On Meta Levels of an Organized Society of KDD Agents

Ning Zhong[1], Setsuo Ohsuga[2], Chunnian Liu[3], Yoshitsugu Kakemoto[4], and Xiaodong Zhang[5]

[1] Dept. of Computer Science and Sys. Eng., Yamaguchi University
[2] Dept. of Information and Computer Science, Waseda University
[3] Dept. of Computer Science, Beijing Polytechnic University
[4] RCAST, The University of Tokyo
[5] HPCS, University of Texas at San Antonio

Abstract. Modeling of KDD (Knowledge Discovery in Databases) process constitutes an important and new research area of KDD, that is, *meta levels* of the KDD process, including formal specification of the process, its planning, scheduling, controlling, management, evolution, and reuse. The key issue is how to increase both *autonomy* and *versatility* of a KDD system. Our methodology is to create an organized society of KDD agents. This means (1) to develop many kinds of KDD agents for different discovery tasks; (2) to use the KDD agents in multiple learning phases in a distributed cooperative mode; (3) to manage the society of KDD agents by multiple meta-control levels. Based on this methodology, a multi-strategy and cooperative KDD system, which can be imagined as a *softbot* and is named GLS (Global Learning Scheme), has being developing by us. This paper focuses on the meta control levels for increasing both *autonomy* and *versatility* of the KDD system.

1 Introduction

Generally speaking, *discovery* is a thinking process of human being in order to form the hypothesis from the observed facts, and test/verify it when he/she meets with surprised facts. In other words, *discovery* is a kind of unsupervised (human or machine) learning for acquiring knowledge. Although the means of scientific discovery can be divided into two types: theory-driven and data-driven, data-driven discovery is more one in the history of science and technology [22]. In particular, the number and size of available databases in many fields are growing so fast that there will never be enough human experts to analyze all data in databases and to discover conceptual knowledge from them. Hence, systems with the capability of automatic knowledge discovery from databases (KDD) will play an increasingly important role [12].

Recently, it has been recognized in the KDD community that the KDD process for real-world applications is extremely complicated [3, 1]. Some researchers have proposed that there are several phases and large number of steps and alternative KDD techniques in the process, iteration can be seen in anywhere and at any time, and the process may repeat at different intervals when

new/updated data come. Basically, the KDD process can be divided into two main types: *human-centered* and *autonomous*. Brachman and Anand proposed a human-centered framework of the KDD process [1]. They consider that knowledge discovery is a knowledge-intensive task consisting of complex interactions, protracted over time, between a human and a database, possibly supported by a heterogeneous suite of tools. Zytkow described a way of increasing cognitive autonomy in machine discovery by implementing new components of the discovery process [22], namely, greater autonomy means more discovery steps in succession performed without external intervention, and external intervention can be replaced by automated search, reasoning and the use of background knowledge-bases. However, no one has begun to describe

- how to plan, organize, control, and manage the KDD process dynamically for different KDD tasks;
- how to get the system to know it knows and impart the knowledge to decide what tools are appropriate for what problems and when.

Solving of such issues needs to develop *meta levels* of the KDD process by modeling such a process. We argue that modeling of the KDD process constitutes an important and new research area of KDD, including formal specification of the process, its planning, scheduling, controlling, management, evolution, and reuse. The key issue is how to increase both *autonomy* and *versatility* of a KDD system. Our methodology is to create an organized society of KDD agents. This means

- to develop many kinds of KDD agents for different discovery tasks;
- to use the KDD agents in multiple learning phases in a distributed cooperative mode;
- to manage the society of KDD agents by multiple meta-control levels.

The methodology is based on the viewpoint of "the society of mind" developed by Marvin Minsky [8]. That is, the society of KDD agents is made of many smaller components that are called *agents*. Each agent by itself can only do some simple thing. Yet when we join these agents in an *organized* society, this leads to implement more complex KDD tasks.

Based on this methodology, a multi-strategy and cooperative KDD system, which can be imagined as a *softbot* and is named GLS (Global Learning Scheme), has being developing by us [15, 17]. This paper focuses on the meta control levels for increasing both *autonomy* and *versatility* of the KDD system. It includes to outline the architecture of the GLS system, to describe several KDD agents that have been developed or have been developing, to discuss how to plan and organize the KDD process, how to manage and control the KDD agents, how to combine autonomous discovery and interactive discovery, and our future work.

2 The Architecture of the GLS System

Fig. 1 shows the architecture of the GLS system. We can see that GLS is divided into three levels: two meta control levels (planning, management) and an object

level. On the meta levels, the main (meta) agents are the *planning* meta-agent and the *controlling* meta-agent respectively. The planning/controlling meta-agents and some ("higher-level") agents on the object level have the capability and responsibility of process planning (globally or locally). First, by means of the planning meta-agent on the top level, a KDD process is planned and organized by communicating with a user or learning from environment according to different discovery tasks. Then the controlling meta-agent dynamically creates, executes, and controls the KDD agents in the object level according to the process plan. On the other hand, on the object level, the KDD agents are divided into three groups, that is, three KDD phases: *pre-processing, knowledge elicitation,* and *refinement*, for actual KDD activities in a distributed cooperative mode.

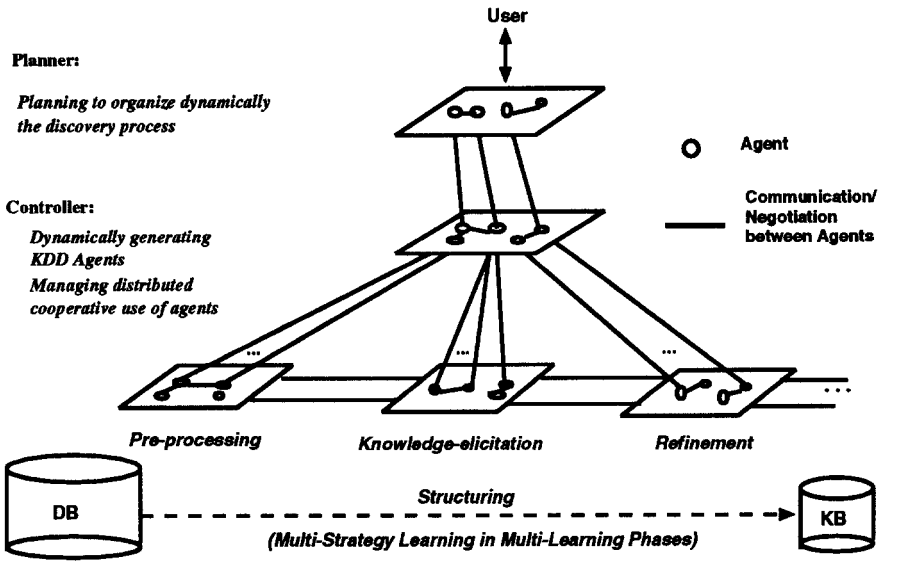

Fig. 1. The architecture of the GLS system

Since uncertainty, incompleteness, and dynamics of data in databases, the discovered knowledge is only the hypothesis, and the hypothesis must be managed and refined. The multiple KDD phases, stated above, which are controlled by the meta levels, serve the KDD process based on the repetitive process of incipient hypothesis generation, evaluation, management, and refinement, as shown in Fig. 2.

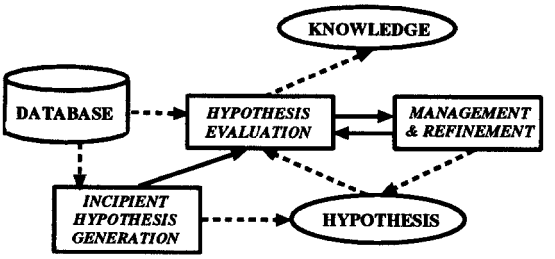

Fig. 2. The process of knowledge discovery from databases

3 KDD Agents and Modeling of the KDD Process

KDD agents in our sense are not static but dynamic, i.e., they are dynamically generated and composed of Intelligent Mail Box. And their IDs are composed of names of workstation and agent. An agent can try to solve a KDD task or a sub-task, or something that serves this task but necessary to another agent's solving of the task. It can interact with other agents to solve the task. Several KDD agents on the object level such as

- agents to collect information from global information sources;
- agents to decompose a large database into several local information sources;
- agents for data cleaning;
- agents for knowledge elicitation;
- agents for refinement and management of the discovered knowledge,

have been developed or have been developing by us. Their development is partly based on the results of our research on several KDD methods and systems [14, 16, 18, ?, 19, 20, 21].

A model of the KDD process consists essentially of a formal description of each type of KDD agents. We introduce a formalism for this purpose in the style of OOER (Object-Oriented Entity Relationship data model). For each type of KDD agents, the types of its input/output and its functionality are explicitly specified in the data model. The most difficult problem in a general-purpose KDD system is that how to choose appropriate KDD techniques to achieve a particular goal in a particular domain. In our method, the combination of the KDD process model and the planner gives an automatic solution to this problem (to some extent, at least). In the hierarchical planning, higher-level agents are gradually decomposed into networks of sub-agents (sub-plans). The type of a higher-level agent has the specification listing the types of its candidate subagents. Given a KDD (sub) goal, the planner reasons on the candidate subagent types to choose the appropriate ones and build the (sub) plan which, when executed, would achieve the (sub) goal. On the other hand, a "lower-level" agent just has an associated KDD algorithm to carry out its task.

4 Meta Control Levels

Our system is not merely a collection of KDD agents stated above and all the rest that will be developed if necessary. For the KDD agents would not work at all unless those agents were linked to one another by a suitable graph of interconnections. As a result of linking the graph, a KDD process is organized dynamically. In other words, we need to use meta control levels, as shown in Fig. 1, for solving some meta-level problems and controlling the meta or object level. The main roles of a meta control level are to generate dynamically KDD agents, allocating resources, adaptive self-configuration of steps in a KDD process, managing interaction/communication among agents in use, synthesizing part-results of discovery, and so on. Furthermore, the meta-level is controlled by a meta-meta-level that is used to plan and organize dynamically the KDD process and to control this process as the coordinator.

4.1 Planning of the KDD Process

Planning to organize dynamically the KDD process is a key component to increase both autonomy and versatility of our system. Typically, a KDD process is the one in which there are more discovery steps performed in succession by using KDD agents in a distributed cooperative mode. The KDD agents can be represented by a graph of discovery goals/subgoals, and a plan describes a way to arrive at a goal state from a given initial state. Unlike classical planning methods, there is not the sharp distinction of planning and execution phases in our system.

The planning meta-agent first sets the top-level plan for the KDD process. Then, coupling with the controlling meta-agent, the global planner decomposes the overall KDD process into a KDD agents network in a hierarchical manner. The basic planning mechanism is a core domain-independent non-linear planner [4, 5] plus a KDD domain specific outer layer as shown in Fig. 3.

Since the KDD process is basically a multi-step process, and KDD agents have been divided into three groups based on three KDD phases, the planning can be done in a multi-step mode. For example, in the pre-processing phase, we can first plan how to decompose a database into several information sources by using cooperatively some KDD agents, and then based on the result of the sub-plan, we do next sub-plan, that is, how to allocate system resources to the decomposed local information sources, and what kinds of KDD agents are used to search hypotheses from the sources, and so on. On the other hand, in many cases, agents' plan may be either conflicting or ambiguous because of incompatible states, incompatible orders of activities, or incompatible use of resources. Hence, we need to use a kind of probabilistic planning and to solve conflicts by a particular agent (coordinator) or through negotiation.

Since the KDD process that is organized is essentially a *repetitive* process as shown in Fig. 2, this implies that revising a plan and re-planning are necessary [5]. On the other hand, the knowledge refinement on data changes is an important component of the KDD process. The process for knowledge refinement is

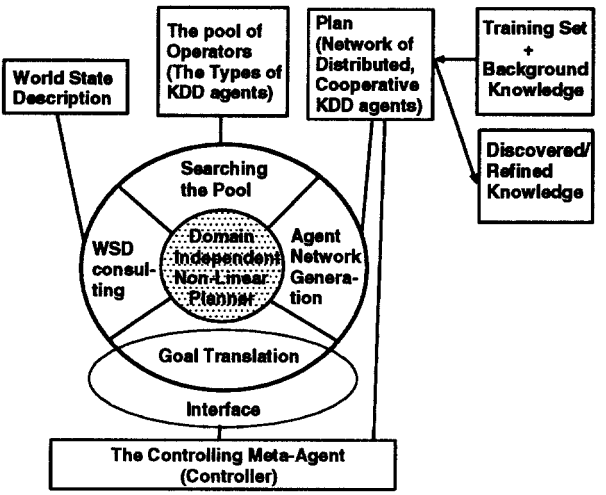

Fig. 3. Coupling of the Planning and Controlling Meta-Agents

similar to the original one. To somehow reuse existing process plans (rather than replanning from scratch) is obviously very desirable. This is achieved by the incremental replanning or case-based planning techniques [5]. Replanning can be designed as a function of the global planner, or we can model the knowledge refinement task by a higher-level KDD agent who takes the process replanning as (part) of its functionality.

4.2 Managing KDD Agents

Managing KDD agents (see Fig. 4) includes to generate dynamically KDD agents in one or more workstations according to the result of planning and organizing a KDD process, and control interaction and communication among agents.

There are two main jobs for the management. The first job is the decomposition and allocation of KDD tasks. To decompose a KDD task into sub-tasks means that these sub-tasks can be solved by using one or more KDD agents that are distributed over different workstations. Thus the decomposition problem leads us to the problem of distributed cooperative system design. We use a method called *Partial Global Planning* (PGP) developed by Durfee and Lesser for this purpose [2]. PGP is designed to have a flexible approach to coordination that does not assume any particular distribution of sub-tasks. Nodes coordinate in response to current situation, each node can represent and reason about the actions and interactions of groups of nodes and how they effect local activities. These representations specify how different parts of the network plan to achieve more global goals. A partial global plan is a general structure for representing coordinated activities in terms of goals, actions, interactions, and relationships.

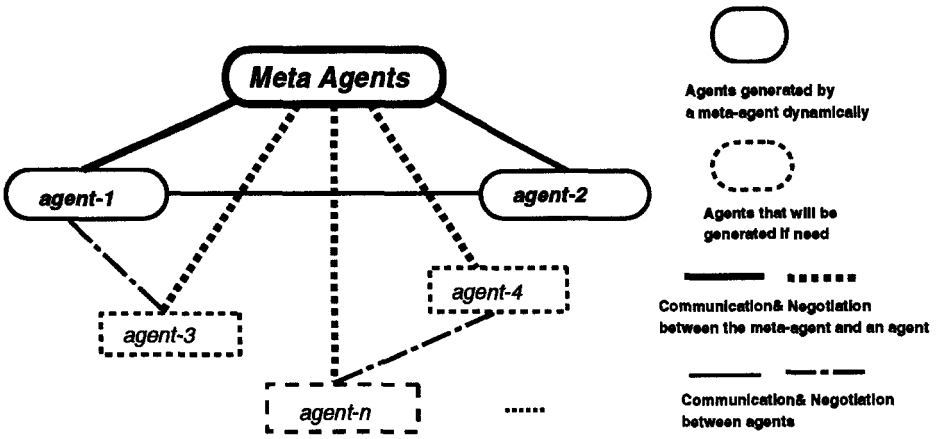

Fig. 4. Managing KDD agents

The PGP maintains a solution-activity map that represents what the nodes are doing, including the major plan steps that the nodes are concurrently taking, their costs, and expected results.

The second job is to allocate resources for KDD agents. Allocating resources to KDD agents is basically a scheduling task that synchronizes activities to avoid and resolve conflicts. The schedule is built in a cooperative fashion through local computation and communication [13]. This method can coordinate parallel KDD agents independently in each workstation based on the co-scheduling principle to ensure that no agent will wait for a non-scheduled agent for synchronization/communication, and will minimize the waiting time at the synchronization points.

5 The Value of Information and User Interface

An interface between users and the organized society of KDD agents is needed for combining autonomous discovery and interactive discovery. The main goal of the GLS system based on the intelligent agent model shown in Fig. 1 is to increase both *autonomy* and *versatility* of the system, although it is necessary that combining with interactive discovery for realistic KDD applications. In GLS, the KDD process is essentially that of information transformation by structuring operations for data in multiple KDD phases. The result of the information transformation is that more generalized form (i.e., knowledge) of information is obtained. In other words, the value of information increases along with the structuring operations in the KDD process. For this purpose, we need the criteria to evaluate quantitatively the progress of structuring operations in different

KDD phases. That is, the criteria are used to evaluate the increasing degree of the value of information.

Here, we should distinguish between two kinds of value of information: *formal* and *semantic*, so that select the useful criteria to increase cognitive autonomy [9]. The formal value of information is the intrinsic value of information; the semantic value of information has relevance to the objective of the user and the use of background knowledge. We argue that the formal value of information should play a leading role for KDD if we distinguish between discovery and invention, "laws of nature" and "laws of science" [22]. For instance, a criterion based on information theory (entropy) can be used to evaluate the formal value of information in the progress of structuring operations [19].

However, when the result of structuring operations is used to form a theory, concept or "law of science" by human, there are two possibilities: correct and incorrect. That is, this result cannot ensure to meet the demands of the user to be a hundred percent sure. Hence, in some cases, background knowledge or the judgment of the user should be used to decide the direction of structuralization (or refinement). That is, the semantic value of information is considered in this case. In other words, what is called the *semantic value* of information is to consider the intention of the user and/or to use a large amount of background knowledge in the KDD process. For this purpose, we need to develop both a large-scale background knowledge-base and an intelligent user interface for our system.

6 Concluding Remarks

We presented a methodology based on an intelligent agent model with multiple meta levels for increasing both *autonomy* and *versatility* of a KDD system, and the framework of our GLS system based on this methodology. In comparison, GLS is mostly similar to INLEN in related systems [7]. In INLEN, a database, a knowledge-base and several existing methods of machine learning are integrated as several operators. These operators can generate diverse kinds of knowledge about the properties and regularities existing in the data. INLEN was implemented as a toolkit like GLS. However, GLS can dynamically plan and organize the KDD process performed in a distributed cooperative mode for different KDD tasks. Moreover, the refinement for knowledge is one of important capabilities of GLS that was not developed in INLEN.

GLS is implemented by KAUS. KAUS is a knowledge-based system shell developed in our group that involves knowledge-bases based on multi-layer logic [10]. Thanks to the useful capabilities such as meta reasoning, multiple knowledge worlds/levels and the model representation in KAUS, and the recent development of distributed KAUS [11], GLS can be easily implemented and extended by KAUS.

Since the GLS system to be finished by us is very large and complex, however, we have only finished several parts of the system and have undertaken to extend it for creating a more integrated, organized society of KDD agents. That is,

the work that we are doing takes but one step toward a multi-strategy and cooperative KDD system.

Acknowledgements

This research was partially supported by International Communications Foundation of Japan.

References

1. Brachman, R.J. and Anand, T. 1996. The Process of Knowledge Discovery in Databases: A Human-Centred Approach. In *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press, 37-58.
2. Durfee,E.H. and Lesser,V.R. 1989. *Negotiating Task Decomposition and Allocation using Partial Global Planning.* Distributed Artificial Intelligence Vol.2.
3. Fayyad,U.M., Piatetsky-Shapiro,G et al (eds.) 1996. *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press.
4. Liu, C. 1991. Software Process Planning and Execution: Coupling vs. Integration. LNCS 498, Springer Verlag, 356-374.
5. Liu, C. and Conradi, R. 1993. Automatic Replanning of Task Networks for Process Evolution in EPOS. Proc. the 4th European Software Engineering Conference (ESEC'93), LNCS 717, Springer Verlag, pp.437-450.
6. Matheus, C.J., Chan, P.K. & Piatetsky-Shapiro, G. 1993. Systems for Knowledge Discovery in Databases. *IEEE Trans. Knowl. Data Eng.*, 5(6):904-913.
7. Michalski, R.S. et al. 1992. Mining for Knowledge in Databases: The INLEN Architecture, Initial Implementation and First Results. *J. of Intell. Infor. Sys.*, KAP, 1(1):85-113.
8. Minsky,M. 1986. *The Society of Mind*, Simon and Schuster, New York.
9. Ohsuga, S. 1970. On the Value of Information and Decision Making. *Trans. of Information Processing of Japan*, Vol.10:97-108.
10. Ohsuga, S. 1990. Framework of Knowledge Based Systems. *Knowledge Based Systems*, 3(4):204-214.
11. Ohsuga, S. 1995. A Way of Designing Knowledge Based Systems. *Knowledge Based Systems*, 8(4):211-222.
12. Piatetsky-Shapiro, G. & Frawley, W.J. (eds.). 1991. *Knowledge Discovery in Databases.* AAAI/MIT Press.
13. Zhang,X. 1996. Co-scheduling Parallel Workloads across Networks of Workstations, invited talk at Yamaguchi Univ. Japan, June 1996.
14. Zhong, N. & Ohsuga, S. 1994a. Discovering Concept Clusters by Decomposing Databases. *Data & Knowl. Eng.*, Elsevier Science Publishers, 12(2):223-244.
15. Zhong, N. & Ohsuga, S. 1994b. The GLS Discovery System: Its Goal, Architecture and Current Results. *Proc. 8th Inter. Symp. on Methodologies for Intell. Sys. (ISMIS'94).* LNAI 869, Springer, 233-244.
16. Zhong, N. & Ohsuga, S. 1995a. KOSI - An Integrated System for Discovering Functional Relations from Databases. *J. of Intell. Infor. Sys.*, KAP, 5(1):20-50.
17. Zhong,N. and Ohsuga,S. 1995b. Toward A Multi-Strategy and Cooperative Discovery System. *Proc First Inter. Conf. on Knowledge Discovery and Data Mining (KDD-95)*, AAAI Press, 337-342.
18. Zhong,N. and Ohsuga,S. 1996a. System for Managing and Refining Structural Characteristics Discovered from Databases. *Knowledge Based Systems*, Elsevier, 9(4):267-279.
19. Zhong,N. and Ohsuga,S. 1996b. A Hierarchical Model Learning Approach for Refining and Managing Concept Clusters Discovered from Databases. *Data & Knowl. Eng.*, Elsevier Science Publishers, 20(2): 227-252.
20. Zhong,N. and Ohsuga,S. 1996c. Using Generalizations Distribution Tables as a Hypothesis Search Space for Generalization. *Proc. 4th Inter. Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD'96)* (1996) 396-403.
21. Zhong,N., Fujitsu,S., and Ohsuga,S. 1997. Generalization Based on the Connectionist Networks Representation of a Generalization Distribution Table, H. Lu, et al. (eds.) *KDD: TECHNIQUES AND APPLICATIONS. Proc. First Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-97)*, World Scientific (1997) 183-197.
22. Zytkow, J.M. 1993. Introduction: Cognitive Autonomy in Machine Discovery. *Machine Learning*, KAP, 12(1-3):7-16.