DOCUMENT RESUME

ED 454 821 IR 020 800

TITLE	ICCE/ICCAI 2000 Full & Short Papers (Educational Agent).
PUB DATE	2000-11-00
NOTE	27p.; In: Learning Societies in the New Millennium: Creativity, Caring & Commitments. International Conference on Computers in Education/International Conference on Computer-Assisted Instruction (Taipei, Taiwan, November 21-24, 2000); see IR 020 792. Some figures contain very small and illegible type.
PUB TYPE	Collected Works - General (020) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Computer Uses in Education; Concept Formation; Distance Education; *Educational Technology; Elementary Secondary Education; *Instructional Design; Intelligent Tutoring Systems; Postsecondary Education
IDENTIFIERS	Knowledge; Social Construction

ABSTRACT

This document contains the full text of the following papers on educational agent from ICCE/ICCAI 2000 (International Conference on Computers in Education/International Conference on Computer-Assisted Instruction): (1) "An Agent-Based Intelligent Tutoring System" (C.M. Bruff and M.A. Williams); (2) "Design of Systematic Concept Learning Model Using Computer Education Search Engine" (Seong-Guk Kang, Young-Houn Lee, and Seong-Sik Kim); (3) "Educational Agents and the Social Construction of Knowledge" (Carolyn Dowling); and (4) "A Real-Time Handwriting Communication System for Distance Education" (Hsin-Chu Chen, Jho-Ju Tu, and Jianping Zhang). An abstract of the following paper is also included: "Strange Creatures in Virtual Inhabited 3D Worlds" (Jens F. Jensen). (MES)

ICCE/ICCAI 2000 Full & Short Papers (Educational Agent)

ICCE/ICCAI 2000

Learning Societies in the New Millennium : Creativity, Caring & Commitments

HOT NEWS

Background
Committees
Call for Papers
Deadlines
Registration
Program
Paper Acceptance
Reviewers
Venue & Travel Info.
Contact Info.
Host & Sponsors
Related Websites
Discussion Board
Proceedings

HOME

Proceedings

Content

▷ **Full & Short Papers (Educational Agent)**

HOME

An Agent-Based Intelligent Tutoring System

C.M. Bruff and M-A. Williams
University of Newcastle,
NSW 2308, Australia
Email: cgcmb@cc.newcastle.edu.au *and* maryanne@eBusiness.newcastle.edu.au

In this paper we describe the architecture of an agent-based intelligent tutoring system. The agent architecture is based on the BDI framework. The BDI framework is based on the use of Beliefs, Desires and Intentions. Our architecture is under construction using an agent oriented programming language called JACK [1]. JACK provides an agent-oriented development environment. It supports the BDI framework and is built on top of the Java development environment. It not only provides all the necessary agent infrastructure for our architecture, but it also allows us to embed previously developed Java modules in an agent environment. In essence our intelligent tutoring system builds and maintains a student model in a dynamic learning environment where new, possibly inconsistent or uncertain, information is obtained through interactions with the student, and where the system may not have complete knowledge when deciding on the next instructional step. Our architecture supports the development of highly individualised student models using techniques in belief revision, nonmonotonic reasoning and possibility theory.

Keywords: **Educational Agent, Intelligent Tutoring Systems, Artificial Intelligence in Education, Belief Revision**

1 Introduction

In this paper we describe the architecture of an agent-based intelligent tutoring system. The agent architecture is based on the BDI framework. The BDI framework is based on the use of Beliefs, Desires and Intentions. Our architecture is under construction using in an agent-oriented programming language called JACK. JACK provides an agent-oriented development environment. It supports the BDI framework and is built on top of the Java development environment. It not only provides all the necessary agent infrastructure for our architecture, but it also allows us to embed previously developed Java modules in an agent environment. In essence our intelligent tutoring system builds and maintains a student model in a dynamic learning environment where new, possibly inconsistent or uncertain, information is obtained through interactions with the student, and where the system may not have complete knowledge when deciding on the next instructional step. Our architecture supports the development of highly individualised student models using techniques in belief revision, nonmonotonic reasoning and possibility theory. This architecture was described in a previous paper [5].

2 Using Intelligent Agents

Agents vary in capability from procedural wizards to information agents which are used for information filtering and retrieval. Herein the term agent is taken to mean: *an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives,* as suggested by Wooldridge [29]. By autonomous we mean agents have control over both their internal state and their behaviour, in other words they can make choices regarding their actions depending on their internal state and the goal they seek to achieve. Objects in the object-

[1] JACK was developed at Agent-Oriented Software (http://agent-oriented.com)

oriented paradigm do not have this capability. Agents can be both reactive or proactive, in other words, they can react to external events and they can pursue goals. Agents can make run-time decisions that were not foreseen at design time.

It has been consistently argued, for example by Wooldridge [29], that an agent-oriented approach to problem solving and software engineering offers substantial benefits for complex systems. In particular, Jennings [14] has argued that the usual tools identified by Booch [4] of problem decomposition, abstraction and organisation acquire more power if an agent-oriented approach is adopted, because the agent-oriented approach supports distributed processing and at the same time reduces the system's control complexity. Decisions about the next action to be performed are devolved to the agents, and this obviates the need for a global controlling module and as a consequence gives rise to more flexibility and better performance.

3 Implementing Intelligent Agents in JACK

JACK is an agent programming language, which in essence provides agent infrastructure to the Java programming language. Agents are designed in JACK and compiled into standard Java code before being executed. Agent-oriented programming is a highly sophisticated paradigm which is highly suited to intelligent tutoring in a real-time environment.

JACK agents are based on the BDI Framework of Rao and Georgeff [20]. They are autonomous software components that execute plans (intentions) to achieve their goals (desires). The plan chosen at any given time depends on the current set of beliefs. JACK agents can also respond to events as well as striving for goals, in other words they exhibit both proactive (goal-driven) and reactive (event-driven) behaviour. Each agent possesses: (i) a **database** (set of beliefs), (ii) a set of **events** it will respond to, (iii) a set of **goals** that it wishes to achieve, and (iv) a set of **plans** that describe the appropriate responses to events or ways to achieve goals.

A JACK agent remains idle until it is given a goal to pursue, or until it is has to respond to an event. The agents are autonomous and they must determine the appropriate response to goals and events, i.e. the appropriate plan to be executed. A JACK Agent is able to exhibit the following behaviours [JACK Manual]:

- A goal-directed focus. The agent focuses on the objective and not the method chosen to achieve it.
- Real-time context sensitivity. The agent will keep track of which options are applicable at each given moment, and makes decisions about what to try and retry based on the present conditions.
- Real-time validation of approach. The agent will ensure that a chosen course of action is pursued only for as long as certain maintenance conditions continue to be true.
- Concurrency. The agent system is multithreaded. If new goals and events arise, the agent will be able to prioritise.

The JACK Agent Language extends Java in the following ways:

- It defines new base classes, interfaces and methods.
- It provides extensions to the Java syntax to support new agent-oriented classes, definitions and statements.
- It provides semantic extensions (runtime differences) to support the execution model required by an agent-oriented software system.

The JACK Agent Language provides the following five main class-level constructs:

- **Agent**: The agent construct is used to define the behaviour of an intelligent software agent. This includes capabilities an agent has, what type of messages and events it responds to and which plans it will use to achieve its goals.
- **Capability**: The capability construct allows the functional components that make up an agent to be aggregated and reused. A capability can be made up of plans, events, databases and other capabilities.

- **Database**: The database construct provides a generic relational database. It has been designed specifically so that it can be queried using logical members. Logical members are like normal data members, except that they follow the rules of logic programming. Agents can also use regular Java data structures for storing information, but the built-in database can generate events when particular changes occur.
- **Event**: The event construct describes an occurrence that the agent must take an action in response to.
- **Plan**: An agent's plans are analogous to functions. They are the instructions the agent follows to try to achieve its goals and handle its designated events.

4 Agent Communication

Our system is a multi-agent system. Agents need to interact to build and manage the student model. As a consequence our agents require the ability to communicate to one another. JACK provides the infrastructure of the communication but does not specify a particular language or protocols, hence designers can choose the most appropriate for their application. We have chosen KQML as our protocol for exchanging information and knowledge. It is based on speech acts theory as described by Searle [23]. One of the main reasons for our choice is that all the information for understanding the content of the message is included in the communication itself. It is defined by the following protocol structure, as outlined by Huhns and Stephens [13]:

```
(KQML-performative
                    :sender      <word>
                    :receiver    <word>
                    :language    <word>
                    :ontology    <word>
                    :content     <expression>
                    ..)
```

The performatives in our systems are: evaluate, achieve, monitor, revise, extract, tell, and ask. In JACK the agents must know one another's name, and when agents that are communicating are running in separate processes, then the JACK network communications layer needs to be used to allow these processes to communicate. KQML-speaking agents behave as clients and servers, and communication can be synchronous or asynchronous.

5 The Architecture

The ability of an Intelligent Tutoring System to deliver appropriate individualised instruction to a student depends heavily on the type and calibre of the information held about the student in the student model. This in turn depends on the type and level of sophistication of the knowledge representation used in the system and on the effectiveness of the methods used to elicit new information about the student and to incorporate the new information into the student model. Problems arise when new information conflicts with information already in the student model; when the student model contains insufficient information for the tutor to decide on the next instructional step; or when there is uncertainty associated with some of the information about the student - for example, there may be more than one way of interpreting an error made by a student in terms of what the student knows or does not know. There have been many approaches to dealing with these problems.

A number of studies (e.g. Mizoguchi et al [19], Kono et al [17], Giangrandi and Tasso [9]) have applied Truth (or Reason) Maintenance Systems (Doyle [7], DeKleer [6]) to overcome the problem of new information conflicting with old. The TMS identifies the conflicts, which must then be resolved by some domain specific reasoning system. A TMS must maintain not only the beliefs of the student, but also the justifications for them, and therefore use of a TMS is computationally very intensive. Huang and McCalla [12], and Huang [11] have developed a "Logic of Attention", a modification of the TMS which overcomes the problem of efficiency by focusing only on the parts of the student model and instructional planner that are relevant to the current sub-goals. Jones and Poole [15] examined how Reiter's default logic [21] could be used to build expert diagnostic systems.

One general approach to coping with uncertain or incomplete information is to assume that student models do not need to be completely accurate and absolutely precise to be successful. In granularity-based recognition of students' problem solving strategies the philosophy is that student behaviour can be recognised at some level of detail, even if this is very coarse (McCalla and Greer [18]). In the "fuzzy" student model approach (eg Hawkes et al [10] and Katz et al [16]), which is grounded in fuzzy set theory, a student might have partial membership in the set of students who are expert in a particular skill, and partial membership in the set of students who are less expert in that skill. Alternatively, application of Bayesian belief networks (e.g. Villano [25], Shute [24], and Reye [22]) deals with the problem of uncertain information and also facilitates prediction of student knowledge and performance, but most likely at the cost of extensive knowledge engineering and programming.

Intelligent tutoring systems will, in general, have to provide mechanisms to deal with four interrelated information modeling problems:

- Uncertainty of information,
- Incompleteness of information, ie all relevant information may not be known.
- Fusion of information, where information is merged from different sources, and
- Revision of an existing knowledge base when new information is obtained. This new information may be inconsistent with the knowledge base.

Information that is uncertain or incomplete may need to be revised as it is refined over time. Hence, revision of a knowledge base is closely related to modelling both the uncertainty and the incompleteness of information.

In a previous paper [5] we proposed an architecture in which the problem of conflicting information is resolved using methods recently developed by Williams [26, 27, 28] based on the AGM paradigm for belief revision (Alchourron et al [1]). We used possibility theory (see Dubois and Prade [8]) to take care of uncertain information, nonmonotonic reasoning, in particular Reiter's default logic [21] and the formalism of Antoniou and Williams [2], to deal with missing information, and Theory Extraction for fusion.

Our new proposed system architecture is illustrated in Figure 1 below. It has been modified to take advantage of the agent-based architecture, and consists of the three component agents: the knowledge management agent, the student agent, and the inference agent.

6 The Agents

The **Knowledge Management Agent**: This agent mainly responds to events which take the form of requests from other agents. Agents request information regarding such things as domain knowledge, typical errors and misconceptions, suggestions for the next task to present the student. The domain knowledge is structured to suit the application at hand and managed by an agent. We are currently exploring two applications; one based on a mathematical dictionary for schools, The MathProbe[2], and a second focusing on database design for our own courses at the University of Newcastle. An agent that knows about common student errors and misconceptions has a better chance of diagnosing problems than a system without this knowledge. The quality of this knowledge is often what sets a good teacher apart. In both the domains that we have selected this knowledge is well known and widely accepted. For example, *students who consistently place foreign keys in the wrong database table normally do not understand the concept of cardinality of relationships between entities.*

The notion of knowledge granularity has been widely used in the literature (eg. McCalla and Greer [18]). In our architecture granularity is used in both the domain knowledge and in the set of common errors and misconceptions. Levels of granularity fit naturally into the agent architecture and can be used to help the agent choose an appropriate plan.

The **Student Agent** Each student is assigned an individual agent instantiation. The main objective of this agent is to manage the evolution of the student model, i.e. a representation of the student's knowledge about

[2] See http://mathresources.com/mathbrow.html

the domain knowledge and the student' s personal goals and preferences. This model is described using the following components:

- The Student' s Goals using JACK Goals
- The Student' s Preferences using the JACK Database,
- Explicit Knowledge about the Student based on their performance so far using the JACK Database.

The student agent is autonomous and responds to input from the student.
The **Inference Agent**: The inference agent manages a team of agents that provide several forms of useful inference mechanisms and sophisticated reasoning operators. It is not necessary for the agents requesting knowledge to know how the Inference Agent generates that knowledge. The Inference agent uses slave agents for deduction[3], abduction and induction. The belief revision agent[4], possibilistic reasoning agent[4], nonmonotonic reasoning agent[4] and theory extraction agent[4] rely exclusively on the slave agents.

The student agent' s goals will typically vary from student session to student session, and can be customised by a third party such as a human tutor. These goals determine the learning strategies and tasks to be used during a given learning session. The learning strategies together with the database describing the current state of the agent and its knowledge about the student' s capabilities will largely control the agent' s behaviour. These strategies are ultimately implemented via an agent that constructs a learning task hierarchy. This learning task hierarchy is constructed at run time. It can be viewed as a sub-hierarchy of the global task hierarchy customised using the current student profile. This sub-hierarchy is designed to provide feedback about the student that can be used to build and manage the student model during a learning session. In addition, it is used to diagnose student problems and subsequently offer remedial action.

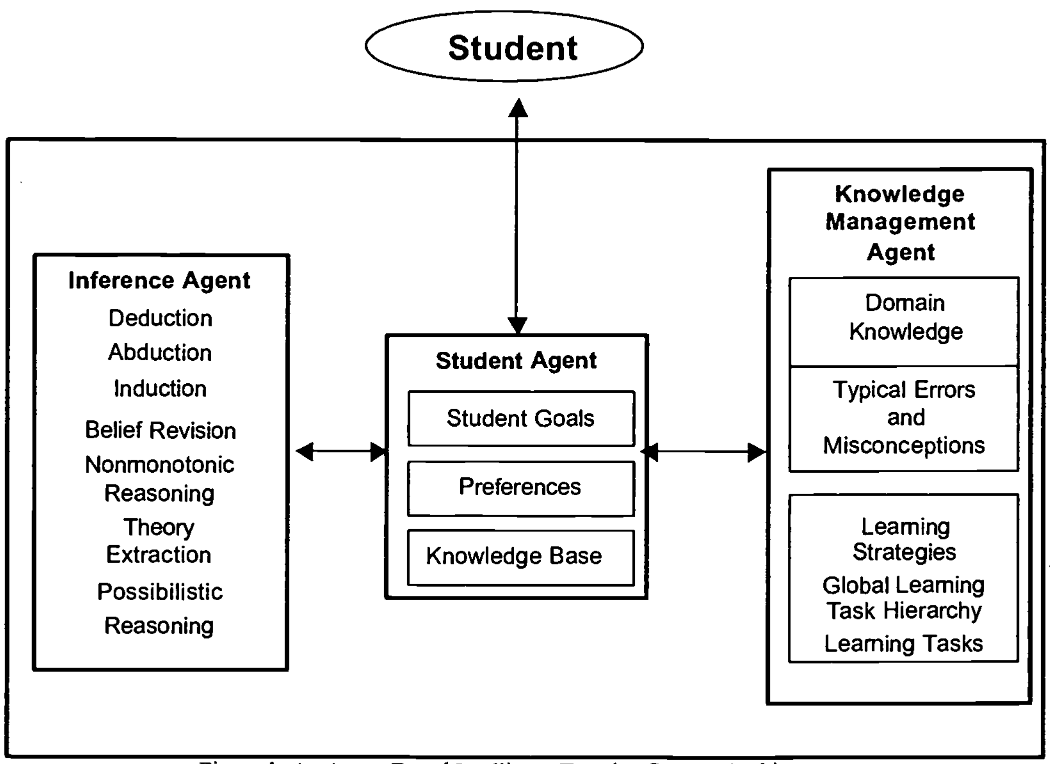


Figure 1: An Agent-Based Intelligent Tutoring System Architecture

[3] See http://ebusiness.newcastle.edu.au/vader
[4] See http://ebusiness.newcastle.edu.au/saten

7 Conclusions

In a previous work we identified several information modeling problems that arise in Intelligent Tutoring Systems: change, incompleteness, information integration, and uncertainty. We described an architecture for an intelligent tutoring system that addressed these problems based on recent developments in knowledge representation and reasoning in the areas of belief revision, possibilistic reasoning, nonmonotonic reasoning and theory extraction.

In this paper we described an agent-based design of the architecture based on the BDI framework. The BDI framework is based on the use of Beliefs, Desires and Intentions. Our architecture is under construction using an agent oriented programming language called JACK. JACK provides an agent-oriented development environment. It supports the BDI framework and is built on top of the Java development environment. It not only provides all the necessary agent infrastructure for our architecture, but it also allows us to use previously developed Java modules in an agent environment.

The main advantage of using an agent-based approach is that the control module has been eliminated. Control has successfully been devolved to the agents, i.e. there is no need for a superagent to oversee the communication and interaction. This leads to better performance and a more customised student learning session.

References

[1] Alchourrón, C., Makinson, D. and Gärdenfors, P. (1985), On the Logic of Theory Change: Partial Meet Functions for Contraction and Revision, *Journal of Symbolic Logic* **50**: 510 - 530.

[2] Antoniou, G. and Williams, M-A., Reasoning with Incomplete and Changing Information, *Journal of Information Science*, 99, **1** & **2**: 83 – 99, (1996).

[3] Antoniou, G. and Williams, M-A., Teaching Intelligent Information Management, in the *Proceedings of the First International Conference on Computational Intelligence and Multimedia Applications*, 380 – 384, (1997).

[4] Booch, G., Object-oriented Analysis and Design with Applications, Addison-Wesley, Reading, MA, 1994.

[5] Bruff, C.M. and Williams, M-A, An Intelligent Tutoring System Architecture based on Knowledge Representation and Reasoning, in *Proceedings of the 7th International Conference on Computers in Education*, IOS Press, 736-743, 1999.

[6] De Kleer, J., An Assumption-based TMS, *Artificial Intelligence*, **28**, 127 – 162 (1986)

[7] Doyle, J., A Truth Maintenance System, *Artificial Intelligence*, **12**, 231 – 272 (1979)

[8] Dubois, D. and Prade, H., Epistemic Entrenchment and Possibilistic Logic, *Artificial Intelligence*, **50**, 223–239, (1992).

[9] Giangrandi, P. and Tasso, C., Truth Maintenance Techniques for Modelling Students' Behaviour, *J. Artificial Intelligence in Education*, **6**, 153 – 202, (1995).

[10] Hawkes, L.W., Derry, S.J. and Rudensteiner, E.A., Individualized Tutoring using an Intelligent Fuzzy Temporal Relational Database, *International Journal of Man-Machine Studies*, **33**, 409 – 429, (1990).

[11] Huang, X., Modelling a Student's Inconsistent Beliefs and Attention, *Student Modelling: the Key to Individualized Knowledge-Based Instruction*, eds. J.E. Greer and G.I. McCalla, 267 – 280, (1994).

[12] Huang, X. and McCalla, G., Instructional Planning using Focus of Attention, *2nd International Conference on Intelligent Tutoring Systems* eds Frasson, C., Gauthier, G. and McCalla, G., Montreal, 443 – 450, (1992).

[13] Huhns, M.N. and Stephens, L.M., Multiagent systems and societies of agents, in *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*, ed. G. Weiss, MIT, 79 – 120 (1999).

[14] Jennings, N.R., On agent-based software engineering, Artificial Intelligence **117**, 277 – 296, (2000).

[15] Jones, M. and Poole, D., An Expert System for Educational Diagnosis based on Default Logic, *Proceedings of the 5th International Workshop on Expert Systems and their Applications*, Vol. **2**, 673 – 683, (1985).

[16] Katz, S., Lesgold, A., Eggan, G. and Gordin, M., Modelling the Student in Sherlock II, *J. Artificial Intelligence in Education*, **3**, p. 495, (1993).

[17] Kono, Y., Ikeda, M., Mizoguchi, R., To Contradict is Human - Student Modelling of Inconsistency, *2nd International Conference on Intelligent Tutoring Systems,* eds Frasson, C., Gauthier, G. and McCalla, G., Montreal, 451 – 458, (1992).

[18] McCalla, G.I. and Greer, J.E., Granularity Based Reasoning and Belief Revision in Student Models, *Student Modelling: the Key to Individualized Knowledge-Based Instruction,* eds. J.E. Greer and G.I. McCalla, 39–62, (1994).

[19] Mizoguchi, R., Ikeda, M. and Kakusho, O., An Innovative Framework for Intelligent Tutoring Systems, *Artificial Intelligence Tools in Education,* eds P. Ercoli and R. Lewis, 105 – 120, (1988).

[20] Rao,A. and Georgeff, M., Modeling rational agents within a BDI-architecture, in the Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, (1991).

[21] Reiter, R., A Logic for Default Reasoning, *Artificial Intelligence*, **13**, 81 – 132, (1980).

[22] Reye, J., A Belief Net Backbone for Student Modelling, *3rd International Conference on Intelligent Tutoring Systems,* Montreal, 596 – 604, (1996).

[23] Searle, J.R., Speech Acts: An essay in the philosophy of language, Cambridge University Press, 1970.

[24] Shute, V., SMART evaluation: Cognitive Diagnosis, Mastery Learning and Remediation, *Proc. Artificial Intelligence in Education,* 123 – 130, (1995).

[25] Villano, M., Probabilistic Student Models: Bayesian Belief Networks and Knowledge Space Theory, *2nd International Conference on Intelligent Tutoring Systems* eds Frasson, C., Gauthier, G. and McCalla, G., Montreal, 491 – 498, (1992).

[26] Williams, M-A., Iterated Belief Revision: A Computational Model, in *the Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, 1541 – 1550, (1995).

[27] Williams, M-A., Towards a Practical Approach to Belief Revision: Reason-Based Change, Luigia Carlucci Aiello and Stuart C. Shapiro (eds), *Proceedings of the Fifth International Joint Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann Publishers, 412 – 421, (1996).

[28] Williams, M-A. Anytime Belief Revision, in the *Proceedings of the Fifteenth International Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, 74 - 80, (1997).

[29] Wooldridge, M., Agent-based software engineering, IEEE Proceedings Software Engineering **144 (1)** 26 – 37, (1997).

Design of Systematic Concept Learning Model Using Computer Education Search Engine

Seong-Guk Kang*, Young-Houn Lee and Seong-Sik Kim*****
Dept. of Computer Education in Korea National University of Education
Tel: +82-043-2303786
Fax: +82-043-2314790
*E-mail: *lovena@comedu.knue.ac.kr, ** lyh@blue.knue.ac.kr,*
****seongkim@cc.knue.ac.kr*

The requirement of education site to apply IT(Information Technology) to class is increasing day by day from development of computer and internet. But, present web-based coursewares are most individual teaching types centered on learners, and each kind of learning theories and models consisting of a main axis of WBI are emphasized on courseware or general, so they cannot be directly applied in education site actually. In this aspect, educationally designed search engine may provide quality information to teachers and be used as good educational tool in class using internet by supporting teaching and learning actively. In this study, we designed computer education search engine and a new learning model introducing it, and we examined that this model may have a positive effect in systematic concept forming process of learners for curriculum concerned.

Keywords: **Computer Education, Search Engine, WBI**

1 Introduction

The requirement of education site to apply IT(Information Technology) to class is increasing day by day from development of computer and internet. But, present web-based coursewares are most individual teaching types centered on learners, and each kind of learning theories and models consisting of a main axis of WBI are emphasized on courseware or general, so they cannot be directly applied in education site actually. The education pattern using internet, which is suggested in • Internet and Education• (Baek, Young-gyun, 1997), got close to school site well relatively, but it will be meaningless if quality sites are not continuously supported. Therefore, plan and learning model to suggest useful site suitable for learning contents to teachers and help them to easily find related information and construct teaching design with those, and to make learners reduce problems of loss of direction and attend class with active attitude. In this aspect, learning model through special search engine may suggest a solution. In the aspect that meaning of knowledge and learning in cognitive constructivism is considered as acquirement process with systematic concept formed on the basis of situation and learner's knowledge, active search process through search engine is very similar to it, and special search engine constructed suitable for curriculum may properly control problems of web-based learning which may occur in learning process.

In this study, we designed computer education search engine and a new learning model introducing it, and we examined that this model may have a positive effect in systematic concept forming process of learners for curriculum concerned. Also, we suggested a design plan of special search engine to support it and were to prepare a base to broaden research range of WBI in school education and to make many people meet with educational information further easily..

2 Systematic Concept Learning Model

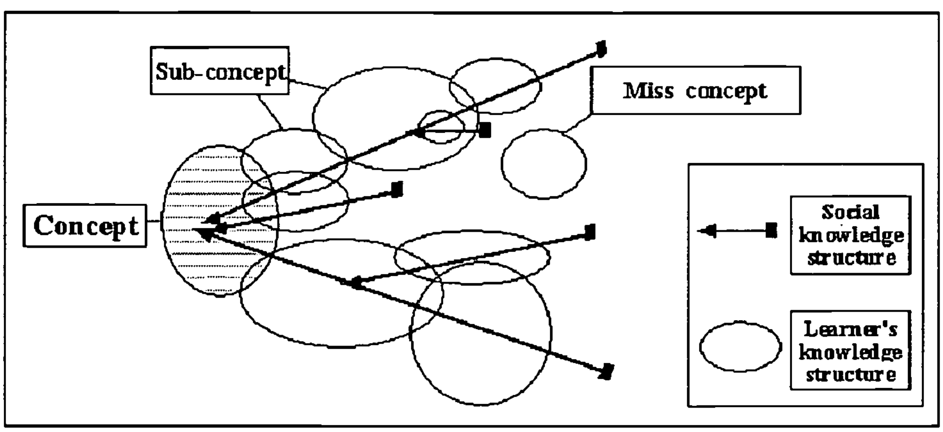
2.1 Meaning of Systematic Concept

Since development of cognitive psychology in 1980s, development of high-technology by pedagogical engineering access and new development of high cognitive ability by development of cognitive psychology have created new-type teaching design concept of constructivism. And, concept of the subject of learning has been changed from intentional concept which focuses on static side and behavioral change to concept of construction of active learner and knowledge to pursue information of volunteer experience and problem solving and reorganize existing knowledge for getting new insight(Mayer, 1992).

The systematic concept is based on concept of knowledge and learning by constructivism, and means complete concept which combines divergent knowledge including originality, creative problem solving and creative transfer of learning and structured knowledge which means partial concept naturally organized in the whole system to which one concept belongs. It accepts basic viewpoints of constructivism - knowledge is structured by the subject of recognition(Duffy; Jonassen, 1991), logically connected(Brown; Duguid, 1989) and formed through social negotiation(ConGlasersfeld, 1989)- in the macroscopic aspect, but it shows the following difference in the aspect that it emphasizes system and transfer of knowledge.

First, semi-fixed opinion of a change in knowledge structure. A change in knowledge structure suggested in constructivism and a viewpoint that it is constructed according to social conditions are right in changes over may centuries. In changes of past education, the structure of knowledge has been changed according to the periodic conditions and change in vision on human, and actually social demands also have been changed. But, structure of changing knowledge is limited if it is premised on a life cycle in which knowledge is constructed, utilized and transferred in human. Furthermore, weight given with meaningful learning activities is not large in a life cycle.

In case of individual knowledge structure constructed intrinsically, the range of individual change may be broad, but generally individual knowledge structure in education is less rational and universal than social knowledge structure. And individual knowledge structure which is far out of social knowledge structure is difficult to be recognized. Therefore, it is assumed that knowledge according to systematic concept follows semi-fixed structure which recognizes partial construct according to individual in socially formed universal structure.



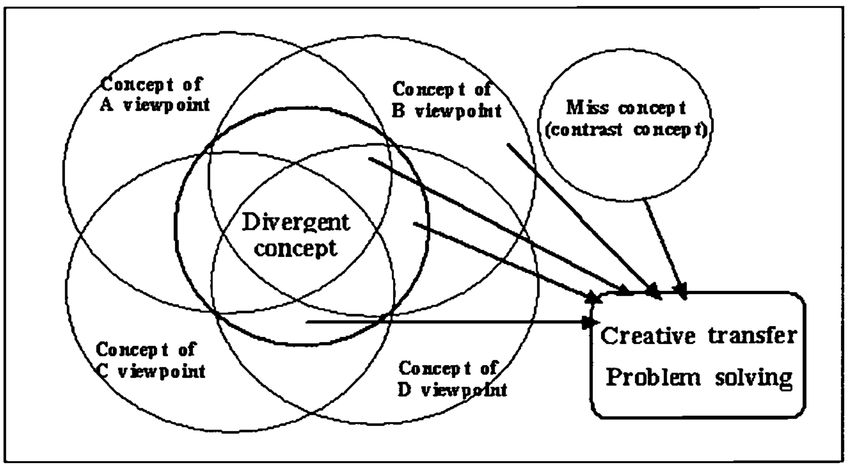
(Figure 1) Structured Concept : Correlation of social knowledge structure and learner's knowledge structure

In (Figure 1), concepts formed in socially standardized knowledge structure and floating knowledge structure formed in learners are diagrammed. The social knowledge structure has a possibility of gradual change according to social conditions, and learner's knowledge structure corrects miss-concept, complements the system and forms complete concept on the basis of social knowledge structure; It is structured concept.

Second, knowledge according to systematic concept does not limited to construction of knowledge, and on the basis of this, it emphasizes that creatively transferable knowledge is complete knowledge. Creative transfer means rapid development into valuable and new knowledge is realized on the basis of existing knowledge. To get a new creature, two processes - building of a knowledge base as a prepared source of creation and transfer of knowledge to problem solving - are required(Kim, Seong-sik, 1999). The base of knowledge may be completed through structured concept and in this process also miss-concept performs accessory role.

Supposing that you perform concept learning of CPU in the concept forming process of computer structure. Concept of CPU in learner is formed rather through CPU position and various visions of CPU - viewpoint of HW developer, computer seller and computer scholar - in computer structure than through direct definition of terms and suggestion of its role. And it forms complete concept corrected and complemented to concept corresponding to miss-concept - CPU in sewage disposal plant. The concept formed as this realizes a source of power for learner to have an ability to cope with various conditions which occur in relation to CPU and to solve a problem creatively and transfer on the basis of the concept of CPU.

As this, concept to make learner oneself construct creative transfer and solutions for various problems is divergent concept. The divergent concept makes concept to be learned be complete in various concepts and systems under social structure, has a synergic effect on obtaining other concerned concepts, and further may be a help to formation of non-concerned concept and improvement of creative thinking power.



(Figure 2) Creative transfer of divergent concept

In (Figure 2), it is diagrammed that divergent concept of learner which is formed through various viewpoints has an effect on creative transfer, problem solving and learning of other concepts. This figure shows that various concepts in learning process and even miss-concept as well as concept formed through learning play an important role in creative transfer.

2.2 Components of Systematic Concept Learning

Components of systematic concept learning are composed of active learner, problem conditions for learning target, learning contents of systematic structure, peripheral concept including miss-concept, teacher as active assistant, media for learning process control and assessment for creative transfer. Roles and contents of these are as [Table 1].

[Table 1] Components and roles of systematic concept learning

Components	Roles and contents
Learner	Learner solves problem conditions as the subject of learning, constructs system of learning contents, and obtains systematic concept.
Problem condition	It is regarded same with learning target and is a final target for acquiring concept. The problem conditions are suggested including a key of problem solving and a motive of learning.
Learning contents	Learning contents are constructed according to social knowledge structure and education course and are constructed structured concept with general nature.
Peripheral concept	It is constructed including miss-concept contrasted with concept of various viewpoints and is suggested in a category which includes learning contents.

Teacher	Teacher prevents breakaway of learner and induces access to structured concept as active sympathizer for problem solving rather than one of knowledge sources. Also teacher constructs peripheral concept connected with learning contents and records it, and assesses the process.
Media	It includes document, video data, web document and clue information connected with these as suggestion form of peripheral concept. And it means virtual space in which learning may be realized.
Assessment	The process to solve problem conditions is recorded and assessed through media. Another problem conditions about which systematically formed concept may generate creative transfer are suggested, and the problem solving process is assessed.

2.3 Basic Premise of Systematic Concept Learning

The systematic concept learning is based on the followings:
First, learner should be prepared well for learning and be motivated properly.

Second, the subject of learning is a community of learners, learners and learners, and learners and teachers.

Third, the structure of concept is semi-fixed, and should be general and proper on the basis of social knowledge structure.

Fourth, problem conditions should include learning process of the target concept, and in consideration of connection between precedent knowledge and transfer, should be set before learning.

Fifth, teacher should select peripheral concept, control learner properly and be responsible for assessment as community which participates in learning actively.

Sixth, learner and teacher should exist in a same learning space. Learning space includes virtual space.

Seventh, tools to suggest and select peripheral concept effectively and control learning process are required.

Eighth, it should focus on concept acquirement and transfer, and in case of learning accompanied with behavior such as practice, special learning procedure should be accompanied.

Ninth, peripheral concept necessary to solve problem conditions and the related information should be fully built.

Finally, systematic concept learning is a guided learning type, is based on learning subject of secondary school or higher, and is not suitable for infant and the lower classes of elementary school relatively.

2.4 Promotive Elements of Systematic Concept Learning

Motivation is an element to promote effective process of systematic concept learning. The promotive element as intrinsic motivation depends on problem conditions and the related interest. Setting of problem conditions similar to real situation induces learner to have an interest on learning itself and maximizes concept acquirement.

And, planning for problem solving, concentration on object, high cognitive recognition of contents to be learned and how to learn it, positive research of new information, clear perception of feedback, self-confidence and satisfaction from achievement, and no anxiety on failure act as elements of learning motivation(Johnson & Johnson, 1985).

The promotive element as extrinsic motivation depends on suggestion method of peripheral concept and compensation for assessment. If it constructs learning process as detective praying which applies search and a motive element reflected in assessment is added to the whole process, extrinsic motivation is formed in learner and natural concept learning may be expected.

2.5 Obstructive Element of Systematic Concept Learning

(1) Uncontrolled Concept
The uncontrolled concepts are never-related concept to obstruct systematic concept formation, incorrect concept, example of too many concepts, and example of too few concepts. The never-related concept in computer education learning process confuses concept structure which is totally different from the existing knowledge structure for the proper part, so it confuses learner. The incorrect concept may break down or damage a well formed concept system in knowledge structure, and too many or too few peripheral concept makes concept system be imperfect.

(2) Non-structured Concept Structure
The social knowledge structure constructed with unsafe structure may have a bad effect on setting direction of concept formation. But, if education course according to educational policy of a nation is reorganized as social knowledge structure, it may not be a significant problem.

(3) Instability of Media
If a form in wh ich peripheral concept is suggested and a space in which learning is performed are unstable, it may decrease learner's motivation and interfere systematic concept formation. Especially, if learning is performed on web, special caution on system stability is required.

(4) Formation of Motivation Contrary to Learning Process
The unexpected motivation according to media or learning method suggested in learning process makes control of learning be difficult. Induction of interest from magnificence and peculiarity of media itself or formation of provocative motivation may act as obstructive factor of learning.

2.6 Web-based Systematic Concept Learning Model

The systematic concept learning theory may have a complete system under the condition that learner and teacher use special search engine and perform class in a same place. Existing web-based learning just depends on web courseware or uses it accessorily, but a characteristic of web-based systematic concept learning model is that web-based learning using traditional class pattern and special search engine is constructed with the same weight.

In this section, components of class design and additional components according to the above are shown. The main design bases are class design theory and problem solving scenario in home and suggestive issues of web-based learning models are referred to. The general model of web-based learning design is inclusively designed, and according to the contents, it is considered that media and construction of learning method may be varied, and selection procedure of these is included in the model itself. But, in case of this web-based concept learning design model, the structure is organized close to concrete class design from several previously selected components such as selection of tool - search engine - and form of learning method. It has the following 8 steps;

2.6.1 Selection of Learning Contents

For class design, first, teacher analyzes what contents should he or she teach with curriculum, related teaching materials, guidebook and reference data. Especially, for computer education, the learning contents are changed very fast, so teacher's role is very important. For example, if operating system is learning content, although learning content in curriculum and teaching materials is Window 95, it is required to be reanalyzed as Window 98 according to learner's level and social conditions. The proper learning target should be set by analyzing characteristics of selected learning contents.

2.6.2 Setting of Learning Target

The learning target should be set on the basis of selected learning contents. Settings - 'classification and definition of concept, and suggestion of base', 'use of effective method for concept acquirement', 'search of concept and application of related technology', and 'understanding of organization and construction of concept and taking an interest' - are possible.

2.6.3 Organization of Learning Concept

This step organizes concepts connected with learning target concretely, analyzes concept structure which has already been organized in learner, and defines position and peripheral concepts in the structure of concept to be learned, and also defines positive miss-concept and negative miss-concept for expectable miss-concept. Teacher uses special search engine as tool in this process, and terms dictionary and related information suggest peripheral concept and miss-concept. And it may be used to find the position in classification system. Uncontrolled concept and non-structured concept which act as obstructive factor of systematic concept learning in organization of concept should be prevented.

2.6.4 Construction of Problem Conditions

Teacher should establish concrete problem conditions to acquire concept to be formed in learner. Problem conditions should be constructed to be able to induce motivation and interest from learner, and it should be constructed as form to purse achievement of learning target and to be far apart from statement of learning target. From this, it may be difficult for learner to recognize a fact that he or she learns. Here, promotive factor of systematic concept should be applied. And, problem conditions to promote creative transfer of concept formed in learner at the final step should be considered.

2.6.5 Construction of Learning Process

Construction of learning process designs the whole structure to be performed in real class. Concentration of learner is induced, and learning target is suggested, and initiative factor is learned, and factors necessary for the whole parts such as stimulation search, learning guide, performance inducement, feedback are designed.

2.6.6 Selection of Learning Method

This step constructs organization and role of learner. This step has two types; first, teacher prepares organization and role previously according to contents, and second, teacher prepares organization and role in the course of class. This step constructs groups to be able to perform cooperative learning and give competitive factors between groups. This process suggests problem conditions designed by teacher or reorganizes it through interaction with learner, and designs concept searching method to solve problem conditions in learner groups.

2.6.7 Teaching and Learning

In this step, teaching and learning are realized. Teacher may suggest prepared document or other learning data, and learner systemizes and acquires concept through concept search process. The special search engine searching process constructed by teacher in the learning process construction step makes learner go through systematic concept search process in partially controlled search structure and record this process, so it is used in learning assessment. Learner acquires systematic concept and peripheral concept and positive miss-concept in search process. This forms formation of extrinsic concept to generate creative transfer. Teacher participates in problem solving process of learner as positive search assistant and performs minimal controlling role.

2.6.8 Control and Assessment of Learning

This step assesses whether concept formation of learner agrees with learning target, suggests problem conditions for promotion and confirmation of creative transfer and solves them. Also, this step assesses concept formation and establishes the complemental policy on the basis of learning process of learner recorded in search engine.

3 Design Plan of Special Search Engine

In this study, we performed required analysis and inhouse study for groups of teachers, students, and specialists in education to develop computer education special search engine as space in which systematic concept learning will be realized. In this, key points are as follows;

First, detail information of site and the reliability aspect. By detail information of site including characteristics of site, key data, subject of use, and speed, learner may greatly decrease access time to necessary educational information. Also, functions to increase reliability are required such as link keeping function to remove non-serviced site and operation of professional surfer by directories.

Second, systematic user-centered classification type. Educational information should be specialized and subdivided into classification by learning ranges and systems, by education courses, and by general ranges. Also, functions that information is sorted and suggested by user classes are required.

Third, provision of terms dictionary. Terms dictionary to be able to correctly suggest concepts that learner requires should be suggested as hypertext or hypermedia type.

Fourth, formation of community. Realtime and non-realtime community tool is needed so that formation of community between learner and learner, learner and teacher be possible in special search engine.

Fifth, design and assessment of learning process. Functions that teacher supports each kind of design necessary for learning process such as motivation factors, key information, expectable learning process, and functions to record and assess search process of learner are required.

4 Conclusion

Recent WBI studies have been centered on web courseware and remote education site based on it, and with this trend, web browser based newest technologies are being developed and applied. It is very positive in remote education and will meet various demands of learners. With this, also we should consider it in school education site. This study suggested an education model which is further closer to demand of the site in computer education, and designed design plans of special search engine necessary for it. We are designing and implementing further concrete real class model and detail functions and DB of search engine, and through this, will examine systematic concept forming process of learner. We hope this study will be a turning-point at which computer education may be activated in curriculum education.

References

[1] Anderson, E. S., K. V. Grude, T. Haug and J. R., "Turner. Goal Directed Project Management", London : Kogan Page, (1987)

[2] Anita E. Woolfolk (1998). "Educational Psycology" , Hakmoon Publisher

[3] Denis dion Jr. antonie Escobar etc, "Develpment of Educational Tools : a Web-Oriented Approach", (http://www.gel.ulaval.ca/~ift19965) (1996)

[4] Glaser, R., "Components of Psychology of Instruction : Toward a Science of Design", Review of Educational Research, (1997)

[5] Harasim, L. Foreword., "In From Bulletin Boards to Electornic Universities : Distance Education", Computer-Mediated Communication, and Online Education : The American Center for the Study fo Distance Education, (1987)

[6] H.A. Latchman etc, "Information technology enhanced learning in distance and convertional education". IEEE trasactions on education, (1999)

[7] James E. Baker etc, "A New Model for Algorithm Animation Over the WWW", (1999)

[8] Jenkins, Charlotte, Mike Jackson, Peter Burden, and Jon Wallis, "Searching the world wide web : an evaluation of available tools and methodologies", Information and Software Technology, pp. 985-994, (1998)

[9] Miller, T., E., and P. C. Clemente. "The 1997 American Internet User Survey : Realities Beyond the Hypertext", GVU's 8th WWW Survey Result, (1997)

[10] R. S. Peters . "Anthropology and Education" , Pedagogical Science Publisher, (1980)

[11] R. S. Peters . "Pedagogical Philosopher Criticism Theory", Seogwang Publisher, (1989)

[12] TaeWook Lee etc, "Computer Curriculum Pedagogy" , Hyongsol Publisher, (1998)

[13] Baek, Young-gyun . "Design of Web-based Learning" , Yangsowon, (1999)

[14] Gang, Mi-ryang, "Effect of Constructivism Teaching-Learning on Social Morality of Children" , master's degree paper of graduate school of education, Kyung Hee Univ., (1999)

[15] SeongSik Kim, "The Computer Education for Creative Transfer" , paper of the Computer Education Society, (1997)

[16] YoungHoun Lee, "Design and Implementation of Search Engine with Web and DB linkage", master's degree paper of Korea National University of Education, (1998)

Educational Agents and the Social Construction of Knowledge: some issues and implications

Carolyn Dowling
Australian Catholic University
115 Victoria Parade, Fitzroy, Victoria 3065, AUSTRALIA
Phone: (61 + 3) 9241 4456
Fax: (61 + 3) 9241 4546
Email: c.dowling@patrick.acu.edu.au

The use of intelligent software agents within computer mediated learning environments is currently an important focus of research and development in both AI and educational contexts. Roles envisaged and implemented include those of tutor, of 'manager', of information seeker and of fellow learner. Each of these raises its own special challenges in relation both to the capabilities of the software and to our understandings in regard to the nature of the learning process. High on the list of factors currently believed to contribute to effective learning is social interaction in the service of knowledge construction. Within many electronic learning environments we are currently witnessing the emergence of a new participant in the social interactions that mediate learning. The substitution of computer programs possessed of varying degrees of intelligence, autonomy and 'personality', for certain dimensions of human presence within the computer based classroom raises a number of questions related to the processes through which knowledge is socially constructed, and to the qualities which are necessary to ensure successful participation in those processes. Through discussion of both theoretical perspectives and practical examples, this paper explores some of these issues.

Keywords: **AI in Education, Educational Agents, Intelligent Tutoring Systems, Interactive Learning Environments, Networked Social Learning, Teaching and Learning Process**

1 Introduction

Developments in computing and information technology in recent years have rapidly propelled the notion of intelligent software agents from concept to implementation. Today, whether or not we are always aware of them, they are an integral part of a growing number of computing environments. From the invisible armies of knowbots and related entities scurrying around the Net in the service of increasingly sophisticated search engines to the cheery little characters who pop up on our screens offering assistance with anything from formatting a date to constructing a complex multimedia presentation, or the 'personalities' with whom we interact in chat rooms in happy ignorance of their purely digital nature, intelligent agents are alive and well and are multiplying rapidly.

An early but still useful conception of a software agent is, "A character, enacted by the computer, who acts on behalf of the user in a virtual environment", useful in mediating " ... a relationship between the labyrinthine precision of computers and the fuzzy complexity of man [10, p. 355]. Later definitions tend to be expressed in more functional terms, such as, "An agent can be viewed as an object which has a goal and autonomously solves problems through interaction, such as collaboration, competition, negotiation and so on" [9]. This definition has some similarities with that offered by Maes [12] who defines an agent as:

"A computational system which:
- is long lived;

- has goals, sensors and effectors;
- decides autonomously which actions to take in the current situation to maximize progress towards its (time-varying) goals" [12, slide 5].

Summarising the writings of a number of researchers, Aroyo and Kommers [1, p. 237] identify four major characteristics of agents as being autonomy, responsiveness or reactiveness, pro-activeness and social ability. Other qualities frequently proposed, but not supported by all researchers or indeed by all users, include the ability to learn from experience and consequently to respond in flexible and possibly unforeseen ways to particular situations, and the possession of a believable 'character' or personality as a basis for social interaction.

It appears that a combination of factors has contributed to the current proliferation of software agents. Apart from the technical developments which have opened up the possibility of implementing what were previously largely theoretical conceptions, there is our very real need for assistance as we operate within computing environments characterised by rapid change, large quantities of extraordinarily complex information, and a lack of common organisational structures through which information may be accessed and managed. As Laurel predicted, there are now many situations in which, in the interests of efficiency, some form of 'intelligent' mediation is required between computer systems and the needs of users.

There are, of course, different forms that this mediation could have taken. The strong propensity for most users to accept assistance in the form of a more or less personified entity as largely unproblematic undoubtedly derives at least in part from the anthropomorphic elements implicit in most computer interfaces from the earliest days of computing. It can be strongly argued that a degree of personification has always been automatically and inevitably conferred as much by a program's use of language as a component of the interface as by our everyday understandings of the 'intelligence', albeit artificial, of computers. Intelligence and language use are, after all, key defining attributes of human beings.

Not only are we accustomed to interacting with computers as though they share with us a degree of 'humanity', but in a number of areas of activity we have been persuaded to value 'social' interaction particularly highly. Education is a good example, given the extent to which our current understandings of learning depend upon an acceptance of the belief that knowledge is to a large extent socially constructed. In the current drive to move teaching and learning online, the notion of agency in computing has found a strong ally and a vehicle for expansion. Unless the social interactions that mediate learning in face to face environments can be shown to have a digital equivalent, proponents of online courses will be forever 'on the back foot', with their products being regarded by most educators as second best. While courses incorporating the communications facilities of the Internet certainly go a considerable way in promoting interactions of various types between teacher and student and also between student and student, the possibility of using software agents to create an illusion of interpersonal interaction so convincing as to achieve pedagogical outcomes equivalent to those deriving from a relationship with another human being is extremely enticing to the designers of electronic learning environments.

2 Some examples of socially interactive pedagogical agents

Johnson [7] has proposed the following definition the role of a pedagogical agent as distinct from those designed for other purposes:

> "Pedagogical agents are autonomous agents that support human learning, by interacting with students in the context of interactive learning environments. They extend and improve upon previous work on intelligent tutoring systems in a number of ways. They adapt their behaviour to the dynamic state of the learning environment, taking advantage of learning opportunities as they arise. They can support collaborative learning as well as individualized learning, because multiple students and agents can interact in a shared environment. Given a suitably rich user interface, pedagogical agents are capable of a wide spectrum of instructionally effective interactions with students, including multimodal dialog. Animated pedagogical agents can promote student motivation and engagement, and engender affective as well as cognitive responses" [7, p. 13].

This is a comprehensive and optimistic vision, incorporating a number of possible roles for software agents within educational environments. Types of agents currently implemented in projects around the world include record keepers, information seekers, testers, facilitators of collaboration, tutors or instructors, fellow learners, and tutees. Of special interest in regard to this paper are those that contribute to the overtly social dimensions of the learning environment. The last three listed most clearly fulfil this criterion.

2.1 Agents as instructors

There is a sense in which perceptions of the role of computers in the learning process have come full circle. Early models of the role of 'computer as tutor' in the form of drill and practice style of instructional software, generally based on Skinnerian principles and incorporating very limited interaction between user and computer, have long been rejected by most educators in favour of a range of other more acceptable guises including that of a learning tool, an information source, and a learning 'space'. With the development of agent technologies, as Johnson suggests, new possibilities now exist for incorporating computers within the learning environment in a range of socially interactive roles, including that of 'tutor', through modes of interaction more in keeping with current pedagogical theory.

It is commonly asserted that the presence of computers in classrooms has itself played a part in modifying the image of the teacher as the 'sage on the stage' in favour of a more collaborative model. Not surprisingly, these changing concepts are well reflected in many implementations of 'agent as teacher'. As Solomos and Avouris [18] write, for instance:

> "The user mental model of the system should be based on the metaphor of the "invited professor" rather than the "knowing everything own tutor". ... Our first findings confirm the observation that today's users, accustomed to hypertext-like interaction, are more likely to accept this collaborative teaching metaphor, according to which their tutoring system is viewed as an intelligent hypertext browser, offering links to other tutoring systems with the right content and at the right time" [18, p. 259].

The increasingly popular concept of the teacher as a facilitator of learning is also reflected in such statements as: "Each student working on the project will have an agent, operating in the background, watching progress, measuring it against the plan, and taking remedial action when necessary" [19, p. 362].

2.2 Agents as fellow learners

A style of agent of special significance in the context of socially constructivist theories of learning is the 'fellow learner', which to differing degrees might be presumed to include all participants within the learning environment. If agents are to gain widespread acceptance in the field of education, this is an important area for research and development. Since the 1980s Chan [2, 3] and colleagues have been working on a range of models of socially interactive agents for learning environments, perhaps the best known being the 'learning companion' – a software entity having limited knowledge of the domain in question, conceptualised as a fellow learner with whom the student may collaborate and even disagree. As in real life, some of these learning companions may be better informed than the student in the relevant domain of knowledge, while others may know less. Perhaps not surprisingly, in learning environments for younger students, animals are a popular choice of persona for such agents, as in this example of a networked learning environment for Taiwanese high school students, as described by Chan:

> "The Dalmation is having the same performance as the student. ... Another animal companion is Dragon, like one of those animal companions in Mulan, a Disney cartoon of this summer. This dragon will "learn" (mainly rote learning) from the student and also from other students on the Net and so may know more than the student. At certain point it'll stop learning and come back to teach the student. In a way, Dragon is protecting the student" [3].

An interesting development of this concept is presented by Sheremetov and Nunez [16, p. 310], who describe the function of a 'monitor agent' as being to modify the role, behaviour or expertise of learning companions from that of strong group leader to a weaker companion or even a passive observer, depending on its interpretation of the degree of guidance required by the learner.

2.3 Agents as pupils

We are all familiar with the common wisdom that we learn through teaching others. At the school level, many educators have long been familiar with the concept of the computer as 'tutee' through the use of the Logo programming language, in which 'teaching the turtle' was a familiar metaphor for the activity of programming. More recently, a number of researchers have explored the translation of this concept into electronic learning contexts where agents exist to be 'taught' by the student user, as in the example from Chan quoted above. A further example is described by Ju [8] who writes of a computer based peer tutoring system employing two categories of agent – an 'expert', and a 'learner':

> "... students become active learners who are guided to learn by teaching a computer. After the students watch how the computer expert solves a set of linear equations [the program] helps the human student act as a teacher in order to learn more about the subject matter. At this time, the computer plays the role of a student ..." [8, p. 559].

3 Some issues for consideration

3.1 Multiple agents

Most agent based systems utilise a number of agents, many of them capable of a complex range of interactions with the student, with one another, and increasingly with agents associated with other programs. Their individual purposes derive from theoretical analyses of the component tasks and activities that are included in the larger scale pedagogical interactions of human beings. As educators, and indeed as students, we may simultaneously enact a range of roles within the educational environment. The apparently unitary activity of 'teaching' involves such elements as demonstrating, guiding, telling, questioning, explaining, testing, motivating, criticising – even learning! Many researchers consider that the electronic medium makes it feasible to identify and separate out these diverse functions. These can then be enacted through different configurations of agents working in relationships which ranging from collaboration to competition.

An example is the Multiple Agent Tutoring System (MATS) described by Solomos and Avouris:

> " MATS is a prototype that models a "one student-many teachers" learning situation. Each MATS agent represents a tutor, capable of teaching a distinct subject. All MATS tutors are also capable of collaborating with each other for solving learning difficulties that their students may have" [18, p. 243].

Strategies for most efficaciously combining the activities of multiple agents such as these necessitate a complex agent architecture, and understandably occupy a great deal of the research agenda in this area. Of interest in relation to their participation in the social construction of knowledge is the fact that one of the most common metaphors employed by a number of researchers and courseware designers is that of a 'society' of agents, a conception reminiscent of Minsky's *The Society of Mind* [14], Gardner's multiple intelligences [6] and other related theories of cognition and behaviour. In describing the different aspects of the design of their "multi-agent, computer-based interactive environment", for example, Costa and Perkusich [4, p. 196], drawing on the work of Franklin and Graesser [5] refer to their aggregation of agents quite specifically as a 'society'.

> "The society [of artificial tutoring agents] is an open multi-agent system made up of a collection of tutoring agents that co-operate among themselves to promote the learning of a certain human learner. This society is designed to be open and dynamic in the sense that it allows maintenance operations such as the entry and the exit of agents, besides eventual modifications in the knowledge and in the inference mechanisms of an agent. Each agent defines an expert tutor in some domain, having the necessary knowledge to solve problems in this domain. These agents are cognitive and possess properties like autonomy, goal-oriented, social ability" [4, pp. 197-198].

While on the one hand, the variety of functions of agents within a multi-agent environment must also be appreciated as an attempt to realise the type of rich user interface which Johnson suggests is necessary if the pedagogical interactions within electronic learning environments are to approximate to any degree to the face to face educational experience, some educators have concerns in regard to the assumptions underlying these practices. They argue that such developments are underpinned by a reductionist rather than a holistic understanding of the processes and relationships involved in teaching and learning. In separating out the different components of pedagogical interactions, are we enabling each part to be realised more effectively, or are we failing to acknowledge that the global act of human teaching may in fact be more than the sum of its component parts? It seems reasonable to suggest that firm judgments on issues such as this must await greater experience of the roles of agents within these learning contexts.

3.2 Personification

Another focus of debate concerns the degree to which personification is helpful in fostering fruitful pedagogical interaction between the human learner and software agents. This question clearly relates more to the 'socially interactive' agents than to those fulfilling more tool-like functions, which arguably require far less in the way of 'personality'. As noted earlier in this paper, there are clear arguments for accepting that a degree of personification of computer interfaces is inevitable. As Shirk puts it:

> "Although there is some dispute among software critics concerning the advisability of having 'personalities' in computer programs, their presence seems unavoidable. Any time there is communication between a computer and a human, the information presented by the computer has a certain style, diction, and tone of voice which impact upon the human's attitude and response toward the software" [17, p. 320].

However the extent to which this should be deliberately fostered is less clear, although many feel intuitively that it should be an important element in the creation of an electronic learning environment characterised by interactions which can reasonably be described as 'social'.

An important aspect of the representation of 'character' or personality is visual appearance. Interestingly, both research and experience suggest that the relationship in the case of software agents is far from straightforward, and that a mismatch between realism in appearance and the apparent knowledge level of the agent can have a deleterious effect on credibility. The more visually realistic the representation, the higher the expectations of the user in relation to the appropriateness and 'intelligence' of utterances and actions. Agents that 'look' smart and 'act' or 'talk' dumb are poorly received by many users, who express a higher tolerance for the limitations of a 'character' more sketchily represented, for instance through cartoon-like graphics. As Masterton, writes, for instance, "A common problem with AI programs that interact with humans is that they must present themselves in a way that reflects their ability. Where there is a conflict between the ability of the system and the users' perception of that ability a breakdown occurs and users may either fail to exploit its full potential or become frustrated with its shortcomings" [13, p. 215]. He goes on to suggest the implementation of a degree of anthropomorphism intended to convey qualities such as friendliness and usefulness, without the implication of possession of full human capabilities [13, p. 211]. He describes the development and role of such an entity in the form of a VTA (Virtual Teaching Assistant) which is able to introduce topics and answer simple questions, the more complex types of exposition and interaction being left to the human teacher. In terms of a traditional scenario at university level, the VTA functions somewhat like a tutor or demonstrator as distinct from a lecturer. "In this way faculty is left free of the guiding and assisting issues of the course and is able to concentrate on more complex questions and higher level issues generated during the course" [13, p. 211].

Further instances of this principle are the examples of agents presented as animals discussed earlier in this paper. Our expectations in regard the cognitive skills of animals may well be more appropriate to the capabilities of software agents than are our experiences of human-to-human interactions.

3.3 Autonomy

Closely related to the 'intelligence' of software agents is the issue of autonomy, in particular the degree to which an agent should be furnished with pre-existing goals which might lead it to take particular action without instruction from the user, and even contrary to what the user might perceive as his or her interests and wishes. Exploring the implications of such entities existing and interacting within virtual reality environments, Loeffler [11], for instance, notes that the unpredictability resulting from significant autonomy might well result in agents who are less 'helpful' to us than we might hope or indeed expect. It is easy to slip from such considerations into the need for a contemporary version of Asimov's laws of robotics as conceived in fictional terms more than 30 years ago!

In educational contexts, the implications of autonomy, particularly in terms of control of and responsibility towards the learner, are potentially extremely complex and difficult to address without more exposure to these types of software, and indeed it is quite likely that such experience may cause community understandings in regard to appropriate relationships between the 'human' and the 'not human' in electronic contexts to develop and change over time. In the short term, current trends in educational thinking which favour giving more control and autonomy to the learner would appear to be more in line with the thinking of researchers such as Schneiderman who favour 'direct manipulation' over the development of interactive agents with a significant degree of independence of action. Where agents are involved, they may be programmed so as to exercise control over the learner on behalf of the creator of the learning environment, or they may be configured so as to be more sensitive to a user model, and more responsive to instruction from the user/student. In the latter instance, the agent would have a greater degree of responsibility to the needs and wishes of the learner, but this may not be in keeping with the pedagogical goals of the teacher.

Trust is another aspect of the teacher/learner relationship that is complicated by the degree of autonomy with which a pedagogical agent is endowed. To the extent that the programmer chooses to delegate certain functions and responsibilities to the agent, it is their problem, but it may also be an issue for students, particularly those with more insight into the nature of the agents with which they are interacting.

A further concern in regard to the autonomy of pedagogical agents relates to the issue of intervention in the learning process. Despite the finding of Aroyo and Commers [1] that pro-activity is a quality frequently sought after in agents, there is an important issue of balance to be addressed in relation to the educational process. It is well accepted that a high degree of unsought assistance whether from a human teacher or an excessively diligent and proactive agent can be quite detrimental, in particular to the metacognitive aspects of learning. Of course this is also an issue for teachers and learners in face to face educational contexts!

3.4 Level of participation in the social construction of knowledge

The belief that it is possible for agents to participate effectively in the social aspects of knowledge construction is central to the work of many theorists and researchers. Sheremetov and Nunez [16], for example, whose works derives overtly from the theoretical frameworks of Piaget and Vygotsky, argue that:
"The design of learning environments, virtual or not, aims to promote productive interactions. In this type of learning a student changes from being a passive information receiver to an active collaborator, interacting with the tutors and colleagues in the learning process. Learning does not only result from acquiring knowledge, solving problems or using tools, but also from interacting about these on-going activities with persons and agents"[16, p.305 – 306].

In relation to their specific project they write: "Our emphasis lies in the role of interactions in an artificial learning community as a group of real and artificial learners, tutors, and facilitators, working, supporting and learning from each other [16, p. 306]. But however personified and autonomous the software agent, can it really be said to participate fully in the social construction of knowledge? It has been argued quite extensively that even the most heavily personified of computer programs suffer from an intrinsic lack of ability to participate in the metacognitive aspects of learning. Pufall [15], for instance, expresses a strong belief that a computer program is unable at any level commensurate with human capacities to modify its own knowledge structures or cognitive processes, and so cannot be regarded as a co-constructor of knowledge in a meaningful sense. While this might well have been the case in relation to earlier computer based learning environments, can we continue to make the same claims with confidence today or in the future? The capacity of software to 'learn' and adapt to experience through the incorporation of new information, the appropriate modification of its representation of the context in which it functions (its 'world') and of its inference mechanisms, is undoubtedly increasing. One way of considering this question might be to look at it in terms of the type of distinction sometimes made between 'hard' and 'soft' notions of artificial intelligence. If our test of full participation depends on an understanding that the agent has 'learnt' in precisely the same way that the human has learnt, then we will have difficulty accepting the electronic entity as genuine co-constructor of knowledge. If, however, we make our claim on the grounds that it appears to the human learner that the agent has participated in the learning that has taken place, then perhaps we can at least tentatively admit such a piece of software to membership of the social milieu which has mediated the educational experience.

Conclusions

It is clear that developments in agent technology have created a range of new possibilities in terms of aligning computers more strongly with prevailing educational theories and philosophies. In considering the many issues which might be raised in relation to the nature and roles of pedagogical agents, there are three overarching questions. Firstly, do agents have the potential to enhance learning, or do they threaten to undermine those aspects of the educational enterprise that we most value? Secondly, to what extent might they assist in the replication of the social dimensions of face to face learning within online environments? Thirdly, do they go further than this, and create new possibilities in regard to the social mediation of learning? To the extent that visions such as those of Johnson [7] are able to be realised, we may be faced one day with the need to re-evaluate our attitudes regarding the relative merits of a human teacher and an electronic entity designed specifically for educational purposes. But while the rhetoric of developers often suggests an ideal surpassing the sometimes imperfect realities of human-to-human pedagogical interactions, the 'jury' of online learners and of educators is still out.

References

[1] Aroyo, L & Kommers, P., "Preface – Intelligent Agents for Educational Computer-Aided Systems", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

[2] Chan, T.-W., "Learning companion systems, social learning systems, and the global social learning club", Journal of AI in Education 7 (2) (1996).

[3] Chan, T.-W. (1998), "The past, present, and future of educational agents", http://www.apc.src.ncu.edu.tw/apc/ppt_chan.html (accessed on February 3, 2000).

[4] Costa, E.deB. & Perkusich, A., "Designing a Multi-Agent Interactive Learning Environment", in Proceedings of International Conference on Computers in Education (ICCE '97), Kuching, Sarawak, Malaysia, December 2 – 6, pp. 196 – 203, (1997).

[5] Franklin, S. & Graesser, A., "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages, Berlin: Springer-Verlag (1996).

[6] Gardner, H., "Frames of mind: the theory of multiple intelligences", New York: Basic Books (1983).

[7] Johnson, W.L., "Pedagogical Agents", Proceedings of, the Sixth International Conference on Computers in Education (ICCE '98), Beijing, China, October 14 – 17, pp.13 – 22, (1998).

[8] Ju, Y. (1998), 'Development and formative evaluation of a computer-based peer tutoring system', Proceedings of, the Sixth International Conference on Computers in Education (ICCE '98), Beijing, China, October 14 – 17, pp. 559 – 566, (1998).

[9] Kinoshita, T. & Sugawara, K., "Agent oriented computing", Tokyo: Soft Research Center (1995).

[10] Laurel, B., "Interface agents: metaphors with character", in Laurel, B. (ed.) The Art of Human-Computer Interface Design, Reading, Massachusetts: Addison-Wesley (1990).

[11] Loeffler, C.E., "Artificial Life of Agents", Proceedings of the IFIP World Conference on IT Tools, Canberra, Australia, September 2 – 6, pp. 48 – 55, (1996).

[12] Maes, P., "Software Agents Tutorial", http://www.media.mit.edu/people/pattie/CHI97 (1997), (accessed on March 18, 2000).

[13] Masterton, S., "Computer support for learners using intelligent educational agents: the way forward", Proceedings of, the Sixth International Conference on Computers in Education (ICCE '98), Beijing, China, October 14 – 17, pp. 211 - 219, (1998).

[14] Minsky, M., "The Society of Mind", London: Picador/Heinemann, (1985).

[15] Pufall, P., "Function in Piaget's system: some notes for constructors of microworlds", in Forman, G. & Pufall, P. (eds) (1988), "Constructivism in the Computer Age", Hillsdale, New Jersey: Lawrence Erlbaum Associates, (1988).

[16] Sheremetov, L. & Nunez, G., "Multi-Agent Framework for Virtual Learning Spaces", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

[17] Shirk, H.N., "Technical writers as computer scientists: the challenges of online documentation", in Barrett, E. (ed.), "Text, Context and Hypertext: writing with and for the computer", Cambridge, Massachusetts: MIT Press (1988).

[18] Solomos, K & Avouris, N. (1999), "Learning from Multiple Collaborating Intelligent Tutors: An Agent-Based Approach", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

[19] Whatley, J., Staniford, G, Beer, M. &Scown, P. (1999), "Intelligent Agents to Support Students Working in Groups Online", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

A Real-time Handwriting Communication System for Distance Education+

Hsin-Chu Chen* Jho-Ju Tu and Jianping Zhang*****
Department of Computer and Information Sciences, Clark Atlanta University, 223 J.P.Brawley Dr., SW, Atlanta, GA 30314. hchen@star.cau.edu.
** *Instructional Technology Center and School of Education, Georgia Southwestern State University, 800 Wheatley St., Americus, GA 31709. tu@canes.gsw.edu*
*** *Department of Computer Science, Utah State University, Logan, Utah 84322. jianping@zhang.cs.usu.edu*

1 Introduction

In this paper we present the design and implementation of a handwriting communication system for real-time graphical information exchange. This system provides an environment for a user to write and erase messages on a computer screen using a light pen or a mouse and to transmit the handwriting message to another user on the Internet in real time. The communication techniques employed for the system include the basic client-server model and peer-to-peer model. The client-server model is mainly for sending handwriting information using the world wide web. The peer-to-peer model, however, is aimed at real-time communications between two end users on the network to conduct instant dialogues. The system is implemented using Java . It can be integrated into many different applications such as collaborative learning, on-line discussions, and distance education.

2 Communication Models

A handwriting communication system may be implemented using a client-server or a peer-to-peer model. Each model has its cons and pros. The client-server model, in which the client sends requests to the server and the server responds to the request [3], works well in situations when immediate responses are not necessary. For a real-time instant dialogue or information exchange, however, the client-server model alone is somewhat restrictive due to its limited degree of interactivity. To achieve a full degree of interactivity for this type of applications, a peer-to-peer model that allows for full duplex real-time communications is more appropriate, since the two end users of the system may send and receive information at the same time, roughly speaking.

In addition to communication models, we must also take into account the nature of the communication protocols and decide which to use. Information exchange carried over the Internet normally requires support from either TCP or UDP, which are the two protocols operated at the transport layer in the TCP/IP protocol suite [1]. When TCP is employed, the information is sent as a data stream, similar to a telephone conversation. Since TCP requires a connection setup prior to transferring data, it incurs an initial time delay. UDP, on the other hand, does not require such a connection setup. However, the delivery of datagram packets, which are independent data units sent individually from the source to the destination, is not guaranteed. Datagram packets may arrive out of order too. For textual information, UDP may not be a bad choice because the user normally can tolerate, to certain degree, occasional loss of packets or misplaced textual data. In our handwriting system, the handwritten information is represented as numerical data which are sensitive to the loss of any single bit of information, therefore, TCP is our natural choice.

+ This work was supported in part by the Army Research Laboratory under grant DAAL01-98-2-D065.

3 System Design

Our design philosophy for the development of the handwriting portion centers around the following principles: interactivity, functionality, reliability, user-friendliness, and portability. A handwriting communication system must possess two important capabilities. The first is the ability to support the writing by a light pen, or a mouse if such a pen is unavailable, and the display of the handwritten data. The second capability should allow users to transmit and receive handwritten data from the network. To achieve these goals, a friendly graphical user interface, which requires the use of abstract windowing toolkit and event handling, is a necessity. In order to make the system a useful communication tool on the Internet, it must support both textual and graphical information exchanges. The system must also provide functions for users to overwrite or modify handwritten message received over the network. In addition, the programming language used for implementation must be platform independent so that the system can be easily ported to other machines with different operating systems.

4 Description of the System and its Applications to Distance Education

As mentioned earlier, we use Java [2, 4] as the programming language and TCP as the transport protocol for transferring handwritten data in our current client-server communication system. A graphical user interface consisting of buttons, radio check boxes, and a handwriting area, as well as the operations associated with the interface have been developed using the abstract windowing toolkit. All main tasks of the system are invoked from within the event handling functions. Our system currently allows users on the Internet to exercise handwriting from within a web page that contains the client code and send the information to the server that accepts the handwritten data. It can also be used to enhance online presentations over the Internet. This is due to the fact that the system allows users to perform handwriting directly on the specified writing area in a web page. By putting the presentation material inside the handwriting area, it is possible to add notes, make corrections, highlight important subjects on the spot during the course of the presentation.

The handwriting communication system has many applications in distance education and on-line collaborative learning. It can be used by an instructor to deliver on-line lectures via the web; the instructor may use one part of the screen to present prepared presentations and another part to highlight the important points of his/her presentation using a light pen. It can be used by fellow students in different locations to solve problems collaboratively and work on team projects. In addition, the instructor and students can use it to conduct on-line class discussions and answer student's questions by employing the communication capability.

5 Conclusions

In this article, we have presented the basic approaches, design considerations, and implementation of a real-time handwriting communication system on the Internet as well as its applications to on-line education. Our design philosophy centers around functionality, interactivity, portability, and user friendliness.

References

1. Andrew S. Tanenbaum, "Computer Networks", NJ: Prentice Hall, 3rd Edition, (1996).
2. David Flaganan, "Java in a Nutshell, Sebastopol", CA: O'Reilly & Associates, (1996).
3. David Gunter, et al., "Client/Server Programming with RPC and DCE", IN: Que, (1995).
4. Patrick Naughton and Herbert Schildt, "JAVA: The Complete Reference", CA: McGraw-Hill, (1997).

Strange Creatures in Virtual Inhabited 3D Worlds

Jens F. Jensen
InterMedia Aalborg, Aalborg University
Niels Jerners Vej, NOVI4-215, DK-9220 Aalborg East, DENMARK
Phone: +4598156279 Fax: +4598142444
E-mail: jensf@hum.auc.dk

This paper discusses the strange creatures that currently populate 3D cyberspace and3D Internet. First, the concept of Virtual Inhabited 3D worlds are discussed and defined. Next, some of the key elements or basic entities that can be found within the horizon of Virtual, Inhabited 3D Worlds are identified and denned. Among these basic elements are objects and agents, differentiated by whether or not their primary function is to carry out an action. Agents (denned as entities, which primary function is to carry out actions) have two main forms, which have been described as relatively sharply differentiable polar opposites. This is done based on questions such as: who is controlling the agents? 'who is doing the driving?' On the one hand there are agents that react independently of the user, but which are controlled by software or AL, the so-called 'autonomous agents' or 'bots'. On the other hand, there are agents, which directly represent and are controlled by users, the so-called 'avatars. Although there is then, in principle, a differentiation, in terms of definition, between bots and avatars, the paper argues that both concepts cover a relatively wide spectrum of very different types of phenomena with differing degrees of control. There also seems to be a tendency toward the appearance of more and more hybrids- in the present context termed 'cyber-hybrids' - combining avatars and bots. Furthermore, these hybrid forms are in many ways the most interesting and most promising in the virtual worlds at the moment. Rather than considering avatars and bots as polar opposites, it may therefore be more productive to consider them as the outer points along a continuum, between which can be found all sorts of combinations or hybrids. Following this line of argument, the paper outlines a new typology of hybrid creatures, which currently populate the continuum between (objects) bots and avatars in Virtual worlds.

*The paper was not available by the date of printing.