

Module-Based Construction Schedule Administration for Public Infrastructure Agencies

R. J. Dzung¹; W. C. Wang²; and H. P. Tserng³

Abstract: Infrastructure projects such as the construction of expressways are often distributed over different regions and must therefore be divided into several tendering packages. The public agencies that administrate such projects manage project packages that must be integrated. All project packages may involve similar work. However, they are carried out by different contractors each with their own scheduling practices. Standardization provides a foundation for more efficient and effective schedule integration, but cannot easily be enforced without objections from contractors. This paper presents a three-stage standardization implementation framework using modularization. A set of network modules that involve normal, repetitive, cyclic, and merging activities was developed for expressway projects. Two computer systems were developed to help contractors use these modules to create schedules, and to help the owner review schedules submitted by contractors. Experiments and a survey were also carried out to validate the proposed framework, demonstrating a significant amount of time saving and errors reduced by using network modules.

DOI: 10.1061/(ASCE)0733-9364(2004)130:1(5)

CE Database subject headings: Infrastructure; Project management; Scheduling; Government agencies; Computer applications.

Introduction

The scope of infrastructure varies from one country to another, but typically involving projects in transportation, communication, services utilities, residential buildings, water conservation, flood prevention, or energy resource development. The planning, design, construction, operation, and maintenance of the infrastructure are usually performed or/and administrated by national or local public agencies. Good construction schedule administration following the award of projects is essential in meeting the publicly announced project completion date. Construction schedule administration includes the following main tasks:

- Defining the required format and content of the schedules submitted by awarded contractors,
- Reviewing and approving the schedules,
- Periodically monitoring updated schedules,
- Generating master schedules by integrating related projects, and
- Reviewing and approving schedule changes due to change orders.

Infrastructure projects tend to be larger than other public and private projects. An expressway project, for example, is often

divided into subprojects and requires several tendering packages based on factors such as location, risk diversification, work interface, the average working and financial capability of the available contractors, and the balance of excavation and filling of soil. The entire project is thus typically carried out by several general contractors who may perform very similar work in a similar time frame at different locations.

The public agencies, or their assisting professional construction management teams, manage similar projects that are carried out by different general contractors but must be integrated. Such integration requires much experience-based human interpretation and cannot be automated without proper standardization since each contractor follows his own scheduling practices, each with different activity names, level of detail, and computer tools.

Public agencies understand that proper standardization results in more efficient and effective schedule administration, but such standardization cannot easily be implemented. Agencies always focus on their primary responsibility—making sure that projects are completed on time—and devote only a little time to establishing publicly acceptable standards. They are also reluctant to enforce rigid standardization (e.g., the use of standardized activity codes or certain software tools) that might promote resistance or any other adverse response from the contractors. The agencies also seek to avoid any decision that could be interpreted as favoring any contractor or software company.

This paper presents the extended results of a funded project in which, over a period of three years, modules rather than rigid standards were employed to promote a public agency in developing a foundation for improving their schedule administration. The tangible outcomes of this project include the following:

- A general conceptual model to implement standardization, and help public agencies move toward automating schedule administration,
- A set of network modules that cover road, bridge, and tunnel construction,
- Computer software, called *Network Builder Assistant* (NBA), which allows users to generate schedules readable by

¹Associate Professor, Dept. of Civil Engineering, National Chiao-Tung Univ., 1001 Ta-Hsieu Rd., Hsinchu, Taiwan 30050. E-mail: rjdzeng@mail.nctu.edu.tw

²Assistant Professor, Dept. of Civil Engineering, National Chiao-Tung Univ., 1001 Ta-Hsieu Rd., Hsinchu, Taiwan 30050.

³Associate Professor, Dept. of Civil Engineering, National Taiwan Univ., No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 10600.

Note. Discussion open until July 1, 2004. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on August 17, 2001; approved on September 25, 2002. This paper is part of the *Journal of Construction Engineering and Management*, Vol. 130, No. 1, February 1, 2004. ©ASCE, ISSN 0733-9364/2004/1-5-14/\$18.00.

Microsoft Project (MS Project), Open Project Planner (OpenPlan), and Primavera Project Planner (P3), by simply selecting the applicable modules,

- Computer software, called *Network Review Assistant* (NRA), which helps agencies review schedules submitted by contractors, and
- Results of two experiments and a survey that show the benefits gained by both contractors and agencies under the proposed schedule administration framework.

Literature Review

In the area of infrastructure project management, much research has been devoted to improving project delivery methods, financial and cost analyses, and maintenance during operation; e.g., Miller (1997), Arditi and Messiha (1996), De La Garza et al. (1998), and Peña-Mora and Tamaki (2001). Little research has addressed the administration of infrastructure construction schedules that are homogeneous in terms of the kind of work involved but represented heterogeneously because different contractors are involved.

Schedule administration for a public agency involves managing schedule generation, review, monitoring, control, and storage. Most scheduling textbooks, such as Potts (1995), reviewed various kinds of scheduling techniques (e.g., linked bar charts, lines of balance, location-time diagrams, network analysis) and the coding system for integrating multiple projects. The textbooks assume that the project owner has enforced a unified coding system without objection from contractors. However, this situation might not hold for a public agency that tries to avoid doing anything that might stimulate any protest from contractors and hinder the progress of the project. Many artificial intelligence planners, such as *BUILDER* (Cherneff et al. 1991) and *HISCHEd* (Shaked and Warszawski 1995), have been developed over the last decade. The knowledge bases behind these planners were intended for schedule generation but could also be used for schedule review. However, these planners were primarily applied to building and plant construction—not expressway construction. De La Garza and Ibbs (1990) also examined methods for eliciting expert knowledge in the construction scheduling domain. Their proposed methods of eliciting and representing knowledge were helpful in developing the network modules presented here. *Case-Plan* (Dzeng and Tommelein 1997) is a planner that stores project schedules with multiple indexes, based on which similar cases can be searched for, and referred to, in making a new schedule. A similar storage and retrieval scheme was also adopted for the automatic review in this research.

Schedule Administration Automation

Background

The Taiwan Area National Expressway Engineering Bureau (TANEEB), an infrastructure administration agency, initially funded this research. TANEEB is a subdivision of the Department of Transportation and Communication in Taiwan and is primarily responsible for the administration of newly developed national expressways. In the 2001 fiscal year, TANEEB carried out expressway projects of approximately \$16,940 (35.82% of the transportation and communication infrastructure spending, or 15.87% of national total infrastructure spending).

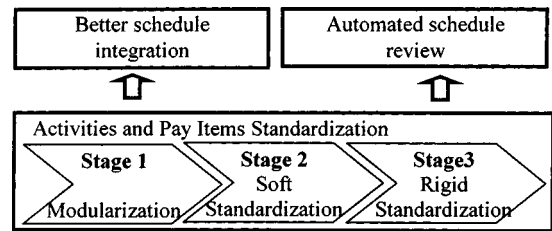


Fig. 1. Three-stage implementation of activity standardization

The current total budget/contract volume of TANEEB's highway and expressway projects in construction is about \$12,472. Bridges and tunnels contribute over half of the total Taiwanese budget for highways and expressways due to the country's geography. Inspecting the projects in the construction phase reveals that many subprojects are tendered within each major project. Many general contractors (ranging from 17 to 68) are involved in each major project. They break down work differently; name activities differently, and include a different level of detail in their schedules. However, each contractor performs work of a similar scope, involving constructing roads and bridges, with or without tunnels. Construction methods are also similar and are applied repetitively in each project and cross projects.

A unified set of construction activity names and codes does not exist, even though work is similar and repetitive. Different activity naming and coding leads to poor communication among the contractors, and between the owner's Architect/Engineer (A/E) representatives and the contractors. Under such conditions, automated integration of schedules is not feasible.

TANEEB demands that submitted schedules be presented as a Precedence Diagramming Network and in electronic form (e.g., as a P3 file). The information in most initially submitted schedules is usually incomplete and misallocated. For example, many contractors do not include associated pay items under each activity, which is required by the owner so that the earned value can be calculated based on actual work progress. Several further verbal communications are often required before a schedule is finally approved.

Consequently, TANEEB faces the challenging task of integrating and quickly reviewing schedules submitted with different formats and incomplete information, usually including over 1,000 activities along with their pay items and the major equipment involved. In practice, the owner can only check samples of so much of nonstandardized information. TANEEB strongly desires for activity standardization to support more efficient communication and automated schedule integration. However, as a public agency, TANEEB is reluctant to enforce rigid standardization that might stimulate protest and hinder project progress, or favor any contractor or software company. The writers suggest that this kind of dilemma is not only faced by TANEEB, but also by many agencies that administrate public work.

Standardization Transition Framework

TANEEB's current primary schedule administration goal is to establish a set of standard codes for construction activities and contract pay items, and to persuade successful and perspective contractors to use these codes. The use of these standard codes would facilitate the integration of related schedules into master schedules and reviews of consistency between earned value schedules and progress schedules. Fig. 1 shows that the standardization of activities and pay items provides a foundation for more efficient

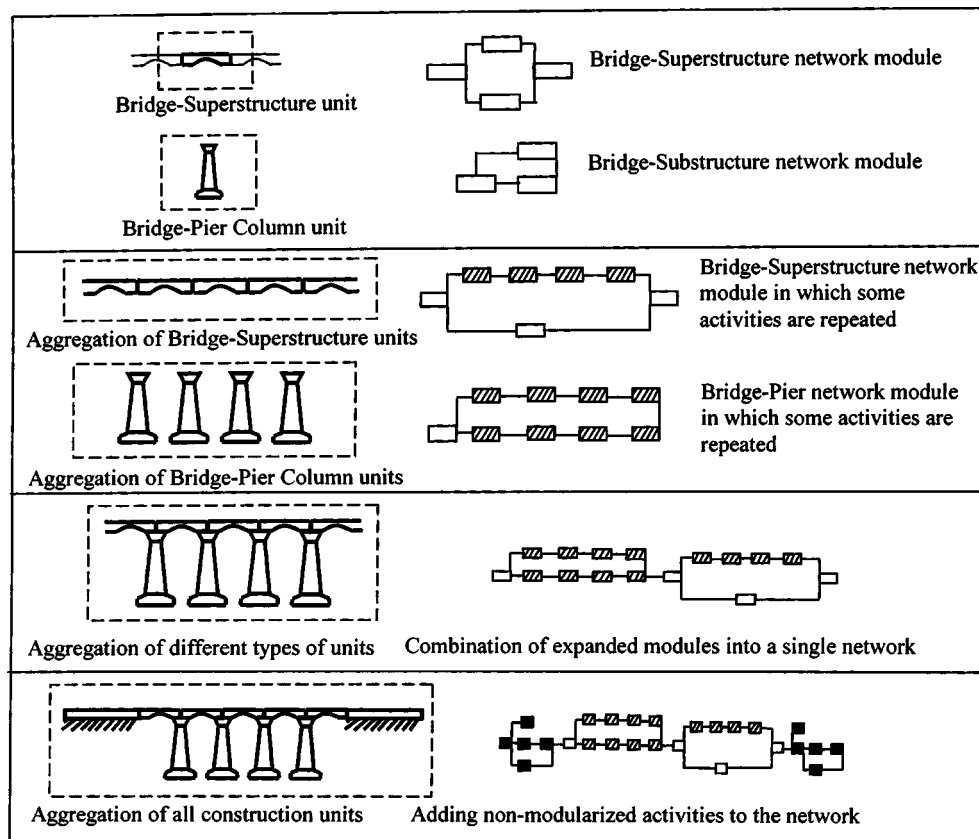


Fig. 2. Mapping between construction units and network modules

and effective schedule integration and automated schedule review. However, our interviews revealed strongly pessimistic and uncooperative attitudes among contractors toward the standardization due to the expected requirement for a large number of schedule inputs.

This work proposes a three-stage implementation rather than directly enforcing a rigid single-stage standardization implementation. The three-stage implementation involves modularization, soft standardization, and rigid standardization to increase contractors' acceptance of the standardization (Fig. 1).

The modularization stage involves a set of modularized activity networks, a software tool to help contractors use these modules to generate schedules, and a further software tool to help TANEEB's staff review schedules. The primary goal of the tool is to motivate users to use these modules by saving time and effort. The soft standardization stage encourages successful contractors to use standard activities by reviewing their earned value and payment schedules faster. Adding the capability to comply to standard codes and present schedules electronically as part of the criteria for evaluating bid proposals is also suggested. The rigid standardization stage seeks to add the use of standard codes and electronic schedule representation as part of the contract provisions. Our goals can be achieved with a smoother transition through these three stages. This paper focuses primarily on the modularization stage and describes in detail the developed modularized networks and facilitative tool. Automated schedule review based on these modularized networks is only briefly discussed due to space limitations.

Definition of Network Module

The term, "activity network module" is similar to the term, "sub-network" in general or "fragmet" in P3, but with the much stronger purpose of motivating contractors to employ it to create the main part of a schedule. Each piece of project work may be broken down into a hierarchy of construction units.

Each type of construction unit is associated with at least one network module that describes how the unit can be constructed. The module may be expanded to describe the aggregation of units of the same type by repeating some of the internal activities. A schedule for a project with different types of units can be generated by linking activities of different expanded modules. A complete schedule typically requires adding some nonmodularized activities such as mobilization and utilities reallocation specific to individual projects. Fig. 2 illustrates an example of the mapping between construction units and network modules for a bridge project that consists of a superstructure and piers.

Each *module* includes the following attributes: *name*, *activities*, activities' *precedence relationships* and *lead times*, *recurring times*, *unit section* (describing the location of the associated construction unit; e.g., "Dashu County Overpass 170–175 km"), and *unit direction* (specifying the associated lane direction; e.g., north).

Each *activity* includes the following attributes: *standard code* (uniquely identifying the class of activity), *counter* (uniquely identifying each activity in the same class; required if the schedule is to be readable by commercial scheduling tools that require a unique code for each activity), *name*, *type* (explained later),

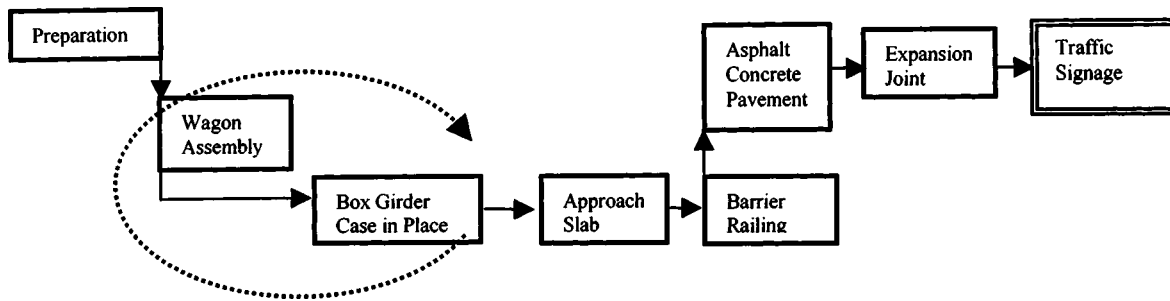


Fig. 3. "Advancing shoring method" module

duration, duration estimation *reference* (formulas, factors, or other experience-oriented duration estimation information), associated *pay items*, and *subactivities* (describing the scope of the work under the activity).

Each pay item includes the following attributes: standard *code* (uniquely identifying the class of pay item), *counter* (uniquely identifying each pay item of the same class), *quantity*, and *unit* (e.g., ton).

The primary purpose of including standard codes in the module is to increase the contractors' acceptance of using the codes as these codes require no human input once they have been encoded. The inclusion of major associated pay items in each activity motivates the contractors to complete all of the associated pay items by just adding situation-based items. The inclusion of subactivities in an activity provides a description of activity at the level of detail preferred by contractors, which is more detailed than that sought by the owner.

Normal, Repetitive, Cyclic, and Merging Activities

A module includes the attribute, *recurring times*, and an activity includes the attribute, *type* to accommodate the repetitive nature of expressway construction and to reduce the need for contractors' further input when they use modules.

An activity may fall under one of the following four types: *normal*, *repetitive*, *cyclic*, and *merging*. A *normal* activity describes the work that is performed once as a continuous process when the module is used. For example, activity "Excavation" in the "Earthwork" module is a *normal* activity, i.e., the construction schedule for a section of road usually involves only a single "Excavation" activity.

A *repetitive* activity describes work that is performed discretely, section by section or unit by unit. The number of repetitions is specified by the value of the *recurring times* attribute of the module. For example, the "Box Girder Segment" activity in the "Balanced Cantilever Method" module is a *repetitive* activity. That is, when the Balanced Cantilever Method is used to construct a bridge, the box girder segments are erected and extended one by one. Such a schedule usually involves repetitive box girder activities.

A *cyclic* activity describes work that with other *cyclic* activities as a cycle is discretely (section by section or unit by unit) performed. The number of cycles is also specified by the *recurring times* attribute. For example, in Fig. 3, both "Wagon Assembly" and "Box Girder Case in Place" activities in the "Advancing Shoring Method" module are *cyclic* activities (shown by a circular dashed arrow); i.e., the Advancing Shoring Method requires "Preparation," several cycles of "Wagon Assembly" and "Box Girder Case in Place," and then "Approach Slab," etc.

A *merging* activity describes work required by several modules but typically performed as a whole and represented as a single activity. For example, the "Traffic Signage" activity in the "Advancing Shoring Method" module (Fig. 3) is a *merging* activity; i.e., even if Advancing Shoring Method is used more than once (e.g., because the project involves two bridges), the contractor may perform most activities separate while treating the installation of traffic signage as a single continuous activity.

Within a project, duplicating is the use of a module more than once. Expanding is to recur or repeat some activities of a module for a certain number of times as specified in the *recurring times* attribute. Figs. 4–6 describe *normal*, *repetitive*, *cyclic*, and *merging* activities in both duplicating and expanding situations. In the figures, "R" represents a *repetitive* activity; "C" a *cyclic* activity; "M" a *merging* activity, and the other letters *normal* activities.

In Fig. 4, the left-hand side describes a situation in which a module with a *repetitive* activity is used twice but does not recur (*recurring times*=0). All activities, including the *repetitive* activity, are duplicated once (performed twice). The right-hand side describes a situation in which a module with a *repetitive* activity is used once and recurs *n* times (*recurring times*=*n*). The *repetitive* activity is performed *n* times sequentially. Fig. 5 describes a situation in which a module with a group of *cyclic* activities (C1, C2, and C3) is used once and recurs *n* times. The group cycles *n* times (i.e., the entire group is duplicated *n*-1 times) sequentially, with its activity relationships unchanged. Fig. 6 describes a situation in which a module with a *merging* activity is used *n* times

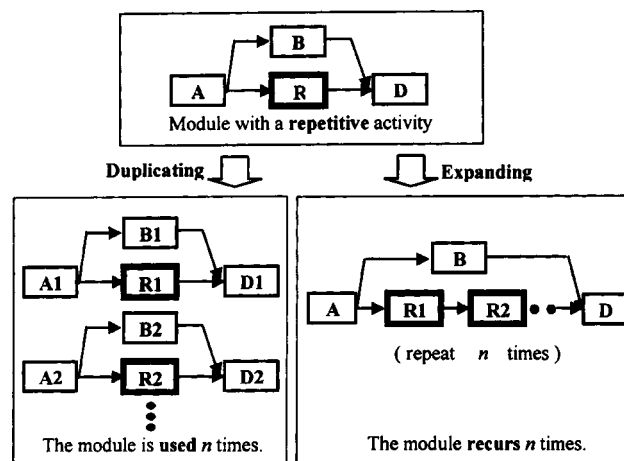


Fig. 4. Duplicating versus expanding a module (normal and repetitive activities)

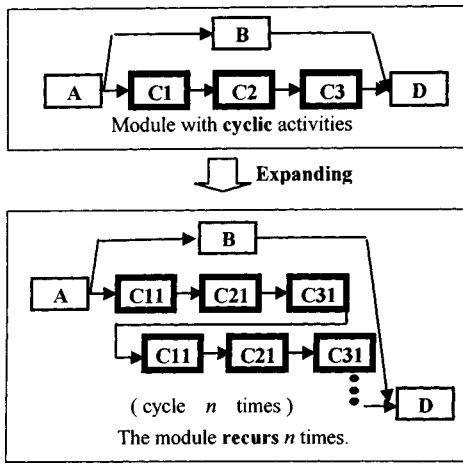


Fig. 5. Expanding a module (cyclic activities)

but does not recur. All activities except for M are performed n times. The precedence relationships with activity M remain after the activities are duplicated.

Development of Modules

The modules developed here are not intended to cover all activities of expressway construction to avoid too many available selections and necessary changes when the modules are used. Only commonly and repeatably performed activities that change little from project to project are covered. Developing the modules involved determining WBS (work breakdown structure), standard codes, and network representation, including the level of detail, activities and their relationships, and major associated pay items. The initial network modules were developed according to a literature review and realistic schedules collected from selected winning contractors. The modules were discussed, revised, and finalized through interviews with experienced schedulers and through formal meetings with representatives of the owner, A/E, contractors, scheduling software companies, and academics. This phase of development led to the following learning:

- Making subjective decisions is easy, but reaching consensus among representatives is difficult. More than eight months

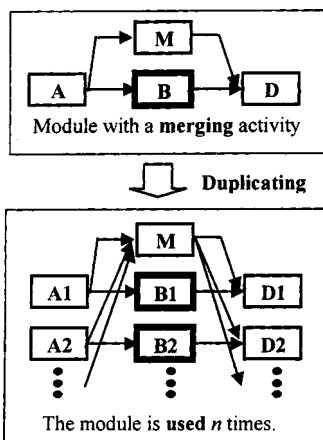


Fig. 6. Duplicating a module (merging activity)

were required to reach initial consensus, but changes to the modules were still being made occasionally during the following year.

- A consensus on the scheme for standard codes was easily reached. Consensus on the level of detail of the representation was the most difficult to secure as everyone's needs were different. For this reason, subactivities (at another level of detail) were included in the modules. Consensus on activities to be included was easily obtained once the desired level of detail in the modules was determined.
- Pay items to be included were determined without a long discussion, not because a consensus was reached, but because everyone agreed that a consensus was impossible to reach and the association of activities and pay items varied across projects. Accordingly, only commonly used pay items were included under each activity and contractors were expected to add situation-based items when they used the modules.

Table 1 summarizes the developed modules. Interested readers may find a complete list in (Dzeng 2000). Table 2 further details an example module, the Advancing Shoring Method.

Applications of Modules

Fig. 7 shows current applications for our network modules, which include Network Builder Assistant (NBA) and Network Review Assistant (NRA). Arrows between diagrams represent the main direction of information flow. NBA enables a system manager to maintain and manage network modules and allows users to generate basic schedules readable by MS Project, OpenPlan, and P3. After the schedule is created, users may continue to work on the schedule using the scheduling software.

NRA uses a critique rule library and case library to help reviewers to review schedules. The schedules should employ the proposed standard codes for activities and pay items, and can be created using NBA or any of the aforementioned scheduling software. The NRA output is a report of listed messages pointing schedule data (e.g., activity names and durations) that may contain potential errors and thus require the users' attention. Information on an activity's average and deviation in duration based on existing similar cases in the system library, and common associated pay items and relationships based on the standard modules is also given as a reference to support the correction of errors.

Network Builder Assistant

The Network Builder Assistant (NBA) was developed using *Microsoft Visual Basic for Applications* and *Access 2000*. It consists of two subsystems (Fig. 7), the *Module Library Management System* (MLMS) and the *Modularized Schedule Builder System* (MSBS). MLMS provides a graphical interface for managing the proposed network modules and includes information about all attributes discussed above. It helped the users create new modules and delete existing modules, and edit existing modules, activities, activity relationships, pay items, subactivities.

MSBS provides users a graphical interface to select and edit modules, and use them to generate a schedule. Its primary functions are as follows:

- Select appropriate modules from a list of modules currently available in the library (Fig. 8),
- Edit selected modules at the module level (e.g., change *recurring times*) or activity level (e.g., change activities' *durations* or *types*),
- Automatically create location-specific activity names by combing the module's *section* and *direction*, and the activities'

Table 1. Summary of Expressway Modules

Type	Module	Number of acts	Number of subacts	Number of pay items
Road	Earthwork	5	22	12
	Pavement			
	Asphalt concrete	5	11	11
	Cement concrete	5	11	15
	Box culvert	6	25	17
Bridge	Superstructure			
	Precast I-beam	8	27	14
	Balanced cantilever	10	58	21
	Incremental launching	8	37	20
	Steel box girder	10	30	22
	In situ shoring	7	26	14
	Span-by-span erection	7	24	15
	Advancing shoring	7	25	15
	Abutment			
	Spill-through	5	35	18
	Cantilever	5	27	14
	Footing-foundation	5	35	16
	Pier			
	Full casing pile	4	36	10
	Reverse circulation pile	4	35	9
	Caisson foundation	3	35	9
	Drilled-shaft	3	31	9
	Foundation			
	Prestressed concrete Pile	4	47	9
Tunnel	Excavation			
	NATM	10	41	70
	TBM	14	23	27
	Portal construction	8	11	34
	Cross connection construction	6	11	19
	Shaft method	7	12	17

Note: NATM=New Austrian tunneling method; TBM=tunnel boring machine.

names and counter. Examples of generated names are, "Wagon Assembly (Section II, North)," "Box Girder Cast in Place (Section II, North)-001," and "Box Girder Cast in Place (Section II, North)-002",

- Inspect and edit the schedule data after activities were created for selected modules but before the activities are repeated or cycled. Users may modify the *section* and *direction* descriptors, *recurring times*, *durations*, and *types* of individual activities before creating the final schedule, and
- Transform the schedule data to formats readable by MS Project, OpenPlan, or P3, and save data.

Evaluation of Implementation

Experiments I and II

Experiment I was an experiment of four groups of schedulers concerning time saved by using NBA to create a schedule for parts of projects with which they are familiar. Experiment II was an experiment of two schedulers concerning the time saved by using NBA to generate a schedule for a full project with which they are familiar. An attempt was made to eliminate the learning-

curve effect and the contemplation time for scheduling, which varied a lot for different schedulers, by undertaking experiments to duplicate a selected schedule using different approaches, not to create a new one. The schedule to be duplicated was chosen and created by the participants for their previous projects. Additionally, approximately one hour was spent in familiarizing the schedulers with the schedules and defining the test scopes.

Experiment I was concerned with the average amount of time saved by using NBA, as opposed to traditional scheduling tools, in various settings. First, seven projects whose submitted schedules appeared to be of high quality (e.g., periodically updated, containing detailed data such as pay items) were chosen. The contractors of the selected projects were asked to send two schedulers familiar with the projects to participate in our experiment. Only four out of seven contractors were able to participate. The participants were divided into four groups by company such that each group consisted of two schedulers from the same contractor.

For comparison, only portions of the projects that could be represented using the standard activities were chosen in this experiment. A test scope of each selected project was determined by considering the amount of work required and whether or not it could be represented by standard activities. As shown in Columns 3 and 4 of Table 3, for Groups A and B, parts of projects involving approximately 15 activities but excluding *cyclic* activities, were chosen. For Groups C and D, parts of projects involving approximately 50 activities including *cyclic* activities, were chosen.

Each participant was required to use both the traditional scheduling tool with which he was most familiar and NBA, to create two identical copies of a schedule for the designated scope. Columns 5–20 of Table 3 show the results of the experiment. The total time spent in creating a schedule was divided into three categories—average time for establishing activities and their relationships, inputting activities' detailed data (e.g., pay items and resources), and adjusting the schedule (e.g., changing relationships). In the headings of these columns, "T" denotes the time spent using the traditional scheduling tool, "NBA" the time spent using NBA, "S" the time saving using NBA, and "%S" the percentage of time saving.

Survey I was a survey of 92 schedulers with experience of using NBA concerning their opinions of the benefits of implementing network modules and using NBA. While Experiment I tested only parts of projects, Experiment II targeted a full project consisting of some activities that were not defined in the standard modules. Only two groups of schedulers could participate due to the amount of time required. The final row (Group E) of Table 3 shows the results.

Although lacking statistical significance due to the sample size, some interesting results are worthy of discussion. For Groups A and B, the test projects involved a similar number of activities and both included no *cyclic* activities. Using traditional scheduling tools, the time taken to define activities/relationships and detailed data was similar (Column 5 versus 9). Little time was required for adjustment (Column 13). The total time used by Group B (Column 17) was slightly more because their project involved more activities. Similar results also applied using NBA.

For Groups C and D, the test projects involved a similar number of activities but a significantly different number of *cyclic* activities. Using traditional scheduling tools, even with fewer activities, Group C required more time to define activities/relationships than did Group D, because Group C's project involved fewer *cyclic* activities. The *copy* and *paste* features of traditional tools were apparently helpful in defining *cyclic* activities. Group C took

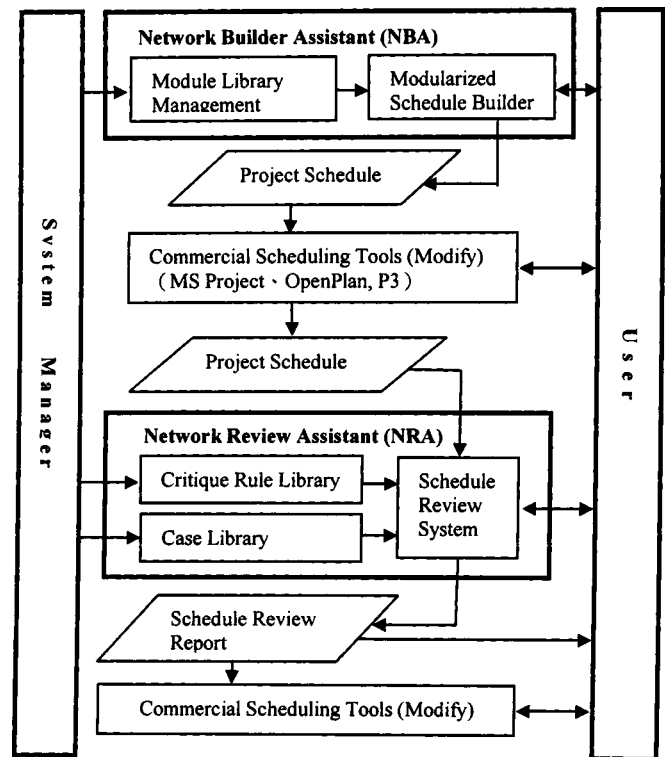
Table 2. Advancing Shoring Method Module

Activity	Major pay items	Subactivities
Preparation		Design, purchase, submittal, site work, materials move-in, access, labor move-in
Wagon assembly		Assembly of LHSF (large-scaled hanging system forms), Installation of LHSF
Box girder cast in place	Prestressed concrete, 350 kg/cm ²	Box girder/T girder cast in place
Approach slab	Backfill concrete, 240 kg/cm ² forms deformed bars, fy=4200 kg/cm ³	Diaphragm and LRB (lead rubber bearing) Back-wall filling Utilities Formwork assembly, reinforcement erecting, concrete placing
Barrier railing	Guard rail	Barrier railing
Asphalt concrete pavement	Dense graded asphalt concrete Open graded asphalt concrete	Asphalt concrete pavement, concrete pavement
Expansion joint	Bridge expansion joint	Saw cutting, expansion joint, assembling, nonshrink, concrete placing, curing, rubber material filling
Traffic signage	Signs Markings Pavement markers Delineator Fencing Glare screen	Traffic signage

slightly less time to define detailed data because their project involved fewer activities. However, the total time taken by Group C was still greater as their project involved fewer *cyclic* activities. Similar results applied using NBA, but the difference between the two groups was smaller when the traditional tools were used.

Consider the relationship between the number of activities/*cyclic* activities and the amount of time saved using NBA. NBA always required more time to adjust the schedule than did the traditional tool. The number of activities/*cyclic* activities was strongly positively correlated with the time saved in defining activities/relationships, and the total time saved. The number of activities and the time saved in defining detailed data were also positively correlated, except for Group D. This inconsistency may be explained by the fact that different groups of participants instead of a single one were tested. Another explanation may be that the *copy* and *paste* features of the traditional tool reduced significantly the amount of time required to define activities/relationships, thereby reducing the time that could be saved by using NBA.

Consider the number of activities/*cyclic* activities and the percentage of time saved using NBA (Columns 3, 4, 8, 12, and 20 in Table 3). With as many as ten activities, the percentage of total time saved was clear (57 and 59%). However, this percentage increased and remained between 61 and 70%, regardless of the

**Fig. 7.** Framework for applying of network modules

number of activities. Similarly, the percentage of time saved in defining activities/relationships was between 90 and 93%.

Although some of the discussed results were expected, some were not. For example, prior to the experiments, we knew that the module-based schedule generation requires less time in defining activities, their relationships, and detailed data than the traditional approach does due to less data entry. The former required more time to make adjustments than the latter did because standard elements were used. Our results also demonstrated that the module-based approach would require less total time than the traditional approach if the schedule was full of repetitive and cyclic activities.

However, prior to the experiments, we could not confirm whether the additional time required to adjust the schedule could compensate for the time saved using the module-based approach. We could also not determine the following: (1) how much more time could the module-based approach save over that of the traditional approach in defining activities and their relationships, as well as activity details; (2) how much more time did the module-based approach require than the traditional approach did in adjusting the schedule; (3) by using the module-based approach, whether the required more time in adjusting the schedule could compensate for the time saved in defining activities, and their relationships as well as details; (4) the amount of time the module-based approach would save over that of the traditional approach if the schedule contains repetitive activities; and (5) whether the module-based approach requires less total time than the traditional approach in the realistic schedules that include nonstandard activities. Answering these questions would help the schedulers to determine when using the module-based approach is better than the traditional approach. They also helped decision makers at TANEEB to determine what information should be included in the modules, and what project is a better candidate for a pilot implementation.

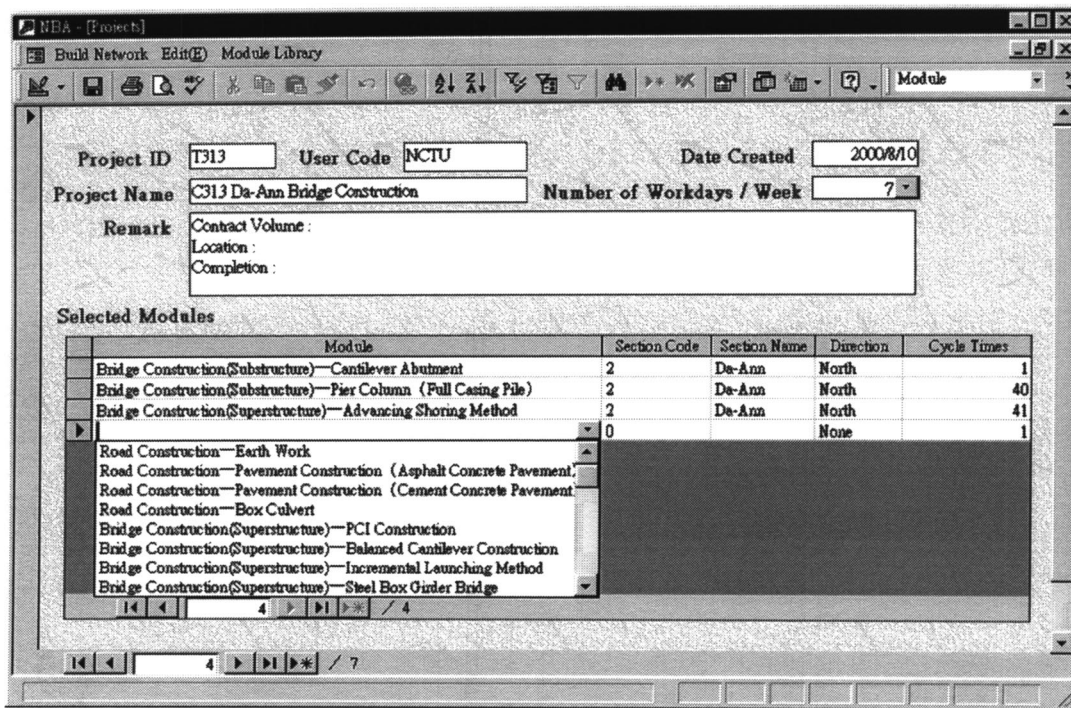


Fig. 8. Network builder assistant's interface for inputting project information and selecting desired modules

Survey I

Survey I had two goals: to understand the general expectation of practitioners of the benefits of using NBA to create a project schedule for TANEED and implement the network module concepts. We arranged two one-day free training courses for the module-based schedule generation concept and NBA. The participants consisted mainly of TANEED's managers and engineers, A/E's schedule reviewers, and contractors' schedulers. The course included an introduction to the concept of network modules, familiarization with the contents of the modules, hands-on use of NBA, and practice of its use on a sample project. We also asked the participating practitioners to fill out a questionnaire at the end of the course. A total of 72 (75.8%) responses were received from the 97 questionnaires sent out.

The questionnaire consisted of the following three parts to obtain, (A) basic information about the participant, (B) the expected benefit of each primary feature of NBA, and (C) the expected benefit of implementing network modules. Part A included questions about,

- Personal details (e.g., age, education and job background, earnings per worked hour, number of years of site experience),
- Company profile (e.g., average annual contract volume, business experience with TANEED),
- The primary scheduling tool used,
- The expected average number of man-hours required for creating a typical project schedule for TANEED, and
- The expected average number of man-hours required for reviewing a project schedule for TANEED.

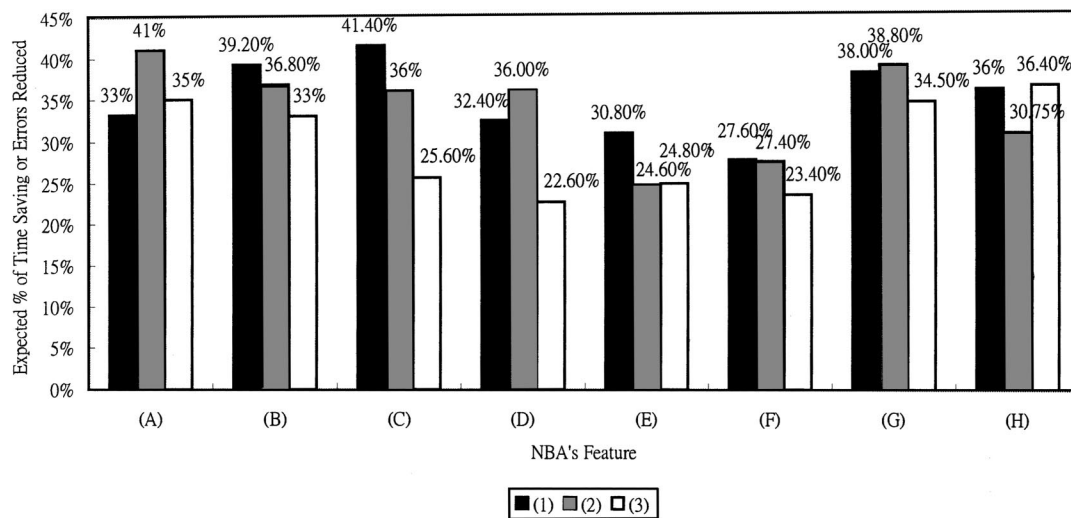
The responses to Part A revealed that the respondents had an average age of 36. Most respondents (91.67%) were at management level, and 40 of them had over three years of scheduling or schedule reviewing experience. The primary scheduling tools used were P3 (87%), Open Plan (12%), and MS Project (1%). On average, generating a schedule for a typical TANEED project required 280 man-hours, and the review required 45 man-hours for A/E and 107 man-hours for contractors.

Part B asked about the expected time saving and error reduction due to the use of each of NBA's primary features, including:

Table 3. Time Saved Using Network Builder Assistant

Test group	Test project	Number of acts	Number of cyclic acts	Amount of time spent (minutes)															
				Activities and relationships				Detailed data				Adjustment				Total			
				T	NBA	S	%S	T	NBA	S	%S	T	NBA	S	%S	T	NBA	S	%S
A	C318 (Box culvert)	13	0	9	4	5	56	10	3	7	70	2	2	0	0	21	9	12	57
B	C311 (Overpass No. 5)	16	0	11	4	7	64	14	5	9	64	2	2	0	0	27	11	16	59
C	C318 (Section Chinshu-Longchin)	50	16	87	6	47	93	113	47	66	58	12	19	-7	-58	212	72	110	66
D	C313 (Bridge Da-An)	54	28	52	5	81	90	121	56	65	54	7	9	-2	-29	180	70	140	61
E	C313 Full project	368	86	190	15	176	92	230	79	151	66	30	40	-10	-33	450	134	317	70

Note: T denotes the time spent using the traditional tool; NBA denotes the time spent using NBA; S denotes the time saving using NBA; and %S denotes the percentage of time saving.



(1) percentage of time saving in generating a schedule, (2) percentage of errors reduced in generating a schedule, and (3) percentage of time saving in reviewing a schedule.

Fig. 9. Percentage of time saved and errors reduced by using network builder assistant

1. Automatically creating activities with standard codes;
2. Automatically creating activities with names specifying activity WBS, location, lane direction, and counter;
3. Automatically repeating, cycling, and merging activities;
4. Automatically defining activity relationships within each module;
5. Providing estimation references for activity duration;
6. Defining subactivities for contractors
7. Automatically creating associated standard codes, names, and units of common pay items for activities; and
8. using NBA over all.

Fig. 9 shows the results of Part B. The x-axis lists NBA's features. Each feature is associated with three bars representing, from left to right, the expected (1) percentage of time saved in generating a schedule, (2) percentage reduction of errors in generating a schedule, and (3) percentage of time saved in reviewing a schedule.

The three features that were expected to save the most time in generating a schedule were (3) automatically repeating, cycling, and merging activities (41.4%); (2) automatically creating activities with names specifying activity class, location, lane direction, and counter (39.2%); and (7) automatically creating associated standard codes, names, and units of common pay items (38%). These features targeted the tasks that required much keying-in, duplicating, and editing.

The three features that were expected most to reduce errors in generating a schedule were (1) automatically creating activities with standard codes (41%); (7) automatically creating associated standard codes, names, and units of common pay items (38.8%); and (2) automatically creating activities with names specifying activity class, location, lane direction, and counter (36.8%). These features targeted the tasks involving much duplication and code inputs that mean little to humans.

The three features that were most expected to reduce the time required for the schedule review were (1) automatically creating activities with standard codes (35%); (7) automatically creating associated standard codes, names, and units of common pay items (34.5%); and (2) automatically creating activities with names specifying activity class, location, lane direction, and counter

(33%). These features also targeted code-related tasks, and were the same as those expected most to reduce the errors in generating a schedule.

Table 4 shows some of the questions and results of Part C. Only 40 respondents with over three years of scheduling or schedule review experience were asked to answer Questions 1 and 2. Questions 3 and 4 were for all respondents. The responses revealed great interest in "promoting modules into standards" and "using free NBA to generate schedules," with 100 and 97.5% of positive responses, respectively. This result shows that practitioners are willing to try standardization if the right tools are provided and transition was gradual. All 40 experienced schedules or reviewers agreed that NBA could reduce the amount of time required to create schedules. Furthermore, 80.5, 88.9, and 93.1% of respondents believed that standard codes and modules were helpful or very helpful in "schedule communication," "conservation of scheduling experience," and "schedule integration," respectively.

In addition to the results shown in the table, the average, expected percentage of time saved was 36%; this value was slightly lower than the measured results from Experiments I and II. The following question was also raised, "Should TANEED continue promoting the modules and NBA, and rigidly require contractors' submitted schedules to comply with standard codes for all of the coming future projects?" Further pivot analysis revealed that three groups of respondents (i.e., TANEED, A/Es, contractors) differed in attitudes toward promoting modules into rigid standards. The A/E group had the highest average rate of most positive support with 6 out of 7 respondents (85.7%) answering "agree," followed by the TANEED group (10 out of 15 respondents, 66.7%), and the contractor group (4 out of 18 respondents, 22.2%).

Above results suggest that the respondents with a great interest in promoting modules into standards did not necessarily agree to do so immediately. Undoubtedly, the A/E group was most receptive to realizing the rigid standardization because it would definitely help their schedule review work and only slightly change

Table 4. Summarized Partial Results of Part C of Survey I

Number	Interested in:	“Very interested”	“Interested”	“Not interested”	“No opinion”
1	Promoting modules into standards	19 (47.5%)	21 (52.5%)	0	0
2	Using free Network Builder Assistant to generate schedules	20 (50%)	19 (47.5%)	1 (2.5%)	0
	Are standard codes and modules helpful in:	“Very helpful”	“Helpful”	“Not helpful”	“No opinion”
3	Schedule communication	17 (23.6%)	41 (56.9%)	0	14 (19.5%)
4	Conservation of scheduling experiences	21 (29.2%)	43 (59.7%)	1 (1.4%)	7 (9.7%)
5	Schedule integration	29 (40.3%)	38 (52.8%)	0	5 (6.9%)

the way they did the work. The contractor group was most reluctant to realize the rigid standardization immediately because it changed their way of doing the work although NBA has reduced such an impact. The contractors who agreed on an immediate standardization were larger companies that had a close business relationship with TANEEB. Most respondents from TANEEB who agreed an immediate standardization are engineers who were involved with schedule review or integration.

Conclusions

A schedule standardization implementation framework including modularization, soft standardization, and rigid standardization, was proposed. This paper has detailed the modularization stage, including the concept and the development of activity network modules and two computer systems—namely, *Network Builder Assistant* (NBA) and *Network Review Assistant* (NRA)—which facilitate generating and reviewing schedules. Experiments concerning the benefits of using NBA and schedule standardization were also carried out.

In contrast to commercial scheduling tools, the proposed modules were designed to motivate users to use them in order to create the main part of a schedule. The inclusion of *normal*, *repetitive*, *cyclic*, and *merging* activities in a module reflects the common nature of expressway construction and makes the modules more flexible. A user of NBA can create an initial schedule consisting of standard, coded activities with location-, lane-direction-, and counter- specific names by simply clicking on the desired modules. Associated subactivities, used by contractors, and pay items with standard codes, are also automatically created when activities are generated. The schedule data are readable by popular scheduling tools such as MS Project, Open Plan, and P3.

The standardization of activities and pay items also makes feasible automatic schedule review. NRA helps the owner, A/E, and contractors review the contractor’s submitted schedules by reporting potential errors that violate the rules or case-based reasoning results. Such a report helps reviewers when the schedule was extensive.

The results of the experiments and survey supported the aforementioned claims. The benefits of using NBA in the time saved were measured for several projects and under different experiment settings. Benefits of using NBA in the expected time saved and a reduction in errors were also surveyed. The results identi-

fied those features of NBA that contributed most to these benefits. Practitioners also showed great interest in trying standardization by using NBA and a gradual transition to standardization.

Acknowledgments

This research expanded on the results of projects funded by TANEEB and National Science Council (89WFA0600182), Taiwan. The writers would like to thank P. C. Chu (Deputy Chief Engineer), M. Y. Lu (Section Chief), H. C. Wang and T. M. Huang (Associate Engineers), and C. M. Hsia (Assistant Engineer) at TANEEB for their participation in arranging the interviews and meetings for this research. The writers would also like to thank our graduate student, G. L. Chang, for his careful collection of data and development of NRA.

References

- Arditi, D. A., and Messiha, H. M. (1996). “Life-cycle costing in municipal construction projects.” *J. Infrastruct. Syst.*, 2(1), 5–14.
- Cherneck, J., Logcher, R., and Sriram, D. (1991). “Integrating CAD with construction-schedule generation.” *J. Comput. Civ. Eng.*, 5(1), 64–84.
- De La Garza, J. M., Drew, D. R., and Chasey, A. D. (1998). “Simulating highway infrastructure management policies.” *J. Manage. Eng.*, 14(5), 64–72.
- De La Garza, J. M., and Ibbs, C. W. (1990). “Knowledge-elicitation study in construction scheduling domain.” *J. Comput. Civ. Eng.*, 4(2), 135–153.
- Dzeng, R. J. (2000). “Development of activity network modules for expressway projects.” *Tech. Rep.*, Taiwan Area Expressway Engineering Bureau, Taipei, Taiwan (in Chinese).
- Dzeng, R. J., and Tommelein, I. D. (1997). “Boiler erection scheduling using product models and case-based reasoning.” *J. Constr. Eng. Manage.*, 123(3), 338–347.
- Miller, J. B. (1997). “Engineering systems integration for civil infrastructure projects.” *J. Manage. Eng.*, 13(5), 61–69.
- Peña-Mora, F., and Tamaki, T. (2001). “Effect of delivery systems on collaborative negotiations for large-scale infrastructure projects.” *J. Manage. Eng.*, 17(2), 105–121.
- Potts, K. (1995). *Major construction works: Contractual and financial management*, Longman Group Ltd., London.
- Shaked, O., and Warszawski, A. (1995). “Knowledge-based system for construction planning of high-rise buildings.” *J. Constr. Eng. Manage.*, 121(2), 172–182.