International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:6, No:3, 2012

Schedule Management of an Enterprise Receiving Orders Considering Dependency between Unit Tasks of a Collaborative Project

Joseph Oh, Bo-Hyun Kim, and Jae-Yong Baek

***Abstract*—**This study suggests how an order-receiving company can avoid disclosing schedule information on unit tasks to the order-placing company when carrying out a collaborative project on the value chain in an order-oriented industry. Specifically, it suggests methods for keeping schedule information confidential, and categorizes potential situations by inter-task dependency. Lastly, an approach to select the most optimal non-disclosure method is discussed. With the methods for not disclosing work-related information suggested in the study, order-receiving companies can logically deal with political issues relating to the question of whether or not to disclose information upon the execution of a collaborative project in cooperation with an order-placing firm. Moreover, order-placing companies can monitor undistorted information, while respecting the legitimate rights of an order-receiving company. Therefore, it is fair to say that the suggestions made in this study will contribute to the smooth operation of collaborative intercompany projects.

***Keywords*—**collaborative project, dependency, schedule management, unit task.

I. Introduction

IN order-oriented industries, an order-placing company is a firm that places an order with another company, while an order-receiving company is a firm that receives an order from another business. Each order made by an order-placing company constitutes a project, and such projects are completed by order-receiving companies. An order-receiving company may also function as an order-placing company that places an order to another firm. In the study, intercompany projects are considered to be collaborative projects. The automotive parts industry is a typical example of an industry in which collaborative projects are common. In the industry, most collaborative projects involve module/parts companies placing an order for producing molds with molding companies [5].An order-placing company and its order-receiving company define the nature of the collaborative project through mutual agreement before actually executing the project. However, it is common for the two companies to display different standpoints when defining the project.

J. Oh, B. H. Kim, and J. Y. Baek are with Korea Institute of Industrial Technology, Ansansi, Gyeonggi-do, South Korea (phone: +82-31-436-8074; fax: +82-31-436-8050; e-mail: josephoh@kitech.re.kr).

In other words, order-placing companies tend to require more information on the progress of the project, while in most cases, order-receiving companies are reluctant to reveal such information, in order to ensure the security of their technology and know-how and avoid unwanted interference from their clients. For this reason, it is imperative to study work-related information disclosure methods for intercompany collaborative projects. From an order-receiving company's point of view, information on project schedule, budget, resource allocation and the like is considered sensitive. Having said that, this study focuses on schedule information only to research methods for keeping project information confidential.

This study suggests how to hide schedule information on each unit task, without distorting information on the Logic Bar Chart managed by an order-receiving firm. A unit task is a task placed on the lowest level of the WBS (Work Breakdown Structure) dividing project tasks in a hierarchical manner. Each unit task has the start time, the termination time and the task period as its property information, and may include inter-task dependency information in addition to the schedule information. Inter-task dependency involves the existence of a precedence relationship between unit tasks and mutual dependency.

This study is structured as follows. In Chapter 2, common project schedule management techniques are introduced to provide background information on Logic Bar Chart, to which the methods for not disclosing schedule information suggested by this study are applied. Chapter 3 describes three methods to avoid the disclosure of a particular unit task. Chapter 4 mentions 16 situations in which dependency between unit tasks can occur, and suggests how to select the best way of not disclosing information in each situation. Finally, Chapter 5 presents the conclusion of the study, and offers several directions for future research.

II. Project Schedule Management Technique

A. Schedule Network Diagram(SND)

A Schedule Network Diagram (SND) is a widely used technique to diagram project inter-task dependency. A project manager can use this to easily comprehend the work flow and the critical path of the project. An SND is comprised of arrows and rectangles, and is designed using either the PDM (Precedence Diagramming Method), in which unit tasks are presented on rectangles, or the ADM (Arrow Diagramming

Method), in which unit tasks are displayed on arrows [8]. In PDM, project tasks or activities are demonstrated with rectangle or circle nodes, and the arrows between nodes illustrate inter-task dependency [6]. For this reason, PDM is also known as AON (Activity-On-Node)[7]. In the study, a Logic Bar Chart, in which a PDM-type SND is combined with a Gantt Chart, is examined [2].

B. Bar Chart(BC)

A BC demonstrates unit task schedule information with bars to show the start date, the termination date, and the estimated task period. Bar charts are easy to understand, and thus are mainly used in management and administration reports. Of the bar chart types, the Gantt Chart is the most common. Generally, in a Gantt Chart, the vertical axis describes project tasks, while the horizontal axis illustrates the schedule information of the tasks. Thanks to its outstanding readability in relation to schedule information, Gantt Charts are commonly used in project management. However, they are not suitable for describing the influence of a task on another task, as they do not show inter-task dependency. In recent years, the Logic Bar Chart, which adds inter-task dependency information to a Gantt Chart, has also been used as a schedule management tool [2]. This study suggests how to keep schedule information undisclosed with a Logic Bar Chart.

III. Methods for Keeping Unit Task Information Undisclosed

When an order-placing company intends to monitor not only the schedule information demonstrated in a Gantt Chart, but also the work flow based on dependency information, the order-receiving firm will need to find a way to maintain the security of schedule information. This study suggests how to keep particular unit task schedule information undisclosed on a Logic Bar Chart involving dependency information. As demonstrated in Fig. 1, there are roughly three methods that can be used to keep unit task information confidential.

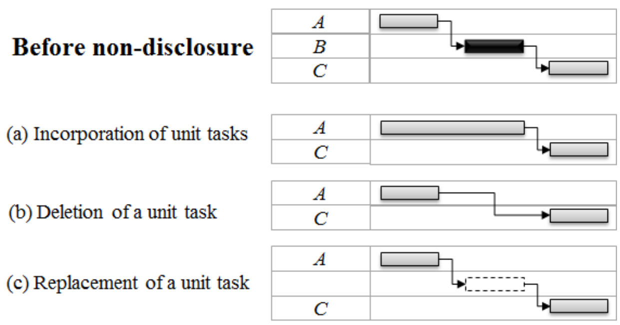

Fig. 1 Non-disclosure methods of a unit task

A. Incorporation of Unit Tasks

Incorporation of unit tasks enables the task period of a unit task for non-disclosure to be merged into the task period of another unit task connected to the unit task with dependency between them. As a result, the schedule of the unit task containing sensitive information remains undisclosed. When there are several unit tasks with dependency, the unit task for non-disclosure shall be incorporated into the one with the highest dependency level. Fig. 1-(a) illustrates an example of unit task incorporation. In this example, unit task B is merged into Unit task A to remain undisclosed. Accordingly, the task period for unit Task A is extended in proportion to the task period for unit task B.

Equation (1) is a numerical expression of unit task incorporation, while equation (2) is the description of the unit task incorporation featured in Fig. 1-(a) in accordance with equation (1).

$$Mer: X, Y \rightarrow Z, \text{ if } X, Y \text{ have a dependency} \quad (1)$$

$$Mer(A, B) = A \quad (2)$$

B. Deletion of a Unit Task

Deletion of a unit task is used to remove a unit task for non-disclosure from the Logic Bar Chart. The dependency related to the removed unit task is also deleted, and another dependency replaces the deleted unit task. However, from an order-placing company's point of view, as a unit task is removed, it may seem that there is no progress in the project. In other words, the order-placing company may think that the order-receiving company has entirely ignored the project, and make a complaint about it. Therefore, this method shall be used as the second best policy following the unit task incorporation method.

Fig. 1-(b) illustrates an example of unit task deletion. In the example, unit task B has been deleted so as to be undisclosed to the order-placing company, and a new dependency status has been established between unit task A and B.

When a unit task to remain undisclosed exists in the critical path, the unit task cannot be removed, because the entire period of the critical path will be reduced if a unit task in the critical path is deleted. As it is not permitted to provide distorted information to an order-placing company, this method shall be applied only to unit tasks not included in the critical path.

In Fig. 1, unit task A, B and C are connected to each other like a chain through dependency among them. Unit task A and B, or unit task B and, have direct dependency between them. Accordingly, unit task A and C have an indirect dependency between them. If unit task B is deleted by the unit task deletion method, unit task A and C come to have direct dependency between them. When a new dependency status cannot be created if a unit task to remain undisclosed is deleted, the unit task must not be deleted.

Equation (3) illustrates unit task deletion with a numerical formula, while equation (4) shows the unit task deletion method applied in Fig. 1-(b) in accordance with equation (3).

$$Del: X \rightarrow \varnothing, \text{ if } X \text{ is not on the Critical Path} \quad (3)$$

$$Del(B) = \varnothing \tag{4}$$

C. Replacement of a Unit Task

With unit task replacement, a unit task that must remain undisclosed is replaced with a dummy unit task. The schedule information of the unit task is disclosed, while the details of the task remain confidential. This method is applicable even when the unit task exists on the critical path. However, it is likely that the order-placing company will make a complaint in this case, because the period of the dummy unit task is still disclosed. Accordingly, of the three methods for non-disclosure, this method has the biggest possibility of resulting in intervention by an order-placing company. Therefore, it is advised to use this method as second best policy, if possible. Fig. 1-(c) illustrates an example of unit task replacement. In the example, the period of unit task B is displayed on the dotted line, but the period is not linked to a particular unit task.

Equation (5) demonstrates unit task replacement with a numerical formula, while equation (6) shows the unit task replacement method applied to Fig. 1-(c) in accordance with equation (5).

$$Rep: X \rightarrow \delta, \ \ \delta \text{ is a dummy task} \tag{5}$$

$$Rep(B) = \delta \tag{6}$$

IV. Schedule Management in Consideration of Dependency

A. Dependencies

Generally, there are four types of dependency statuses depending on the start point and the finish point of the unit task. (See Fig. 2) Fig. 2-(a) demonstrates the most common type, the FS (Finish to Start) relationship, in which unit task B shall start upon the completion of unit task A. Fig. 2-(b) shows the FF (Finish to Finish) relationship, in which unit task B is completed only when unit task A is finished. Fig. 2-(c) illustrates the SF (Start to Finish) relationship, which is used when there is a unit task to be completed in advance to meet an important future schedule. In this case, unit task B is a pre-arranged task, secondary task, or ancillary task to unit task A. Fig. 2-(d) describes the SS(Start to Start) relationship, in which unit task B can be started only when unit task A is commenced. In the FF relationship or the SS relationship, unit tasks are carried out in a parallel manner [1-5].

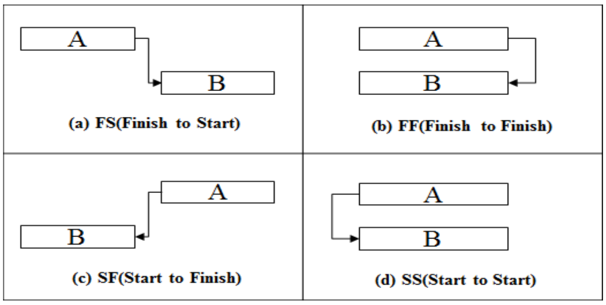

Fig. 1 Dependencies

Equation (7) to (10) in Fig. 2 illustrate the dependency status of (a) to (d), respectively.

$$A \xrightarrow{FS} B \tag{7}$$

$$A \xrightarrow{FF} B \tag{8}$$

$$A \xrightarrow{SF} B \tag{9}$$

$$A \xrightarrow{SS} B \tag{10}$$

B. Categorization of situations in consideration of inter-task dependency

In the study, when keeping a particular unit task undisclosed, the potential situations are sorted by dependency. Depending on the dependency status at the start/finish point of the unit task to remain undisclosed, one of 16 situations can develop (See Fig. 3). Task B is a unit task to remain undisclosed, and task A and C are unit tasks connected to task B, with dependency at the start point and the finish point of task B, respectively. With A-B dependency or B-C dependency, there are four possible situations. With A-B dependency FS, SF, SS(A→B), or SS(A←B) can happen, while with B-D dependency FS, SF, FF(B←C), or FF(B→C) can happen. Consequently, depending on the dependency of the start/finish point of unit task B, one of 16 situations can occur.

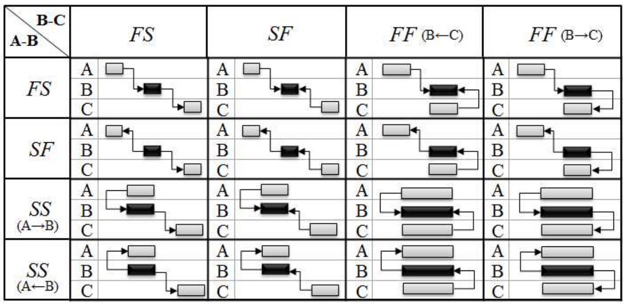

Fig. 2 Categorization of situations based on dependency

C. Application of the Schedule Non-disclosure Scheme in Consideration of Given Situations

An order-receiving company shall apply the schedule non-disclosure scheme in consideration of the characteristics and the dependency status of the unit task. This study suggests how to keep schedule information undisclosed in the 16 situations demonstrated in Fig. 3. Fig. 4 illustrates the information non-disclosure scheme that can be applied in the 16 situations. All three of the non-disclosure methods can only be applied in certain situations. The unit task replacement method can be used in every situation, while the unit task deletion method is selectively applicable. With the unit task incorporation method, both the incorporation of unit task A into B and that of unit task B into C are applicable. In some cases, there is another condition to be met to merge a unit task to another.

When the A-B dependency is FS and the B-C dependency is FS, all three of the non-disclosure methods are applicable. With the unit task incorporation method, it is possible to either combine A and B or B and C. As a result, there are four methods that can be used to keep unit task B undisclosed. Firstly, by merging A to B, the combination obtains FS dependency with C. Secondly, by combining B with C, the combination acquires FS dependency with A. Thirdly, unit task B can be deleted. In this case, the dependency status of the start/finish point of unit task B is also deleted, but the new dependency status connecting A to B replaces B. Lastly, unit task B can be replaced with a dummy unit task, leaving the dependency status unchanged.

When the A-B dependency is SS(A←B) and the B-C dependency is FF(B→C), the unit task incorporation method or the unit task replacement methods can be used. It should be noted that the incorporation method is applicable only when certain conditions are met. For merging task A to B, the finish point of task A should be earlier than that of task B, unlike other unit task incorporation cases. This condition maintains the dependency between task B and C. If the finish point of task A is later than that of B, the B-C dependency will be affected by the incorporation of A and B. Therefore, it is necessary to consider the finish point of task A and B so that task A can be merged to task without affecting the B-C dependency. For combining task B with C, the start point of task B shall be earlier than that of task C. This is to prevent the incorporation of task B and C from affecting the A-B dependency.

When the A-B dependency is SS(A←B) and the B-C dependency is FF(B→C), the unit task deletion method cannot be used, because the new dependency status to replace unit task B cannot be defined. To set the new dependency status, the dependency of the start/finish point of unit task B shall be in the same direction. In this case, the dependency has conflicting directions, and thus the new dependency status cannot be defined. Consequently, the unit task deletion method is inapplicable.

D.Selection of the Optimal Method for Keeping Schedule Information Undisclosed

After examining several methods, this study suggests how to select the optimal method that can be used by order-receiving companies to keep information undisclosed based on the given situation. Table 1 demonstrates the method selection procedure, which is comprised of four steps.

In Step 1, find the applicable non-disclosure methods by using the non-disclosure method table in Fig. 4. In Step 2, check whether the unit task deletion method is applicable or not. If the unit task to remain undisclosed exists on the critical path, delete the unit task deletion method from the available options. In Step 3, the non-disclosure method is selected according to the priority order. The unit task replacement method is the least preferred option, as it can lead to a direct complaint from the client on a task with disclosed schedule information. With the unit task deletion method, the order-placing company may make a complaint on the seemingly inactive period due to the deleted task. However, the schedule information of undisclosed unit tasks is not explicitly displayed, and thus this method is the second most preferred option. The unit task incorporation method is the most preferred, as it keeps schedule information undisclosed, and thus the order-placing company is unlikely to make a complaint on schedule issues. If there are a number of cases in which the unit task incorporation method is applicable, Step 4 shall be followed. When there are several options for unit task incorporation, the task with the highest dependency level shall be used for incorporation.

TABLE I
PROCEDURE FOR SELECTING THE OPTIMAL WAY TO KEEP SCHEDULE INFORMATION UNDISCLOSED

Step	Description
1	Identify non-disclosure methods applicable to the given situation from the non-disclosure method table
2	Delete the unit task deletion method if the task to remain undisclosed exists on the critical path
3	Select the method from among remaining options in the following priority order: 1st - unit task incorporation, 2nd – unit task deletion, 3rd - unit task replacement
4	When there are several unit tasks that can be incorporated, select the unit task with the higher dependency level

V.CONCLUSION AND FUTURE WORK

The study has covered various approaches to keeping the schedule information of unit tasks of an order-receiving company undisclosed on a Logic Bar Chart when carrying out a collaborative project on an order-oriented value chain. Specifically, three methods for preventing the disclosure of schedule information have been suggested, and 16 situations that can occur depending on inter-task dependency have been mentioned. Finally, the study proposed how to select the best non-disclosure method in 16 situations. With the methods for preventing the disclosure of work information discussed in this study, order-receiving companies can logically deal with business issues on information non-disclosure when executing a collaborative project with an order-placing company. In addition, order-placing companies can monitor undistorted information, without disturbing the legitimate rights of their order-receiving companies. For these reasons, the suggestions made in the study are useful for the smooth operation of intercompany collaborative projects.

Future studies will examine the dependency such as lag and lead, etc that can occur upon the execution of a project, see if there are other applicable types in addition to the three unit task non-disclosure methods suggested in the study, and research approaches to applying non-disclosure schemes to cases in which the dependency status is structured in a parallel or complex manner, in contrast to the serial connection examined in this study.

B-C / A-B	*FS*	*SF*	*FF* (B←C)	*FF* (B→C)
FS	• $Mer(A,B) \xrightarrow{FS} C$ • $A \xrightarrow{FS} Mer(B,C)$ • $A \xrightarrow[Del(B)]{FS} C$ • $A \xrightarrow{FS} Rep(B) \xrightarrow{FS} C$	• $Mer(A,B) \xleftarrow{SF} C$ • $A \xrightarrow{FS} Mer(B,C)$ • $A \xrightarrow{FS} Rep(B) \xleftarrow{SF} C$	• $Mer(A,B) \xleftarrow{FF} C$ • $A \xrightarrow{FS} Mer(B,C)$ $, B.start \prec C.start$ • $A \xrightarrow{FS} Rep(B) \xleftarrow{FF} C$	• $Mer(A,B) \xrightarrow{FF} C$ • $A \xrightarrow{FS} Mer(B,C)$ $, B.start \prec C.start$ • $A \xrightarrow[Del(B)]{FF} C$ • $A \xrightarrow{FS} Rep(B) \xrightarrow{FF} C$
SF	• $Mer(A,B) \xrightarrow{FS} C$ • $A \xleftarrow{SF} Mer(B,C)$ • $A \xleftarrow{SF} Rep(B) \xrightarrow{FS} C$	• $Mer(A,B) \xleftarrow{SF} C$ • $A \xleftarrow{SF} Mer(B,C)$ $, B.start \prec C.start$ • $A \xleftarrow[Del(B)]{SF} C$ • $A \xleftarrow{SF} Rep(B) \xleftarrow{SF} C$	• $Mer(A,B) \xleftarrow{FF} C$ • $A \xleftarrow{SF} Mer(B,C)$ $, B.start \prec C.start$ • $A \xleftarrow[Del(B)]{FF} C$ • $A \xleftarrow{SF} Rep(B) \xleftarrow{FF} C$	• $Mer(A,B) \xrightarrow{FF} C$ • $A \xleftarrow{SF} Mer(B,C)$ $, B.start \prec C.start$ • $A \xleftarrow{SF} Rep(B) \xrightarrow{FF} C$
SS (A→B)	• $Mer(A,B) \xrightarrow{FS} C$ $, A.finish \prec B.finish$ • $A \xrightarrow{SS} Mer(B,C)$ • $A \xrightarrow[Del(B)]{SS} C$ • $A \xrightarrow{SS} Rep(B) \xrightarrow{FS} C$	• $Mer(A,B) \xleftarrow{SF} C$ $, A.finish \prec B.finish$ • $A \xrightarrow{SS} Mer(B,C)$ • $A \xrightarrow{SS} Rep(B) \xleftarrow{SF} C$	• $Mer(A,B) \xleftarrow{FF} C$ $, A.finish \prec B.finish$ • $A \xrightarrow{SS} Mer(B,C)$ $, B.start \prec C.start$ • $A \xrightarrow{SS} Rep(B) \xleftarrow{FF} C$	• $Mer(A,B) \xrightarrow{FF} C$ $, A.finish \prec B.finish$ • $A \xrightarrow{SS} Mer(B,C)$ $, B.start \prec C.start$ • $A \xrightarrow[Del(B)]{SF} C$ • $A \xrightarrow{SS} Rep(B) \xrightarrow{FF} C$
SS (A←B)	• $Mer(A,B) \xrightarrow{FS} C$ $, A.finish \prec B.finish$ • $A \xleftarrow{SS} Mer(B,C)$ • $A \xleftarrow{SS} Rep(B) \xrightarrow{FS} C$	• $Mer(A,B) \xleftarrow{SF} C$ $, A.finish \prec B.finish$ • $A \xleftarrow{SS} Mer(B,C)$ • $A \xleftarrow[Del(B)]{SS} C$ • $A \xleftarrow{SS} Rep(B) \xleftarrow{SF} C$	• $Mer(A,B) \xleftarrow{FF} C$ $, A.finish \prec B.finish$ • $A \xleftarrow{SS} Mer(B,C)$ $, B.start \prec C.start$ • $A \xleftarrow[Del(B)]{FS} C$ • $A \xleftarrow{SS} Rep(B) \xleftarrow{FF} C$	• $Mer(A,B) \xrightarrow{FF} C$ $, A.finish \prec B.finish$ • $A \xleftarrow{SS} Mer(B,C)$ $, B.start \prec C.start$ • $A \xleftarrow{SS} Rep(B) \xrightarrow{FF} C$

Fig. 3 Application of a non-disclosure scheme depending on the given situation

REFERENCES

[1] Cheng, J., Gruninger, M., Sriram, R., and Law, K. "Process Specification Language for project scheduling information exchange," International Journal of IT in Architecture, Engineering and Construction, Vol. 1, pp. 307-328, 2003.

[2] PMI, A Guide to the Project Management Body of Knowledge, the 3rd ed., Project Management Institue, Inc, PA, 2004.

[3] Meredith, J. R., and S. J. Mantel, Project Management: A Managerial Approach, 7th ed., John Wiley & Sons, NY, 2009.

[4] Kerzner, H., Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 9th ed., John Wiley & Sons, NJ, 2006.

[5] Chu, S., and B. Cesnik, "Improving Clinical Pathway Design: Lessons Learned From a Computerised Prototype ," International Journal of Medical Informatics, Vol. 51, No. 1, pp. 1-11, 1998.

[6] Earl T. Glenwright Jr., Let's Scrap the Precedence Diagramming Method, 2004 AACE International Transactions, 2004.

[7] Lu, M., and H. Li, "Resource-Activity Critical-Path Method for Construction Planning," Journal of Construction Engineering and Management, Vol. 129, No. 4, pp. 412-420, 2003.

[8] Galloway, P. D., "Comparative Study of University Courses on Critical-Path Method Scheduling," Journal of Construction Engineering and Management, Vol. 132, No. 7, pp. 712-722, 2006.