

Computing and Informatics, Vol. 26, 2007, 565–576

AN ARCHITECTURE OF AN INTELLIGENT TUTORING SYSTEM TO SUPPORT DISTANCE LEARNING

Marcia T. MITCHELL

*Computer and Information Sciences Department
Saint Peter's College
2641 Kennedy Blvd.
Jersey City, New Jersey, 07306
United States
e-mail: mmitchell@spc.edu*

Manuscript received 3 October 2005; revised 19 February 2007
Communicated by János Fodor

Abstract. This paper outlines a design framework of an intelligent tutoring system (ITS). ITS focuses on a newer and more comprehensive distance learning (DL) process as compared to the established traditional DL programs practiced today. The DL model presented in this paper (CHARLIE) is a high level software based tutorial that has the ability to encompass a wide variety of current DL technologies in a single DL session. CHARLIE's architecture has four components: Control Component (responsible for the interaction between software agents and the operating system); Instructional Component (concerned with the instructional aspects of an ITS session); Text Analysis Component (analyzes the partial syntax and partial semantics of the text in the session); Student Modeling Component (analyzes a student's progress and determines the best model for learning during a session). Each component is serviced by a set of software agents to accomplish its mission. Three additional entities in CHARLIE are two separate databases and an explanation facility. Six agents have been implemented in CHARLIE to create a DL course in C++ programming. Much of CHARLIE remains to be completed which opens many areas for research.

Keywords: Intelligent tutoring systems, intelligent agents, distance learning, artificial intelligence

1 INTRODUCTION

The distance learning (DL) model presented in this paper (CHARLIE) is a high-level software based tutorial that has the ability to encompass a wide variety of the technologies mentioned above in a single DL session.

CHARLIE's proposed architecture is designed as a Distributed Reasoning System (DRS) [11, p. 433]. A DRS is one that is composed of a set of separate modules, often called software agents, which act as problem-solving entities controlled by a set of communication paths coordinating them. CHARLIE's agents communicate with each other, two databases, and an explanation facility using a message passing system i.e. one agent sends messages (both requests for services and information as well as replies to such requests) to one or more other agents [11, p. 439]. Six of the following agents (denoted by **) have been programmed in CHARLIE to provide the kernel of a DL course in learning C++.

2 INTELLIGENT AGENTS

Franklin and Graesser [3] have defined intelligent agents as "software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires" (p. 2). Furthermore, they say, a rational agent (one that performs correct tasks), must be able to make decisions and inferences.

Don Gilbert [4] notes that an agent must be goal-driven (the agent has purpose); must be reactive (the agent senses changes in its environment and responds in a timely fashion to those changes); must be adaptive (the agent learns or changes its behavior based on previous experience); and must be believable (the agent represents an entity visible or audible to the user) (p. 4).

3 COMPONENTS AND AGENTS IN CHARLIE'S ARCHITECTURE

3.1 CHARLIE's proposed architecture has 4 interacting components serviced by 16 agents

Control Component. Agents that control the interaction with other agents and the operating system.

Instructional Component. Agents servicing the instructional aspects of an ITS session.

Text Analysis Component. Agents that analyzes the partial syntax and partial semantics of the text in the session.

Student Modeling Component. Agents concerned with analyzing a student's progress and determining the best model for learning.

The following table indicates the associated agents (column entries) servicing the associated components (column headings).

Control Component	Instructional Component	Text Analysis Component	Student Modeling Component
Observer Agent	Domain Agent	Analogy Agent	Neural Network Agent
Supervisor Agent	Learning Style Agent	Fuzzy Logic Agent	Student Model Agent
Textbox Agent	Pedagogy Agent	HTML Agent	
	Strategy Agent	Natural Language Processing Agent	
	Subject Matter Agent	Partial Semantic Agent	

Table 1. Component and associated Intelligent agents in CHARLIE

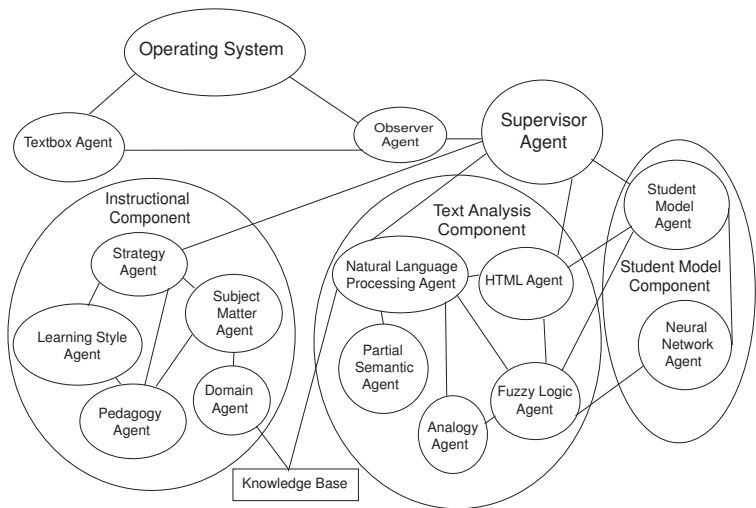


Fig. 1. Architecture of CHARLIE

Some of the above agents in CHARLIE have only been partially constructed, namely the Supervisor Agent, HTML, Parser, Domain and Natural Language Processing Agents and Partial Semantic Agent. The knowledge base, in addition, is only partially constructed.

4 CONTROL COMPONENT

4.1 Observer Agent

The Observer Agent monitors the students' behavior, keeping track of how they interact with CHARLIE, and report it to the Supervisor Agent, which is a supervisory

agent. Supervisor Agent passes all new information from the Observer Agent to one or more of the other appropriate agents. Observer Agent analyzes the students' progress and passes data back to the Supervisor Agent.

The Observer Agent would obtain the data from user thru the textbox and sends the data to the Supervisor Agent. The Supervisor Agent determines where the data will next be sent. There are four possibilities: the Strategy, the Natural Language Processing, the HTML, or the Student Model Agent. If the data is for the Partial Semantic Agent, then the Supervisor Agent sends the data thru the Natural Language Processing Agent for processing. After processing, the Natural Language Processing Agent passes the information to the proposed Student Model Agent which then continues the activities of the tutor.

4.2 Supervisor Agent ()**

As noted, CHARLIE is designed to be a message passing system. The Supervisor Agent is a supervisory agent that delegates tasks to different agents. For example, the Supervisor Agent may take the web page from the Observer Agent and send it to the HTML Agent for processing. It is this agent that interacts with all other agents upon request.

4.3 Textbox Agent

The Textbox Agent allows students to input their responses in sentence form. The Observer Agent takes the sentence from the tutor agent and sends it to the Natural Language Processing Agent. The Natural Language Processing Agent then sends the sentence to the Partial Semantic Agent, which parses the sentence and sends the result to Supervisor Agent. The Supervisor Agent passes the sentence to the Fuzzy Logic agent for analysis.

The Textbox Agent could also be used to dialogue with the tutor. The students use the Textbox Agent to type in questions, answers, or requests for information from the tutor. The Textbox Agent responds to the students' input by displaying requested information. The Textbox Agent is displayed as a window in the ITS.

5 INSTRUCTIONAL COMPONENT

5.1 Domain Agent ()**

The domain is the subject matter of the topic being taught by the intelligent tutor. The Domain Agent within CHARLIE determines the content to be presented. If a web page is used, the relevance of its content is determined using frequency analysis and is added to the tutor's knowledge base. This agent uses a pattern-matching technique to match the knowledge base content to the data from the web page. The

content will be presented as a web page for the student. However, the content is saved as plain text within the knowledge base.

5.2 Learning Style Agent

The Learning Style Agent will be called upon to evaluate appropriate data to determine the learning style of the student. The result of this evaluation will be used to determine whether the student is placed in self-directed mode or tutor-directed mode [8]. ITS delivers one-on-one instructions while the student are in tutor-directed mode. Once the student has successfully completed assigned tasks given by the tutor, the student could select to be put in self-directed mode. A task could be for the student to read a text and answer questions related to the text. The self-directed mode allows the student to surf the World Wide Web and view web pages that are related to the topics outlined in the curriculum.

5.3 Pedagogy Agent

The Pedagogy Agent is responsible for the student's learning while in tutor mode. The Pedagogy Agent sets up each lesson to present to the student through the Subject Matter Agent. The Pedagogy Agent also sets up goals and objectives for the student while in tutor mode. In addition, it evaluates the students' progress to determine whether their curriculum objectives and goals are being met. This evaluation could be done through non-adaptive exercises consisting of true/false, multiple choice, and short one-sentence answers. The ITS could sit in the background and observe what pages the student is viewing.

The Pedagogy Agent could set up the style of learning as directed by the Strategy Agent. If the Strategy Agent indicates the student can progress to the next level, it could direct the Pedagogy Agent to select new subject matter and pass it to the Subject Matter Agent to present. The Pedagogy Agent is also responsible for testing the student on the concepts that they have learned. Adaptive exercise techniques could be used. Adaptive exercise involves a collection of items pertaining to a topic. As an example the computer could randomly generate questions based on the topics, and present questions to the student. If the student correctly answers the question, then the computer could select slightly harder questions for the student. The student who fails to answer the questions correctly should be given slightly easier questions. (For a discussion of adaptive exercise techniques see Jacobs and Chase [6].) Giving student slightly easier questions could help ensure that the student learns the basic concepts before moving to a higher level.

5.4 Strategy Agent

The Strategy Agent is responsible for determining whether student are able to accomplish a particular task based on the student's performance [5]. If the Strategy

Agent determines that the student can accomplish the task at hand, then the student is allowed to continue with the lesson using the strategy defined by the Strategy Agent (self-directed or tutor mode). If the Strategy Agent determines that the student is not able to accomplish a task then the Strategy Agent will evaluate the student's data and send the information to Partial Semantic Agent for further analysis and recommendations. The Strategy Agent sets up goals and objectives for the student who are in self-directed mode.

5.5 Subject Matter Agent

The Subject Matter Agent will be responsible for the content of the lesson. The Subject Matter Agent will create lists of goals for the student and provide data to the Observer Agent. Once the student is put in self-directed mode, the Subject Matter Agent will present the lessons to the student. The student will be free to choose what s/he wants to learn and when s/he wants to learn it. During this stage, the Observer Agent will observe the student's progress and pass the information to the Supervisor Agent which will then pass the information to the Strategy Agent for evaluation. If a student is having difficulty with the lesson, then the Strategy Agent will be consulted, e.g. one indication of difficulty would be if a student has selected to listen to or display a lesson more than two times. The Subject Matter Agent will evaluate this information and make recommendations to the student regarding the lesson. The Subject Matter Agent will also be consulted if the student requests help. If the student has difficulty in the lesson, the Subject Matter Agent will provide explanations about the subject whether the student is in tutor or self-directed mode.

6 TEXT ANALYSIS COMPONENT

6.1 Analogy Agent

The Analogy Agent will be used to make analogies to help explain the subject matter. This feature is similar to the X Model ([10], which is used to supply a method for problem solving when domain specific methods are lacking or are inadequate.

6.2 Fuzzy Logic Agent

If the student answers a question in the Textbox Agent, the answer will then be passed to the Natural Language Processing Agent. In order for the system to better understand the student's rationalization processes, the appropriate information will also be passed to a Fuzzy Logic Agent. The reason for the inclusion of the Fuzzy Logic Agent is that the data from the course objects may not provide an accurate picture of the student's learning. The data values from the ITS may be either yes or no, true or false. These data values will not give the Supervisor Agent a true understanding of how the student is interpreting the information presented

or whether the student is able to apply the learned knowledge to a new situation. The Fuzzy Logic Agent could help the Natural Language Processing Agent interpret the student's behavior to better evaluate the student's progress.

An HTML document contains tags that are used to format text, hyperlinks, and graphical images as well as text. The title tags

((title)and(\\title))

are examples of HTML tags. The content between the title tags is used to display what the document is about. The HTML Agent removes the tags from the web page and saves the plain text in a file to be used by other agents. The HTML Agent also identifies and catalogues each sentence on the web page for future use.

6.3 Natural Language Processing Agent ()**

The Natural Language Processing Agent is used to facilitate the communication in the form of free flowing dialogues or conversations between the ITS, the student, and other input sources such as web pages being displayed on the browser used by the student. The Natural Language Processing Agent determines the content of the displayed web page, and performs partial semantics of the text in the knowledge base as well as the web page.

6.4 Partial Semantic Agent ()**

A parsing technique is a method of analyzing a sentence to determine its grammatical structure. The Partial Semantic Agent inputs sentences from a file that contains contents of the subject matter. This may be input accumulated from the web or questions asked by the student using the Textbox Agent. The Stanford Log-Linear Part-Of-Speech Tagger [14] determines the part of speech for each word in the input line. The first version of the Partial Semantic Agent used Eric Brill Part-of-speech Tagger [1] determines the part of speech for each word in the input line. The individual words are checked to determine if they are used correctly in the sentence, and thus to determine if the structure of the sentence is syntactically correct. This process ensures that the partial semantics is accurate. Currently the Partial Semantic Agent has been programmed with encouraging results. The Partial Semantic Agent can determine partial semantic for declarative sentences for both scientific and non-scientific sentences.

7 STUDENT MODELING COMPONENT

7.1 Neural Network Agent

Neural networks have been designed to be trained using an inductive learning algorithm. Inductive learning can be applied to incremental learning [12]. The Neural

Network Agent determines the behavior patterns of the student. The Neural Network Agent receives data regarding the student's performance and builds a model of the student's knowledge using an inductive algorithm. The Neural Network Agent could be called upon to interpret the data and make recommendations based on its findings. The Neural Network Agent also receives feedback from the Supervisor Agent regarding its interpretation in order for it to learn more about the student. In addition, the Fuzzy Logic and Neural Network Agents both play a part in the modeling of the student's knowledge. If-then rules and statistical techniques could be used by the Fuzzy Logic Agent to interpret data sent by the Neural Network Agent. According to Jang, Sun, and Mizutani [7], fuzzy if-then rules form the key components of a fuzzy inference system that can effectively model human expertise in a specific application. Inclusion of neural networks in CHARLIE creates a better model for a changing environment.

According to Ira Sack [13], "information pattern is the ability of an agent to see beyond manifest data and data patterns and apply the (information) pattern in support of a still higher level pattern such as found in a business service (e.g., a customer response pattern)." The neural network could be trained to recognize the knowledge patterns derived from data collected from a variety of sources. One source could be messages from the Student Agent that may contain information about the student's behavior. Another source could be the processed data from the Supervisor Agent as well as the processed data from the Natural Language Processing Agent. For example, students may view a presentation (lesson), and, after the presentation, they could be asked whether they have understood the concepts. The student model could be used to gather this data and send the data to the neural network in order to determine the students' knowledge patterns. The neural network could analyze the students' knowledge patterns to determine whether the students have in fact understood the material, in the instance when understanding means that the students could be able to successfully apply the recently acquired knowledge.

7.2 Student Model Agent

The Student Model Agent will interpret the behavior of the student. The student model contains information such as the student's learning style. Neural networks are called by the Student Model Agent to help interpret the data that have been gathered by the tutor. The reason why neural networks are employed in the user modeling is that the traditional approach to knowledge base techniques, according to Finlay and Beale [2], is not adequate for student modeling. Rather, Finlay and Beale [2] point out that student modeling should contain the following features:

- facts and procedural knowledge about what is and is not known;
- possible paths of valid user actions;
- the immediate context of the user's action;

- the user's learning preferences and the style of teaching most appropriate;
- the user's social background;
- the user's goals.

8 ADDITIONAL ENTITIES OF CHARLIE

8.1 Knowledge base

Intelligent tutoring systems have “(two) separate data bases, or knowledge bases, (one) for instructional content (specifying what to teach), and (the other) for teaching strategies (specifying how to teach) and attempt to use inferences about a student's mastery of topics to dynamically adapt instruction” [9]. The Subject Matter Agent uses the information in the knowledge base to instruct the student. The Domain Agent is responsible for adding new information to the knowledge base. This new knowledge must be consistent with the type of information already present in the knowledge base.

8.2 Explanation Facility

When students are in tutor mode they will be given hints and explanations as they progress through a session. Each statement of problems will have a set of correct responses along with explanations. The students in “tutor” or “self-directed mode” may request hints or explanations through the Textbox Agent. Explanation of the topic could be done in conjunction with the Natural Language Processing Agent and the Domain Agent.

9 CONCLUSIONS

The ITS presented here describes a tutor/self-study model that coordinates agents to make it easier for student to self-learn. As noted previously, six agents presented have been implemented in CHARLIE to provide the kernel of a DL course in C++ programming. The kernel of the DL course includes the Supervisor, HTML, Parser, Domain, Natural Language Processing, and Partial Semantic Agent. Much of CHARLIE remains to be completed, and this will open many areas of research.

Acknowledgement

I wish to express my gratitude to Alessandro C. Calianese, Ph.D. Saint Peter's College for his many helpful suggestions.

Glossary

Artificial Intelligence (AI): “Have the capacity to reason, learn, understand, and solve problems”, 2006,
https://www.spider.hpc.navy.mil/index.cfm?RID=TTE_OT_1000008.

Agent: “Anything that can be viewed as perceiving its environment through sensors and acting upon the environment through effectors” (Russell and Norvig, 1995, p. 31).

Extended ASCII Character Code Representation: The technique of using extended ASCII character codes to represent C++ keywords and programming concepts in order to facilitate processing with formal models. These formal models include frequency analysis, reduction techniques and first-order logic. Extended ASCII character code are used to translate natural language into symbols so that formal mathematical models can be applied.

C++ Keywords: Reserve words in the C++ programming language. Examples of C++ keywords include class, delete, friend, public and private.

C++ Statement Agent: Used by CHARLIE to find which C++ statements are on the web page. The C++ Statement Agent is composed of several agents that are used to examine the web page, and work in conjunction with the HTML Agent. IF, WHILE, FOR, and FILE are some of the agents in the C++ Statement Agent.

C++ Statements: Instructions for the computer in C++ programming language.

CHARLIE: A proposed web-based tutor designed by the author in order to deliver instruction to students on the World Wide Web.

Distance Learning (DL):

Domain Agent (DA): The subject matter expert for the topic being taught by the tutor CHARLIE. The domain contains the tutor’s knowledge base. The domain is sometimes referred to as the Expert Model of the tutor. The Domain Agent within CHARLIE determines the contents of the web page using frequency analysis, and uses a pattern-matching technique to match the programming code to the template it has in its knowledge base.

Supervisor Agent (SA): A supervisory agent within CHARLIE that delegates tasks to different agents.

HTML Agent: Removes the tags from the web page and saves the plain text and programming code in separate files in CHARLIE. The HTML Agent also identifies each sentence on the web page.

HTML: An acronym that stands for hypertext mark up language.

Intelligent Tutoring System (ITS): A computer system that processes artificial intelligence (AI). “AI capabilities enable the computer to process natural

language, machine learning and speech. With the use of AI, an ITS can identify the students' strengths, weaknesses, and preferred style of learning", 2006 https://www.spider.hpc.navy.mil/index.cfm?RID=TTE_OT_1000008.

Intelligent Agents (IA): "Software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires" (Franklin and Graesser, 1996, p. 2).

Natural Language Processing Agent (NLPA): Helps CHARLIE to determine the content of the web page, perform partial semantics and concise summarizing of the text in the knowledge base as well as the web page.

Observer Agent (OA): Monitors the students' behavior, such as keeping track of how they interact with CHARLIE, and reports it to the Supervisor Agent, which is CHARLIE's supervisory agent.

Partial Semantics (PS): Pertains to the meaning of a sentence. Concise meaning, after which the sentence no longer conveys "sense". World Wide Web Tutor: Tutors students while they are on the web.

REFERENCES

- [1] BRILL, E.: Eric Brill Part-of-speech Tagger. <http://www.cs.jhu.edu/~brill/>, 2002.
- [2] FINLAY, J.—BEALE, R.: Pattern Recognition and Classification in Dynamic and Static User Modelling. In Russel Beale and Janet Finlay, Eds.: *Neural Networks and Pattern Recognition in Human-Computer Interaction*. New York: Prentice Hall Inc., 1992, pp. 65–85.
- [3] FRANKLIN, S.—GRAESSER, A.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Third International Workshop on Agent Theories, Architectures, and Languages. [Internet]: Retrieved [6 March 1998], from the World Wide Web: <http://www.msci.memphis.edu/~franklin/AgentProg.html>, 1996.
- [4] GILBERT, D.: *Intelligent Agents: The Right Information at the Right Time*. NC: IBM, Corporation, Research Triangle Park, NC.
- [5] ITOH, Y.—TAKAHASHI, I.—KONISHI, T.: On Evaluating Students' Knowledge by Observing their Actions in ILE. In B. du Boulay and R. Mizoguchi, Eds.: *Artificial Intelligence in Education*, Amsterdam: IOS Press, 1997, pp. 495–598.
- [6] JACOBS, L. C.—CHASE, C. I.: *Developing and Using Test Effectively*. San Francisco: Jossey-Bass Publisher, 1996.
- [7] JANG, J. S. R.—SUN, C. T.—MIZUTANI, E.: *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, N. J., Prentice Hall, Inc., 1997.
- [8] LINTON, F.: Learning to Learn from an ITS. In B. du Boulay and R. Mizoguchi Eds.: *Artificial Intelligence in Education*, pp. 317–323, Amsterdam: IOS Press, 1997.

- [9] MURRAY, T.: Authoring Knowledge Base Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. *Journal of the Learning Sciences*, Vol. 7, 2000, No. 1, pp. 5–64. [Internet]: Retrieved [3 May 2000], from the World Wide Web: http://www.leaonline.com/doi/pdf/10.1207/s15327809jls0701_2, 1998.
- [10] PIROLI, P.: Explanation, Analogy, and Transfer in an Intelligent Tutoring System Paradigm. In Marshall J. Farr, Joseph Psotka: *Intelligent Instruction by Computer Theory and Practice*, Washington: Taylor and Francis, 1992.
- [11] RICH, E.—KNIGHT, K.: *Artificial Intelligence*. New York: McGraw-Hill Inc, 1991.
- [12] RUSSELL, S.—NORVIG, P.: *Artificial Intelligence: A Modern Approach*. Upper Saddle River, N. J., Prentice Hall, 1995.
- [13] SACK, I.: Personal Communication. Stevens Institute of Technology, 1999.
- [14] Stanford Natural Language Processing Group: Stanford Log-linear Part-Of-Speech Tagger and the Stanford Parser. Stanford University, <http://nlp.stanford.edu/>, 2006.



Marcia T. MITCHELL, Ph.D., associate professor of computer science at Saint Peters College in the United States. She has recently completed a term as chair of the Computer and Information Sciences Department. Her topics of interest include intelligent agents, knowledge representation, natural language processing and programming languages.