Distributed Coordination of Project Schedule Changes by Using Software Agents

소프트웨어 에이전트를 이용한 건설공사 공정관리의 분산화

김 기 수*
Kim, Kee-soo

Abstract

In the construction industry, projects are becoming increasingly large and complex, involving multiple subcontractors. Traditional centralized coordination techniques used by the general contractors become less effective as subcontractors perform most work and provide their own resources. When subcontractors cannot provide enough resources, they hinder their own performance as well as that of other subcontractors and ultimately the entire project. Thus, construction projects need a new distributed coordination approach wherein all of the concerned subcontractors can reschedule a project dynamically. To enable the new distributed coordination of project schedule changes, I developed a novel agent-based compensatory negotiation methodology, which allows software agents to simulate negotiations on behalf of their human subcontractors. This research formalizes the necessary steps that would help construction project participants to increase the efficiency of their resource use, which in turn will enhance successful completions of whole projects.

Keywords : Scheduling, Project management, Software agent, Distributed coordination

1. Introduction

1.1 Practical Motivation

Despite the ubiquity of change in large, complex construction projects, current approaches to change coordination are mostly reactive, and therefore lead to less than optimal solutions. If, however, changes in a given schedule were coordinated prior to execution, then better solutions could be found. Previous researchers have explored the various causes of schedule changes in construction projects. Discrepancies between the needed resources for activities and the resources available to subcontractors are one major cause of change. The resource discrepancies occur when the timing of the activities is not well matched with the available resources, i.e., when subcontractors have different perspectives of scheduling.

Soon after the general contractor awards subcontracts according to the master project schedule, subcontractors often want to change the master schedule because resource discrepancies cause additional costs either through over-utilizing currently available resources or importing new resources (OBrien and Fischer 2000). Therefore, the subcontractors may try to change the project schedule in order to accommodate their wishes. Changes are likely to cause schedule conflicts among subcontractors because any move affects the activities of other subcontractors in tightly coupled construction project schedules.

In most cases, these schedule conflicts cannot easily be resolved simply by delaying the succeeding activities since such delays would affect the resource profiles of succeeding subcontractors, which would cause additional costs for them. Delays could also extend the project completion beyond the deadline. Therefore, there is a need for a methodology to handle subcontractors resource-driven schedule changes.

1.2 Points of Departure

Numerous research papers have recognized a major problem in the Critical Path Method (CPM) network approach, which assumes unlimited resource supplies, and have provided frameworks to address various limited-resource issues in construction planning and

---

* 일반회원, (주)대우건설 정보전략팀 차장, 공학박사

scheduling, however, few current frameworks address the difficulties of gathering information in the coordination of subcontractors' resource-driven schedule changes. The existing centralized frameworks are insufficient because the information needed for centralized resource-based scheduling, such as the resource constraints, is usually kept private by subcontractors (Choo and Tommelein 2000). Therefore, subcontractors need a distributed coordination framework for project schedule changes (DCPSC) that includes a monetary conflict-resolution methodology, while maintaining schedule logic and keeping their information private.

Current distributed frameworks in construction and in the broader project management and AI research literature have inadequately addressed challenges for distributed coordination of project schedule changes. They have not provided a monetary conflict-resolution mechanism, which is their main shortcoming, even though some of them have provided various conflict-resolution mechanisms for interactions between participants (Khedro et al. 1993; Jin and Levitt 1993). ProcessLink (Petrie et al. 1998) identifies dependencies among activities and participants but does not specify a conflict-resolution mechanism.

1.3 Research Objectives

To overcome past research limitations for DCPSC, the overall purpose of the research was threefold:

(1) To formalize and generalize a DCPSC framework
(2) To formalize and generalize an agent-based compensatory negotiation (ABCN) methodology to enable the DCPSC framework, and
(3) To implement a distributed subcontractor agent system (DSAS) to demonstrate the DCPSC framework

The next three sections describe the DCPSC, ABCN methodology, and DSAS. For more details, refer to Kim (2001).

2. DCPSC

2.1 Formalization of DCPSC

In order for subcontractors to consider their own activities, and also enhance global outcomes, I developed the distributed coordination framework for project schedule changes (DCPSC), based on the social welfare function. I define a social choice function, E, such that E represents the group choice, as follows:

$$E = \sum_{i=1}^{n} \sum_{j=1}^{m} Cost_{(i,j)}$$

where E and $Cost_{(i,\ j)}$ = the sum of the subcontractors extra costs for all m activities of n subcontractors and the jth activity, which belongs to the ith subcontractor.

To increase individual utility and social welfare together, therefore, I set the objective of distributed coordination of project schedule changes so as to lower E, i.e., the sum of subcontractors costs associated with their resource constraints, subject to the precedence relationship among project activities:

$$lower \quad E = \sum_{i=1}^{n} \sum_{j=1}^{m} Cost_{(i,j)}$$

subject to:

$$\forall x, Finish_x \le \min_{y \in S_x} \{Start_y - 1\}$$

where E and $Cost_{(i,\ j)}$ = the sum of the subcontractors extra costs for all m activities of n subcontractors and the jth activity, which belongs to the ith subcontractor, respectively; $Finish_x$ = finish date of activity x; $Start_y$ = start date of activity y; and S_x = set of activities which must succeed activity x.

This DCPSC framework reveals three important issues: distributed coordination by competitive subcontractors, socially rational decision-making, and maintaining the logical sequence of the work.

2.2 Agent-Based Negotiation Approach

I introduce an agent-based negotiation approach to overcome the difficulties stemming from these issues. In the context of this paper, I define agent-based negotiation as the process of resolving conflicts among affected agents by increasing knowledge about others intentions through the structured exchange of relevant information.

By adopting agent-based negotiation approach, I can model the subcontractors as software agents performing a task on behalf of human subcontractors, while modeling the interactions among subcontractors as agent negotiation protocols based on agent communication language. Software agents can communicate rapidly with each other over the Internet, which allows subcontractors to coordinate project schedule changes with the agent-based compensatory negotiation methodology to be summarized in the next section.

3. Agent-Based Compensatory Negotiation Methodology

In this section, I formalize three main aspects of the agent-based compensatory negotiation (ABCN) methodology: (1) the compensatory negotiation strategy based on utility to the agents; (2) the multi-linked negotiation protocols by which agents interact with other agents; and (3) message-handling mechanisms for agents to evaluate alternatives and simulate the decision-making.

3.1 Compensatory Negotiation Based on Utility

Each agent calculates the utility for each activity k using the following function:

$$Utility_k = AC_k - \sum_{x \in all\_suceeding\_activities_k} DC_x$$

where AC_k is the extra acceleration cost for accelerating the *kth* activity; DC_x is the extra delay cost for delaying the succeeding activity x. Note that the agent, which has activity k, knows *ACk*, but does not know the summation of *DCx* until getting *DC*s from the succeeding activities.

By calculating utility of timing, agents can evaluate the impacts of its schedule changes and compensate other agents for disadvantageous agreements through a utility transfer scheme. After getting *DC* through negotiation, if *AC* is more than *DC*, i.e., there is positive utility, the agent decides to make an extension, and transfers the *DC* portion of the utility to other agents for compensation of disadvantageous agreements.

3.2 Multi-Linked Negotiation Protocols

Negotiation protocols govern the interaction among agents by constraining the way the agents interact. In this research, agents need negotiation protocols to get *DC*s from succeeding agents and transfer utility to other agents for compensation of disadvantages agreements. This is simple when agents can reschedule their activities without affecting others or the counterpart agent is one, which is the case of pair-wise negotiation. In a more complicated case, an agent needs to negotiate with another agent, which in turn need to negotiate with a third, and so on, until the last agent. I call it multi-linked negotiations.

Multi-linked negotiation protocols are needed because of the tightly coupled nature of construction project schedules. The multi-linked negotiation differs from multilateral negotiation (auction) protocols because multi-linked negotiation allows agents to negotiate with other agents within precedence relationships rather than restricting them to negotiate solely with an auctioneer. The negotiation protocols provide the performatives, which are shared primitive message types for agents to use in negotiation, and conversation sequence, which decides who to talk with and how to initiate and maintain the communication.

3.3 Message-Handling Mechanisms

The negotiating agent reacts according to what message it gets. Therefore, the negotiating agent should have the functionality of handling messages for each type of multi-linked message protocol. When the agent handles a message, it should also make a decision accordingly.

Message-handling mechanisms use the Critical Path Method (CPM) for coordination of message passing among agents so that agents exploit the sequence logic in the project schedule for coordinating message passing and to ensure successful completion of distributed computation. I assume that project schedules have fixed work logic and precedence relations among activities. Therefore, the message-handling mechanisms do not allow agent to change the work logic in project schedule, but only to find a better schedule within the fixed work logic.

3.4 Summary of ABCN Methodology

I conclude that the proposed ABCN methodology facilitates the distributed coordination of project schedule changes by meeting practical challenges as follows:

- By using schedule-change options based on utility of timing, an agent can compensate other agents for disadvantageous agreements.
- By employing multi-linked negotiation protocols, agents can identify schedule conflicts, consider alternatives, and resolve schedule conflicts in a tightly coupled network of related activities
- By directing message-passing based on the CPM, agents can maintain work logic and ensure convergence of distributed coordination

In the next section, I will present a multi-agent system that implements the agent-based compensatory negotiation (ABCN) methodology for distributed coordination of project schedule chnages (DCPSC).

4. Distributed Subcontractor Agent System

To test the effectiveness of the DCPSC-based ABCN methodology, I need to build a multi-agent system, wherein all of the concerned subcontractors can reschedule a project dynamically through negotiations with help of software agents.

4.1 Requirements for DSAS

The requirements for developing the multi-agent system called distributed subcontractor agent system (DSAS) are as follows:

(1) Subcontractor agents should have the functionalities of ABCN methodology.

(2) Human subcontractors can interact with their agents to provide them with the needed information for negotiations and to get the negotiation results that needed to reschedule the project, which the objective of the DCPSC.

4.2 DSAS Architecture

According to the aforementioned requirements, I designed and implemented a multi-agent system called the distributed subcontractor agent system (DSAS). DSAS consists of multiple subcontractor agents that have functionalities of the ABCN methodology, multiple Graphic User Interfaces (GUIs) for human subcontractors to interact with their subcontractor agents, and the Agent Message Router (AMR), which routes messages between agents over the Internet.

Subcontractor agents, on the basis of the options input by the users, simulate decision-making on behalf of human subcontractors. The subcontractor agents consist of three important classes, the *Subcontractor* class, the *BookkeepingAgent* class, and the *NegotiatingAgent* class, as well as of other helper classes. The *Subcontractor* class is the body of the subcontractor agent. It invokes the *BookkeepingAgent* class when the subcontractor agent receives messages from human subcontractors. It invokes the *NegotiatingAgent* class when the subcontractor agent receives any other messages from subcontractor agents. The *NegotiatingAgent* class conducts actual compensatory negotiations. Only the *Subcontractor* class sends and receives the messages because it has the necessary name and password.

DSAS provides each human subcontractor with a GUI to interact with its subcontractor agent. The GUI has the functionality to input typed messages for the subcontractor agent to handle.

In the DSAS, the subcontractor agents and GUIs can communicate with other agents and with the GUIs. However, if the intended receiving agent does not exist at the time of communication, the communication will be lost. In fact, agents cannot be assumed to exist all the time in the distributed coordination framework for project schedule changes. Therefore, I needed an AMR that buffers and forwards messages, much like an email server. The function of the AMR is to update the addresses of registered agents and to route messages between agents.

4.3 Supporting State-of-Art Technologies

There are many computer environments in various domains for agent development. Among them, I chose to use JATLite (Java Agent Template *Lite*) (Jeon *et al.*, 2000), which was developed by the Center for Design Research (CDR) at Stanford University, to create the DSAS. JATLite is a package of programs written in the Java language that allow users to quickly create new software agents that communicate robustly over the Internet.

Currently two standards exist for the agent communication language: Knowledge Query and Manipulation Language (KQML) (Finin *et al.*, 1994) and FIPA ACL. I chose to use KQML because JATLite, which is the choice of environment for agent development, currently uses KQML for its standard agent communication language. KQML is a language and protocol for exchanging information and knowledge.

4.4 DSAS Implementation

I implemented the subcontractor agents in the Java language, which is object-oriented and portable across platforms, by extending the JATLites agent template. Consequently, subcontractor agents can run on any machine that supports JDK. The JATLite also facilitated development of DSAS, which provides GUIs and the AMR.

Since the GUI complies with the JATLite AMR, human subcontractors can download the GUIs from Internet web browsers, such as Microsoft Internet Explorer, Netscape Navigator, or Microsystems appletviewer. Thus, human subcontractors can interact with their agents without geographic restrictions.

5. Evaluations

This section demonstrates the effectiveness of the ABCN methodology for DCPSC through evaluation tests. It compares two centralized coordination methodlogies used in current practice to DCPSC-based ABCN methodology in terms of extra costs and project duration. I conducted charrette tests of the DSAS to test the effectiveness compared to manual centralized processes. I also conducted a series of experimental tests with different schedules to measure the system performance of DSAS.

5.1 Comparison Tests

I examined two methodologies of centralized coordination -- tight and loose -- to DCPSC-based ABCN methodology in terms of extra costs and project duration.

Under Tight "Iron-Fist" Centralized Coordination (TCC), the objective is to finish the project on time and the subcontractors are instructed to finish their activities before the latest finish date of each activity respectively. Under TCC, the GC can coordinate the subcontractors to finish the project on time. However, some subcontractors might experience cost overruns when their available resources differ from their resource requirements.

Under Loose "Laissez-Faire" Centralized Coordination (LCC), the objective is to match the resources available to produce a workable schedule. Under LCC, activities are finished when enough resources are provided. Without knowing subcontractors resource availability, the GC instructs subcontractors to start their activities when the preceding activities have been finished and when enough resources are available; i.e., the job is ready for it and its work can proceed unimpeded. LCC usually delays the project and some subcontractors might experience cost overruns due to delays of preceding activities as well as their resource deviations. The GC also incurs liquidated damages due to the project delay.

After comparing the results on a case scenario, I found that ABCN can find a solution that is better than or equal to any of the results from the centralized coordination methodologies. Kim (2001) shows the generalization of these evaluation results with mathematical proofs.

5.2 DSAS Charrette Tests

In order to test the effectiveness of DSAS, I used the charrette test method (Clayton et al. 1998), which the Center for Integrated Facility Engineering (CIFE) at Stanford University has used to test the effectiveness of software systems. I conducted the charrette tests to compare two processes: one was a manual centralized coordination process and another was a computer-aided ABCN process on DSAS. The propositions to be tested are whether a computerized DSAS coordination produces the lower cost solution faster than a manual centralized coordination. The task of the participants was to find a better project schedule from schedule options, which were given to participants, in terms of costs and time taken. The reason why I used the charrette test method was that it could test the effectiveness of the prototype system from the human perspective.

After two sessions of charrette tests, computerized DSAS coordination produced a lower cost solution faster than any of the manual centralized coordination efforts by two groups. The reason for finding a soution faster is that computerized DSAS coordination used software agents that could communicate rapidly, and reasoning mechanisms inside software agents made decisions automatically. If the number of subcontractors and activities in schedules grows, the power of DSAS to produce a solution quickly will be more evident. The reason for finding a lower-cost solution is that computerized DSAS coordination considered more schedule-change options than manual centralized coordination because humans bounded rationality limited them. However, I did not conclude that computer agents always perform better than humans in any case because the human rationality is required in negotiation over soft issues that cannot be reduced/translated to mathematical models and discrete numerical values.

5.3 System Performance of DSAS

I also measured DSAS system performance on the five CPM schedules from various sources, ranging three to twenty-seven activities; with zero to maximum changes. A change means that an activity does not have enough resources for following the initial schedule.

The test results showed that the number of messages exchanged among subcontractor agents does not grow exponentially with the number of activities or with the number of changes. I estimated that the worst-case computational complexity of DSAS is $O(n^3)$, where O is the approximate running time of DSAS, measured as a function of the number of activities, n, in a schedule. However, under three problems, which is common at a time in real cases, the common computational complexity is $O(n^2)$.

6. Conclusions

The paper shows a distributed approach to coordination of project schedule changes and demonstrates agent-based compensatory negotiation as a vehicle for enabling this approach. I conclude that the agent-based compensatory negotiation methodology facilitates the distributed coordination of project schedule changes by enabling subcontractors to compensate the affected subcontractors for disadvantageous agreements; by allowing subcontractors to identify and resolve schedule conflicts in a tightly coupled network of related activities; and by enabling subcontractors to maintain work logic and ensure convergence of distributed coordination. This in itself is a significant departure from prior and recent research, particularly in the area of construction project planning and scheduling, which has traditionally attempted to centralize the coordination process to enhance a project schedule.

In addition to this theoretical work, I designed and implemented a new Java-based DSAS to demonstrate the effectiveness of the DCPSC framework through a series of comparison tests, charrette tests, and measurements. This research formalizes, implements, and tests the necessary steps to help subcontractors coordinate schedule changes in order to increase the efficiency of their resource use, which in turn enhances successful completion of whole projects.

References

1. Choo, H. J., and Tommelein, I. D. (2000). "Interactive coordination of distributed work plan." *Proc., 6th Constr. Congress*, Orlando, FL, ASCE, 11-20
2. Clayton, M. J., Kunz, J. C., and Fischer, M. A. (1998). "The charrette test method." *Technical Rep.*, No. 120, CIFE, Stanford Univ., Stanford, Calif.
3. Finin, T., R. Fritzson, D. McKay, and R. McEntire (1994), "KQML as an agent communication language." *Proc. 3rd International Conference On Information and Knowledge Management*, ACM Press, New York, 456-463
4. Jeon, H., C.J. Petrie, and M.R. Cutkosky (2000), "JATLite: A Java agent infrastructure with message routing." *IEEE Internet Computing*, IEEE, 4(2), 87-96
5. Jin, Y., and Levitt, R. E. (1993). "i-AGENTS: Modeling organizational problem solving in multi-agent teams." *Intelligent Systems in Accounting, Finance and Management*, John Wiley & Sons, Ltd., New York, 2, 247-270
6. Khedro, T., Genesereth, M. R., and Teicholz, P. M. (1993). "Agent-based framework for integrated facility engineering." *Eng.ineering with Computers*, Springer Verlag, New York, 9(2), 94-107
7. Kim, K. (2001). "Distributed coordination of project schedule changes: An agent-based compensatory negotiation approach," *Technical Rep.*, No. 130, CIFE, Stanford Univ., Stanford, Calif. (http://www.stanford.edu/group/CIFE/online.publications/TR130.pdf)
8. OBrien, W., and Fischer, M. A. (2000). "Importance of capacity constraints to construction cost and schedule." *J. Constr. Eng. Manage.*, ASCE, 126(5), 366-373
9. Petrie, C., Goldmann, S., and Raquet, A. (1998). "Agent-based project management." *Technical Rep., #19981118,* Center for Design Research, Stanford Univ., Stanford, Calif.

요 약

최근 건설 프로젝트가 대형화 및 복잡화됨에 따라 다수의 하도급업체들이 참여하는 형태로 진행되고 있다. 특히 하도급업체들이 자체 자원을 운용하여 공사를 수행함에 따라, 기존의 원 도급업체가 주도하는 중앙집중식 조정방식은 실효성이 저하되고 있다. 하도급업체의 자원수급이 건설공사의 공정과 일치하지 않을 경우 해당업체의 공사가 지연될 뿐만 아니라, 다른 하도급업체의 공사들과 더 나아가 건설공사의 지연을 초래한다. 이에 따라, 관련 하도급업체들이 해당공사에 공정변경이 생길 경우에 이에 맞추어 건설공사 공정을 조정하는 새로운 건설공사 공정변경의 분산조정에 관한 연구가 필요하다. 연구자는 건설공사 공정변경의 분산조정방식 및 이를 위한 소프트웨어 에이전트를 이용한 보상협의 방법을 정의하였다. 본 연구는 현재의 건설공사 주체인 하도급업체의 자원수급의 효율성을 향상시키는 데 필요한 방법을 정의하고, 구현하고, 검증함으로서 하도급업체의 이윤추구 및 건설공사의 성공적인 수행을 함께 달성할 수 있도록 한다.

키워드: 공정관리, 프로젝트 매니지먼트, 소프트웨어 에이전트, 분산조정