge about the domain and about students learning the domain [10]. The tutoring component contains a representation of the teaching strategies of the system. Finally the user interface is responsible for translating between the system’s internal representation and an interface language understandable to the student.

ITSs are quite good at providing dynamic aspects to the reasoning ability of educational applications. This is mainly due to their student modelling component that aims at gaining an understanding of how a student learns and what the student’s misconceptions may be.

However, ITSs have often been criticised that they are very difficult and expensive to build and that they are so few because their construction is immensely complex (e.g. [11, 12]). Moreover they cannot be reused for a domain other than the one they had originally been constructed for. Therefore, recently, a lot of research energy has been put into the development of knowledge-based authoring tools for the construction of multiple ITSs. Such authoring tools provide authoring environments where human instructors may be given a combination of facilities to produce visually appealing, interactive screens and a deep representation of content and pedagogy [13]. Moreover, prospective authors do not have to be software or knowledge engineers to produce their own ITS.

Authoring tools may be used for multiple domains by a multitude of instructors. Hence their reusability is ensured. However, the fact that they have to be useful for a multitude of domains and a multitude of authors, renders their own construction more difficult than the construction of ITSs. The main problem is that the methods which may be used in the authoring tools have to be as domain-independent as possible and at the same time the resulting ITSs should support the individual learners’ needs in every particular domain.
This paper investigates the application of a cognitive theory as a domain-independent reasoning mechanism for providing user modelling capabilities to a knowledge-based authoring tool. The authoring tool is called NEITS-Author, which stands for Negotiating ITS Author. NEITS-Author may be used by instructors to create ITSs where students may be presented lessons in an attractive way. Then the students’ knowledge may be assessed by tests which consist of questions. Students are expected to answer questions and justify their reasoning. In this way, the resulting ITSs may have the ability to negotiate with students on the students’ answers to questions of tests. This may be achieved by the resulting ITSs because they are able to model both the students’ knowledge of the domain being taught and their reasoning ability within this domain. In this sense, the authoring tool constructs environments where there is opportunity for a negotiating teaching-learning dialogue between the ITS and the students.

The cognitive theory aims to model human reasoning and is called Human Plausible Reasoning [14], henceforth referred to as HPR. HPR formalises the plausible inferences based on similarities, dissimilarities, generalisations and specialisations that people often use to make plausible guesses about topics that they know partially. These inferences may lead to either correct or incorrect guesses; in any case these guesses are plausible. HPR has been adapted and used previously in learning environments for novice users of UNIX [15, 16] and for novice users of a Graphical User Interface [17, 18]. However, it has not been previously incorporated in an authoring tool to provide domain-independent reasoning abilities.

The incorporation and adaptation of this theory in the authoring tool aims at providing ITSs with the ability to follow the students’ reasoning and create teaching-learning dialogues. Collaborative discourse is an issue that has attracted a lot of research energy in the recent years (e.g. [5, 19, 20, 21, 22]). As Ohlsson [23] points out, the analysis of epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.) are more relevant for higher order learning epistemic activities (arguing, describing, explaining, etc.)

II. THE COGNITIVE THEORY

The core theory consists of a set of primitives and a set of inference rules.

1. A set of primitives. A summary of the primitives is given in Table I.

| arguments $\alpha_1, \alpha_2, f(\alpha_1)$ | e.g. Sam, whale, Sam’s food |
| descriptos $d_1, d_2$ | e.g. size, animal-type |
| terms $d_1(\alpha_1), d_2(\alpha_2), d_3(d_1(\alpha_1))$ | e.g. animal-type(Sam), size(whale), size(animal-type(Sam)) |

| referents $r_1, r_2, \{r_3, \ldots\}$ | e.g. whale, large, large plus other sizes. |

The primitives can be classified into four groups:

- **Statements representing people’s beliefs about the world.**
- **Statements involving relations** (i.e. GEN, SPEC, SIM, DIS). These represent different relationships among concepts in hierarchies.
- **Relational expressions** which are either mutual implications or mutual dependencies. These represent people’s approximate knowledge about what depends on what, which can be specified with more or less precision.
- **Certainty parameters** that act to condition these three kinds of expression and which affect the certainty of the different inferences described in the next two sections.

The primitives consist of:

- **Basic expressions**: statements and relational expressions.
- **Operators**.
- **Certainty parameters**.


- **Statement transforms**.
- **Transforms based on dependencies and implications**.

The simplest class of inference are called statement transforms. If a person believes some statement such as that the flowers growing in England include daffodils and roses, which translates to

| flower-type(England) = daffodils, roses,… |

there are eight statement transforms which allow plausible conclusions to be drawn. The argument transforms move up, down or sideways in the argument hierarchy using GEN, SPEC, SIM or DIS respectively. The referent transforms do the same in the referent hierarchy.

The formal representation of the similarity statement transforms, which are quite important is the following:

**SIM-based argument transforms**

$\phi = r_1 \cdot \gamma_1, \mu_0$

$\alpha \xrightarrow{\text{SIM}} a \text{ in } CX(A, D(A)): \sigma, \gamma_2$

$D(A) \leftrightarrow d(A): \sigma, \gamma_3$

$A, \alpha \xrightarrow{\text{SPEC}} A: \gamma_4, \gamma_5$

---

**TABLE I: HPR’S ELEMENTS OF EXPRESSIONS**

| arguments $\alpha_1, \alpha_2, f(\alpha_1)$ | e.g. Sam, whale, Sam’s food |
| descriptos $d_1, d_2$ | e.g. size, animal-type |
| terms $d_1(\alpha_1), d_2(\alpha_2), d_3(d_1(\alpha_1))$ | e.g. animal-type(Sam), size(whale), size(animal-type(Sam)) |

| referents $r_1, r_2, \{r_3, \ldots\}$ | e.g. whale, large, large plus other sizes. |

The primitives can be classified into four groups:

- **Statements representing people’s beliefs about the world.**
- **Statements involving relations** (i.e. GEN, SPEC, SIM, DIS). These represent different relationships among concepts in hierarchies.
- **Relational expressions** which are either mutual implications or mutual dependencies. These represent people’s approximate knowledge about what depends on what, which can be specified with more or less precision.
- **Certainty parameters** that act to condition these three kinds of expression and which affect the certainty of the different inferences described in the next two sections.

The primitives consist of:

- **Basic expressions**: statements and relational expressions.
- **Operators**.
- **Certainty parameters**.


- **Statement transforms**.
- **Transforms based on dependencies and implications**.

The simplest class of inference are called statement transforms. If a person believes some statement such as that the flowers growing in England include daffodils and roses, which translates to

| flower-type(England) = daffodils, roses,… |

there are eight statement transforms which allow plausible conclusions to be drawn. The argument transforms move up, down or sideways in the argument hierarchy using GEN, SPEC, SIM or DIS respectively. The referent transforms do the same in the referent hierarchy.

The formal representation of the similarity statement transforms, which are quite important is the following:

**SIM-based argument transforms**

$\phi = r_1 \cdot \gamma_1, \mu_0$

$\alpha \xrightarrow{\text{SIM}} a \text{ in } CX(A, D(A)): \sigma, \gamma_2$

$D(A) \leftrightarrow d(A): \sigma, \gamma_3$

$A, \alpha \xrightarrow{\text{SPEC}} A: \gamma_4, \gamma_5$
d(a) = r: γ = f(γ1, φ, µ, σ, γ2, α, γ3, γ4, γ5)

E.g. livestock(West Texas) = cattle, ..., γ1 = high, φ = high, µ = high
Chaco SIM West Texas in CX(region, vegetation(region)): φ = moderate, γ2 = moderate
vegetation(region) ↔ livestock(region): α = high, γ3 = high
West Texas, Chaco SPEC region: γ4 = high, γ5 = high
livestock(Chaco) = cattle, ..., γ = moderate

SIM-based referent transforms

d(a) = r: γ1, φ, µ, σ, γ2, α, γ3, γ4, γ5
r' = SIM r in CX(d, D(d)): σ, γ2
D(d) ↔ A(d): α, γ3
A SPEC A: γ4

d(a) = r': γ = f(γ1, φ, µ, σ, γ2, α, γ3, γ4, γ5)

E.g. Sound(wolf) = howl, ..., γ1 = high, φ = high, µ = low
Bark SIM howl in CX(sound, means of production(sound)): σ = high, γ4 = high
Sound(wolf) = bark, ..., γ = moderate

Human Plausible Reasoning has been applied in the authoring tool as an underlying reasoning mechanism that simulates learner’s reasoning when they answer questions and they do not have a ready answer. Statement transforms are used to show what possible errors s/he may have made. Certainty parameters are used to represent the degree of confidence of the system concerning the inference beliefs.

III. OVERVIEW OF NEITS-AUTHOR

NEITS-Author works at two levels. On the first level it takes input from the instructor and it creates the knowledge-base of the domain to be taught. This knowledge-base is incorporated into the ITS that will be used by the students. At the second level, the resulting ITS interacts with the students and provides feedback adapted to their individual needs. This feedback is based on the error diagnosis performed by the ITS on the students’ answers to questions. This diagnosis concerns both the student’s knowledge level of the domain and his/her reasoning skills within this domain.

All domain-dependent data is given by the human instructor to the system during the authoring procedure which is illustrated in Fig. 1. NEITS-Author provides a user-friendly environment where the author may create lessons to be presented to students and may provide a concept-map of the domain which is stored in the form of hierarchies in the knowledge-base about the domain.

Fig. 1 Authoring procedure

The resulting ITS contains a knowledge-base about HPR which is the same for all ITSs since it is based on the domain-independent reasoning mechanisms of the cognitive theory. Moreover, it contains the knowledge-base about the domain which has been created during the authoring procedure. These knowledge-bases are used to process the input of students while they answer questions to tests as illustrated in Fig. 2.

Fig. 2 Student interaction with the resulting ITS

IV. CREATING INTELLIGENT TUTORING SYSTEMS

The initial input to the authoring tool is a description of knowledge concerning a specific domain given by a human teacher who is acting as an author. At first, the domain has to be described in terms of hierarchies, which constitute the knowledge representation of HPR. Therefore the author has to decide what the main concepts of the lesson are, that may be represented in hierarchies. Then s/he may create hierarchies by giving data to a dialogue box of the system. After this input has been given, the tool constructs a knowledge base concerning the specific domain in the form of hierarchies. Then the author inserts facts that s/he wishes to be taught to students and which are relevant to the main
The resulting Intelligent Tutoring Systems may be used by students who can be shown facts from the knowledge base. In addition, students may test their knowledge by answering the questions that the authoring tool has formed, based on the data given by the author. The questions that assess the students’ knowledge and reasoning skills, may be answered by a student when s/he is working in one of two different modes, the instructor mode or the co-learner mode. The underlying reasoning used for both modes is practically the same and is based on HPR. The difference of these modes is restricted to the interface level.

The student is asked to give an answer and then is also asked to give a justification for this answer. The student modelling component examines the correctness of the students’ answers in terms of the students’ factual knowledge and reasoning that they have used. Information about each student concerning his/her knowledge and reasoning ability, is recorded in his/her long term student model [24]. The long term student model keeps a history record of the student and is updated every time the student completes a test. The long term student model is used to adapt the presentation of lessons to the particular student’s knowledge and possible weaknesses.

V. STUDENTS’ JUSTIFICATIONS

When a student is asked a question by the ITS, s/he is expected to give both an answer and a justification for this answer. For example, in the ITS for zoology that will have resulted from the above authoring example, the student may be asked the following question: “Do ducks have broad wings?” Then the student may give either a positive or negative answer. In any case the student is also expected to provide a justification for that answer.

If the student believes that s/he knows the answer, s/he gives a justification of the following type: “Yes” or “No”, “because I know so”. If the student does not have a ready answer because s/he may not possess the appropriate piece of factual knowledge for this answer, s/he is asked to make a plausible guess and justify it. For example, the student may answer: “My guess is yes. I know that geese have broad wings; geese are similar to ducks with respect to their characteristics. Therefore, it is likely that ducks have broad wings too.” The student’s guesses may be correct or incorrect. However, the reasoning that s/he uses may reveal whether the student has a good knowledge of zoology and whether s/he is able to use it plausibly.

The correctness of immediate answers, guesses and/or justifications are recorded to the long term student model. By the end of the test, the system can have a good idea of what the student has learned and how well s/he uses this information to answer questions that s/he may not know precisely. In particular, there may be several cases of combinations in terms of the correctness of a guessed answer and the correctness of the justification:

1. Both the guessed answer and the justification are correct.
2. The guessed answer is wrong but the reasoning in the justification is correct.
3. In this case the system informs the student about the mistake but admits that the reasoning was plausible and the answer was close to the correct one.
4. The guessed answer is correct but the justification is incorrect.
5. In this case the system informs the student that the answer was correct by chance, therefore the student does not get any credit for this answer.
6. Both the guessed answer and the justification are incorrect.

For the cases 3 and 4 where the justification is incorrect, there are two possibilities as to the reason why the justification is incorrect:

1. The justification is incorrect because the reasoning has been completely incorrect.
2. The reasoning is partly correct but the factual knowledge of relevant topics that has been used is incorrect. For example, if the student was asked whether ducks had long-pointed wings and the student answered: “My guess is yes because geese have long-pointed wings and ducks are similar to geese with respect to their characteristics”. The student’s reasoning would have been based on incorrect facts. In fact ducks are similar to geese but ducks do not have long-pointed wings.

In this way, for each question the ITS takes into account a broader part of the student’s knowledge and skills than appears in the question alone. Indeed, as Andriessen and Sandberg point out [25], the process of becoming an expert in a certain domain should no longer be solely viewed as the acquisition of a representation of correct knowledge; the knowledge to be acquired should flexibly manage open problems. Thus, if designers of future tutoring systems wish to capitalise on the knowledge gained from human tutoring studies, the next generation of tutoring systems will incorporate pedagogical agents that engage in learning dialogues [22].

VI. VIRTUAL INSTRUCTOR AND CO-LEARNER

In the resulting ITS, the reasoning of the system may be used by two different animated agents, the virtual co-learner and the virtual instructor. Both agents operate at the interface level. In particular, in the instructor mode, there is an animated agent figuring a tutor that provides rather formal messages and is responsible for assigning a grade to the student whereas in the co-learner mode, there is an animated agent figuring a peer that provides messages in a more casual way.

The mode of the co-learner has been considered quite important by many researchers for the purpose of improving the educational benefit of tutoring systems. For example, Van Lehn et al. [26] argue that students can improve their learning in collaboration with a simulated student, because the simulated student can be simultaneously an expert and a co-learner, it can scaffold and guide the human’s learning in subtle ways.

Furthermore, in general, the presence of an animated speaking character in the interface of an educational application attracts the students’ attention and thus increases their engagement and motivation. Indeed, previous studies [27, 28] have shown that the presence of a life-like character in an interactive learning environment – even one that is not expressive – can have a strong positive effect on student’s perception of the learning experience. When these life-like characters are combined with an underlying reasoning mechanism that is based on human plausible reasoning then the interaction may become more human-like and appealing to students than it currently is for many systems.

VII. CONCLUSIONS

In this paper we described NEITS-Author which is a knowledge-based authoring tool. NEITS-Author provides a user-friendly environment to instructors who wish to create their own ITS on a certain domain. The resulting ITSs incorporate a reasoning mechanism which is based on the domain-independent cognitive theory, called HPR. The reasoning mechanism is used in the knowledge bases and inference mechanisms of the ITSs and may assess both the students’ knowledge on a certain domain and their reasoning skills. In this way students are given the opportunity to argue, describe, explain and predict about facts in a certain domain and thus become experts in this domain while they practice their human plausible reasoning skills which may be useful to them for other domains as well.

The incorporation of the domain-independent cognitive theory in the authoring tool may be useful both for students in many domains and for instructors of many domains. Students who use an ITS in a certain domain may be helped to develop their own domain-independent cognitive skills which are useful for any domain. On the other hand, instructors who teach these students may have a detailed knowledge of what their students know. At the same time the authoring tool allows the construction of many cost-effective ITSs and ensures reusability in the knowledge-based software engineering process.

REFERENCES


