Communication Content Ontology for Learner Model Agent in Multi-agent Architecture

Weiqin Chen and Riichiro Mizoguchi

Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
Email: {wendy, miz}@ei.sanken.osaka-u.ac.jp

In this paper we describe domain-independent concepts in message content in a multi-agent architecture of Intelligent Educational Systems (IES). Although it is well accepted that a common ontology holds the key to fluent communication between agents, most researchers believe that the common ontology is domain ontology. This has been one of the causes of the fact that the work on agent communication has concentrated on general-purpose languages with communication performatives. In IES research, however, we can find task-dependent but domain-independent content, which we could "ontologize". By domain, we here mean subject domains such as geometry, chemistry, and mathematics, etc. In the multi-agent architecture of IES, we examined the information exchanged between the learner model agent and other agents, and abstracted the domain-independent concepts. We represent them explicitly and domain-independently, to build an ontology. By so doing, we believe that we can move toward a standard for the communication between agents in educational systems, so that the modules/agents can be reused. In this paper we discuss in detail the ontological issues in the message content, and give some examples to show the ontology, that we have designed.

Keywords: **Agent architecture, communication content ontology, learner model agent**

1 Introduction

In the last few years, communication between agents has been extensively studied, and fruitful results have been achieved. One of the most well known achievements is the Knowledge Query and Manipulation Language (KQML) developed by the ARPA-sponsored Knowledge Sharing Effort [5]. Although it is well accepted that a common ontology holds the key to fluent communication between agents [5][7], most researchers believe that the common ontology is domain-dependent. This has been one of the causes of the fact that the work on agent communication has concentrated on general-purpose language with communication performatives. The situation is similar in the field of Intelligent Educational Systems (IES). Multi-agent architecture and agent communication technology have been extensively discussed [1][15][16][17][18], but not much effort has been put to the message content, which is transferred from agent to agent. However, we can find task-dependent but domain-independent content, which we could "ontologize" in agent communication of IES. By domain, we here mean subject domains such as geometry, chemistry, and mathematics, etc. Identifying the domain-independent concepts makes a first step towards a standard for agent communication content.

The vocabularies used in the learner model and teaching activities are defined in educational ontology [12]. An ontology is an explicit specification of a conceptualization [8]. Educational ontology includes the entities, relations and functions needed to describe the procedure of modeling a learner, making instructional decisions, interacting with a learner and communicating between agents.

The main purpose of this paper is to discuss the domain-independent concepts in message content for the agent communication in IES, to answer "What concepts are necessary for the communication?", "How can these concepts be organized?" and "What kind of role does an ontology play in agent communication?". First, we will present a multi-agent architecture of IES as a reference model and discuss "What agent is needed in IES?", "What functions does each agent have?", "What information is needed from other agents?", and "What information can be provided by a certain agent?". These questions are fundamental considerations for communication between agents in IES. Later the communication content ontology, which we obtained thus far, will be discussed in detail.

2 Multi-agent architecture of IES

As a starting point for the identification of agents in IES, we reviewed the research in IES, ITS, ILE and social learning systems in the last two decades, and re-identified the following list of agents in general IES:

- learning material agent
- learning support agent
- learner modeling agent
- interface agent
- learner model agent

Among them, the learning support agent or pedagogical agent [10][11] and learner modeling agent [15] have been studied by many researchers in the intelligent education area. It is obvious that this architecture itself is not new. However, what we're doing here is to make one step towards a standard for the communication between agents or modules in educational systems, so that all the agents or modules that have been developed can be reused.

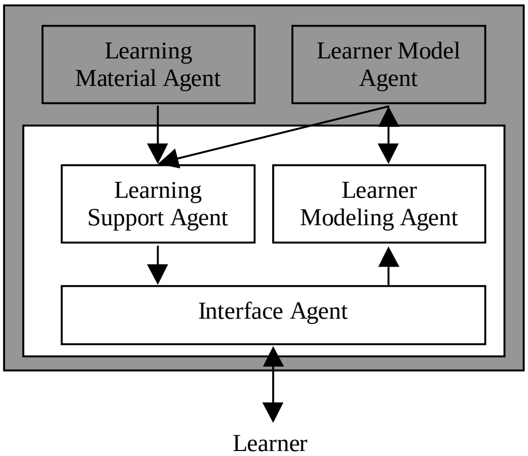

Fig.1 Multi-agent architecture of IES

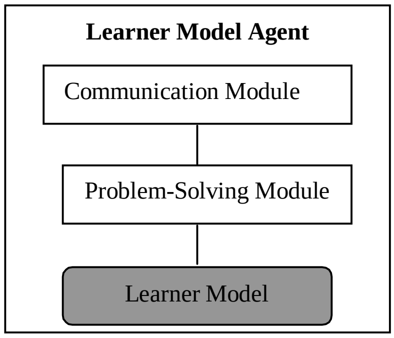

Fig. 2 Structure of Learner Model Agent

Each agent in Fig.1 consists of several major functional modules. An example of a learner model agent is shown in Fig. 2. The communication module is common to all the agents. It is responsible for the composition and interpretation of messages. The learning material agent takes charge of domain knowledge, its content and organization. It answers queries from other agents that need domain knowledge to accomplish their functions. For example, the learning support agent takes domain knowledge as its object to plan the instruction. It is responsible for planning instructions and creating feedback. It selects appropriate knowledge to teach, and makes decisions on teaching strategy application and learning support actions. The learner-modeling agent is responsible for the construction of a learner model. It derives the learner's information from the interaction between the learner and the system. When the learner-modeling agent needs extra information about the learner to construct a learner model, it asks the learning support agent to show relative problems to the learner, so that the interface agent can transfer the learner's actions/answers to the learner-modeling agent. The interface agent is responsible for the interaction between a learner and the system. It monitors the learner's activities, and passes the messages to other related agents, such as the learner modeling agent and the learning support agent. The learner model agent is responsible for answering queries from other agents about the learner's information.

3 Communication between agents

The communication between agents involves more than just simple messages. In this section, in order to get the feel of how the various agents interact with each other, we present a detailed description of communication between agents in IES, answering questions such as “how is the communication specified?” and “what information can a certain agent provide to other agents?”.

Concerning the problem of communication specification, we start by looking at KQML. KQML is both a message format and a message-handling protocol to support run-time knowledge sharing among agents [5]. It can be thought of as consisting of three layers: a protocol layer, a content layer, and an ontology layer. The domain-independent performatives in protocol layer, such as **tell**, **ask** and **reply**, describe the communication actions between agents. KIF (Knowledge Interchange Format) [6] defines the syntax and semantics for the expression of communication content. Although KQML and KIF define the performatives and content format, the concepts used in the content are not specified in ontology layer. In order to address the problem in multi-agent educational systems, we will start by examining the information each agent can provide to other agents.

Take the learner model agent as an example. The learner model agent is a very important part in intelligent educational systems. An educational system that keeps a learner model, and can adapt its behavior to the content of the model, is thought to be “intelligent”. The learner model agent in our multi-agent architecture consists of a communication module, a problem solving module and a learner model (Fig. 2).

The learner model contains both facts and abstract information (hypothesis about learner's understanding and mental state). The communication module controls the communication with other agents, including determining message performatives, sending out messages, and receiving and interpreting messages performatives. The problem-solving module is responsible for answering other agents' queries. With an inference engine, it can infer answers from the content of the learner model. The main functions of the problem solving module are: 1) analyzing the message content and determining the task; 2) solving the problem by reasoning using the learner model content; 3) constructing a message content for answering a query or informing other agents. A learner model agent can provide three kinds of information to other agents: 1) Information about the learner, including static information and dynamic information. 2) Information about the model, including the type of the model, the attributes of the model and the model representation. 3) Information about the learner assessment. Fig. 3 lists questions that the learner model agent should be capable of answering.

1. What is the name of a learner?
2. What is the ID of a learner
3. Is a learner a male or female?
4. How old is a learner?
5. What is the social status of a learner, upper, middle or lower (for language learning)?
6. What is a learner’s education status, no degree, bachelor, and master or doctor degree?
7. What is the motivation of a learner, high, middle or low motivation?
8. Does a learner have experience in a certain topic?
9. What's the learning style of a learner, principle-oriented or example-oriented, general-to-specific or specific-to-general?
10. What kind of media does a learner prefer?
11. What type of exercise does a learner prefer?
12. What knowledge a learner has mastered? Has a learner mastered a certain topic?
13. What knowledge a learner has not mastered? What knowledge a learner has missed?
14. Of what knowledge does a learner have incorrect understanding?
15. At which part of knowledge is a learner weak/strong?
16. What is the cause of a learner's error? or what is the bug?
17. What kind of bug does a learner have?
18. Where is the bug?
19. What is the plan of a learner for problem solving?
20. What answer did a learner give to a certain question?
21. What question did a leaner ask?

22. What examples were given to a learner?
23. What problems were given to a learner?
24. What learning objects were given to a learner?
25. What answer will a learner give to a certain problem / what will a learner do in his/her next action?
26. What score did a learner get in the test?
27. What is the average score of a learner?
28. How many examples were given to a learner?
29. How many problems were given to a learner?
30. How many times did a learner try before answering a question/solving a problem correctly?
31. What is the rate of success and failure?
32. How much time did a learner spend on a question?
33. Which phase in a learning process is a learner in, initial acquisition, assimilation or mastery (through its use) of long-range process, or problem identification/formulation, hypothesis formation, or verification of short-range process?
34. What media does a learner use for interaction?
35. How interested is a learner in a certain topic?
36. What is the learner's concentration like, high, middle or low concentration?
37. How is the overall performance of a learner?
38. How well does a learner master a certain topic?
39. How confident is the system in evaluating a learner?
40. What type of model does a learner model agent have?
41. Is the learner model runnable or executable?
42. How reliable is the learner model?
43. How is the learner model represented, by rule, frame network, or by numerical method?

Fig. 3 Queries supposed to be answered by learner model agent

Obviously, a learner model agent must sometimes protect the privacy of learners. It must therefore have a facility to decide what kind of queries should be answered and what information can be provided. For example, if an agent asks a learner model agent about a learner's age or gender, the learner model agent is supposed to make a decision whether or not it should answer.

4 Communication content ontology

The ontological issue is a fundamental one in agent communication. It defines the necessary concepts needed to exchange messages. Furthermore, because an ontology has different abstraction levels, the message content can have different grain sizes. It can be general, and it can also be specific. For example, in communication with learner model agent, another agent can ask about both the general knowledge status of a learner and the knowledge of a specific topic.

During the communication between agents in the multi-agent architecture of IES, both domain-dependent concepts and domain-independent concepts are required. Those that define the domain terms, relations and functions needed to model a certain domain are specified in a domain ontology. For example, in an operator training system in a substation of the power system [3], the domain ontology consists of equipment such as *main circuit equipment*, *connecting line*, *power cable*, etc and operations such as *confirm*, *record*, *reset*, etc. Such domain ontologies are outside of the scope of this paper, which focuses on the domain-independent concepts in communication. Those concepts that define the domain-independent terms, relations and functions needed to model a specific task are specified in task ontology. In the multi-agent architecture of IES, we have identified a series of domain-independent but task-dependent concepts that can be used in agent communication.

Considering the domain-independent concepts for the learner model agent [12][14]. Suppose other agents can ask all the questions listed in Fig.3. The message content is constructed by arranging the concepts defined in the ontology according to the syntax and semantics defined by KIF. Fig. 4 shows the concept hierarchy for the communication between learner model agent and other agents in IES.

With the domain-independent concepts, it is easy to represent a buggy model. While to represent an overlay model, which is still popular in intelligent educational systems, we

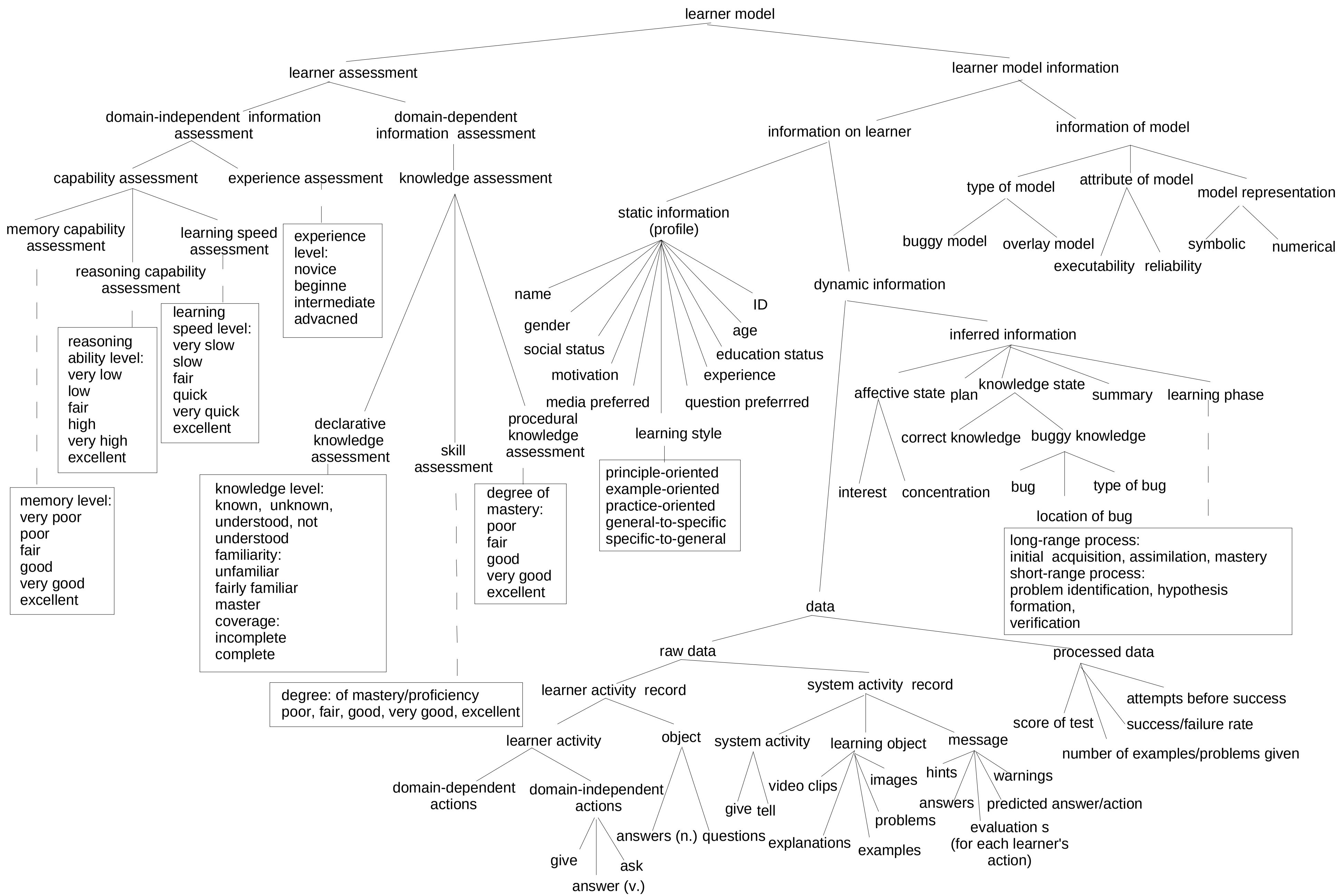

Fig. 4 Learner model ontology
Note: Some of theleaf nodes contain incomplete lists, for example, "affective state" can have more aspects than "concentration " and "interest".

need to combine the correct knowledge in the information of learner with the knowledge assessment, because the overlay model does not separate the learner model and the assessment of the learner for each learning item.

5 Communication example

Once an ontology has been explicitly incorporated into the communication, it can be used to construct the message content, one of the key points for communication. In this section, we will give some examples of communication messages in the multi-agent architecture of IES. With the examples we will show how the ontology is used in the message content construction.

The communication process starts when the learning support agent sends a message to the learner model agent, asking about the learning style of a learner. In this example, we assume that the type of the learner model is a simple overlay model.

Message1, learning support agent—>learner model agent,
"What is the **learning style** of learner <L1>?"
Message2, learner model agent—>learning support agent,
"The **learning style** of learner <L1> is **example-oriented**."
Message3, learning support agent—>interface agent
"Show a *multiple-choice question* <M1> of *topic* <T1> to learner <L1>."
Message4, interface agent—>learner modeling agent
"The **answer** of learner <L1> to the *multiple-choice question* <M1> is <A1>."
Message4, learner modeling agent—>learner model agent
"Learner <L1> fails to answer the *multiple-choice question* <M1> of *topic* <T1>"
Message5, learning support agent--->learner model agent
"What is the **mastery degree** of learner <L1> of *topic* <T1>?"

```
(ask        :sender learning-support-agent
            :receiver learner-model-agent
            :reply-with t-l-1
            :ontology learner-model-ontology+learning-material-ontology
            :content (val (mastery-degree learner1) (topic T1)))
```

Message6, learner model agent—>learning support agent
"The **mastery degree** of learner <L1> of *topic* <T1> is **low."**

```
(tell       :sender learner-model-agent
            :receiver learning-support-agent
            :in-reply-with t-l-1
            :ontology learner-model-ontology+learning-material-ontology
            :content (= (val (mastery-degree learner1) (topic T1)) low)
```

Message7, learning support agent—>learning material agent,
"What's the *example* of *topic* <T1>?"
Message8, learning material agent—>learning support agent,
"*Topic* <T1> has an *example* <E1>."
Message9, learning support agent—>interface agent
"Show *example* of *topic* <T1> to learner <L1>."

Note that the **bold words** are defined in the learner model ontology, the *italic words* are defined in learning material ontology, and the underlined words (learner in the examples) are defined in the ontology of participant of IES.

6 Related work

Concerning the communication between agents in IES, three related works are worthy to note. In ITS'96 Workshop on Architecture and Methods for Designing cost-effective and Reusable ITSs, Paiva [15] discussed the communication between user/learner modeling agents and application agents. She described the learner modeling tasks, which can be regarded as the functions of learner modeling agent. She provoked three main problems in communication between the user/learner modeling agent and the application agents: the protocol problem, the communicative act problem and the ontology problem. She proposed to address the first two problems with the KQML and KIF languages respectively. She claimed the third problem is more complex because it involves defining conceptualization and language which both the applications and the user/learner modeling system should be able to share. She, therefore, did not consider the third problem in her paper. However, the paper did make an important step toward a communication consensus between user/learner modeling system and application agents.

In CIKM'95 Workshop on Intelligent Information agents, Cheikes [1] presented an agent-based architecture for intelligent tutoring systems--GIA (a Generic Instructional Architecture). He hypothesized that ITSs can effectively be decomposed into collections of independent agents that collaborate and exchange information using an expressive formal language. In GIA, he adopted a federated architecture for agent communication including three types of agents: a facilitator, a kernel agent and an interface agent. The kernel agent, consisting of five independent agents, provides the essential services of intelligent tutoring systems. In order for the agents to collaborate and exchange information, GIA adopts a subset of KQML performatives. With respect to the ontology problem, Cheikes defined four ontologies: the system ontology, the pedagogy ontology, the domain ontology, and the student ontology, which can capture properties of ITS systems and domains of instruction and can be reused across many different ITS applications. His work on the agent architecture of ITS and the definitions of the four types of ontology is also valuable, although he did not give details about the conceptualization.

Ritter and Koedinger [17] are attempting to build learning environments that incorporate tutoring elements into pre-existing software packages. Their basic idea is similar to ours in the sense that they intend to move toward a set of standards for tutor agents that interact with complex tools. The communication between the tutor agent and tools is domain-independent. They accomplished this domain-independent communication by attaching an accompanying translator, which is domain-dependent, to each of the tools. With the help of translator, tools in different domains are able to share the same tutor agent.

7 Conclusions

In this paper we have tackled a problem that has been neglected by most researchers in the agent communication field. We discussed the domain-independent, but task-dependent, communication content ontology for a learner model agent in the multi-agent architecture of IES. We have analyzed the information that the learner model agent can provide to other agents during the communication, abstracted the domain-independent, but task-dependent, concepts, and organized them into a concept hierarchy. The concepts can compose all the messages necessary to answer the queries in Fig. 3. Additionally some examples have been given at the end of the paper to show how communication content ontology is used in the message content construction.

We would like to draw several general conclusions from the preceding discussion:
1. Domain-independent communication content ontology helps to make the communication process among agents explicit. This process has been implicit in the conventional architecture of IES.
2. Domain-independent ontology helps agents interpret the messages that they exchange, so that the agents/modules can be shared and reused.
3. Communication content ontology provides access to information to be shared, at different levels. The message content can have different grain sizes, and can be both general and specific. With the granularity organization of the domain ontology and learner model ontology, the learner model is organized from coarse grained to fine grained so that the learner model agent can provide information about the learner in grain sizes from coarse to fine.

Overall, we argued for the continued exploration of domain-independent, but task-dependent, communication content ontology in IES, “the key to fluent communication in IES”. The problems are hard and closely related to the functions of agents in the architecture of IES. The study presented in this paper is only at an early stage. The implementation and evaluation of the communication content ontology will follow it. Our long-range vision is the construction of a series of ontologies to facilitate the communication among agents within a single IES, and furthermore between multiple IESs.

Acknowledgements

We would like to thank Dr. Mitsuru Ikeda and Dr. Vladan Devedzic for their valuable comments.

References

[1] Cheikes, B., “GIA: An Agent-Based Architecture for Intelligent Tutoring Systems”, Proc. CIKM’95 Workshop on Intelligent Information Agents, (1995)

[2] Cheikers, B., “Should ITS Designers Be looking for a Few Good Agents?”, Proc. AIED’95 Workshop on Authoring Shells for Intelligent Tutoring Systems, (1995)

[3] Chen, W., Hayashi, Y., Jin, L., Ikeda, M., and Mizoguchi, R., “An Ontology-based Intelligent Authoring Tool”, Proceedings of ICCE'98, vol.1, pp. 41-49 (1998)

[4] McCalla G., and Greer, J., “Granularity-Based Reasoning and Belief Revision in Student Models”, Student Modeling: the Key to Individualized Knowledge-Based Instruction, Spring-Verlag, pp. 39-62 (1994)

[5] Finin, T., Fritzson, R., McKay, D., and McEntire, R., “KQML as an Agent Communication Language”, Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press (1994)

[6] Genesereth, M., and Fikes, R., “Knowledge Interchange Format, Version 3.0 Reference Manual”, Knowledge Systems Laboratory, KSL-92-86 (1992)

[7] Genesereth, M., and Ketchpel, S., “Software Agents”, Communication of ACM, vol. 37, no. 7, pp. 48-53 (1994)

[8] Gruber, T., “A Translation Approach to Portable Ontology Specifications”, Knowledge Acquisition, vol. 5, no. 2, pp. 199-220 (1993)

[9] Ikeda, M., and Mizoguchi, R., “FITS: A Framework for ITS--A Computational Model of Tutoring”, JI. Of Artificial Intelligence in Education, vol. 5, no. 3, pp. 319-348 (1994)

[10] Johnson, L., “Pedagogical Agents”, Proceedings of ICCE’98, vol. 1, pp. 13-22 (1998)

[11] Mengelle, T., De Lean, C., and Frasson, C., “Teaching and Learning with Intelligent Agents: Actors”, Intelligent Tutoring Systems, Lecture Notes in Computer Science 1452, Goettl, B. P., Halff, H. M., Redfield, C. L., and Shute, V. J. (Eds), Springer, pp. 284-293 (1998)

[12] Mizoguchi, R., Sinita, K., and Ikeda, M., “Task Ontology Design for Intelligent Educational/Training Systems”, Proc. ITS’96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs (1996)

[13] Mizoguchi, R., “A Step Towards Ontological Engineering”, http://ei.sanken.osaka-u.ac.jp/english/step-onteng.html (1998)

[14] Murry, T., “Toward a Conceptual Vocabulary for Intelligent Tutoring Systems”, http://www.cs.umass.edu/~tmurray/papers/conceptual_vocab/conceptual_vocab.html (1996)

[15] Paiva, A., “Towards a consensus on the Communication Between User Modeling Agents and Application Agents”, Proc. UM’96 Workshop on Standardization of User Modeling (1996)

[16] Paiva, A., “Communication with Learner Modeling Agents”, Proc. ITS’96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs (1996)

[17] Ritter, S., and Koedinger, K., “An Architecture for Plug-in Tutor Agents”, J. of AIED, vol. 7, no. 3/4, pp. 315-347 (1996)

[18] Ritter, S., “Communication, Cooperation, and Competition among Multiple Tutor Agents”, Artificial Intelligence in Education, du Boulay, B. and Mizoguchi, R. (Eds.), IOS Press, pp. 31-38 (1997)