Constraint-Directed Scheduling for Multi-Storey Building Projects

Pasit Lorterapong* and Osama Moselhi**

* Department of Civil Engineering, King Mongkut's Institute of Technology Thonburi,
Pracha u-tid Rd., Ratburana, Bangkok 10140, Thailand

** Professor, Centre for Building Studies, Concordia Univ., 1455 de Maisonneuve Blvd.
West, Montreal, H3G 1M8, Canada.

ABSTRACT

This paper describes the development of a scheduling prototype system that assists schedulers in managing constraints that are normally encountered in the scheduling of multi-storey building projects. Constraints such as technological constraints, time constraints, resource limitations, space congestion, safety and managerial constraints are treated in this study. The system breaks down a building project into levels of abstraction based on a hierarchical model consisting of work categories and work locations. Scheduling constraints can be defined at different levels in the hierarchy. The effects of those constraints specified at a higher level are automatically propagated to those generated in the lower levels. The system is highly interactive providing a wealth of information throughout the data preparation and scheduling processes. The system can be used as a decision support tool for facilitating and improving the generation of construction schedule for multi-storey building projects.

1. INTRODUCTION

Construction planning and scheduling generally involve the identification of project activities or tasks, their logical relationships, resource requirements, durations, and their associated start and finish times. Planning and scheduling can be regarded as two separate but dependent project management tasks. The inputs for these processes generally include, for example, the project specifications and drawings, information pertaining to site visits, the availabilities of resources both in-house and outside organizations. In addition, there are several types of constraint that must be incorporated in order to produce and/or maintain a practical schedule. These constraints may be of technical in nature (e.g. technological constraints).They may be of managerial type (e.g. to comply with the interim payments schedule specified in the contract). Others may be related to safety or space congestion requirements. These constraints naturally have varying degrees of significance to the project. Therefore they should be treated accordingly. Disregarding the type or the significance, each constraint must properly be represented in the scheduling process. In addition, their influences must correctly be propagated to the associated scheduling entities.

The complexity of today's construction projects has made the preparation of schedule a challenging task for the project team. This is due to the fact that several parties (e.g. the owner, suppliers, subcontractors, etc.) may be involved in executing a project. Each party could impose its constraints to the project, in addition, perhaps, to those mentioned earlier. Failure to incorporate all of the necessary constraints could render the schedule generated impractical, necessitating frequent schedule updating or replanning. Imposing redundant constraints, on the other hand, could eliminate the chance of obtaining a good or practical schedule.

2. PREVIOUS WORK

Among various project management functions, scheduling is considered to be one of the very first to be computerized. Today, there are numerous commercial scheduling software packages that are readily available in the market[2]. Most of the software tools are, however, domain independent. They are not specifically designed to accommodate the unique characteristics of construction projects such as buildings. In addition, these software systems have often been criticized for their vast demands for inputs and for their lack of intelligence. To overcome of these limitations, a number of research has been conducted, utilizing the advanced computer technology called artificial intelligence (AI) to identify project activities and their relationships. Examples of AI planning systems include CONSTRUCTION PLANEX[1], ESCHEDULER[3]. Once a plan is developed, some systems continue to generate a schedule using the conventional technique such as CPM or the combinations of LOB and CPM[4,5]. Majority of these systems have focused on the automation of construction planning and scheduling. While automated planning systems provide clear benefits, their uses are still limited in the real operation. The main reason, perhaps, is that the majority of scheduling constraints have not been incorporated in those systems.

3. WORK BREAKDOWN STRUCTURE

In this paper, scheduling is viewed as constraint satisfaction problems. Emphasis has been placed on the identification and representation of various types of constraints that are normally encountered in the construction of multi-storey building projects, as well as the propagation of their effects on other scheduling entities during the scheduling process. In this study, scheduling entities that are commonly found in the construction of multi-storey building projects can be represented in a hierarchy that is divided into two main axes: 1) Work Categories, and 2) Work Locations as depicted in Fig. 1. The work categories include: sitework, substructure, super-structure, mechanical work, electrical work, interior finishing, exterior finishing, and miscellaneous. Each work-category can further be broken-down into trades such as those specified in the MASTERFORMAT. On the other hand, one could also divide the project site into different locations along the function of the facility. In multi-storey building projects, the function of the facility may include, for example, the building tower, parking garage, swimming area, recreation area, landscaping, and other supporting areas. These areas are classified mainly into two groups. The first group involves the work that progresses, both, vertically and horizontally (e.g. the tower and parking garage). The second group on the other hand, includes the work that progresses mainly in the horizon direction (e.g. on-ground swimming pool, side-walk, earthwork, etc.). The work classified in the first group could be further subdivided into a number of floors that subsequently can be broken-

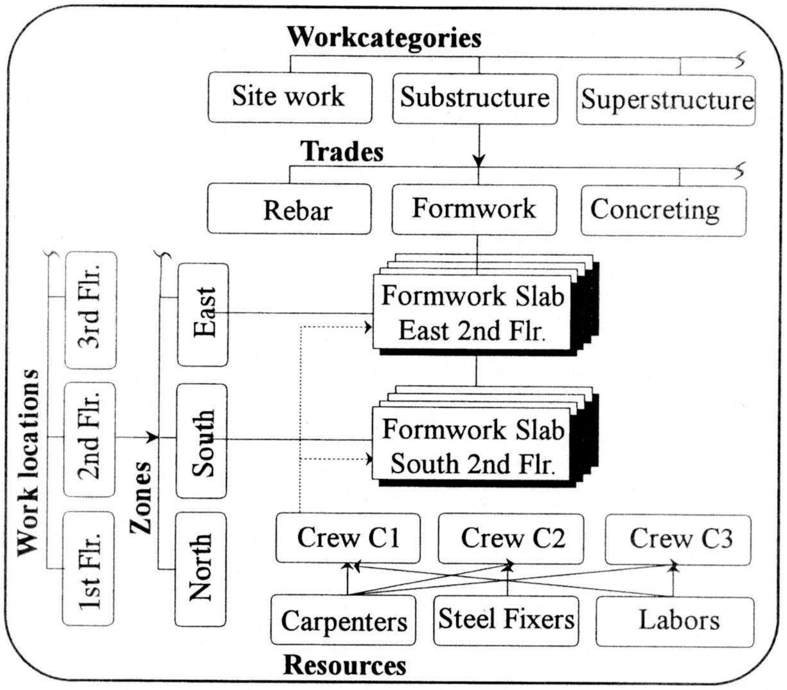


Figure 1 Work Breakdown Structure

down into various zones. The work included in the second group, on the other hand, can be divided directly into various zones, if deem necessary. Resources used for the execution of a project can be defined along the resource axis (see Fig. 1). These resources are then used to form various construction crews, which later on will be allocated to project activities. A subcontractor is treated as one type of the main contractor's resources.

The above entities usually have some sort of relations or restrictions either to themselves or to others. Such relations or restrictions can also be regarded as scheduling constraints. The constraints treated in this study include technological constraints, time constraints, resource limitations, space congestion, safety and managerial constraints. Technological constraints are those that represent the logical relationships between any two activities in order to assemble a design element. For example, placing concrete in a column can start after formwork and rebar placements are finished. Time constraints are usually related to intermediate milestones, set to conform to the interim payments specified in the contract, or any contractual dates made between the owner and the contractor. Resource limitations are the imposed quantities and their associated availability periods assigned to the project. Space constraints are introduced to prevent any trade interference, resulting in a decreasing productivity due to work interruptions. Safety constraints are imposed to safeguard against the avoidable accidents that could happen in the construction site. Managerial or preferred constraints are those specified to meet certain project or organization objectives from the time and cost point of view.

All of the above constraints can be imposed on the scheduling entities defined at all levels of abstraction in the work breakdown structure. The effects of the constraints imposed on the entities that are defined in the higher level will be propagated to their associated entities defined in the lower level of the hierarchy. This form of representation therefore facilitates effective schedule generations and maintenance.

4. CONSTRAINT SATISFACTION SYSTEM

The main focus of this study is to develop a scheduling prototype that assists the project team in generating schedules for the construction of multi-storey building projects. Scheduling is viewed as constraint-satisfaction problems where the scheduling entities and constraints are continually being added, deleted, and modified throughout the construction of the project. The system generates and maintains the schedule mainly by reasoning about the dependencies between scheduling entities and constraints that are involved in the decision making at time. Scheduling entities are modeled as objects. An object contains a number of attributes and methods that modify its attribute values. The system consists of four modules and a database. These modules are: *User Interface*, *Constraint Manager*, *Resource Manager*, and *Scheduler*. The database contains information related to specific projects and the organization. The architecture of the proposed system is depicted in Fig. 2. The system is being developed on a PC, using a windows-based, object-oriented programming environment. The functions of the system modules are described below.

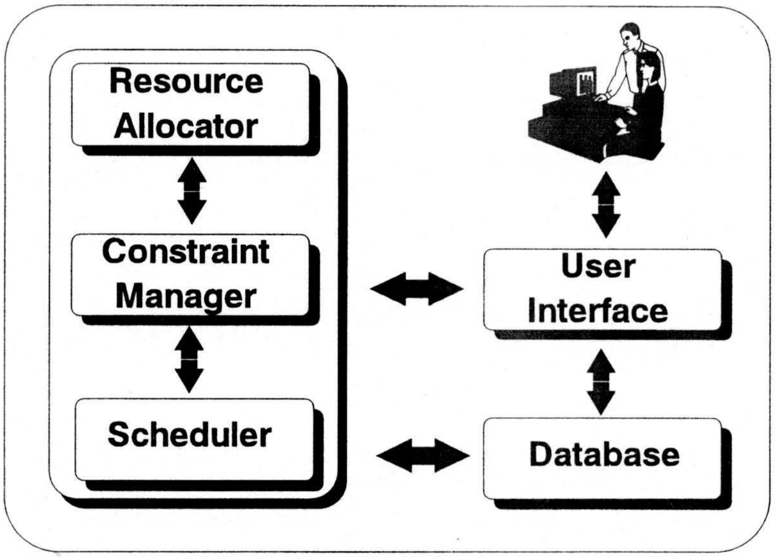


Figure 2 System Architecture

4.1 User Interface Module

The main function of *user interface* are: 1) to prepare all the data required by the system at the input stage, and 2) to facilitate information exchange between the user and the system during the schedule generation process. *User interface* first asks the user to define resources that are available to the project. Next, the system assists the user for breaking-down the project into eight work categories. Each work category is further subdivided into a number of user-defined trades. For example the work category "Substructure" can be subdivided into various trades such as "Formwork", "Rebar", and "Concrete placement", etc. Upon completion of inputing the trade, the module guides the user to breakdown the project into various work locations according to the function of each facility (e.g. building tower, parking garage, recreation areas, etc.). The work that progresses mainly along the vertical axis (e.g. the tower, parking garage, etc.) is subdivided utilizing the floor numbering system shown in Fig. 3. This numbering system helps minimize the user effort required for further subdividing each floor into working zones, as well as for creating construction activities.

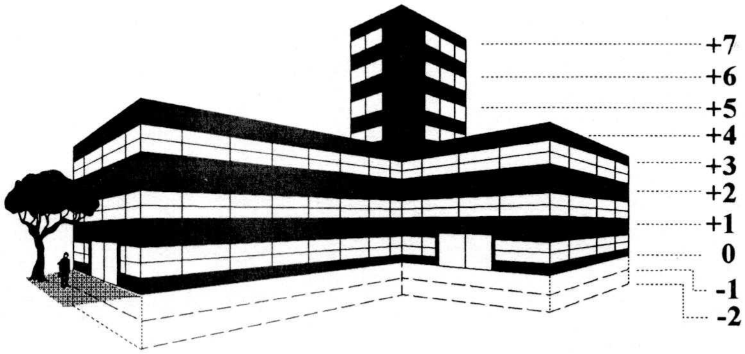
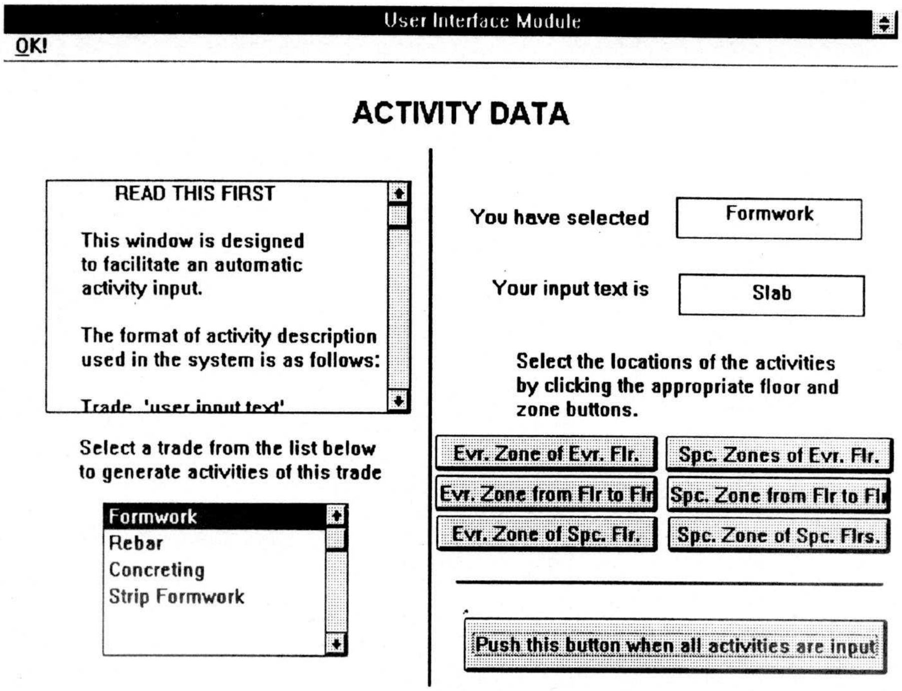


Figure 3 Work Breakdown Structure

Fig. 4 shows an example of the screen designed to enter project activities that will be performed on the tower of a multi-storey building project. The default format for the description of activity is *[Trade_"user input text"_Floor_Zone]*. For example, if activity "Formwork slab" will be performed at the "North" zone of every floor, the user will select first the trade "Formwork from the list. Next, the user will type "Slab" for the user input text. The system provides six options for specifying the locations of this activity. They are: 1) Every zone of every floor, 2) Specific zones of every floor, 3) Every zone from floor to floor, 4) Specific zones from floor to floor, 5) Every zone of specific floors, and 6) Specific zones of specific floors. In this example assuming that the user selects option 2. The system will bring a new screen to the user for selecting the designated zone (see Fig. 5). After the selections of the zone has completed, the system automatically generates a set of activities "Formwork Slab" that have different *Floor* and *Zone* in their descriptions. This feature is designed to help minimize the user input effort during the data preparation process.

Once all scheduling entities have been input, constraints can then be imposed. Like the scheduling entities, constraints are modeled as objects. These constraints can be numeric or symbolic. Numeric constraints are used to set a limit on the values of objects' attributes (e.g. number of carpenters available to the project). Symbolic constraints are mainly used to define



ACTIVITY DATA

READ THIS FIRST

This window is designed to facilitate an automatic activity input.

The format of activity description used in the system is as follows:

Trade 'user input text'

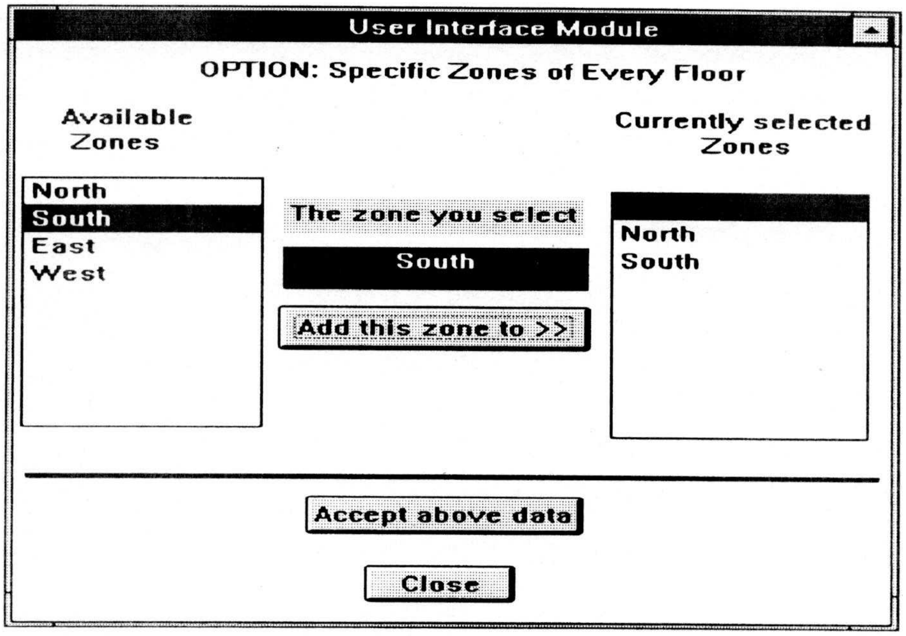
Select a trade from the list below to generate activities of this trade

You have selected: Formwork

Your input text is: Slab

Select the locations of the activities by clicking the appropriate floor and zone buttons.

Figure 4 Example of the screen for entering activity data



OPTION: Specific Zones of Every Floor

Available Zones: North, South, East, West

The zone you select: South

Currently selected Zones: North, South

Figure 5 Example of the screen for zone selection

precedence relationships between scheduling entities or objects. Various precedence relationships are offered by the system. Examples of these relationships are "immediately after", "after", "not after", "concurrently", and "not concurrently", etc. These types of relationships can be imposed at different levels of abstractions. The effect of constraints specified at the higher level can be propagated to the entities defined in the lower level, if desired. For example, if the relationship "immediately after" is imposed between trades "Formwork" and "Rebar", and the user has specified that this constraint is applicable to all design elements. The system will search for activities that belong to these two trades and have the identical "*user input text*", "*Floor*", and "*Zone*" in their descriptions. When found, the system automatically imposes the constraint "immediately after" on these two activities. This could relief the user from having to repeatedly enter the same input data. In addition, it could help minimize human errors that could compromise the practicality of the schedules.

Once, all the data are entered, the system performs network scheduling in two steps. In the first step, the system activates *Constraint Manager* and *Scheduler* to produce a time-constrained schedule. The user will actively be involved during the process to assure that a satisfactory schedule is obtained. In the second step, the three modules *Constraint Manager*, *Scheduler*, *Resource Allocator* and the user will jointly produce a resource-constrained schedule. This process is performed in such a way to minimize any changes made to the preliminary schedule.

4.3 Scheduler Module

The main function of *Scheduler* is to generate a CPM project schedule that satisfies mainly the time constraints provided by the user. This process requires coordination between *Constraint Manager* and the user. The module first requests information related to precedence constraints from *Constraint Manager*. The module then performs forward and backward passes calculations. Upon completion of the computations, the start and finish times as well as the criticality associated with each activity will be obtained. In addition, *Scheduler* also identifies all paths in the project network. Each network path is modeled as an object. The attributes of these objects are the activities that constitute the path. Path identification is of crucial since it facilitates rapid constraint checking without the need to recalculate the entire project network every time any modification is made to activity objects.

If the user is satisfied with the schedule generated, *Scheduler* will save this schedule to the database and consequently activates *User Interface* for further actions. On the other hand, if the user is not satisfied with the schedule (e.g. certain milestones are not met), the module will display a list of constraints that are violated. Next, the user selects the constraint to be modified. Accordingly, Scheduler provides a list of activities that could contribute to that constraint. The user then modifies the duration of the activity in question as desired. *Scheduler* then repeats the above process until a satisfactory schedule is obtained.

4.3 Constraint Manager Module

During the schedule generation, *Constraint Manager* performs the following tasks: 1) constraint identification, 2) constraint checking, 3) constraint updating, and 4) constraint relaxation. At any time when an activity is to be scheduled, *Constraint Manager* identifies the constraints that are relevant to the situation being considered. The module then performs constraint checking to investigate whether all the preconditions have been met before such an activity can start. If those conditions were met, the activity is entitle to start. Otherwise, that activity will be postponed. Once a new decision pertaining to scheduling is made, the module updates the information of those constraints being considered as well as propagating their effects to the scheduling entities involved. *Constraint Manager* also keeps track on the constraints that could possibly conflict one another. If detected, the user will be asked to relax some of the attributes of those constraints that are experiencing the conflict.

4.4 Resource Allocator Module

Resource Allocator is called upon to perform resource aggregation for the schedule generated by the scheduler. Unlike currently available project management software systems, the system allows the user to participate in the resource allocation process. This way, the user's knowledge and experience are utilized in the schedule preparation, enhancing the practicality of the schedule produced. Resource conflicts are identified through the communications between *Resource Allocator* and *Constraint Manager*. Those conflicts will be presented to the user, asking for further actions. Several resource allocation strategies are provided by the system. They are:

a) Ignore the conflict.
b) Use fewer resources.
c) Substitute by another type of resources.
d) Recall resources from other activities.
e) Allow for the use of overtime

The above strategies can be performed upon confirmation made by the user. During the resource allocation process, the user can view the status of the scheduling constraints. For those that are violated, *Constraint Manager* will immediately notify the user. The user can use the overtime option for the remaining activities if desired.

5 CONCLUSIONS

This paper describes a framework for modeling the construction of multi-storey building projects. Scheduling is viewed as a constraint satisfaction problem where constraints are continually being identified, checked, updated and modified until satisfactory schedules are obtained. A new scheduling system has been developed to assist the project team starting from breaking down the project into manageable entities, specifying constraints, and preparing preliminary and final schedules. Considerable attentions have been focused on facilitating user interactions with the system.

An object-oriented programming method has been used and it proved to be suitable for representing the multi-storey building project, which consists of various kinds of scheduling entities. The overall system has been designed in such a way to simulate a typical construction meeting where several parties exchange their information and expertise trying to resolve the arisen problem. The system has been developed in a PC environment. The system allows user involvement in the schedule generation process. This feature will significantly improve the practicality of the schedule obtained.

5. REFERENCES

1. Henrdickson, C., et al. "Expert System for Construction Planning." J. of Constr. Engrg. and Mgmt., ASCE, Vol. 1, No. 4, 1987, pp. 241-252.
2. Johnson, R.V., "Resource Constrained Scheduling Capabilities of Commercial Project Management Software, Project Management Journal, Vol. 22, No.4, 1992, pp. 39-43.
3. Moselhi, O. and Nicholas, M.J., "Hybrid Expert System for Construction Planning and Scheduling." J. of Constr. Engrg. and Mgmt., ASCE, Vol. 116, No. 2, 1990, pp. 221-238.
4. Morad, A.A. and Beliveau, Y.J., "Knowledge-Based Planning System," J. of Constr. Engrg. and Mgmt., ASCE, Vol. 117, No. 1, 1991, pp. 1-12.
5. Shaked, O. and Warszawski, A. CONSCHED: Expert System for Scheduling of Modular Construction Projects," J. of Constr. Engrg. and Mgmt., ASCE, Vol. 118, No. 3, 1992, pp. 221-238.