Design of a Coordinating Mechanism for Multi-Level Scheduling Systems in Supply Chain

Jung Seung Lee* · Soo Kim**

Abstract

The scheduling problem of large products like ships, airplanes, space shuttles, assembled constructions, and automobiles is very complex in nature. To reduce inherent computational complexity, we often design scheduling systems that the original problem is decomposed into small sub-problems, which are scheduled independently and integrated into the original one. Moreover, the steep growth of communication technology and logistics makes it possible to produce a lot of multi-nation corporation by which products are produced across more than one plant. Therefore vertical and lateral coordination among decomposed scheduling systems is necessary.

In this research, we suggest an agent-based coordinating mechanism for multi-level scheduling systems in supply chain. For design of a general coordination mechanism, at first, we propose a grammar to define individual scheduling agents which are responsible to their own plants, and a meta-level coordination agent which is engaged to supervise individual scheduling agents. Second, we suggest scheduling agent communication protocols for each scheduling agent topology which is classified according to the system architecture, existence of coordinator, and direction of coordination. We also suggest a scheduling agent communication language which consists of three layers : Agent Communication Layer, Scheduling Coordination Layer, Industry-specific Layer. Finally, in order to improve the efficiency of communication among scheduling agents we suggest a rough capacity coordination model which supports to monitor participating agents and analyze the status of them.

With this coordination mechanism, we can easily model coordination processes of multiple scheduling systems. In the future, we will apply this mechanism to shipbuilding domain and develop a prototype system which consists of a dock-scheduling agent, four assembly-plant-scheduling agents, and a meta-level coordination agent. A series of experiment using the real-world data will be performed to examine this mechanism.

Keywords : Coordinated Scheduling, Agent Communication Language, Supply Chain

Received : 2011. 12. 02. Final Acceptance : 2012. 01. 27.

* Department of Business Administration, Hoseo University, e-mail : jslee@hoseo.edu
** Corresponding Author, College of International Studies, Dankook University, e-mail : sookim@dankook.ac.kr

1. Introduction

The scheduling problems of large products like ships, airplanes, space shuttles, assembled constructions, and automobiles are very complex in nature. To reduce inherent computational complexity, we often design scheduling systems, so that the original problem is decomposed into small sub-problems, which are scheduled independently and integrated into the original one.

As our experience in real-world problems, DAS (D Shipbuilding Scheduling System) has adopted a two-layered hierarchical architecture. In the hierarchical architecture, individual scheduling systems composed of a high-level dock scheduler, DAS-ERECT and low-level assembly plant schedulers, DAS-PBS, DAS-3DS, DAS-NPS, and DAS-A7 try to search the best schedules under their own constraints.

Moreover, the steep growth of communication technology and logistics enabled introduction of distributed multi-nation production plants by which different parts are produced by designated plants. Therefore vertical and lateral coordination among decomposed scheduling systems is necessary.

In scheduling research area, there are a lot of scheduling systems. However, no standard coordination mechanism of multiple scheduling systems exists. Past researches about the coordination mechanism are external convertsation without capacity model, which is too simple, domain independent, and possible to fall into infinite loop.

In agent research area, there are a lot of researches about agent-based coordination. However, no scheduling domain exists. Past researches about the agent-based scheduling are internal coordination of scheduling process, which is toy problem and is not efficient.

The coordination problem among individual scheduling systems can be generalized to the shape of Supply Chain Network. Raw material handling can be solved by MRP and distribution problem is an area of logistics. However the researches about vertical coordination and lateral coordination among each-level schedulers aren't sufficient.

Finally, why are agent technologies imported in this research? These days are agent era. The characteristics of agent are autonomy, social ability, reactivity, and pro-activeness. By wrapping legacy systems in agents or re-coding legacy systems to the shape of agents, agent technologies can be implemented on legacy systems.

The objective of our research is to design a standard coordination mechanism of multiple scheduling systems and develop an agent-based coordination system. The standard coordination mechanism can be summarized as follows :

- Scheduling agent definitions
- Scheduling agent topology
- Scheduling agent communication protocol
- Scheduling agent communication language
- Rough capacity coordination model

Therefore, we design a general coordination framework of multiple scheduling systems, but the other hand we restrict the scope of prototype system to the shipbuilding domain.

2. Related Research

2.1 Agent-based Scheduling

Designs for real-world agent-based systems must reflect domain requirements as well as the technical capabilities of agents. This needs-driven approach is being applied in AARIA [Autonomous Agents for Rock Island Arsenal; Parunak, Baker and Clark, 1997], an industrial-strength agent-based factory scheduling and simulation system being developed for an Army manufacturing facility. A review of the operations of Rock Island in the light of broader industrial needs yields seven requirements. After introducing the AARIA agent community, they summarize each of these requirements and how AARIA supports it. More information is available at http://www.aaria.uc.edu.

Negotiation among Knowledge-based Scheduling Agents (Lessor) results in the development of the DENEGOT architecture. This architecture represents a general approach for guiding the distributed search of multiple agents through negotiation to arrive at satisfying solution to a possibly over constrained problem. The domain is that of an Airline Resource Manager whose role is to insure that each flight receives the resources (gates, baggage handling, catering, fuel, cleaning, etc.) that it requires in time to meet its arrival and departure deadlines.

Agent-based Meeting Scheduling [Jennings and Jackson] describes the design and implementation of a distributed meeting scheduling system in which each user has an intelligent agent in their computer desktop which is responsible for arranging meetings. Knowing the references and commitments of their user, the agents negotiate with one another to find the most acceptable meeting times.

2.2 Agent-based SCM

Participants in the coordination process are a customer agent, an enterprise wide logistics agent that manages work at the highest level of the supply chain, and a number of enterprise agents, like plants and transportation agents that will execute parts of the work. There are vertical coordination between the logistics agent and the enterprise agents and horizontal coordination between the customer and logistics agents, as well as between plant agents. They use finite capacity scheduling at both vertical levels by logistics and by each involved enterprise agent in part.

The work management processes that permit :

- Negotiation of work between customer and logistics agents
- Negotiation of work between logistics and plant agents.
- Monitoring of execution.
- Solving of unexpected problems occurring dynamically during execution (like breakdowns that make it impossible for a plant agent to finish the committed work, or lack of capacity due to overloading) by re-negotiation with the customer or at the enterprise level.

3. General Framework of Coordination

3.1 Agent Definition

The agent definitions are classified according to system architectures : distributed architecture and hierarchical architecture. In distributed architecture, there are two types of agents which consist of individual scheduling agents and a meta-level coordination agent. In the hierarchical architecture, individual scheduling agents are ordered according to the level of hierarchy: high-level scheduling agent and low-level scheduling agent. A high-level scheduling agent is equal to the union of an individual scheduling agent and a meta-level coordination agent, and a low-level scheduling agent is equal to individual scheduling agent.

3.1.1 Agent Definitions in Distributed Architecture

It is assumed that each individual scheduling agent already has both generative scheduling module and reactive scheduling module. Roles of the individual scheduling agent are summarized as follows :

- Generate schedules for assigned tasks
- Report the schedules to another scheduling agent or the meta-level coordinator
- Request to adjust the task delivery dates for unscheduled tasks
- Request to transfer the task for unscheduled tasks
- Revise schedules for requested tasks and proposal of them
- Report the revised schedule to another scheduling agent or the meta-level coordinator
- Report the status of the charging plant

Roles of the meta-level coordination agent are summarized as follows :

- Recommend or guide scheduling agents to find which agents can adjust the task delivery dates
- Recommend or guide scheduling agents to find which agents can transfer the tasks
- Request to adjust the task delivery dates for unscheduled tasks
- Request to transfer the task for unscheduled tasks
- Monitoring the status of scheduling agents periodically
- Integrate analyze and the status of scheduling agents

3.1.2 Agent Definitions in Hierarchical Architecture

It is assumed that the hierarchy such as high-level or low-level comes into existence in the hierarchical architecture. A high-level scheduling agent consists of an individual scheduler and a meta-level coordinator as it should manage low-level scheduling agents.

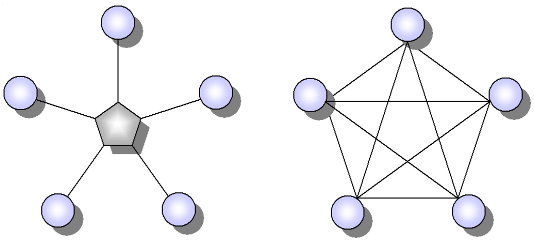
Roles of the high-level scheduling agent are equal to the union of roles of the individual scheduling agent and roles of the meta-level coordination agent. Roles of the low-level scheduling agent are equal to the roles of the individual scheduling agent.

3.2 Scheduling Agent Topology

3.2.1 Lateral Communication in Distributed Architecture

If there is a coordinating agent, one individual scheduling agent communicates with the other individual agent indirectly through the coordinator. In this case, when there are n individual scheduling agents, only n communication channels are necessary.

On the other hand, if there isn't any coordinating agent, one individual scheduling agent should communicate with the other individual agent directly. In this case, when there are n individual scheduling agents, nC_2 communication channels are necessary.

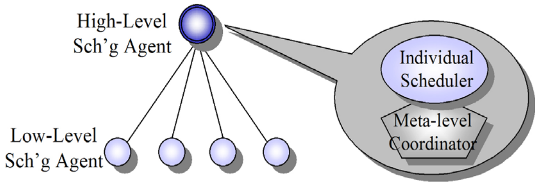


〈Figure 1〉 Lateral Communication in Distributed Architecture

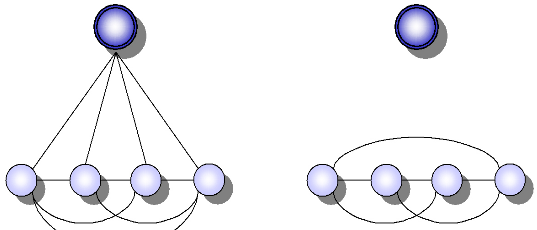
3.2.2 Vertical and Lateral Communication in Distributed Architecture

In hierarchical architecture, there are two types of communications : vertical communication and lateral communication. A high-level scheduling agent communicates with low-level scheduling agents vertically.

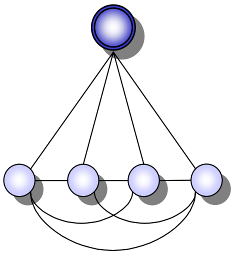
Moreover one low-level scheduling agent communicates with the other low-level scheduling agent laterally. If a low-level scheduling agent uses the meta-level coordinator of the high-level scheduling agent, a low-level scheduling agent communicates with the other scheduling agent indirectly through the high-level scheduling agent. If a low-level scheduling agent doesn't use the meta-level coordinator of the high-level scheduling agent, a low-level scheduling agent should communicate with the other scheduling agent directly.



〈Figure 2〉 Vertical Communication in Hierarchical Architecture



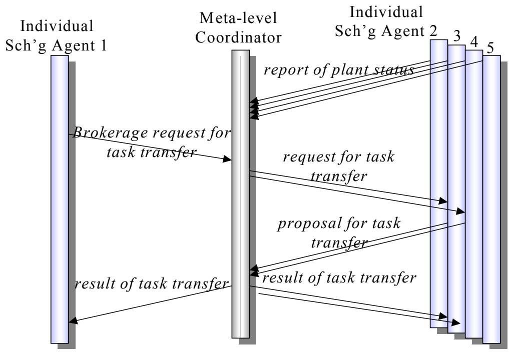
〈Figure 3〉 Lateral Communication in Hierarchical Architecture



〈Figure 4〉 Vertical and Lateral Communication in Hierarchical Architecture

3.3 Scheduling Agent Communication Protocol

Lateral communication protocols are communication protocols among individual scheduling agents. If there is a meta-level coordinator, we can use Lateral Transfer Protocol with a Brokerage Coordinator or Lateral Transfer Protocol with a Recommending Coordinator. If there isn't any meta-level coordinator, we can use Lateral Transfer Protocol without any Coordinator.



<Figure 5> Lateral Transfer Protocol with a Brokerage Coordinator

Conversation plans of Lateral Transfer Protocol with a Brokerage Coordinator are represented as follows :

- Each individual scheduling agent report its plant status to meta-level coordination agent.
- An individual scheduling agent requests brokerage for task transfer to meta-level coordination agent.
- The meta-level coordination agent nominates participant agents and requests task transfer to the participant agents.
- The participant agents propose task transfer to the meta-level coordination agent.
- The meta-level coordination agent tells the result of task transfer to participant agents.

And vertical communication protocols are communication protocols between high-level scheduling agents and low-level scheduling agents. We can use Vertical Assignment Protocol and Vertical Adjustment Protocol.

3.4 Scheduling Agent Communication Language

Communication language standards facilitate the creation of interoperable software by de-coupling implementation from interface. As long as programs abide by the details of standards, it does no matter how they are implemented. Problems arise when it becomes necessary for programs that use one language to inter-operate with programs that use a different language. Agent-based software engineering attacks such problems by mandating a universal communication language, one in which inconsistencies and arbitrary notational variations are eliminated. For this purpose, the formal and common representation of messages, so called Agent Communication Language (ACL) is suggested, which consists of two part; In order to communicate with other agents, an agent should generate messages and interpret the received ones. For this purpose, the

<Table 1> Scheduling Agent Communication Protocols

		Vertical Communication Channel	Lateral Communication Channel
Distributed Architecture	With Coordinator		• Lateral Transfer protocol with a brokerage coordinator (5 types of messages) • Lateral Transfer protocol with a recommending coordinator (6 types of messages)
	Without Coordinator		• Lateral transfer protocol without any coordinator (3 types of messages)
Hierarchical Architecture	With Coordinator	• Vertical assignment protocol (1 type of message) • Vertical adjustment protocol (3 types of messages)	• Lateral Transfer protocol with a brokerage coordinator (5 types of messages) • Lateral Transfer protocol with a recommending coordinator (6 types of messages)

formal and common representation of messages, so called an agent communication language (ACL) is necessary. ACL consists of two part; outer language and inner language. In this paper, we adopt the Knowledge Query and Manipulation Language (KQML) [Finin, et al., 1993] as the outer language, while UNIK-OBJECT (an object-oriented tool) as inner language.

KQML is developed as one of the principal outcome of DARPA Knowledge Share Effort, and is performed by the External Interface Working Group. KQML expresses the communicating attitude about information such as querying, stating, believing, requiring, achieving, subscribing, and offering [Finin, et al., 1993]. Since KQML is indifferent to the contents of the information, it often employs another inner language (or content language) like KIF or UNIK-OBJECT. The structure of KQML message consists of a performative and its parameters.

The performative of Agent A is ask-if which means asking the receiver (agent B) if the content of this message is right or not, and its parameters are language whose value define the language of the message content, ontology whose value is the application area, reply-with whose values is the message identifier of responding message, and content whose value is the message content. Similarly, agent B responds the message whose performative is reply with similar parameters except in-reply-to for message identifier [Finin, et al., 1993].

ACL is devised for the communication among agents independent of domain. Therefore, to apply ACL specifically to the scheduling coordination domain, we should additionally define an inner language which can suitably express scheduling coordination. To implement this idea, we adopt three message layers :

- ACL layer : domain independent outer language layer
- Scheduling coordination layer : terms necessary for scheduling coordination
- Industry-specific layer : the layer that express the industry specification

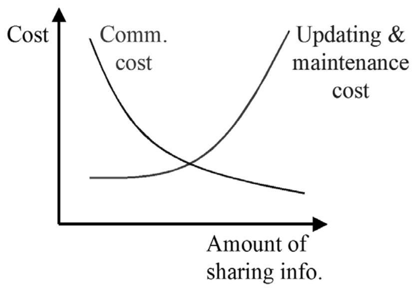
```
(broker                                                          ACL Layer
    : sender          individual_sch'g_agent_1
    : receiver        meta_level_coordination_agent
    : reply-with      20110428_1
    : ontology        agent_based_sch'g
    : language        KQML
    : Content
        (ask
            : ontology        agent_based_sch'g
            : language        UNIK-FRAME
            : Content
                (request_for_task_transfer                    Sch'g Coordination Layer
                    (process_ID       task_transfer_20110428_1)
                    (coordination_type)
                    (requirement
                        (task
                            (task_name          1099-224)
                            (specifications                 Industry-specific Layer
                                (part_of        proj_1099)
                                (consist_of     S9 M9 R1 H1 H3 G7 G9 N1 N9)
                                (front_tasks    1099-221 1099-222)
                                (rear_tasks     1099-233 1099-234)
                                (joint_length   (= 1950.00))
                                (man_hour       (= 785.00))
                                (weight         (= 105))
                                (duration       (= 15))
                                (start_time     (>= 2011/05/21))
                                (finish_time    (<= 20/06/13)))))))
```

〈Figure 6〉 Brokerage Request for Task Transfer

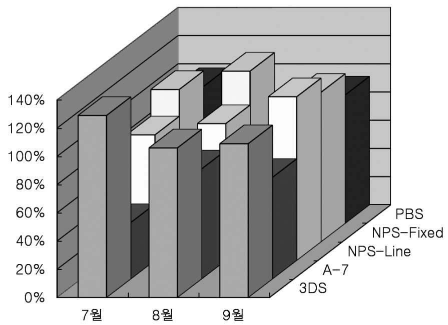
3.5 Rough Capacity Coordination Model

To build the rough capacity coordination model, the status of participating agents should be monitored. Most of all, the amount of sharing information should be considered. If complete information is collected, updating and maintenance cost of sharing information will be increased though the frequency of communication will be decrease. Therefore the level of detail and updating period of sharing information should be decided contingently.

An illustrative example of summary report from participant agents is 3-Dimensional Work Tightness Model. Available capacity and used capacity which are useful to calculate work tightness of each stage are reported by plant and stage according to the time units.



〈Figure 7〉 Amount of Sharing Information vs. Costs



〈Figure 8〉 The Work Tightness by Plants According to Months

4. Prototype Development

With the standard coordination mechanism, we can easily model coordination processes of multiple scheduling systems. In the future, we will apply this mechanism to shipbuilding domain and develop a prototype system which consists of a dock-scheduling agent, four assembly-plant-scheduling agents, and a meta-level coordination agent.

The individual scheduling agent is largely divided into two parts : the communication part and the problem-solving part. In the communication part, there are communication controller, message handling knowledge base, and message database. In the problem-solving part, there are generative scheduling module, reactive scheduling module, scheduling knowledge base, and schedule database. On the other hand, the problem solving part of the meta-level coordination agent consists of rough-capacity coordination model, coordination knowledge base, and summary database. By wrapping legacy scheduling systems to the shape of agents, we will be able to implement scheduling agents in shipbuilding domain.

5. Conclusions and Future Research Issues

The major contributions of this research can be summarized as follows :

- We suggest a general framework for agent-based coordination of multiple scheduling systems. For design of a standard coordination mechanism, at first, we propose a grammar to define individual scheduling agents which are responsible for their own plants, and a meta-level coordination agent which is engaged to supervise individual scheduling agents.

- We suggest scheduling agent communication protocols for each scheduling agent topology which is classified according to the system architectures, existence or non-existence of coordinator, and directions of coordination. We also suggest a scheduling agent communication language which consists of three layers : Agent Communication Layer, Scheduling Coordination Layer, Industry-specific Layer.
- In order to improve the efficiency of communication among scheduling agents and avoid possible infinite loops, we suggest a rough capacity coordination model which supports to monitor participating agents and analyze the status of them.
- With this standard coordination mechanism, we can easily model coordination processes of multiple scheduling systems.

In the future, we will apply this mechanism to shipbuilding domain and develop a prototype system which consists of a dock-scheduling agent, four assembly-plant-scheduling agents, and a meta-level coordination agent. A series of experiment using the real-world data will be performed to examine this mechanism.

References

[1] Jennings, N. R. and Jackson, A. J., "Agent-based Meeting Scheduling : A Design and Implementation", *Working Paper, Dept. of Electronic Engineering*, Queen Mary and Westfield College.

[2] Lesser, V., Negotiation among Knowledge-based Scheduling Agents, http://dis.cs.umass.edu/research/arm.html.

[3] Lim, Y. H., "Special Issue : Agent System", *Korea Information Science Society Review*, Vol. 15, No. 3, March, 1997, pp. 6-68.

[4] Parunak, H. V. D., Baker, A. D., and Clark, S. J., "The AARIA Agent Architecture : An Example of Requirements-Driven Agent-Based System Design", Proceeding of the First International Conference on Autonomous Agents (ICAA), Marina del Rey, CA, February 6-8, 1997.

[5] Parunak, H. V. D., "MASCOT : A Virtual Factory for Research and Development in Manufacturing Scheduling and Control", ITI Tech Memo, 93-02, 1993.

Author Profile

Jung Seung Lee

Jung Seung Lee is a currently assistant professor in department of business administration at Hoseo University in South Korea. He has been served as a vice-president of Korea Intelligent Information System Society and a director of Korea Database Society. He received a B. A. and M. S. in Management Science and Ph.D. in Management Engineering from KAIST. He established two venture companies such as Good Friends (oldboy.co.kr, an internet community) and Best Money (bestmoney.co.kr, a financial consulting site for individuals). His primary research areasare supply chain planning and scheduling, supply network analysis, and business rule and process management.

Soo Kim

Dr. Soo Kim received her Ph.D degree from the Pennsylvania State University and is currently an associate professor of International Business Administration at Dankook University in South Korea. Prior to joining Dankook University, Dr. Kim was a tenured associate professor of management at Montclair State University, NJ, USA. Dr. Kim's research interests center around leadership, emotions, employee work behaviors, social network analysis, and IT. She is the author of numerous articles published in a variety of journals.

◆ 이 논문은 2011년 12월 02일 접수하여 2012년 01월 27일 게재확정되었습니다.