DOCUMENT RESUME

ED 477 042 IR 021 770

AUTHOR	Kim, Youn-Gi; Lee, Chul-Hwan; Han, Sun-Gwan
TITLE	Educational Application of Dialogue System To Support e-Learning.
PUB DATE	2002-06-00
NOTE	7p.; In: ED-MEDIA 2002 World Conference on Educational Multimedia, Hypermedia & Telecommunications. Proceedings (14th, Denver, Colorado, June 24-29, 2002); see IR 021 687.
AVAILABLE FROM	Association for the Advancement of Computing in Education (AACE), P.O. Box 3728, Norfolk, VA 23514. Tel: 757-623-7588; e-mail: info@aace.org; Web site: http://www.aace.org/DL/.
PUB TYPE	Reports - Research (143) -- Speeches/Meeting Papers (150)
EDRS PRICE	EDRS Price MF01/PC01 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; *Computer Mediated Communication; *Distance Education; Educational Resources; *Educational Technology; *Geometry; *Instructional Design; Instructional Materials; Mathematics Education; Mathematics Instruction

ABSTRACT

This study is on the design and implementation of an educational dialogue system to support e-learning. The learning domain to apply the dialogue system used the subject of geometry. The knowledge in the dialogue-based system for learning geometry was created and represented by XML-based AIML. The implemented system in this study can understand the student's context of the dialogue. Moreover, the system can answer the student's questions by referring and saving the previous knowledge while having a conversation with a student. To refer and save a student's state of knowledge, an overlay student model was used in this system. An educational dialogue system was evaluated to test the efficiency of the designed and implemented system with geometry learning. (Contains 13 references and 4 figures.) (Author)

Educational Application of Dialogue System to Support e-Learning

Youn-Gi Kim, Chul-Hwan Lee, Sun-Gwan Han,
Dept. Computer Education, Inchon National University of Education, Korea

Abstract
This study is on the design and implementation of an educational dialogue system to support e-Learning. The learning domain to apply the dialogue system used the subject of the geometry. The knowledge in dialogue-based system for learning the geometry was created and represented by XML-based AIML. Implemented system in this study can understand the student's context of the dialogue. Moreover the dialogue system can answer the student's question by referring and saving the previous knowledge while having a conversation with a student. To refer and save a student's state of knowledge, we used an overlay student model in this system. Finally, we evaluated an educational dialogue system to test the efficiency of the designed and implemented system with geometry learning.

1. Introduction

The existing web-based learning systems have provided the learner the contents with a simple feedback. The learner interacts to the system through an interface screen simply. These methods do not support an intelligent and efficient learning. The interaction between the learner and the learning system advances on automation by an intelligent dialogue agent.

Established intelligent dialogue systems were used the Natural Language Processing system based on the Expert systems. Because these systems interact with a guided question and restricted answer in system, the system and the learner do not carry on a free dialogue concerning the related learning[8]. Therefore, we design and implement a natural language-based learning system using an intelligent dialogue agent. In order to implement the learning system, we used an engine that is a so-called Tutor-bot using the Pattern Matching(PM) method. We selected the learning domain about the geometry learning and extracted the knowledge base of the Plane Euclidean geometry.

Chapter 2 would introduce the basic concept behind the PM method, and introduce the readers to AIML. Chapter 3 discusses about the design of an intelligent dialogue e-learning system. Chapter 4 talks about the study an implementation with the use of tags. A sample code from *geo.aiml* is included in this chapter to allow the reader to fully understand the customization process. Then this study ends with Chapter 5, which talks about the future implementation of dialogue-based learning system.

2. Dialogues System for Tutoring

Pattern Matching and AIML

The Pattern Matching(PM) method is usually driven by fairly simple mechanisms. There is several in an existence. The first one was ELIZA created by Joseph Weizenbaum[13], which mimicked a psychotherapist. It made quite a name for itself because it succeeded in fooling some people it was a person. For the most part, they are implemented by rules that map simple pattern matches on the left-hand side to canned responses on right-hand side. Alice is a chatter-bot developed by Dr. Richard S. Wallace[12]. Alice is an acronym for Artificial Linguistic Internet Computer Entity. It's programmed using an XML DTD called AIML(Artificial Intelligence Markup Language). When the user enters statements at a prompt on WebPages, Alice fits that statement to the most 'specific' category it can find, and returns the response associated with that category. It is capable of understanding natural language syntax. However, the natural language is extremely complex. A person is presumed to have implicit or meta-knowledge about the world when having conversation. In human-to-human conversations, implicit and meta-knowledge is always used. This in someway poses as a huge challenge when customizing the Tutor-bot for e-learning system.

The Tutorbot was implemented by AIML(Artificial Intelligence Markup Language), which is a non-standard evolving markup language to create the chat robots. The primary design feature of AIML is

minimalism. Compared with other chat robot languages, AIML is perhaps the simplest. The pattern matching language is very simple, for example permitting only one wild-card ('*') match character per pattern. AIML is an XML language, implying that it obeys certain grammatical meta-rules. The choice of XML syntax permits integration with other tools such as XML editors. Another motivation for XML is its familiar look and feel, especially to people with HTML experience. An AIML tutor agent begins and ends with the <alice> and </alice> tags respectively. Typical Alice category tags have two component tags.

*-A pattern, eg: <pattern> I HAVE A * <pattern>*

-A template, eg:<template>Where did you get it?<template>

The pattern matching used by Tutor-bot starts with analyzing the words in the phrase entered by the learner[9]. These words are in natural language, separated by spaces and converted to uppercase to enable case independent matching. If a matching category is found, it is possible that the category is a recursive category. If this is the case, the target phrase will be altered and evaluated again by the pattern matcher. This adds the wildcard symbols ... and ' ' to the pattern language. Many AIML patterns start with the same words:

{WHAT IS A CIRCLE..., WHAT IS A ... , WHAT IS ... , WHAT OVAL..., WHAT...}

All these patterns overlap in one or more words. This fact suggests the use of a tree to store the patterns in. Alice uses a rooted, directed tree. Each node in the tree contains a hash table, which is used to store its successor nodes. This way, the pattern matching time is almost constant independent of the number of categories. It only depends on the length of the sentence; the longer the sentence, the longer the time to find a matching pattern. Yet, this relation is linear. All patterns are stored in alphabetical order.

Related Works

Numerous experiments have been performed with natural language interfaces. Several domains have been tested with varying success. Here, a few projects are discussed to determine which domains are suitable and which are not. Biermann [3] conducted an experiment to test the feasibility of natural language for programming environments. Subjects had to solve linear equations and calculate averages on a natural language system, which allowed them to manipulate tables and matrices. About 80% of the users' sentences were processed correctly. These results were compared with a similar experiment using a formal programming language called PL/C. Even though the subjects who used the natural language system used up to 50% more characters, they completed the problems using less time than the subjects who used the PL/C. This result can be explained in two ways: either natural language is easier to use and understand, or it is more verbose. As discussed before, verbosity is not a mayor problem and it must be said PL/C is more suitable for these kinds of problems than natural language because of the tables and matrices. If a domain had been chosen that was not obviously better suited for a mathematical notation, shorter lengths of the natural languages sentences should be expected.

Hauptman and Green [6] compared three interface styles; a command line language, a natural language and a menu-based interface. Each subject was given three hand-sketched diagrams of graphs that they were supposed to reproduce using the interface. It appeared that the natural language mode was not more difficult than any of the other two. It must be said though that this task as well as the previous one are good examples of tasks that are inappropriate for natural language interaction. Domains involving musical notations, algebra and graphical content are other unsuitable domains. Kelley [7] investigated the use of natural language for retrieving information froma calendar. He found that between 86% and 90% of the questions asked by the users were answered correctly. This result may look promising, but it must be noted that the domain is very simple and fairly limited.

3. Intelligent Dialogue Learning System

Overview of Dialogue-based Tutoring System

Figure 1 shows the overview of the intelligent dialogue-based e-learning system architecture. Dialogue-based Tutoring System is composed of 3 parts, learner interface by dialogue system, intelligent dialogue tutoring engine and intelligent tutoring system with learner base. The interface module is the part that learner proceed learning through the web. Dialogue module is embedded in learning content for communication between system and learner. The dialogue module consists of client and server. Moreover, this module is the core of this study. Server of this module contains the pattern matching engine and the knowledge base. Learner's knowledge level has diagnosed through learner profiles and dialogue among learning. The knowledge base saves domain knowledge on geometry information. The intelligent tutoring module is typical

Intelligent Tutoring System. ITS is consisted of student module, tutor module, and expert module. The student model was designed the overlay model, bug model, hybrid model[5].

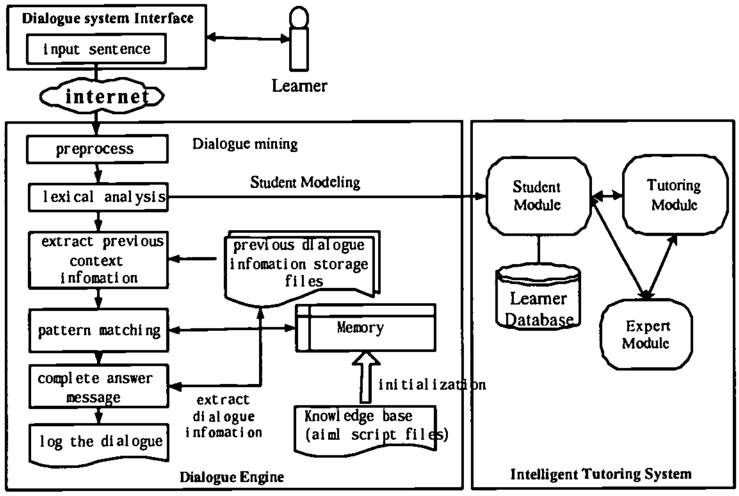


Figure 1. Overview of Dialogue-based Tutoring System

If learner inputs question, Tutorbot makes an analysis context through preprocessing. At this time, a learner's knowledge and a mistake are stored in learner database. For reasoning a whether last question is related former questions, the pattern matching engine extracts stored former contexts at temporary space. Passing through pattern matching processing, dialogue-based system understands learner's question and shows a suitable reply to learner. Finally, the system stores the result in learner database and repeats.

The reason of former question among the rules of the pattern matching uses recursive method and temporary space tag. This reasoning method is necessary for constructing student model. AIML(Artificial Intelligence Markup Language) contains a simple yet powerful XML markup tag called <srai>. The <srai> tag is the symbolic reduction tag. This allows minimalism. <srai>X</srai> is simple: The <srai> tag always appears in the *response* template, but the tutorbot treats X just like an *input* to the tutorbot. The tutorbot scans through its memory and finds the best response for X. The only tricky part is, the response to X may itself contain more <srai> tags.The best way to understand the recursive action of the AIML <srai> tag is by example.

Student: You may say that again Tutorbot
Tutorbot: Once more? "that"

The Tutorbot has no specific response to the pattern "You may say that again Tutorbot." Instead, the Tutorbot builds its response to the learner input in four steps. This simple sentence activated a sequence of four categories linked by <srai> tags. The tutorbot constructed the reply "Once more? "that" recursively as each sub sentence triggered the next matching pattern. <srai> allows the translation of Natural Language to keywords. On the other hand, the tag "that" in Tutorbot refers to whatever the tutorbot said before a user input. Conceptually the choice of "that" comes from the observation of the role of the word "that" in dialogue fragments like:

Tutorbot: Today is yesterday.
Student: That makes no sense.
Tutorbot: The answer is 3.14159
Student: That is cool.

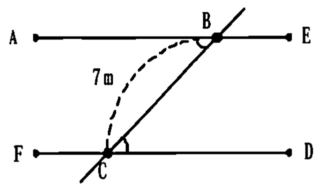
In AIML the syntax <that>...</that> permits an optional "ThatPattern" to match the tutorbot's "that" expression.

Representation of Geometry learning knowledge for AIML Modeling

We should need that analyzes the scheme of geometry knowledge for performing geometry learning. Because the dialogues on the knowledge of the geometry learning are to teach knowledge of geometry domain, the domain knowledge of geometry is represented with a definite rule or context. Geometry object is core that is diagram information. The diagram information is divided the quantity and quality information. The quantity information contains presented independence information in learning spaces as an absolute coordinates, number of pixel, an inclined degrees etc. The quality information represents the ontology of

geometry objects like dot, line, triangle, circle, as well as the proposition alike a connected relation among each geometry objects, and various knowledge for the form of a proposition. Such information gets to be main content for constructing learning and presents problem to learner as well as a tutor system. Consequently, this diagram information determines an efficiency of dialogue-based tutoring system.

The quality information of geometry mainly is divided the structural information and the geometric information as followed figure 2. The structural information represents knowledge about relation among the objects and the features of geometry object. The geometric information is an added proposition on the basic structure. This information is reasoned by the geometric or algebraic knowledge. Efficient dialogue-based tutoring system needs suitable knowledge representation about the quality information of geometry. Main knowledge constructed with AIML is represented by structural information; on the other hand, the used knowledge by answer of dialogue is presented by the geometric information.



Structural information: *The line AB is composed of the dot A and dot B*
The line FD is composed of the dot F and dot D
The line BC is composed of the dot B and dot C
The dot B locates on the line AE ...
Geometry information: *Lines AE and FD are parallel*
The distance between the dot and the dot C is 7m
Alternate interior angles ABC and BCD is same ...

Figure 2. Classification of Geometry information

Student Model and Dialogue Representation

The geometric knowledge is constructed variously by the learner's knowledge state and learning process for using intelligent dialogue. Intelligent dialogue method is different complexity by who take the hegemony of dialogue. The hegemony of dialogue classifies by a case of system-initiative, case of learner-initiative and hybrid case. A case of system-initiative, system asks to a learner a question and a learner answer mostly. This method is same the one of an expert system and system reason the learner's knowledge in the course of a conversation.

On the other hand, a case of learner-initiative doesn't use planed dialogue but system replies the most suitable answer by learner's question. Because this method doesn't exist a constructed knowledge base previously, the number of answer is increase exponentially. As a result, system processes dialogue with restriction the band-with of learner's question.

Hybrid case, if the learner has a mistake or does not understand the learning content, the dialogue method executes with a system-initiative mode, if learner has understanding or tries review learning, learner-initiative drives. A learner's knowledge through dialogue is stored by student model and diagnoses a learner' level. The system-initiated method uses overlay model and fixes up a learner's knowledge and the pre-made dialogue path. While the learner-initiated method uses bug model to find learner's mistakes. Hybrid method uses a bug library model namely machine learning for storing new bug.

AIML Design for Geometry Learning

Tutorbot is able to understand the flow of conversations by capturing the questions and going into a particular topic. <topic> allows Tutorbot to prefer responses that deal with the topic currently being discussed. This creates topical conversation, yet still has the ability to move from one subject to another. For example, if the questions were asked in relation to Plane Euclidean Geometry, the file *Geo.aiml* is that will be put into use.

Example: *<Student> Hi!*
<Tutorbot> Hi! there [general category]
<Student> Can you please tell me more about a quadrangle ?
<Tutorbot> Lets move on the geometry section [geometry category]
<Student> How many lines do a quadrangle have?
<Tutorbot> Four.
<Student> How many dots do a quadrangle have?
<Tutorbot> Four, too.

<Student> What is an angle?
<Tutorbot> That is the degree of an angle formed where two lines cross each other.
<Student> Thanks you.
<Tutorbot> You 're welcome. [general category]

Tutorbot is capable of throwing in random responses along with the proper response. This in a way is like a freebie guide that may assist students. This type of random responses tied with logical responses is feasible in this type of a setting. This gives some form of personality to Tutorbot. It would be like dialogue with an intelligent tutor and from past experience, an intelligent tutor always throws in additional hints to assist students when making decisions about their future.

<category>
<pattern>CIRCLES</pattern><template>
<random>
*
You will have fun in geometry learning.
*
*
Circle is very interesting figure. *
*
What is difference a circle and an oval? *
*
Ask me anything you want about circle
*
*
Did you know that circles' features for Geometry?
*
</random>
</template>
</category>

4. Implementation and Experience

Implementation of intelligent dialogue system

Figure 3 shows the prototype of a dialogue-based e-learning system. This system was designed to learn the plane Euclidean geometry learning with intelligent dialogue method. A learner can learn geometry learning through screen interface and can ask the toturbot a question by using dialogue interface. For reducing the bandwidth of learner's question, the context of question is limited in related learning contents.

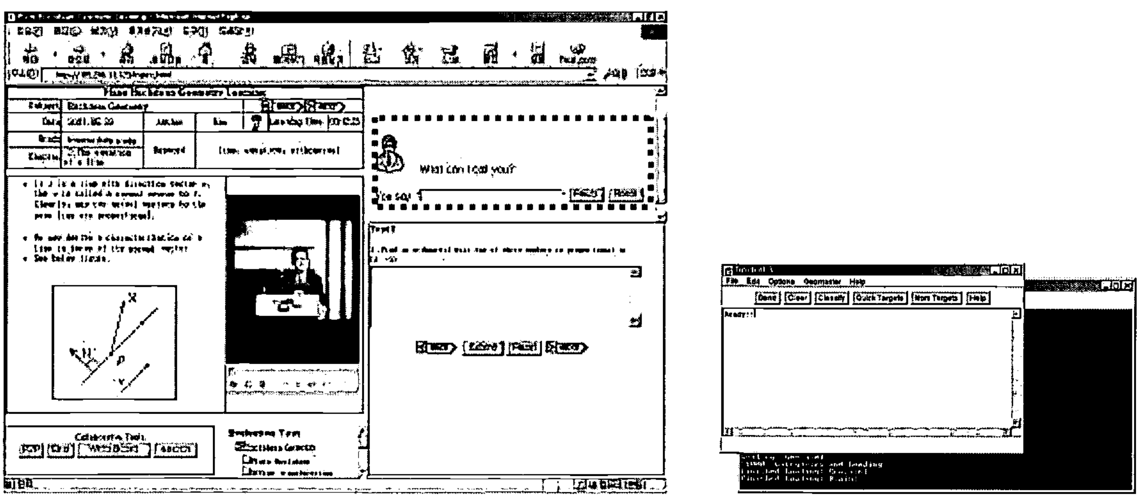


Figure 3. Screenshot of intelligent dialogue-based e-learning system

The dialogue-based engine used JDK 1.3 program and dialogue server executes by servlet. When server drives by servlet, the dialogue engine loads category items of geo.aiml file in memory. Moreover, server creates log file for grasping the learner's information. Right in figure 3 shows server screenshot of Tutorbot. H/W by using research is Windows 2000server. If a Tutorbot server executes, dialogue port is used the 2001 port with HTTP and all learner can communicate with system. Tutor-bot server in figure 3 can see that geo.aiml was loaded in memory by driving server

Sample of Conversation with Tutor-bot

The category on geometry learning was made with 23000-domain knowledge and knowledge was stored in geo.aiml file. A learner can communicate with a special knowledge in geo.aiml file and a general

knowledge in general.aiml file The following Figure 4 was recorded between Tutorbot and a learner. A learner asks a system a question about the quadrangle and we can see that a tutorbot replies a suitable answer. When learner uses an approximated question or an illegal grammar, a tutorbot introduces to change a direction of dialogue.

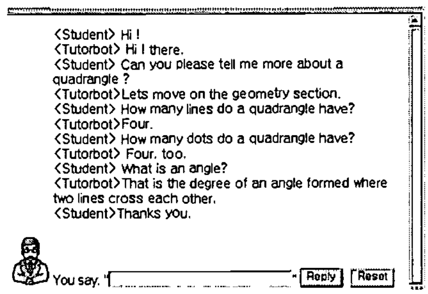


Figure 4. Conversation with Tutor-bot

5. Conclusion and future work

Dialogue systems are becoming very common in many web sites today. They serve to give personality and intelligent information to web sites. Specially, a dialogue system approves a merit in e-learning system. Typical technique of intelligent tutoring system is intelligent dialogue based student model. We discussed on dialogue technique for such intelligent tutoring system in this paper. Moreover we extracted the domain knowledge of geometry learning and geometry analyzed structural information and geometric information. This information and knowledge could use effectual for intelligent dialogue system. We hope for developing intelligent dialogue-based tutoring system by our study.

Futures works need that study on temporal reasoning method by changing a learner's knowledge as well as research on virtual interface by using 3D and virtual reality. Furthermore, for improving the accuracy of dialogue, we will study on the reasoning method of an ambiguity and uncertain knowledge.

References

[1] Anderson, J. (1983) Acquisition of proof skills in geometry, Morgan Kauffman.

[2] Bates, J. (1994). The Role of Emotion in Believable Agents, Technical Report CMU-CS-94-136, School of Computer Science, Carnegie Mellon University, April 1994. Also to appear in Communications of the ACM, Special Issue on Agents, July 1994.

[3] Biermann, A. W. et al. (1983). An experimental study of natural language, International Journal of Man-Machine Studies 18, 71–87.

[4] Boyce, S. J. (2000). Natural Spoken Dialogue Systems for Telephony Applications, Communications of the ACM, Vol. 43, No. 9, 29–34.

[5] Burns, H. Parlett, J. Redfield, C.(1991) Intelligent tutoring System: Evolution in design, Lawrence Erlbaum.

[6] Hauptman, A. G. and Green, B. F. (1981). A comparison of command, menu selection and natural language computer programs, Behaviour and Information Technology, Vol. 2, No. 2, 163–178.

[7] Kelley, J. F. (1983). An empirical methodology for writing user-friendly natural-language computer applications, Proceedings of the CHI '83 Conference on Human Factors in Computing Systems, ACM, New York, 193–196.

[8] Mandl, H. Lesgold, A.(1988) Learning issues for intelligent tutoring systems, Springer-Verlag.

[9] Mykowiecka, A. (1991). Natural language generation – an overview, International Journal of Man-Machine Studies 34, 497–511.

[10] Patrick J. Ryan, (1986) Euclidean and non-Euclidean geometry, Cambridge university press.

[11] Turing, A. M. (1950). Computing machinery and intelligence, Mind, Vol. 59, No. 236, 433–460.

[12] Wallace, R. S. (2000). Don't read me - A.L.I.C.E. and AIML documentation, available online 4 .

[13] Weizenbaum, J. (1965). Eliza - a computer program for the study of natural language communication between man and machine, Communication of the Association for Computing Machinery 9: 36–45.

NOTICE

Reproduction Basis

X	This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.
	This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").