An Agent-Based Approach to Dynamic Adaptive Learning

Shanghua Sun, Mike Joy, and Nathan Griffiths
Department of Computer Science
University of Warwick
Coventry CV4 7AL
{S.Sun, M.S.Joy, N.E.Griffiths}@warwick.ac.uk

ABSTRACT

Dynamic adaptive learning and teaching strategies are increasingly demanded in order to improve the efficiency and effectiveness of the education process, but few learning systems exist which are dynamic and able to satisfy individual students. In an attempt to overcome these limitations, we propose a multi-agent architecture that provides the key functions of a dynamic and adaptive learning system. Learning style schemes are used to adapt to students' individual needs, and learning objects provide effective reuse and structuring of material. The incorporation of agents and learning objects is based on learning style — a pedagogic foundation for adaptivity — and this is one of the main contributions of this research. The system has been analysed through several functional experiments, and the analysis of a simulation study indicates that the approach is able to handle individual students' requirements and improve the dynamic adaptivity in education systems.

1. INTRODUCTION

Advanced information technologies are increasingly used in higher education to facilitate learning and teaching, but inadequacies exist in current systems, materials, and pedagogy. The application of similar learning strategies to all students in a class can be ineffective. For example, a failure to include design skills in introductory modules may become apparent later in a student's course. Students often tend to treat a course as a series of mechanical exercises rather than systemic concepts [32], and a specific framework to support the change process is often lacking [26]. Currently, much of the courseware and software resources used in higher education are unstructured and are isolated from each other.

People learn in different ways. It is important to be aware of the differences between learners, and this is especially relevant during the current expansion of tertiary education to a greater proportion of the population. New delivery mechanisms are required, including online, open and distance learning [4]. These issues can be partially resolved by providing student-centred, self-paced, highly interactive teaching materials and introducing automatic and asynchronous teaching methods. Although there are many educational technology projects, both stand-alone learning systems and Web-based tools using techniques such as multimedia interaction, learning models and asynchronous learning, there is as yet no integrated approach to the design of pedagogic information architectures [32].

Such intelligent learning systems must be adaptive, able to learn and dynamic [29]. Agent technology can provide a dynamic adaptation not only of domain knowledge, but also of the behaviour of individual learners, and so can be used to address the challenges of modern day education [2]. In this paper, we propose an agent-based architecture that is student-centred, adaptive, able to learn and dynamic. Our solution takes a multi-disciplinary approach, combining learning theory with agent-based systems. Thus, at the conceptual level, adaptivity is achieved by the use of learning style schemes to tailor the presentation of learning objects to individual students. Conversely, at the practical level, this adaptivity is achieved through a set of agents that use a combination of pre-built and acquired knowledge to determine the learning styles and learning objects that are appropriate for individual students.

In the remainder of this paper, we begin by introducing the foundational concepts on which our proposed architecture is based. Section 3 describes our proposed approach in detail. A prototype system is discussed in Section 4, and finally we offer our conclusions in Section 5.

2. RELATED TECHNOLOGIES

Our work is based upon three key concepts: agent technology, learning objects, and learning style theories. In this section we introduce these concepts and the work related to these technologies.

2.1 Agent Technology

Depending on the roles that agents take in their deployed environments, their abilities may vary significantly. However, we can still identify essential and commonly agreed properties of agents, namely: autonomy, proactiveness, responsivity, and adaptivity. Additionally, agents should also know users' preferences and tailor their interactions to reflect these [20]. It is generally accepted that an agent is an entity that is capable of carrying out flexible autonomous activities in an intelligent manner to accomplish tasks that meet its design objectives, without direct and constant intervention and guidance of humans.

Multi-agent systems contain many agents that communicate with each other. Each agent has control over certain parts of the environment, so they are designed and implemented as a collection of individual interacting agents. Luck et al. remark that, "Multi-agent systems provide a natural basis for training decision makers in complex decision making domains [in education and training]" [23]. Furthermore, multi-agent systems can substantially contain the "spread of uncertainty", since agents typically process information locally [14]. In the context of our education system architecture, agents provide a means to manage the complexity and uncertainty of the domain.

2.2 Learning Objects

Many learning materials are distributed using Web technologies, and most materials are currently developed for a specific purpose. For example, courseware is usually for a specific module, and its contents will probably not be reused or will only be reused infrequently. To address the issue of reuse, from both the perspective of educators and learners, the concept of a *learning object* has been proposed.

A learning object is a "self-standing, reusable, discrete piece of content that meets an instructional objective" [1]. Learning objects may be tagged with metadata so that their identity and content are available to other systems. The decomposition of educational content into learning objects is analogous to an object-oriented decomposition of a program, and permits an individual learning object to be used in a variety of educational contexts. Learning objects can form individual learning paths for students, to achieve a student-centred adaptive learning environment [30].

2.3 Learning Style Theories

Individuals learn in different ways. The concept of *learning style* has been introduced by educationalists, and is the subject of increasing research interest. The term is used as a "description of the attitudes and behaviours that determine our preferred way of learning" [17]. Learning styles depend on a variety of factors, and are individual to different people. Even for an individual, their learning style can change over time. Learning styles may also differ according to gender, age, and cultural background [6]. In this paper, we restrict our view of learning styles to those applicable for students in higher education.

There are several existing models that are used to classify students' learning styles including Kolb's Learning Style Inventory [21], Gardner's Multiple Intelligences [12], and the Felder-Silverman Learning Style Model [11]. Our proposed architecture does not dictate the model of learning styles used, however, in the system described in Section 4 the Felder-Silverman Learning Style Model is adopted. This model situates a student's learning style preference within a four-dimensional space, with the following four independent descriptors:

- "sensing (concrete thinker, practical, oriented toward facts and procedures) or intuitive (abstract thinker, innovative, oriented toward theories and underlying meanings);
- visual (prefer visual representations of presented material, such as pictures, diagrams, flow charts) or verbal (prefer written and spoken explanations);
- active (learn by trying things out, enjoying working in groups) or reflective (learn by thinking things through, prefer working alone or with a single familiar partner);
- sequential (linear thinking process, learn in small incremental steps) or global (holistic thinking process, learn in large leaps) [11]".

The reasons for choosing this model are that it has been validated by pedagogy research [11, 41], and that the number of dimensions of the model is constrained, improving the feasibility of its implementation.

2.4 Adaptive e-learning systems

Adpativity can be achieved in different ways from various perspectives, in [8], Brusilovsky and Peylo point out that Web-inspired technologies used in adaptive learning applications can be divided into five groups: adaptive hypermedia, for example KBS-Hyperbook [16]; adaptive information filtering, for example MLTutor [33]; intelligent class monitoring e.g. HyperClassroom [27]; intelligent collaborative learning such as, COLER [10]; and intelligent tutoring such as, ELM-ART [39].

Among these e-learning systems, several intelligent technologies have been used to achieve adaptivity and intelligence, rather than non-web based learning systems, which usually focus on single intelligent technology. The adaptivity has been enhanced from diverse perspective, however, these pioneer systems usually improve adaptivity on a specific type or aspect of web-based learning, i.e. adaptive presentation, assisting students to navigate themselves in learning materials, emphasizing on relatively small amount of learning contents, helping collaborative learning, or introducing a learning companion instead of a tutor, etc.

At the same time as the new and higher requirements for web-based e-learning systems, new intelligent technologies are increasingly incorporated with other technologies [8]. Learning objects increase personalization, interoperability, and flexibility [22]. People have their own preferences of how they can learn effectively, and to support a personalized learning strategy the differences between learners must be recognized [19]. Customizing learning materials as learning objects can support students with different learning styles. Although this idea has been proposed elsewhere [34], the incorporation of learning objects and learning style theories to support adaptivity is still a research problem. Agent technology gives a dynamic support for distributed learning applications, and deals well with crucial issues [38]. T In this paper, to enhance the most adaptivity from these three key technologies, we present an adaptive e-learning system, which incorporated these advanced e-learning technologies to facilitate adaptivity.

Some systems have adopted learning style theories, and explored the delivery of learning materials adapted to students' learning styles. The system developed by Carver et al. presents a list of links to each student based on their learning style, leaving the individual student to select the material to use [9]. Paredes and Rodriguez use two dimensions of the Felder-Silverman Learning Style theory [28], and progress has been made on the mechanism elsewhere [36, 15, 18]. They have incorporated learning style theory into their system and learning material design; however, the pedagogies and technologies are not suited to dynamic adjustment to students' learning styles. The knowledge is still delivered in a static way and the learning materials are more or less preset for a certain type of learning style or preference, and will not be changed or adjusted according to a change of learning style of the user over time. The pedagogy that incorporates learning objects and learning style, which we have used in the system, is able to dynamically organise and deliver learning materials to satisfy individual learning requirements, and agent technology gives dynamic support.

2.5 Pedagogical Agent Systems

In the context of adaptive education, agent technology can provide a dynamic adaptation not only of domain knowledge but also of the behaviour of individual learners, and has already been used in a number of educational tools. However, most systems incorporating agent technology, such as [3, 7, 25, 29, 31], have decoupled the agents from the pedagogic foundations of the system. Existing systems tend to emphasise a particular aspect, such as training, group work, or human resources requirements. Beer and Whatley report the initial design of agents to support students undertaking group projects in health care education [3]. For each group of students, they provide a local agent to monitor the project, and enhance the communication between members of the group. The use of agents is emphasised as providing dynamic support for synchronous collaboration.

Each of the current approaches has its individual ways of organising the learning materials, and few have considered the effect of different learning styles. For example, in Shang *et al.*'s system, the students' learning styles are stored in personal agents at the beginning of a student's use of the system, and are not changed dynamically during the learning process [31]. However, learning styles will change during the time students are using the system. In our proposed multi-agent system, students' learning styles are updated during the learning process. Shang *et al.* organise agents according to different courses [31], while Boicu *et al.* use agents that are implemented according to specific learning topics [7]. In our system, however, the agents are decomposed by their function in the teaching and learning process. The use of learning objects in such systems is rare, although the technology has begun to be used in non-adaptive training software. Garro and Palopoli's system is designed to assist finding appropriate employees and measuring the skill gaps between the employee and the requirements of the organisation from a human resources perspective [13].

The functionality of current intelligent learning systems used for adaptive education can be classified as follows:

Student aspect which includes communication and information storage,

Teaching and learning aspect which includes the modelling of students and their learning requirements, and methods of organising learning materials, and

System aspect which includes the system communication and quality control of the output.

We have developed a novel multi-agent approach to the problem of dynamically supporting adaptive learning, that distinguishes between these three aspects [37].

3. THE MULTI-AGENT APPROACH

From a technical viewpoint, the adaptivity requirement suggests that the set of interactions and communications within the system should be dynamic. The use of intelligent agents allows us to abstract data at a higher level than that which would be appropriate for conventional software technologies, and enables us to conceptualise the system in a natural fashion.

Our proposed multi-agent based learning system is functionally constructed by five agents, as shown in Figure 1: the Student Agent, Record Agent, Modelling Agent, Learning Object Agent, and Evaluation Agent. Each agent is designed to satisfy a certain functional requirement that contributes to the purpose of the overall learning system, namely to provide dynamic and adaptive learning materials to individual users. Agents allow the system to be functionally divided, since each agent in the multi-agent system is autonomous and has its own social ability. Agent autonomy (the ability to take charge of its own actions and internal states) also increases system maintainability. The architecture of each agent can be looked as a plug-in mechanism, according to the specific environment, the architecture can be updated to the most appropriate type. The reactivity and pro-activeness characters give the multi-agent system maximum flexibility and compatibility for different situations.

The Student Agent is responsible for communicating with students; the Record Agent maintains information about each student; the Modelling Agent creates models of students' skills and learning objectives; the Learning Object Agent manages the set of learning objects; and the Evaluation Agent ensures that learning objects are presented in individual and adaptive learning paths to each individual student. In the remainder of this section, we describe each of these agents in more detail, and discuss how students interact with the system.

3.1 Student Agent

The Student Agent is responsible for communicating with students, and provides the interface between the system and human users. The function of the agent is to fulfil the communication and data collection requirements, and to provide information from the user to other agents in the system.

When a student first logs in to the system, the Student Agent provides an initial questionnaire to ascertain the student's knowledge level, and to obtain information about their learning requirements (such as module details or the specific subject that the student wishes to study). During the time that the student is logged in to the system, the Student Agent records all of their actions, including the time they spend engaged in each activity presented to them, clicking times, whether they are active or not, etc.

The Student Agent has a clear functional division, and its functional operation can be naturally represented by the BDI notions of beliefs, desires, and intentions. Its knowledge includes students' preferences, the available learning materials in the system, and students' knowledge levels — provided by the students themselves and expanded by the system. This knowledge can be mapped to a set of beliefs, and the agent's interactions with students and the other system components can be mapped to its desires. According to its beliefs certain desires will be triggered, and plans for these are adopted as intentions in the BDI-based implementation. For example, according to the knowledge level provided by a student, appropriate learning materials can be sent. Beliefs are partially based on the information provided by students, and so they may not be completely accurate (e.g. students might exaggerate their abilities). Thus, not all of the desires can be necessarily be achieved, since the choice of intention may be based on inaccurate information.

The Student Agent is implemented as a BDI-based agent [14], which makes decisions according to its knowledge, and is able to reason about its actions. It comprises the three main components shown in Figure 2: a communication interface to students; a repository of beliefs, desires, intentions, and a plan library, together with communication with the other agents within the system. The Student

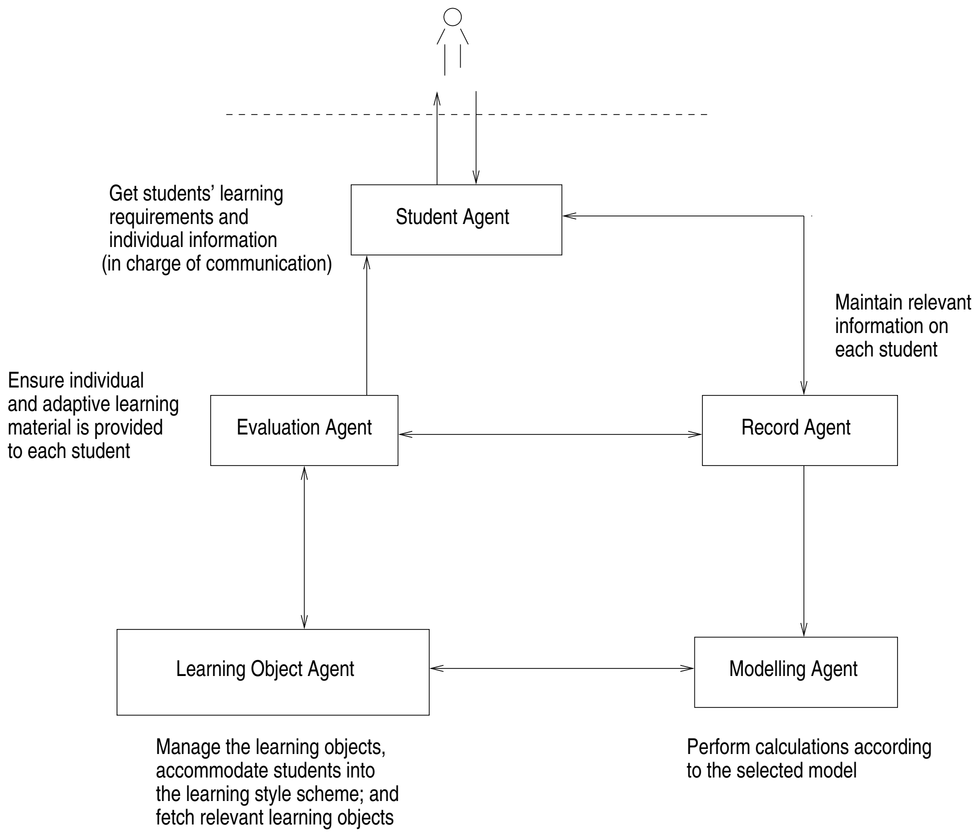

Figure 1: Overall System Architecture.

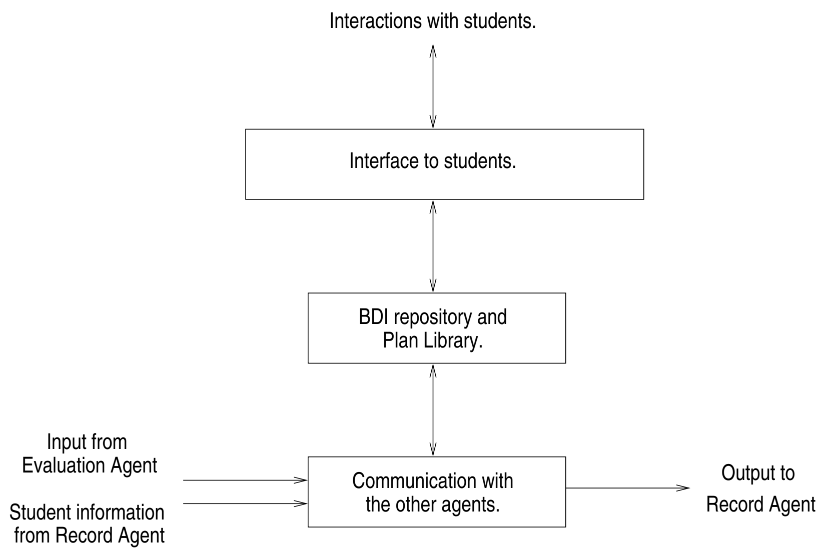

Figure 2: Student Agent.

Agent uses means-ends reasoning to determine an appropriate plan for how to achieve a particular goal [40]. Given a goal, the current state of the environment (its beliefs), and the actions available to the agent, it will update its beliefs, determine an appropriate goal to achieve, use means-ends reasoning to find a plan to achieve the goal from its plan library, and finally will execute the plan. For example, the answers to the questionnaire will first be analysed, and then be immediately sent to the Record Agent. In general, requirements and intentions from students are first mapped to beliefs, and then a course of action is determined and executed. The inputs and outputs for the Student Agent are as follows.

Inputs: Dialogue with students (including answers to questions, comments on learning materials, etc.) and series of learning objects from the Evaluation Agent. Student data from the Record Agent.

Outputs: Questionnaire and learning objects to the student, and student feedback to the Record Agent.

3.2 Record Agent

The Record Agent maintains all of the information about each student. It is the main data storage centre of the system; most of the data contained in the system is stored within this agent, in particular it encompasses:

- answers to questionnaires,
- feedback from students during the learning process,
- the series of learning objects sent to students, and
- the times of logging in, clicking times and the length of time in each stage, communication times, whether the student is active, etc.

Similarly to the Student Agent, the Record Agent must also reason toward a course of action, and so is also a BDI-based agent (as shown Figure 3). Both the Student Agent and the Record Agent monitor their environment and, according to their beliefs (which include information received from other agents), perform actions in order to change the environment. The Record Agent is more than simply a database — it is able to process and draw inferences from the data provided by other agents, and can intelligently provide other agents with information in response to its reasoning, even without information being requested. For example, after the Record Agent categorises the data from student, and realises that the student has spent almost one hour on each single learning unit, which is supposed to finish in 20 minutes, then this record will be sent to the Evaluation Agent without being requested. Data received from the Student Agent or the Evaluation Agent is incorporated into the Record Agent's beliefs. Changes in these beliefs give rise to desires and subsequently to adopted intentions, that cause the communication of suitable information to the Modelling Agent, the Student Agent, or the Evaluation Agent. The function of the Record Agent includes processing data from the other agents and making inferences based on this to update its own beliefs. For example, regardless of the knowledge level the student has provided themselves, it will develop its own beliefs about a student's knowledge by making inferences from the data provided by the other agents. The inputs and outputs for the Record Agent are as follows.

Inputs: Information from the Student Agent, feedback from student via the Student Agent, and the series of learning objects from the Evaluation Agent. After the first time a student uses the system, when they subsequently log in again the Student Agent will request the student's data from the Record Agent. Prior to performing any evaluation, the Evaluation Agent requests the Record Agent to provide the relevant student data.

Outputs: The filtered student data is sent to the Modelling Agent, and on request to the Evaluation Agent.

The Student Agent and the Record Agent each make decisions according to their individual knowledge, and their reasoning is directed toward actions, and so a BDI-based approach [14] is natural. A deductive reasoning agent, was also considered, however it is doubtful whether such logic-based agents can react effectively within our time-constrained environment.

There are several reasons why the Student Agent and Record Agent are separated into two agents. Firstly, the Student Agent is responsible for communication, while data repositories are the focus of the Record Agent. If they were combined together into a single agent, there would be two major functional parts, of reasonable independence of each other. Secondly, these BDI-based agents use mean-ends reasoning to select their actions, and if they were combined into a single agent, their plan libraries would also be joined, leading to a significant increase in the complexity of plan selection and reduced maintainability of the plan libraries. Since the data repositories of the system as a whole are fairly large, it is worthwhile assigning a single agent to its management.

3.3 Modelling Agent

The Modelling Agent is responsible for performing calculations according to the general *pedagogical* modelling approach, and providing the student models required by the Learning Object Agent. In addition to student specific information, the Modelling Agent utilises knowledge about how to perform modelling tasks, this modelling knowledge is stored in its own knowledge base. The Modelling Agent models individual students' needs and their knowledge background, based on its selection of suitable data for the model from the information provided by the Record Agent. Once constructed, the resulting models are stored by the Record Agent (although the information contained in them can be subsequently obtained and updated in the future by the Modelling Agent).

In order to provide a clear distinction between the nature of the knowledge and processing that the Modelling Agent undertakes, it is based upon a hybrid architecture (as shown in Figure 4). Its functions are divided into two levels (avoiding an unmanageable plan library that would result from a BDI-based approach). Its knowledge is separated into different parts, and the reasoning is carried out individually in the two layers. When modelling work is not required, then the communication layer is responsible for the agent's behaviour. The Record Agent regularly sends student data (as required for input to the pedagogical model) to the communication layer of the Modelling Agent, which in turn maps relevant data to the general knowledge component. Based on this information the Modelling Agent can determined suitable goals (within the communication layer), that correspond to the information that should be sent to the modelling layer. When the modelling layer receives information, it is mapped to its modelling knowledge and the goals in the modelling layer, using Bayesian networks to perform its calculations. The results of modelling, such as the index

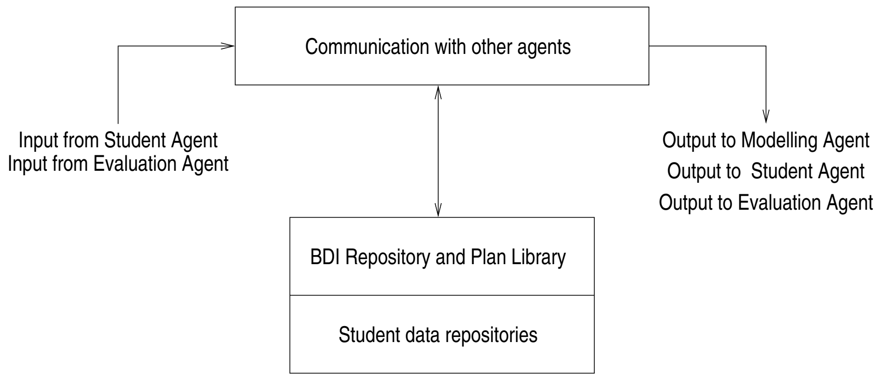

Figure 3: Record Agent.

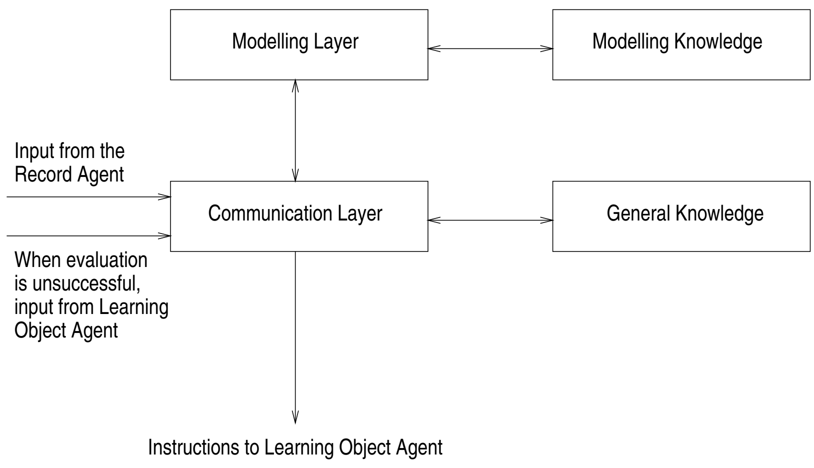

Figure 4: Modelling Agent.

of each student's relative knowledge level, and suggestions for the level of learning materials to be sent to the student, are passed via the communication layer to the Learning Object Agent. The information and functionality required by the modelling layer is separate to that required by the communication layer, and the separation offered by a layered approach provides a more manageable solution than a single-layered approach. The Modelling Agent's inputs and outputs are as follows.

Inputs: Student information for the modelling process (provided by the Record Agent), and where the series of learning objects are not suitable for the student, the re-model instruction and student information is provided by the Learning Object Agent.

Outputs: Results of modelling, e.g. index of student's relative knowledge level, and the difficulty level of learning materials to the Learning Object Agent.

3.4 Learning Object Agent

The Learning Object Agent manages the learning objects, which are organised according to the learning style scheme. In response to instructions from the Modelling Agent, the Learning Object Agent provides different learning style students with relevant learning objects. The function of the Learning Object Agent is to organise the adaptive learning materials for users based on the information that the system has collected.

The Learning Object Agent is a hybrid agent, and has an architecture in which its subsystems are arranged into a hierarchy of layers as shown in Figure 5. The results from the Modelling Agent are transferred to the learning path layer by the communication layer, which in turn maps them to an appropriate learning path for a student. The Learning Object Agent communicates with the other agents through its communication layer. Decisions are sent to the learning object's management layer, which is in charge of managing all of the learning objects in its repository. The learning objects repository is organised into different levels, according to the learning style scheme. Finally, the learning objects management layer selects a series of learning objects, which are transmitted to the Evaluation Agent through the communication layer.

The layered architecture ensures that the Learning Object Agent can be reactive when it is necessarily to respond to changes in the environment, and deliberative to achieve its goals, whilst maintaining a separation between the knowledge and functionality embodied by each layer. The learning path layer is responsible for accommodating the student within a learning style and organising learning materials according to the student's learning style description. The learning objects management layer has specific knowledge of the learning objects. Thus, while these two layers and the communication layer are distinct from each other, they are able to exchange their information. All three of these layers cooperate together, to achieve the functionality of the Learning Object Agent.

The learning path layer adopts the Felder-Silverman Learning Style Model [11] to organise learning objects to fulfil different students' requirements. The learning objects in the repository are categorised by the learning style model. Organisation of the learning materials as learning objects, based on a pedagogic learning style scheme in an agent environment, is a distinct characteristic of this architecture which distinguishes it from existing pedagogic agent-based systems. The Felder-Silverman Learning Style Model categorises learning schemes according to four distinct dimensions, each of which has four layers. The inputs and outputs for the Learning Object Agent are as follows.

Inputs: Instructions from the Modelling Agent and evaluation results from the Evaluation Agent. If the evaluation is successful, the Learning Object Agent will simply be informed of this, otherwise it will receive the unsuccessful results, instructions to rearrange the learning objects, and information about the student.

Outputs: A series of learning objects suitable for the student are sent to the Evaluation Agent, or when the Learning Object Agent is informed that the evaluation is unsuccessful, it will give the data for this student from the Evaluation Agent to the Modelling Agent, and request it to model again.

3.5 Evaluation Agent

The Evaluation Agent ensures that learning objects are presented in an individual and adaptive learning path to each student. This is achieved by using the student data contained in the system to evaluate the learning objects which are sent to students. The Modelling Agent uses the information from the Record Agent according to the modelling knowledge and functionality contained in its modelling layer. Thus, it is constrained by modelling specific knowledge. The Evaluation Agent helps to ensure that the best use is being made of the student data available in the system. If the selected learning objects are evaluated as appropriate for the student, the series of learning objects are sent to the Student Agent directly, otherwise the Evaluation Agent requests the Learning Object Agent to resend learning materials, or if more information is required, then relevant actions will be given to the student first, then the result will be evaluated again. For example, if the student spends a very short period of time in a specific learning unit, compared to the expected time, then the Evaluation Agent will initiate a dialogue about whether the learning materials are too easy for the student, then depending on the answer, further action and decisions will be taken. If learning objects need to be reorganised then the Learning Object Agent will ask the Modelling Agent to model again by using additional data and suggestions from the Evaluation Agent. At the same time, the knowledge of these agents and the student's profile will be updated in order to dynamically adapt to student's change in learning behaviour.

The Evaluation Agent is one of the most important agents in this system. It needs to be able to use all of the data available to decide which learning objects will be sent to each student. It should be capable of reactive and proactive behaviours, and can be considered to be a hybrid agent. It has a vertical layered architecture similar to InteRRaP [24], in which each layer interacts with each of the others, and the main types of interactions are bottom-up activation and top-down execution. This layered approach allows the division into behaviour-based layers, and at the same time reduces the number of possible interactions between the layers. The Evaluation Agent comprises an information interface, along with communication and evaluation layers, as shown in Figure 6. Each layer has its own knowledge base of information. The lower (communication) layer deals with interactions with other agents, and maps its current goal and knowledge base to new goals; the higher (evaluation) layer deals with analysing the information and making final evaluations for each student. The layers are naturally decomposed by the agent's functionality. The Evaluation Agent's inputs and outputs are as follows.

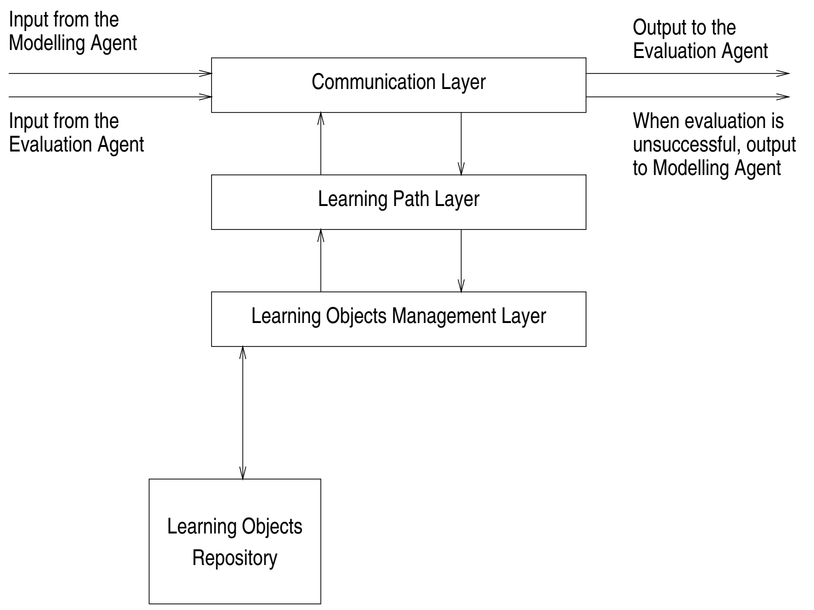

Figure 5: Learning Object Agent.

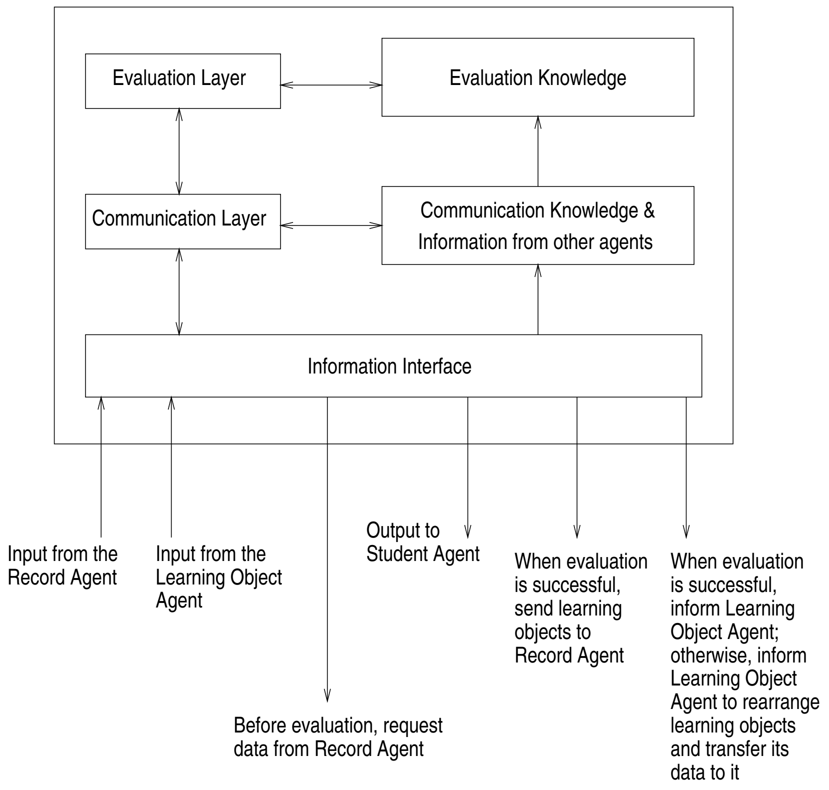

Figure 6: Evaluation Agent.

Inputs: Student information from the Record Agent, and a series of learning objects from the Learning Object Agent.

Outputs: Results of evaluating, given to the Learning Object Agent and the Student Agent. When evaluation is successful, the series of learning objects is sent to the Student Agent and the Record Agent. The Learning Object Agent also will be informed. Otherwise if the evaluation is unsuccessful, the Evaluation Agent will send its data about the student and instructions for rearranging learning objects to the Learning Object Agent.

3.6 Using the system

When a student first logs in to the system, the Student Agent enters into a dialogue with the student to ascertain the student's learning requirements. After initially analysing the results, the Record Agent is informed of the student's learning requirements together with a suggested knowledge level for the student. These items of information are recorded and then passed to the Modelling Agent, which then sends results and instructions to the Learning Object Agent. This in turn arranges the first batch of learning objects to be sent to the Student Agent according to the results of learning style analysis (which occurs in the learning path layer) and difficulty level of the learning objects, which are also organised according to the learning style scheme. These learning objects are first sent to the Evaluation Agent, which checks the student's data from the Record Agent to evaluate whether the learning objects are suitable for this student. If the evaluation is successful, the series of learning objects is sent to the Student Agent (and then to the student) and recorded by the Record Agent. Otherwise, the Evaluation Agent asks the Learning Object Agent to provide alternative learning objects. After the student has used the learning objects, response data is returned to the Student Agent, which transmits them to the Record Agent.

4. EVALUATION

Before implementation, several case studies have been used to verify the consistency of the proposed architecture, including consideration of several first year undergraduate programming topics such as introductory Java programming and UNIX shell programming.

A prototype of the system has been developed using JADE [5]. Prior to the implementation of the complete multi-agent system, the Learning Object Agent has been evaluated. As mentioned above, in order to deliver the learning objects according to different learning styles, its evaluation is divided into three parts: accommodating students into the learning style scheme, categorising learning objects according to the learning style scheme, and delivering Learning Objects. Taking an abstract view, this functionality can be structured as shown in Figure 7.

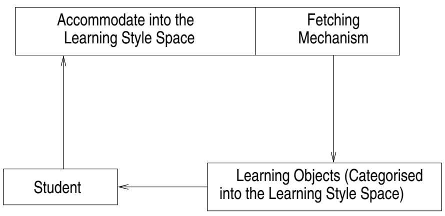

Figure 7: Abstract Method of the Learning Object Agent.

We have chosen an algorithm, which has been evaluated by comparing the results for a sample of students with those generated by Felder and Silverman's original questionnaire [35], to accommodate students into the learning style scheme. A Spearman's rank correlation coefficient statistical analysis has been performed on the normalised data, and indicates a strong correlation between the two data sets. This suggests that our algorithm is sufficient to categorise a student's learning style.

The learning objects stored in the repository of the Learning Object Agent are also organised into the learning style space, the granularity of the categorisation is pragmatically determined, and seems appropriate for the learning objects available to us, but may be refined in the light of future experience.

The multi-agent adaptive learning system stores each student's current learning style (which may change over time) and the style attributes of each learning object as co-ordinates into the learning style space. The algorithm used to deliver learning objects to students involves matching the style attributes of (appropriate) learning objects to the current style preferences of the individual student. In the Learning Object Agent, this is supported by the learning path layer to realise the algorithm and implement the process. The learning object management layer will then search the repository of learning objects, to fetch appropriate learning objects with similar (but not necessarily identical) descriptions. The selected objects are then presented to the student, and the subsequent interactions between the student and these learning objects may be used to modify the student's learning style attributes. It should be noted that both the categorisation of a learning object and the assignment of a learning style to a student are necessarily approximate.

Since it is almost impossible to find students with all possible combinations of the learning style scheme, a simulation has been run on the system. The simulation has covered all of the possibilities — four dimensions, each on a five-point scale ($5^4 = 625$), and the evaluation indicates that our approach is capable of delivering different learning objects to different students according to the learning style category.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a multi-agent based adaptive learning system which incorporates reusable learning objects and uses a learning style scheme as the pedagogic foundation for adaptivity to dynamically adapt to individuals in education. A prototype of the system has been developed, and the Learning Object Agent has been evaluated. In addition to the implementation of the complete system, future work also includes optimising the architecture, and an evaluation of the system effectiveness and efficiency.

6. REFERENCES

[1] Academic ADL Co-Lab, University of Wisconsin System, and Wisconsin Technical College System. What are learning objects?, 2002. url: http://adlcolab.uwsa.edu/lo/what.htm (accessed Dec. 2003).

[2] L. Aroyo and P. Kommers. Special issue preface, Intelligent agents for educational computer-aided systems. *Interactive Learning Research*, 10(3/4):235–242, 1999.

[3] M. Beer and J. Whatley. A multi-agent architecture to support synchronous collaborative learning in an international environment. In *Proceedings of the 1st International Conference on Autonomous Agents*, pages 505–506, 2002.

[4] H. Beetham. Understanding e-learning. e-tutoring for effective e-learning, 2002. url:http://www.ics.ltsn.ac.uk/pub/elearning/ (accessed Nov. 2002).

[5] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa. Jade — a white paper, Sept. 2003. url: http://jade.tilab.com/.

[6] J. Blackmore. Learning styles: Telecommunications for remote work and learning, 1996. url: http://www.cyg.net/ jblackmo/diglib/styl-a.html (accessed May 2004).

[7] M. Boicu, G. Tecuci, B. Stanescu, D. Marcu, M. Barbulescu, and C. Boicu. Design principles for learning agents. In *Proceedings of AAAI-2004 Workshop on Intelligent Agent Architectures: Combining the Strengths of Software Engineering and Cognitive Systems*, 2004.

[8] P. Brusilovsky and C. Peylo. Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education, Special Issue on Adaptive and Intelligent Web-based Educational Systems*, 13(2–4):159–172, 2003.

[9] C. A. Carver, R. A. Howard, and W. D. Lane. Enhancing student learning through hypermedia courseware and incorporation of student learning styles. *IEEE Transactions on Education*, 42(1):33–38, 1999.

[10] M. A. Constantino Gonzalez, D. Suthers, and J. G. Escamilla De Los Santos. Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education*, 13(2–4):261–294, 2003.

[11] R. M. Felder and J. E. Spurlin. Applications, reliability, and validity of the index of learning styles. *International Journal of Engineering Education*, 21(1):103–112, 2005.

[12] H. Gardner. *Multiple Intelligences: The Theory in Practice*. Basic Books, 1993.

[13] A. Garro and L. Palopoli. An XML multi-agent system for e-learning and skill management, 2002. url: http://www.netobjectdays.org/pdf/02/papers/malceb/-0623.pdf (accessed Feb. 2003).

[14] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge. The belief-desire-intention model of agency. In J. Müller, M. P. Singh, and A. S. Rao, editors, *Proceedings of the 5th International Workshop on IntelligentAgents V : Agent Theories, Architectures, and Languages*, pages 1–10, 1999.

[15] J. E. Gilbert and C. Y. Han. Arthur: A personalized instructional system. *Journal of Computing in Higher Education*, 14(1), 2002.

[16] N. Henze and W. Nejdl. Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education*, 12(4):325–350, 2001.

[17] P. Honey. *Honey and Mumford Learning Styles Questionnaire*. PeterHoney Learning, 2001. url: http://www.peterhoney.com/ (accessed Apr. 2004).

[18] H. Hong and Kinshuk. Adaptation to student learning styles in web based educational systems. In *ED-MEDIA 2004*, pages 491–496, 2004.

[19] T. Jenkins. On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, pages 53–58, 2002.

[20] N. R. Jennings and M. Wooldridge. Applications of intelligent agents. In *Agent Technology Foundations, Applications, and Markets*. Springer-Verlag, 1998.

[21] D. A. Kolb. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice-Hall, 1984.

[22] W. Longmire. A primer on learning objects, learning circuits, ASTD's source for e-learning. url: http://www.learningcircuits.org/mar2000/primer.html. (accessed Mar. 2003).

[23] M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing A Roadmap for Agent Based Computing*. AgentLink, 2003.

[24] J. P. Müller. A cooperation model for autonomous agents. In *Proceedings of the 3rd International Workshop on Intelligent Agents — Theories, Architectures, and Languages*, volume 1193, pages 245–260, 1997.

[25] T. J. Norman and N. R. Jennings. Constructing a virtual training laboratory using intelligent agents. *International Journal of Continuous Engineering Education and Life-Long Learning*, 12(1–4):201–213, 2002.

[26] M. B. Nunes and M. McPherson. Managing change in continuing professional distance education (CPDE) through action research. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, pages 20–25, 2002.

[27] T. Oda, H. Satoh, and S. Watanabe. Searching deadlocked Web learners by measuring similarity of learning activities. In *Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems*, 1998.

[28] P. Paredes and P. Rodriguez. Considering learning styles in adaptive web-based education. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, pages 481–485, 2002.

[29] M. A. Razek, C. Frasson, and M. Kaltenbach. Toward more cooperative intelligent distance learning environments. In *Software Agents Cooperation Human Activity*. 2002. url: http://www-perso.iro.umontreal.ca/ãbdelram (accessed Feb. 2003).

[30] P. Semmens. The potential for learning objects to support flexible learning in higher education. In *Technology newsletter, IEEE Computer Society Technical Committee on Learning Technology*, pages 23–27, 2004.

[31] Y. Shang, H. Shi, and S. Chen. An intelligent distributed environment for active learning. In *Proceedings of the 10th International Conference on World Wide Web*, pages 308–315, 2001.

[32] H. Shi, Y. Shang, and S. Chen. A multi-agent system for computer science education. In *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and technology in computer science education*, pages 1–4, 2000.

[33] A. S. G. Smith and A. Blandford. MLTutor: An application of machine learning algorithms for an adaptive web-based information system. *International Journal of Artificial Intelligence in Education*, 13(2–4):233–260, 2003.

[34] R. smith. Guidelines for authors of learning objects. url: http://www.nmc.org/guidelines/. (accessed Jul. 2004).

[35] B. A. Soloman and R. M. Felder. Index of learning styles. url: http://www.ncsu.edu/felder-public/ILSpage.html (accessed Apr. 2004).

[36] M. Specht and R. Oppermann. ACE adaptive courseware environment. *The New Review of Hypermedia and Multimedian*, 4:141–161, 1998.

[37] S. Sun, M. Joy, and N. Griffiths. The use of learning objects and learning styles in a multi-agent education system. In *Proceedings of ED-MEDIA 2005 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2005.

[38] P. van Rosmalen, F. Brouns, C. Tattersall, H. Vogten, J. van Bruggen, P. Sloep, and R. Koper. Towards an open framework for adaptive, agent-supported e-learning. url: http://dspace.learningnetworks.org/retrieve/94/. (accessed Jan. 2005).

[39] G. Weber and P. Brusilovsky. ELM-ART: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education*, 12(4):351–384, 2001.

[40] M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[41] M. S. Zywno. A contribution to validation of score meaning for felder-soloman's index of learning styles. In *Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition*, 2003. session 2351.