Development of an intelligent tutor dialogue using finite automatas

Raúl Ernesto Gutierrez de Piñerez Reyes, Milena Maiguel Villalba, Pedro E. Vera
Departamento de Ingeniería de Sistemas
Facultad de Ingenierías y Arquitectura, Universidad de Pamplona
Campus Universitario, Pamplona, Colombia
Telephone: (0975)5685303 Ext. 155 e-mail: {dsistemas, milena_maiguel}@unipamplona.edu.co

Abstract. This paper show the importance of the application the finite automata in the construction of user's interfaces as integral component of an Intelligent Tutor Dialogue (I.T.D).

The recognition of expressions based on finite automatas are main part of a interface module for an intelligent tutor dialogue developed that guides and orient in the demonstration of base theorems in the theory of groups.

Introduction

The construction of Intelligent Tutorial Dialogues (ITD) as support to teaching and learning tasks is nowadays one of the big challenges of the Artificial Intelligence (A.I), their effectiveness even to check, open a space in the investigation as support to the design, development and setting in these intelligent tutors' march. Since the first developments the *Excheck* (McDonald, 1981) and the *Teacher Apprentice* (Lewis & Anderson, 1985) the treatment of natural language has been one of the stones in the shoe, the control of knowledge of the every the system should be subordinated to the inputs and answers of the interaction student – machine, all the modules of an intelligent tutor should act around the interface module.

Under this premise we began the construction of the interface module for an Intelligent tutorial dialogue carrying out certain analyses to patterns of answers that are presented by the students when making a demonstration of some basic theorems of the theory of groups in the University of Pamplona, achieving for this develop an recognition of expressions based on finite automata non deterministic so that gave the possibility to the tutor of having an interface more guided to the treatment of natural language.

On the other hand in this paper make reference to the use of components of intelligent software for the development of intelligent dialogues, like a technology more than it allows to centralize and to control in a better way the knowledge of the system, also integrating languages of high level and languages of logical programming as Prolog to facilitate with this the creation of friendly interfaces and with a bigger action radio as for the domain of knowledge. In this paper, we first describe the development of the intelligent tutor dialogue, playing important topics as the delimitation of the domain of knowledge, the categorization of errors, the architecture of the dialogue, and the description of the dialogues, and finally we enter to explain the educational design of the ITD.

1. Description of the Intelligent Tutor Dialogue development.

For development of the ITD it is taken into account the phases of identification of the problem, search of the public and private knowledge, definition of the proof, specification of the knowledge domain, architecture of the dialogue, classification of errors for structuring the base of the knowledge and the description of the dialogue.

1.1 Search of the public and private knowledge

The private knowledge managed in the ITD is based on the experience of teachers with more than 20 years in the demonstration of theorems of group theories at the University of Pamplona. To support the public knowledge we applied a proof to students for the demonstration of the theorem: $A \subseteq B \Leftrightarrow A \cap B = A$. following with the specification of the knowledge, we entered to define the domain.

1.2 Domain of knowledge

The domain of the knowledge is based on three demonstrations of the theorem $A \subseteq B \Leftrightarrow A \cap B = A$, for the direct and contradiction method. Defining for each demonstration method a dialogue. In the direct method it is demonstrated that if $A \subseteq B$ then $A \cap B \subseteq A$ and on the other way if $A \subseteq B$ then $A \subseteq A \cap B$.

Now by contradiction it is demonstrated that if $A \subseteq B$ then $A \cap B \neq A$. Initially delimiting the ITD to a basic theorem of the theory of groups as the described previously.

1.3 Category and description of errors of the proof

According to the obtained results of the proof carried out we categorized the errors in three big types:

- The first error type with respect to the inadequate identification and interpretation of the hypothesis and thesis of the theorem had a representation of 55% of the sample, where 40% of errors was made when the student could not distinguish the hypothesis and the thesis of the theorem.
- The second error type with respect to the ignorance of a demonstration method had a representation of 40% of the sample, due to the students made the error of demonstrating the hypothesis starting by the thesis.
- The third error type with respect to the ignorance of the double contenencia in the demonstration, had a representation of 5% corresponding to the total sample, where the student was limited to try to demonstrate so alone an a part of the contenencia .

With base in these classification of errors we opt to build the mathematical expressions and the rules for the types of answer and for the information of diagnose of the student.

1.4 Clasification of the answers

As an example of the possible answers that the student will be able write in the first line of the dialogue for the demonstration: $A \subseteq B \Leftrightarrow A \cap B = A$, the mathematical expression is $x \in A \cap B$, we define the following expressions, that are answers for the student:

- Optimal answers: x A B , Let's suppose that x   A B, Be x   A B, We have that x   A B , let's begin the demonstration saying that exists an element x in A B ,…..
- Correct answers: Let's be x A , Let's suppose that x A y x B,   That x B y x A,...
- Correct answers but undue: _______x ___ A _____B, let's be ____ x   B_____ y_______ x   __A, let's be _____that_____ x ___ A, , Let's be x___   (____ A___   B),.....
- Incomplete answers: x A, Let's say that x B,.......
- Incorrect answers: X A, x A o x B, Let's be x a, then x b,.....

We make clarity that this type of expressions means another type of answers in other lines of the dialogue, although the same classification of answers is conserved, this allows that the finite automatas be used for the recognition of these expressions.

1.5 Architecture of the Intelligent Tutor Dialogue

Our ITD is a dialogue that guide and orient in the demonstration (for the methods direct and contradiction) of the theorem **A B   A B=A,** the tutor comprise of a pedagogic module in charge of the managing of every the system, furthermore they are; the subsystem of dialogue, a module of diagnostic information of the student in charge of establishing the state of student's knowledge, a knowledge base that has the knowledge on the demonstration and the recognition of the automata, the subsystem   of explanation and lastly the interface module.

In the intelligent dialogue, the dialogue subsystem is the that carries out all the calls of input and output through of extended predicates in prolog, that in a bidirectional sense they go to the users interface and the pedagogic module. The user interface takes the input expressions of the student, makes the recognition of the expression and carries out the filtration of the expression determining which of these is the correct, and this way appeal to the explanation subsystem for its respective answer, in this moment the student model generates the partial and total information for the diagnostic of the knowledge stated, everything realized through of the tutor module.

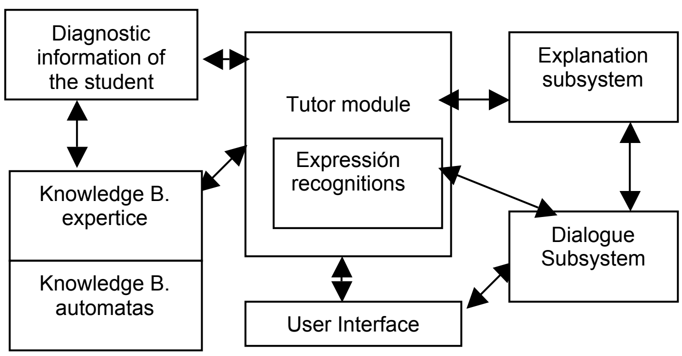

Figure 1. Architecture of the ITD

Although the recognition of expressions appears included in the tutor module, exists an independence between the knowledge control of the dialogue and the knowledge control of the recognition of expressions, this because the inference mechanism possesses rules to determine different actions as: the feedback, helps, orientation in the appropriate moment depending of the answer. In the figure 1 we can see the architecture of the dialogue.

1.5.1 Definition of an architecture metalevel integrated for the tutor dialogue.

The architecture model of the Intelligent dialogue maintains two types of knowledge that in certain form they are independent; *the knowledge for the control of the dialogue and the knowledge for control of the recognition of expressions.*

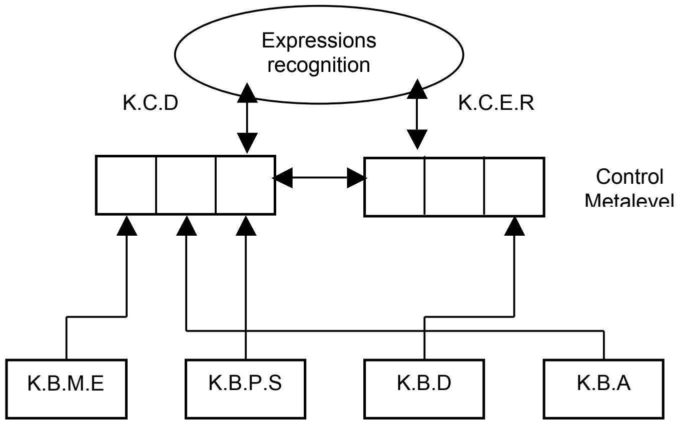

Figure 2. Architecture metalevel of the dialogue

The knowledge of the student model and the knowledge of the pedagogic strategies are managed for the level the knowledge for the control of the dialogue. The knowledge for the control of the automatas and the domain of knowledge is managed for the knowledge for control of the expression recognitions, this proposal recaptures the concept of classic control metanivel applied for intelligent Tutoring System, see figure 2.

K.C.E.R: knowledge for control of the recognition of expressions.
K.C.D: knowledge for control of the dialogue.
K.B.D: knowledge base of the domain.
K.B.A: knowledge base of automatas.
K.B.M.E: Knowledge base the student model
K.B.P.S: knowledge base of the pedagogic strategies.

The architecture of the ITD makes emphasis in the following types of knowledge:

- *Knowledge base of the domain (KBD)*: This embraces the basic concepts and the existent relationships among the concepts of the basic theorems to demonstrate.
- *Knowledge base of the pedagogic strategies (KBPS)*: Has the information of the pedagogic strategies that in our case is the dialogue, besides the strategies as feedback, initiation of the dialogue and helps.
- *Knowledge base of automatas (KBA):* Here are all the finite automatas non deterministic that serve for the recognition of the expressions classified by the types of answers.
- *Knowledge base the student model (KBSM):* this knowledge base has the student's record, its state of knowledge in certain times.
- *Knowledge for control of the dialogue(KCD)*: Refers to the reasoning form and inferences that is applied to the knowledge of the student model and pedagogic strategies and to every the dialogue.
- *knowledge for control the recognition of expressions (KCER)* This is the knowledge for control of the knowledge domain and the knowledge base of automatas.

1.5.2 Model of the student in the ITD

The system for the diagnose of a student requires of the definition of three types of users, whose tasks are specifies and different. For this reason we define two independent subsystem, this way; t*he evaluation subsystem,* the user is the student; t*he subsystem of diagnose and control,* in the diagnose the user is the teacher, and in the control the user is the administrator.

For the handling of the student's information and of the professor we defined four types of databases that are manipulated through the inference motor or pedagogic module, they are: information for the diagnose, the knowledge base for student profile, the student's additional Information and the professor's additional Information that are specified here (see figure 3):

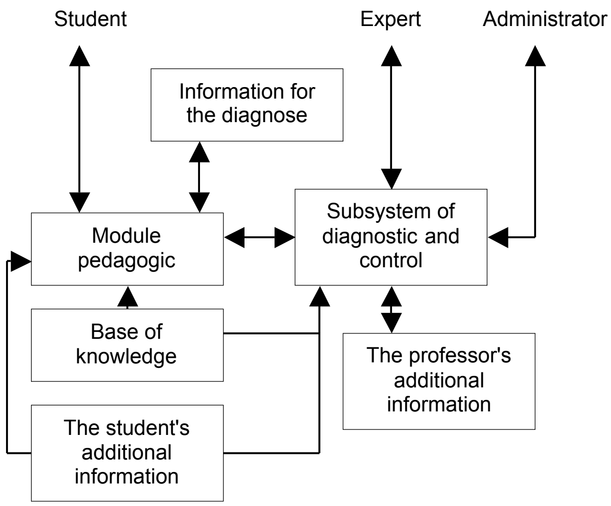

Figure 3. Model of the student in the dialogue

- *Information for the Diagnose.* The information contained in this database is generated starting from the evaluation subsystem and starting from the module pedagogic, later on it is used in the module of diagnostic and control to generate the student's profile
- *Knowledge base of the student profile.* In a base of knowledge they are the facts and rules required to describe the student's behavior in a level of knowledge.
  The base of knowledge managed by student's subsystems and professor, contains the facts and rules for the calculate and elaboration of the student's profile.
- *Additional information of the Student.* Here it is stored: the student's information that the professor includes to work with the system, the information used as help or explanation in the subsystem of the student's evaluation and the information required in the handling of screens that are presented in this subsystem.
- *Additional information of the Teacher´s.* Through this information it is possible to manage: the data of the professors with their access keys to the system and the variables that allow that the system is open and flexible, with the possibility of changing some parameters in the existent information.

1.6 *Description of the dialogue*

For the ITD we defined three dialogues for each demonstration method; two dialogues to demonstrate that If A□B□ A□B□ A and that If A□ B □ A□A□B and another dialogue to demonstrate it for contradiction, now we describe the dialogue for demonstrate that if A□B then A□B □A.

For the dialogue we manage three lines of form that if the answer is optimal for part of the student, the dialogue will be finished of an excellent way, otherwise in each one of these lines, depending of the answers it can be extended to one or more lines. For each one of these lines we manage the type of answers according to the classification: Optimal answers, correct answers, undue correct answers, incomplete answers, incorrect answers.

1.6.1 Model of the dialogue

This is an example of a session of dialogue not optimal of a student.

Dialogue: If A□B then A□B □A.

Student: let's suppose that x □ A.

Dialogue: The hypothesis is incomplete, !remember! that is the hypothesis to demonstrate.

Student: Then x□ A y x□ B.

Dialogue: Correct. You applied directly the intersection concept. Although the step previous should be x □ A □ B.

(Explanation: This is a correct answer for this lines, but it's not the optimal) .

Student: Therefore x □ A □ B.

Dialogue: It is correct but the answer is not appropriated.

(Explanation: for this second line it's not an indicated answer but this is a correct answer.)

Student: Therefore it is concluded that x □ A.

Dialogue: Excellent!!. It has concluded this part of the demonstration.

Now let's observe an example of an optimal dialogue.

Dialogue: If A□B then A□B □A.

Student: Let's be x □ A □ B.

Dialogue: Correct.

Student: Then x□ A y x□ B.

Dialogue: Correct.

Student: Therefore x □ A.

Dialogue: Very well!!!. This part has been displayed. Therefore A□B □A.

1.7 Use of Components of Intelligent Software to the Development of Intelligent Dialogues.

The dialogue was implemented using the intelligent components that are based on prolog, this component possesses a logic server that can be incrusted in a language of high level as visual basic to create better educational environments by computer, calling to extended predicates that allows the integration between the language of high level and prolog.

The advantages that we can have programming with intelligent components as Amzi Prolog Logic Server is the of being able of development interfaces with any programming language JAVA, delphi, visual basic and the inference in prolog. (see figure 4).

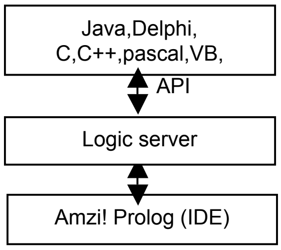

Figure 4. Amzi prolog with server logic

2. An Educational Model for the Development of the Intelligent Dialogue.

The educational model is based on the learning strategy of the tutor dialogue, initially we have developed this model for a modulate interface, oriented more at the expression recognitions, but with the time we obtained an intelligent tutor dialogue, for the handling and generation of knowledge through of the dialogue as strategy main of learning.

Furthermore the dialogue is based, on the integration of learning theories like; the problem solution theories, Skinner reinforcement programs, the Bloom taxonomy and teaching theory of Godofredo Caballero.

The defined educational model has five components: The dialogue, the tutor, the learning theory, the teaching theory, the student. The following figure schematizes the model:

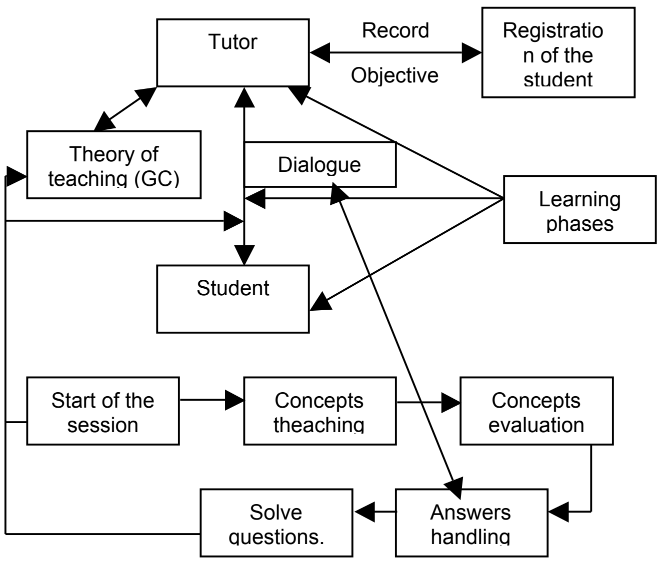

Figure 5. Educational model of the dialogue

The importance of the model locates in the integration of the dialogue; first of all with the phases of teaching concepts, evaluation concepts and ***answer handlings*** of the Caballero theory and second with the learning phases of Gagñe, Skinner, Bloom and learning for rules. Furthermore of learning activities as; the start of the session, the feedback, the question solutions, etc. In conclusion the tutor is behavioral, due at it guides to the student step by step in the development of the showing; starting the tutor with the presentation of goals and arrivals, at once the development of the dialogue where are included the strategies like: feedback, helps on the opportune moment, concept presentations, question solves and answer handlings and finally concept evaluation and end of the dialogue.

2.1 Theories of Teaching and Learning Applied to the ITD

According to Gagñe and his learning phases, the applying to the ITD is given of the following way:

- *Attention:* An event of teaching of the tutor dialogue that forward this phase, is the announce that the session will begin.
- *Expectations:* As application in the tutor dialogue will be informed on the objectives of the lesson.
- *Recovery:* An educational event of the tutor dialogue that forward this phase, is to introduce to the student some rules and subordinate concepts that are considered basics for the demonstration of theorems.
- *Selective perception:* A teaching strategy of the ITD that helps to this phase, it is the presenting of examples, proposed exercises, concepts and rules for the solution processes in the demonstration of theorems of the group theories.
- *Semantic Code:* Each dialogue generates rules through of the answers.
- *Recovery and answer:* A teaching incidence that forward this phase in the dialogue, it is to propos to the student that applies the procedure rules to demonstrate the theorem to specific problems where the student can apply the theorem.
- *Reinforcement :* An educational event that benefits to this phase is the confirmation of the accuracy of the student answers; the dialogue corrects or congratulates to the student after question has been answered.

2.2 Theory of teaching de Godofredo Caballero applied to the dialogue

According to Godofredo Caballero, the methods or technical of teaching more used for a tutor are:

- *Beginning of the tutorship:* It is used to motivate and to prepare the student, stimulating it to learn making them interesting the topics (referring to their real application), enunciating objectives and giving importance to the contents.
- *Teaching of concepts* : The ITD present concepts in sequential form increasing progressivly its complexity, in the presentation of examples and helps.
- *Evaluation of Concepts* **:** The tutor dialogue can make an evaluation with the following ends: An analyze and observation the state of the student's knowledge, to take measured correctives, to locate these topics or to motivate it in the learning of concepts. Certain levels can be demanded to evaluate different capacities like: memory, understanding, analysis, synthesis, application and evaluation.
- *Handling of Answers*. The tutors maintain and feed the discussions among their students by means of questions that require something more than superficial answers. The dialogue has as main component the answer such of the student as of the tutor dialogue. For this the handling of answers in the dialogue is realized of rigorous way, taking into account the classification given to the type of answers.
- *To solve questions.* It is consequence of a process of Teaching - Learning, it is the student's initiative of formulating questions to the tutor. The tutor resolves the student questions through of the dialogue development, each answer of the computer originates a question in the student.

3. The Finite Automatas and their Application to the Intelligent Tutorial Dialogues.

The finite automatas intervene in the Intelligent dialogue in the interface module as abstract machines of recognize of mathematical expressions happened in the demonstration of **A□ B □ A□ B=A.**

¿What are the advantages of the use of finite automatas in the development of intelligent dialogues? There are several possible reasons:

- At level of the pedagogic module is had a better control and handling of the knowledge that is shared in the interaction student-machine and maintain an independence with the knowledge of control of the learning strategies. And like an important advantage, the reutilization of automatas assure us a better manipulation of the answers.
- At level of the interface module, the interactions student-machine may be obtained with treatment of natural language at syntactic level.
- At level of the student's model, is had information in real time, referent to answers of the student and the computer.

3.1 An example of a finite automata in the dialogue.

Let's suppose that the student will demonstrate that **A□ B □ A□ B□ A,** as example the dialogue we will take the expression **x□ A□ B** that is derived of the first line, for it is defined an automata that will recognize these expressions; for it we defined a finite automata non deterministic:

- Q : state sets
- F: final state sets
- s_0: initial state
- □ : the alphabet

The white space, that is different to the empty string also is recognized by the automata.

Q = {q1, q2, q3,..., q19}
F = {q12,q17}
s_0= {q1,q5}
□ = {x, (,), A, B, □ , □ ,_}
_:white space.

This finite automata recognize expressions as: x□ A□ B, x_□ A□ B, x_□ (A□ B), etc, these expressions can have or non spaces in white and belong to the type of optimal rules; here they are specified by the line of below, the following figure specifies the automata: (the lock in the states represent the aceptation of the white space)

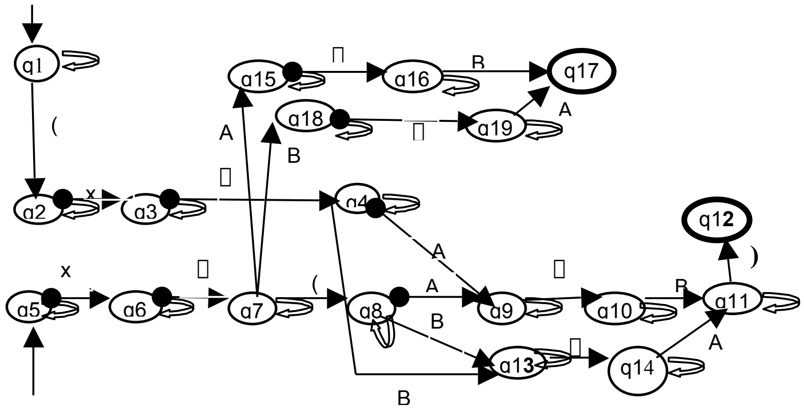

Figure 6. Finite automata that recognize x□ A□ B

In the dialogue if the expression that the student writes is not **x□ A□ B** another automata will recognize this answer type, for each type of categorized error has defined an automata set. Also each line in the dialogue is a finite set of automatas that will recognize a finite number of expressions written by the student.

References

[1] AMZI! Inc. (1997) Prolog user's guide & reference. Lebanon: AMZI.

[2] AMZI! Inc.(1997) Prolog logic server user's guide & reference. Lebanon: AMZI.

[3] Brokate, Arturo. (1996). Módulo pedagógico para un sistema inteligente de enseñanza asistida por computador (SIEAC). Bucaramanga: UIS.

[4] Brookshear Glenn (1993). Teoría de la Computación. Lenguajes formales autómatas y complejidad. Ed Addinson-wesley iberoamericana.

[5] Corredor, Martha V.(1992) Principios de inteligencia artificial & sistemas expertos. Bucaramanga: UIS,

[6] Gutierrez de Piñerez , R. E. (2000). Módulo tutor Inteligente como aplicación pedagógica de la ley distributiva. *Revista de enseñanza y tecnología (ADIE).* # 18. pag 5-10. España.

[7] Gutierrez de Piñerez R.E & Parra C. (2001). Uso de tutores inteligentes como apoyo para la enseñanza de la ley distributiva en matemáticas dentro del marco del pensamiento sistémico. *Articulo publicado en el Tercer Simposio de Informática educativa realizado en 3SIE;* Viseu. Portugal.

[8] Karsley, Greg. (1987) Artificial intelligence & instruction. Applications and methods. : Addinson-Wesley.

[9] Kelley, Dean. (1995) Teoría de Autómatas y Lenguajes Formales. Ed Prentice Hall. P160g 48. España.

[10] Kolman & Busby & Ross. (1997). Estructuras de Matemáticas Discretas para la Computación. Ed Prentice Hall. México.

[11] Maes, P & Nardi, D. (1988). Meta-Level Architecture and Reflection. North Holland: Elsevier Science Publisher.

[12] Richmond, Barry. (1993). Systems Thinking at the 90s and beyond. System Dynamic Review Summer.

[13] Rolston (1989). Principios de inteligencia artificial y sistemas expertos: McGraw-Hill.

[14] Wenger, Etienne.(1987). Artificial intelligence and tutoring systems: Morgan kaufmann.