CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Formalizing Construction Sequence Constraints for the Rapid Generation of Scheduling Alternatives

By

Bonsang Koo

CIFE Technical Report #155
JANUARY 2004

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

FORMALIZING CONSTRUCTION SEQUENCE CONSTRAINTS FOR THE RAPID GENERATION OF SCHEDULING ALTERNATIVES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING
AND THE COMMITTEE OF GRADUDATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Bonsang Koo

December 2003

ABSTRACT

Construction planners today use CPM-based schedules to represent the planned logical sequence of activities to perform a project. A construction schedule typically represents the sequence of multiple trades that perform individual work while sharing common workspaces or resources. During the course of a project, planners frequently modify activity sequences to meet changing project demands. More than one sequencing alternative can exist, and planners need to develop and evaluate alternative sequences to make well-informed decisions.

When developing sequencing alternatives, planners need to understand the “role” an activity plays for following activities. They also need to assess the “status” of activities, i.e., whether activities may or may not be delayed. Planners infer the role and status of activities by conceptually classifying the rationale for constraints between activities with respect to their role and flexibility.

For example, planners infer that an activity is “enabling” a following activity when a “supported by” constraint exists between the two activities. Planners also realize that the activity cannot be delayed because the “supported by” constraint is typically inflexible.

However, the current CPM framework only distinguishes the temporal aspects of constraints and only distinguishes the time-criticality of activities. Consequently, determining the role and status of activities can only be performed in the planner’s minds. Correspondingly, developing sequencing alternatives using today’s CPM-based scheduling tools is an error-prone and time-consuming process.

Thus, the goals of this research were threefold: 1) develop a representation of sequencing rationale that enable planners to describe their rationale for constraints and the classifications they make for different types of sequencing rationale, 2) develop a mechanism that leverages the representation to infer the role and status of activities automatically, and 3) develop a process that supports planners in utilizing the representation and mechanism to develop sequencing alternatives correctly and rapidly.

Accomplishing these goals posed unique challenges. The representation needs to model a classification schema that correctly classifies the different types of specific constraints (e.g., damaged by, protected by, etc.) in construction schedules with respect to their role and flexibility. The mechanism needs to identify unique paths between activities in a CPM network and correctly classify activities based on the role and flexibility of the individual constraints in these paths. The process needs to model generically how planners use the role and status of activities to identify which activities to delay and to prioritize activities when developing sequencing alternatives.

I addressed these challenges by formalizing (1) an ontology that models the conceptual classification planners make for construction sequencing rationale, (2) a "classification" mechanism that uses a network chain search algorithm and inference rules to infer the role and status of activities automatically given a CPM network schedule for which sequencing rationale has been explicitly represented using the ontology, and (3) a formal process that integrates the classification mechanism and ontology to guide planners in developing sequencing alternatives correctly and rapidly.

I demonstrated the power and generality of the formalizations by performing three retrospective test cases and one charrette test using prototype software, CLCPM. The retrospective cases validated that the ontology correctly classifies the different types of constraints that exist in construction schedules. The cases also validated that the classification mechanism and formal process could be used to identify and re-sequence activities correctly for different types of construction work. The charrette test demonstrates that planners identify and re-sequence activities more accurately and consistently using CLCPM than with conventional scheduling tools.

Practically, the formalizations provide an environment that provides planners with a better basis to make re-sequencing decisions. The environment enables multiple project members to communicate and discuss the logic of activity sequences, better understand the individual role and impact between project members, and evaluate multiple sequencing scenarios more quickly and consistently within group settings.

ACKNOWLEDGEMENTS

I always knew in my heart that I could successfully complete Ph.D. level work. What I didn't know was that along the way I would need the help of many people to make my Ph.D. journey an enjoyable and cherished experience. I would like to take this opportunity to thank them for their support and friendship.

First of all, I would like to thank my advisor Professor Martin Fischer for his continuous support and steadfast guidance throughout the Ph.D. process. He has helped me to understand the research process, develop my research skills, and convey complex ideas in a clear way. More than a teacher, he has been a friend, a big brother, a confidant, a person of integrity and honesty. He is and always will be a mentor for me in the truest sense of the word.

I am also indebted to my committee members. Professor Ray Levitt has provided keen insights and direction in formulating my research. Professor Boyd Paulson has provided me with a wealth of knowledge pertaining to the related research in my field of study. Dr. John Kunz's constructive criticism has continuously challenged me to improve and clarify my research. I thank each of them for lending me their intellect and experience.

My colleagues at the Center for Integrated Facility Engineering have helped me get through some of the toughest times and I would like to thank them for their support and friendship. I would like to thank the members of our 4D research group: Florian Aalami, Ragip Akbas, Burcu Akinci, Sheryl Staub-French, John Haymaker, Calvin Kam, Jonghoon Kim, Seungkoon Lyu, and Arto Kiviniemi. I also extend my warm regards to my Korean friends: Ilchan Ahn, Joonsoo Kim, and Jungung Min.

I have been fortunate enough to work with several professionals who have shared their insights and shown genuine enthusiasm for my work. In particular, I would like to thank Dean Reed of DPR Construction for patiently lending his valuable expertise to demonstrate the value of my research. I would also like to thank Art Stout of Intel Corporation for giving me the opportunity to gain experience working on his job site. A special thanks goes to Professor Manfred Breit for his sincere advice and lovely dinners during my stay in Switzerland.

I am grateful for my family back home. My brother Bonjune Ku and his wife Wonhee Park are two of my best friends, and I thank them for their unrelenting support and encouragement.

Most of all, I am indebted to my parents Jayoung Ku and Yangja Chi. They have always been supportive, loving, believing and cheering me on from start to finish. They are a true inspiration for my life and I extend my love to them. I dedicate this dissertation to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF TERMS

Candidate activity: Activity selected by user to delay.

Classification conflict: A classification conflict occurs when two or more network chains return differing classification values for the role and status of activities.

Critical network chain: A network chain populated by activities whose total float values are zero.

Driving activity: An activity that cannot be shifted forward (i.e., delayed) by using target float or by relaxing constraints between the activity and the candidate activity. Activities that do not meet these conditions are non-driving.

Enabling activity: An activity that enables an activity that in turn enables the target activity. Activities that do not meet these conditions are impeding.

Flexibility: Flexibility defines whether a constraint can or cannot be relaxed.

Network Chain: A single succession of activities and constraints between a typical activity and the target activity.

Overriding constraint: Constraints used to classify activities when multiple constraints exist between two activities.

Related activity: Activities linked to the target activity by one or more network chains.

Role of activity: Collective term for describing whether an activity is enabling or impeding.

Role of constraint: The functional property or characteristic that planners implicitly associate with a particular constraint.

Sequencing conflict: A sequencing conflict occurs between two activities when the activity being delayed does not have total float (i.e., is critical) but is linked to the target activity.

Shift backward: with respect to time, i.e., to expedite.

Shift forward: with respect to time, i.e., to delay.

Status of activity: Collective term for describing whether an activity is driving or non-driving

Target activity: Activity requiring earlier execution.

Target float: An activity's total float value calculated with respect to the target activity.

Unrelated activity: Activities not linked to the target activity by network chains.

CHAPTER 1. INTRODUCTION

"The more constraints one imposes, the more one frees one's self."

- Igor Stravinsky

Stravinsky's quote reminds me of the importance of the different constraints that exist in building a construction project: a beam needs to be placed only after columns are in place, slabs need to be placed after frames are erected, trades need to work sequentially to share workspace and limited resources on site. Construction planners need to respect such constraints imposed upon a project when developing construction plans and schedules. Only within the confines of correctly defined constraints do planners have the true freedom to develop activity sequences resulting in a successful plan for delivering a project on time and under budget.

As a composer, Stravinsky uses notes and symbols to describe the different types of constraints that impose the sequence and tempo of his music. Similarly, construction planners today use CPM-based schedules to describe the initial planned sequence and timing of work. However, the current CPM framework does not enable planners to describe explicitly the *rationale* behind the different types of constraints in construction projects.

The absence of sequencing rationale in CPM-based schedules makes it difficult for construction planners to interpret the logic behind activity sequences, or determine whether certain constraints may or may not be relaxed. In addition, the limitation makes it difficult for planners to modify existing activity sequences to meet a particular project requirement during the course of a project. Planners frequently need to re-sequence activity sequences to expedite intermediate milestone activities or activities for installing major project components.

Consequently, planners today modify activity sequences in an ad hoc and manual way, without thoroughly exploring the possible sequencing alternatives that may exist to expedite particular activities throughout the lifetime of a construction project. The ad hoc approach can lead to poorly developed sequencing alternatives, ultimately resulting in additional rework and loss of productivity.

Realizing the problems associated with the current ad hoc approach for developing sequencing alternatives, in this Ph.D. research, I focused on developing a formal approach that enables planners to describe the rationale behind constraints explicitly, and also supports planners in developing sequencing alternatives in CPM-based schedules correctly and rapidly.

This introductory chapter overviews the overall research and provides a guide to the succeeding chapters of this thesis. It summarizes the specific engineering problem through a motivating case example, discusses the research questions that address the core research challenges, presents the formal approach developed in this research for developing sequencing alternatives, and the methodology used to perform this research.

1.1 MOTIVATING CASE EXAMPLE

My motivation for this Ph.D. research stems from my experience working for a cleanroom construction project, specifically Intel Corporation's FAB 22 project. In addition to being a design-build project, the owner requested the project to be accelerated from the initially contracted 15 months' duration to 12 months. Furthermore, frequent design changes, differing site conditions and procurement delays constantly required the schedule to be modified and revised throughout the course of the project. My test case limits the scope of the discussion to one of the main buildings of the project, the Central Utility Building (CUB), which houses much of the process equipment (i.e., chiller/boiler pipes, east/west boiler) required to support the main Fabrication Building (FAB) (Figure 1.1). In particular, construction work in the FAB could not start until the process pipes were installed and connected between the two buildings. Hence, the process pipes needed to be installed as early as possible.

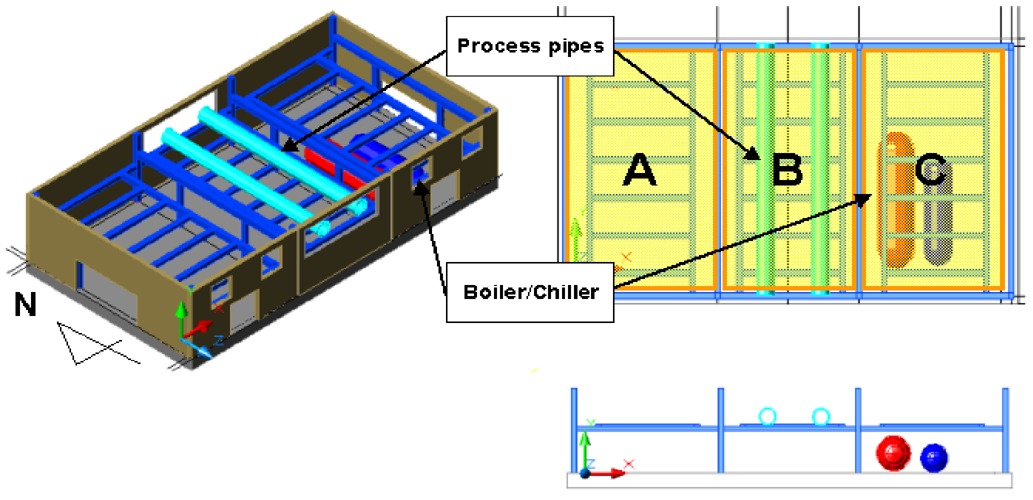


Figure 1.1: Overview diagram of Central Utility Building (CUB).

Figure shows a 3D, plan and front view of the Central Utility Building. A, B and C denote the zones sectioned for planning and coordination purposes.

Figure 1.2 shows the initial schedule and snapshot of the 4D visualization for constructing the foundation, structural frame and process pipes of the building. The project manager has sectioned the CUB building into three major zones (i.e., A, B and C) for planning and coordination purposes. The figure also denotes the respective trades performing different types of work. The conventional bar chart format used to represent the schedule uses precedence relationships (i.e., FS, SS, etc.) to describe the sequence logic between activities. On viewing the schedule, the project manager determined that process pipes needed to be installed earlier than the planned start date (day 12 in Figure 1.2a).

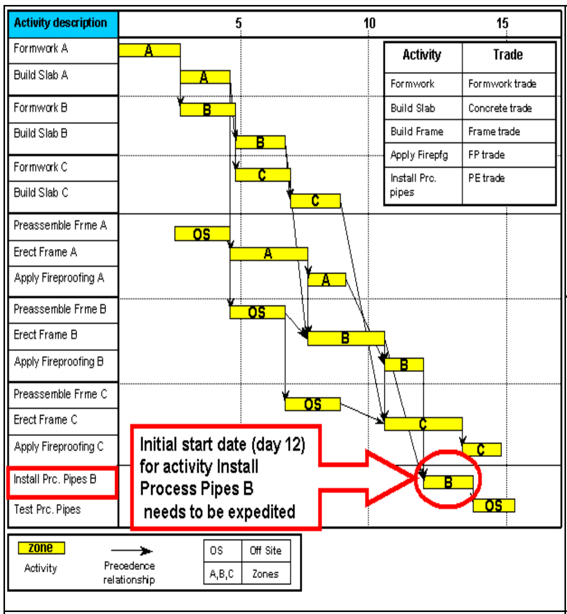


Figure 1.2a: CPM schedule of initial sequence.

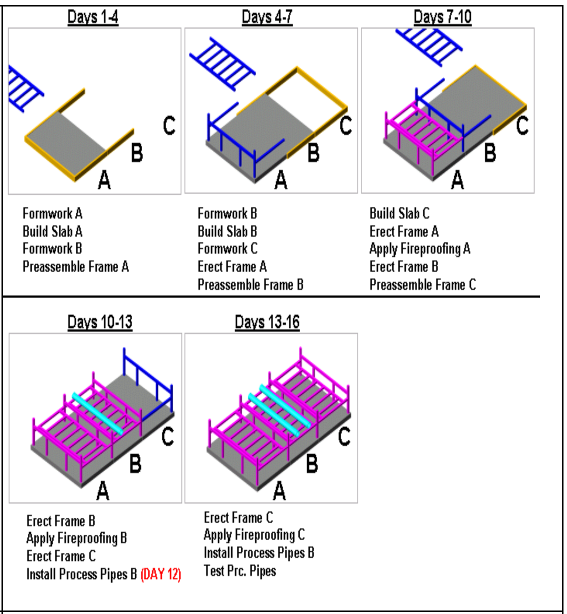


Figure 1.2b: 4D visualization of initial sequence.

Figure 1.2: Initial schedule and 4D visualization for Central Utility Building.

Schedule shows installation sequence of foundation, structural frame, fireproofing, and process pipes. In addition, trades work through zone A to B and C. With this schedule, the projected start date of process pipe installation is day 12.

The solution used by the project manager was to "switch" the sequence between the activities Apply Fireproofing B and Install Process Pipes B. Figure 1.3 shows the schedule and 4D visualization of the modified sequence. The change enabled process pipe installation work to start on day 11. The change also required the Fireproofing trade to wrap the process pipes to provide protection from fireproofing material. However, this alternative was not the product of a thorough investigation of possible sequence alternatives available to the project manager. In addition, the project manager did not create this alternative using the existing CPM schedule, but rather made the decision in the field.

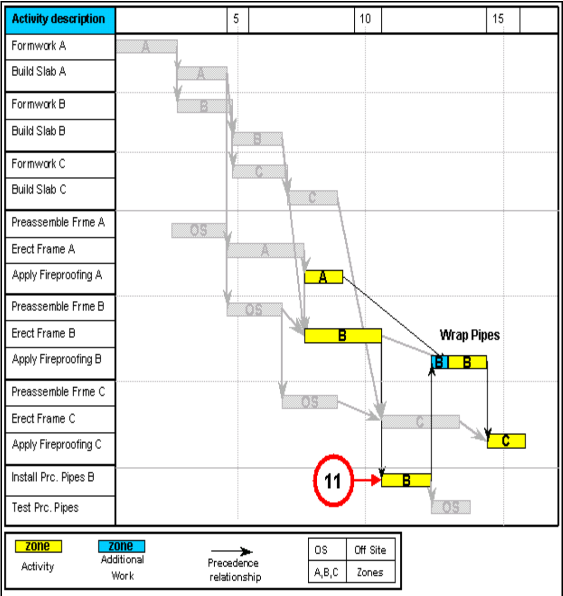


Figure 1.3a: CPM schedule of modified sequence.

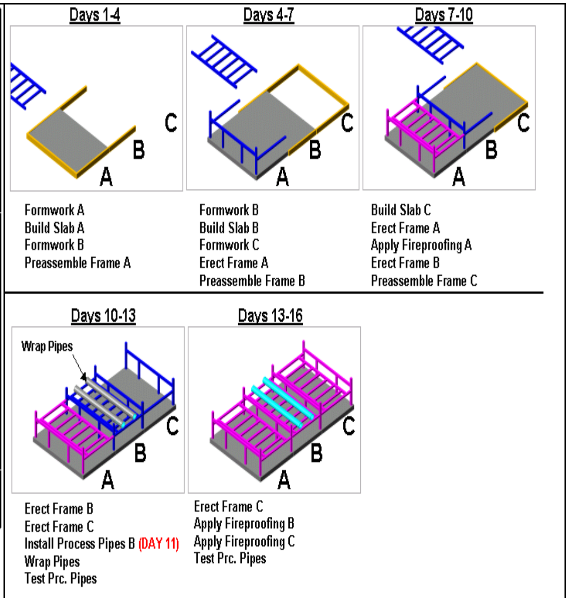


Figure 1.3b: 4D visualization of modified sequence.

Figure 1.3: Modified schedule and 4D visualization of Central Utility Building.

Schedule shows that with the modified sequence, process pipe work can start at day 11.

As the example shows, construction planners frequently have to modify activity sequences to coordinate multiple trades that share limited resources and workspaces (Riley and Sanvido, 1995). The reasons to modify existing sequences can include having to expedite the installation of major components, meeting intermediate milestones, or simply to catch up on a delay.

More than one alternative exists to modify the activity sequence, and hence investigating different sequencing alternatives would allow planners to make better-informed decisions. Each alternative requires planners to identify activities that can or cannot be delayed, update the change in the schedule and evaluate the impact of the change. However, planning decisions are often made without the aid of CPM-based scheduling tools and without the evaluation of possible sequencing alternatives. This is in part due to time constraints that require planners to make moment-by-moment decisions, but it is also due to the difficulty in generating and evaluating sequencing alternatives using existing CPM-based scheduling tools. It is difficult to modify sequences using CPM-based tools because of the absence of sequencing information required to identify and develop feasible sequencing alternatives correctly and rapidly.

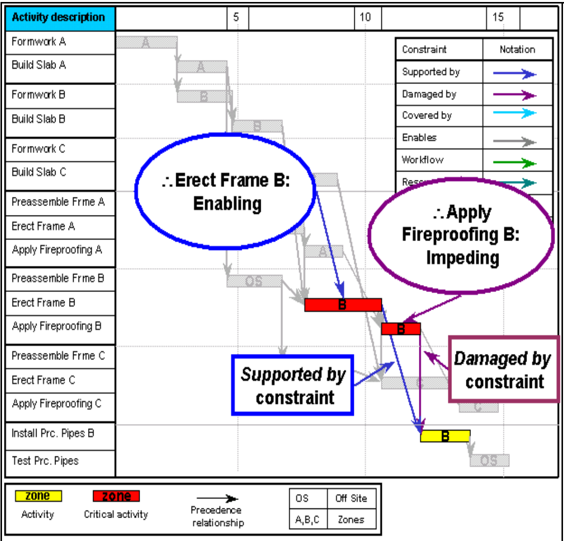


Figure 1.4a: Role of activities based on rationale of constraints.

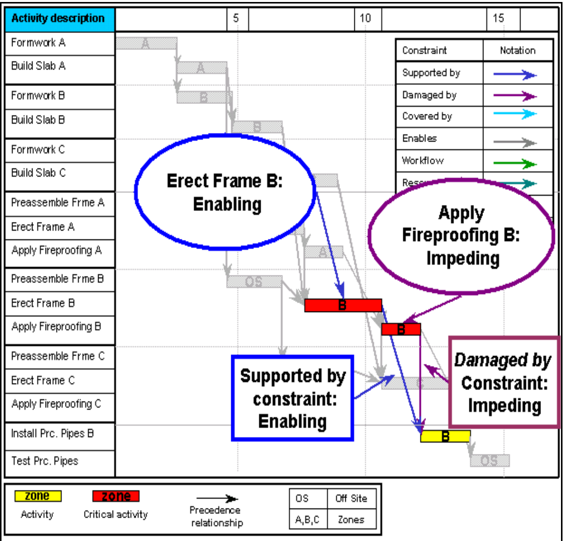


Figure 1.4b: Role of constraints distinguished as enabling and impeding.

Figure 1.4: Role of activities classified based on the role of constraints.

To identify sequencing alternatives, planners need to understand the initial rationale for activity sequences. As shown in Figure 1.4, planners use constraints or, more specifically, precedence relationships to describe activity sequences. However, the precedence relationships only describe the planned temporal dependencies between activities. For example, in the test case, the initial rationale for sequencing pipe installation work after fireproofing frames in zone B is to prevent damage to the pipes. The rationale for sequencing pipe installation work after frame erection in zone B is because frames provide support for the process pipes. Borrowing nomenclature commonly used in literature (Darwiche et al., 1988; Echeverry et al., 1991; Kähkönen, 1993; Aalami and Fischer, 1998), Figure 1.4a shows the rationale for these activity sequences denoted as *damaged by* and *supported by* constraints, respectively. As the figure shows, distinct types of constraints can be defined by distinguishing constraints with respect to their rationale.

Planners need to understand the initial rationale for constraints to determine the "role" activities have on following activities. For example, the *supported by* constraint implies that the activity Erect Frame B is "enabling" as it provides physical support for process pipes (Figure 1.4a). Conversely, the *damaged by* constraint implies that the activity Apply Fireproofing B is "impeding" the earlier execution of process pipes.

As the example shows, planners need to know the rationale for constraints to infer the role of activities with respect to the activity requiring earlier execution. Given a specific constraint, planners conceptually classify the constraint with respect to its role. In the case example, the *supported by* constraint is an “enabling” type, and the *damaged by* constraint is an “impeding” type of constraint. I call this characteristic the “role” of constraints (Figure 1.4b).

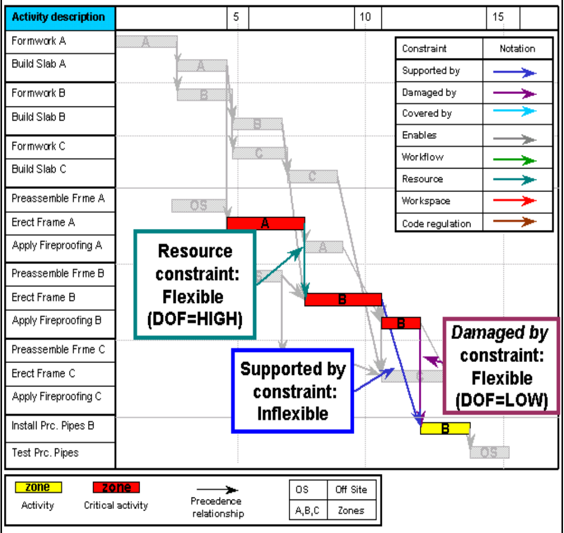


Figure 1.5a: Flexibility of constraints. DOF=degree of flexibility (DOF=HIGH means easiest to relax).

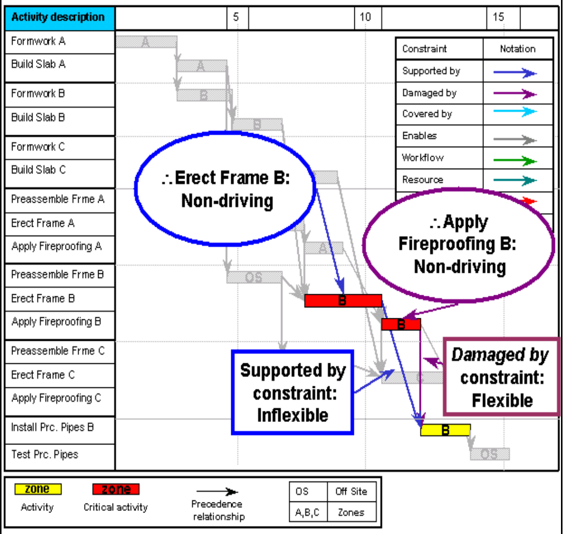


Figure 1.5b: Status of activities based on flexibility of constraints.

Figure 1.5: Status of activities classified based on the flexibility of constraints.

In addition to the rationale of constraints, planners also need to know whether a constraint may or may not be relaxed. The “flexibility” of a constraint depends on the particular circumstances (e.g., availability of labor and materials) of a project. That is, the flexibility of a constraint is project-specific. For example, the *supported by* constraint for the test case is inflexible (Figure 1.5a). However, the constraint may be flexible if temporary support could be provided. In addition, flexible constraints can have varying levels of flexibility (Echeverry et al., 1991). For example, relaxing the *damaged by* constraint of the test case requires performing additional work (i.e., providing protection for the process pipes). Comparatively, the rationale for sequencing activity Erect Frame A and activity Erect Frame B sequentially (instead of concurrently) is because of limited resources. That is, the frame trade can only work in one zone at a time. Figure 1.5a shows the rationale denoted as a *resource* constraint. Relaxing the *resource* constraint may only be a simple matter of the frame trade starting work at a different location. In this case, the *resource* constraint has a relatively higher “degree of flexibility” (DOF) than the *damaged by* constraint. Hence, constraints need to be distinguished as either flexible or inflexible. For constraints that are flexible, the degree of flexibility needs to be distinguished qualitatively.

Planners need to understand the flexibility of constraints to determine whether an activity is "driving" or "non-driving" with respect to the activity requiring earlier execution. "Driving[1]" is a term frequently used in the Architecture/Engineering/Construction (AEC) industry to describe activities that may or may not be delayed. More formally, I define driving activities as activities that cannot be delayed either by using float or by relaxing constraints between activities and the activity requiring earlier execution. As shown in Figure 1.5b, the activity Apply Fireproofing B is a critical activity (i.e., zero float). However, the *damaged by* constraint is flexible, and hence Apply Fireproofing B can be delayed. Hence, the activity is "non-driving." Similarly, the activity Erect Frame B is also a critical activity. Although the *supported by* constraint between activities Erect Frame B and Install Process Pipes B is inflexible, the activity can still be delayed by relaxing the *damaged by* constraint. Hence, the activity Erect Frame B can be delayed and is also a "non-driving" activity. I use the term "status" to describe collectively whether an activity is driving or non-driving.

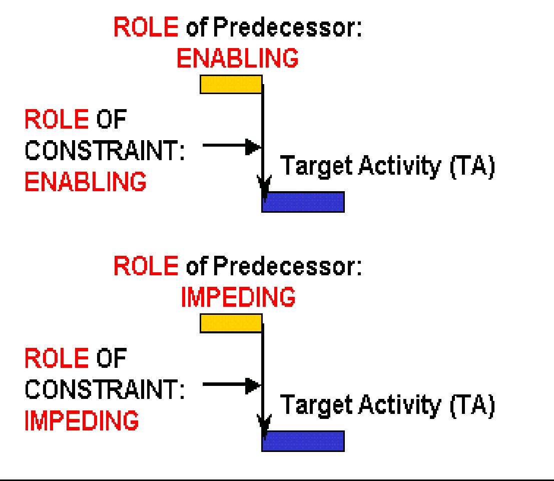


Figure 1.6a: Role (i.e., enabling or impeding) of a predecessor based on the role (i.e., enabling or impeding) of the constraint.

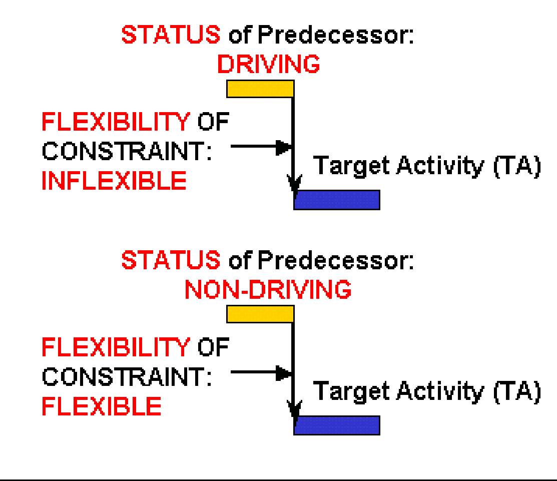


Figure 1.6b: Status (i.e., driving or non-driving) of the predecessor based on the flexibility (i.e., inflexible or flexible) of the constraint.

Figure 1.6: Generalization of the relationships between the role and flexibility constraints and the role and status of immediate predecessors of a target activity.

Figure 1.6 shows my generalization of how planners infer the role and status of immediate predecessors using the rationale of constraints with respect to the activity requiring

[1] Although practitioners commonly use the term "driving" to describe activities that may or may not be delayed, I have not found a formal definition of the term in the AEC literature. Hence, I provide my own definition for this research.

earlier execution, which I call the target activity.[2] Figure 1.6a shows that the role of an immediate predecessor is dependent on the role of the constraint. Figure 1.6b shows that the status of an immediate predecessor is dependent on the flexibility of the constraint.

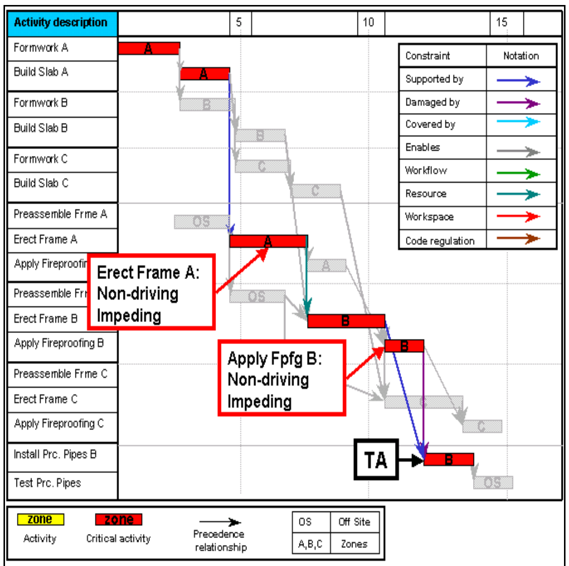


Figure 1.7: Examples of non-driving and impeding activities in the Intel CUB schedule with respect to the target activity (TA).

By contrast, the current CPM framework represents precedence relationships, the temporal aspect of the constraints. Consequently, CPM schedules only distinguish the time-criticality of activities. For example, Figure 1.7 shows the activities that are critical with respect to the activity Install Process Pipes. I call this activity, an activity that is the focus of managerial attention, the “target” activity. The criticality of these predecessor activities implies that they cannot be delayed without affecting the critical path. However, classifying the role and status of the critical activities enables planners to determine opportunities for expediting the target activity by delaying one ore more of the target activity’s predecessors. In Figure 1.7, the activity Erect Frame A is a critical activity that is linked to the target activity by the series of activities: Erect Frame B, Apply Fireproofing B, and Install Process Pipes B. Similarly to the activity Apply Fireproofing B, the activity is also a non-driving, impeding activity. Hence, delaying the activity Erect Frame A can also expedite the activity Install Process Pipes B, if the project manager changes the precedence.

[2] The generalization is only applicable for immediate predecessors of the activity requiring earlier execution. I introduce a more general set of “inference rules” in chapter 3.

In summary, planners need to classify the role (i.e., enabling or impeding) and status (i.e., driving or non-driving) of activities to determine which activities can be delayed to expedite the target activity. Classifying activities in turn requires planners to distinguish a specific constraint with respect to their role and flexibility. As the CPM framework only distinguishes the time-criticality of activities, the process of classifying activities can today only be performed in the planners' minds.

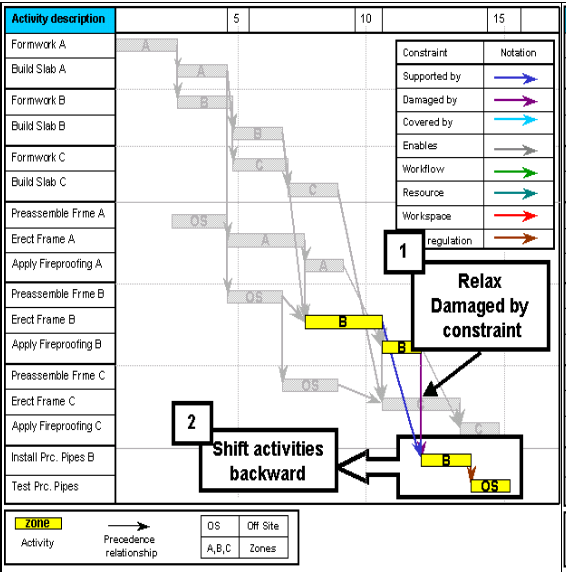


Figure 1.8a: Steps 1 and 2.

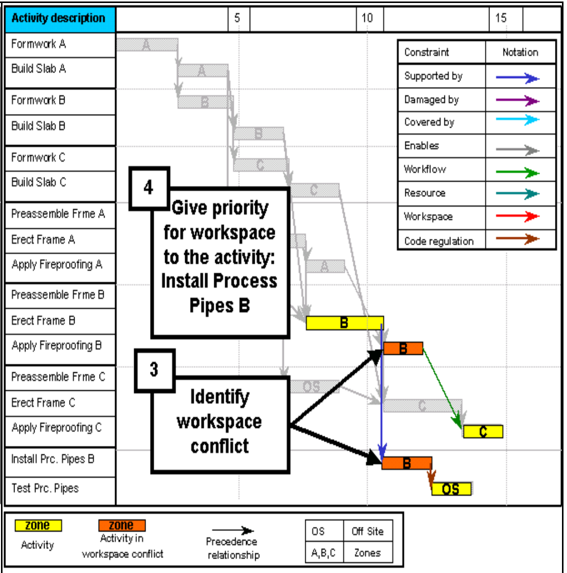


Figure 1.8b: Steps 3 and 4.

Figure 1.8: Steps (1 to 4) required to expedite the activity Install Process Pipes B by delaying activity Apply Fireproofing B.

Having identified activities that can be delayed, planners can make the change in the CPM schedule. However, implementing the sequence change using existing scheduling tools (e.g., Primavera P3) requires planners to perform several steps manually. For example, figures 1.8 and 1.9 show the steps required to recreate the sequencing alternative used by the project manager. The planner relaxes the *damaged by* constraint and shifts the activity and its successors backward[3] (Figure 1.8a, steps 1 and 2). He identifies a workspace conflict (Figure 1.8b, step 3). As the intent is to expedite the activity Install Process Pipes B, he gives priority for workspace to the activity Install Process Pipes B, a driving activity (Figure 1.8b, step 4). Consequently, the planner delays the activity Apply Fireproofing B using the float available with respect to the

[3] Backward with respect to time (i.e., expedites).

activity Install Process Pipes B. I call this float the activity's "target float[4]" (Figure 1.9a). The planner shifts the activity forward[5] until he resolves the workspace conflict (Figure 1.9a, step 5). Finally, he specifies a *workspace* constraint to retain the sequence logic (Figure 1.9b, step 6). Consequently, the activity Install Process Pipes B can now start on day 11.

As the example shows, the role and status of activities is also required during the re-sequencing process to ensure that workspace or resource conflicts are resolved correctly.

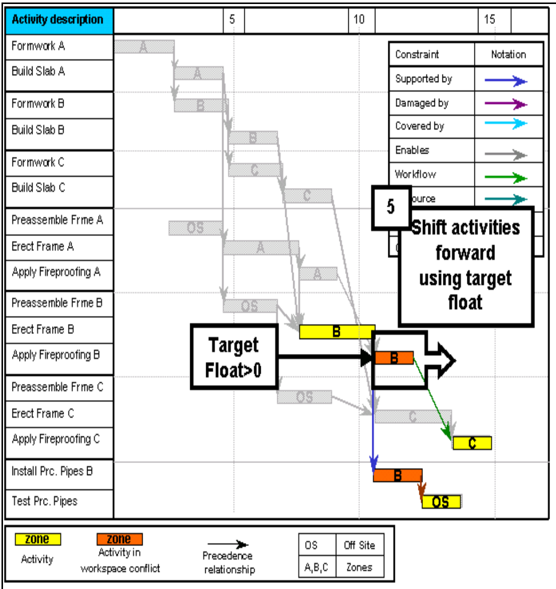


Figure 1.9a: Step 5.

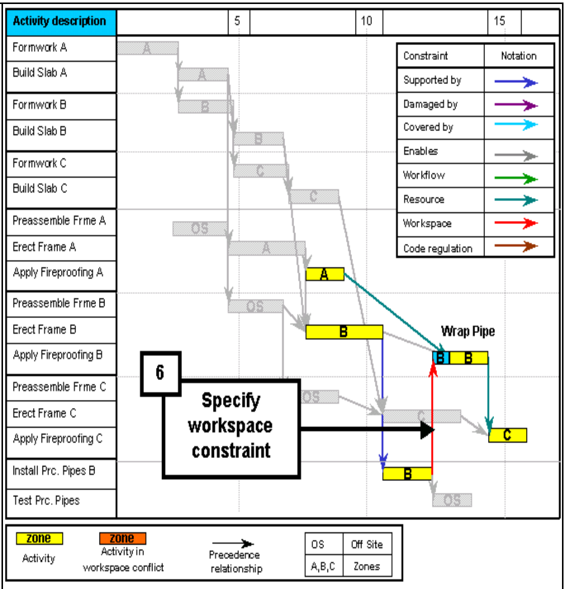


Figure 1.9b: Step 6.

Figure 1.9: Steps (5 and 6) required to expedite the activity Install Process Pipes B by delaying activity Apply Fireproofing B.

In summary, identifying and developing sequencing alternatives requires planners to classify the role and status of activities. Planners classify or infer the role and status of activities based on the rationale and flexibility of constraints. As discussed, however, the existing CPM framework only distinguishes constraints with respect to the temporal relationships between activities. Consequently, the identification and re-sequencing process is manual and ad hoc. The following section describes these limitations in more detail.

[4] The target float can be calculated using CPM's method of calculating activities' total float (TF). However, the calculation is performed with respect to the target activity, not to the end of the project. In the test case example, the activity Apply Fireproofing B is no longer a predecessor of the target activity Install Process Pipes B. In this case, I assume that the activity Apply Fireproofing B has positive target float.

[5] Forward with respect to time (i.e., delays).

1.2 LIMITATIONS OF CURRENT PRACTICE AND RELATED GOALS

The test case illustrates the limitations of CPM-based schedules for modifying activity sequences correctly and quickly. This section summarizes these limitations and the primary goals of this Ph.D. research.

(1) Representation of sequencing rationale not formalized: As shown in the case example, planners implicitly distinguish precedence relationships as different types of specific constraints (e.g., supported by and damaged by, etc.), and conceptually classify specific constraints with respect to their role and flexibility. Subsequently, they use this classification to determine the role and status of activities. However, the current CPM framework only distinguishes the temporal aspects of constraints, and hence planners cannot correctly describe the rationale for constraints and the classifications they associate with specific constraints. In addition, the CPM framework does not provide a way for planners to describe sequencing rationale consistently i.e., use the same specific constraints to describe the identical rationale for activity sequences that may exist within a single project, or in different projects.

The inability to describe sequencing rationale explicitly can make it difficult for multiple project members to interpret and keep track of the sequencing logic behind construction schedules. It also makes it difficult for planners to keep track of the individual classifications they associate with the specific constraints.

Hence, the first goal of this research was to develop a formal representation of sequencing rationale that enables planners to describe correctly and consistently the different types of specific constraints and the classifications they associate with the specific constraints.

(2) *Process of classifying the role and status of activities not formalized:* As shown in the case example, planners classify the role and status of activities to identify suitable activities that when delayed expedite an activity requiring earlier execution (i.e., target activity). In addition, planners prioritize the activities based on their role and status to re-sequence activities correctly. The test case also demonstrated that planners infer the role and status of activities based on the classification (i.e., role and flexibility) they make for specific constraints. As discussed, the CPM framework does not explicitly represent the rationale for activity sequences and consequently only classifies the time-criticality (i.e., zero total float) of activities. Thus, the process of inferring the role and status of activities today can only be performed in the planners' minds. Practically, manually inferring the role and status of activities for large and complex CPM networks can become time-consuming and error-prone.

Hence, the second goal of this research was to formalize the planners' classification process, so that a computer system can utilize a formal representation of sequencing rationale to automatically classify the role and status of activities in a CPM network schedule.

(3) Process of identifying and developing sequencing alternatives not formalized: As shown in the case example, planners conceptually use the role and status of activities to identify activities to delay and to prioritize activities when developing sequencing alternatives. However, the current CPM framework does not support planners in re-sequencing activities with the goal of expediting specific target activities. Consequently, using CPM-based schedules to identify and re-sequence activities is today a manual and an ad hoc process, which limits planners in exploring different sequencing scenarios for expediting particular activities during the course of a project.

Consequently, the third goal of this research was to develop a formal identification and re-sequencing process that guides planners in correctly and rapidly developing sequencing alternatives.

This section summarized the limitations of current practice in using CPM schedules to develop sequencing alternatives and the primary goals of this research. The following section describes the related research questions addressed in this Ph.D. research.

1.3 RESEARCH QUESTIONS

This Ph.D. research addresses the following three primary research questions associated with the three goals discussed in the previous section:

RQ (1) *How can sequencing constraints be formalized to enable planners to describe their rationale for activity sequences <u>correctly and consistently</u>?*

RQ (2) *How can the process of classifying activities using the formal representation of RQ 1 be formalized to <u>automatically</u> classify activities in a CPM network?*

RQ (3) *How can the identification and re-sequencing process be formalized to assist planners in developing sequencing alternatives <u>correctly and rapidly</u>?*

There is a bi-directional relationship between the three research questions. The first research question explores the need for a formal representation of construction sequencing rationale that enables planners to describe the different types of specific constraints and also classify the specific constraints individually with respect to their role and flexibility. The second research question builds on the first. It investigates how to formalize a mechanism that automatically classifies the role and status of activities given a CPM network where the rationale for activity sequences has been explicitly described using the formal representation of the first research question. Hence, the design of the representation for sequencing rationale formalized in the first research question determines how the mechanism in the second research question is formalized.

The formal representation and mechanism are developed with the goal of enabling planners to develop sequencing alternatives correctly and rapidly. Hence, the third research question addresses the need for integrating the formalizations of the first two research questions in a formal process that supports planners in identifying appropriate activities to delay and re-sequencing activities to resolve potential resource or workspace conflicts.

The following three sections below articulate each of the research questions in detail.

RQ (1) *How can sequencing constraints be formalized to enable planners to describe their rationale for activity sequences <u>correctly and consistently</u>?*

The first research question addresses the need for a formal representation for sequencing rationale that enables planners to describe sequencing rationale explicitly in CPM schedules.

As discussed in Section 1.2, a formal representation of sequencing rationale needs to enable planners to not only describe the particular sequencing rationale using specific constraints, but also <u>describe explicitly the classification</u> that planners make for a specific constraint with respect to its role and flexibility.

Hence, the research challenge was to develop a classification schema that is “disjoint” (i.e., exclusive) enough to represent <u>correctly</u> the unique role and flexibility that planners assign for a specific constraint, and also exhaustive (i.e., comprehensive) enough to represent the role and flexibility planners assign for the different types of specific constraints that exist in construction schedules. In addition, a specific constraint can be used several times within a project, and on different projects. Hence, the specific constraints need to be reusable and “specializable” (i.e., customizable) so that planners can describe sequencing rationale <u>consistently</u>.

Previous research studies on classifying sequencing rationale (e.g., Wiest and Levy, 1969; Paulson, 1971; Birrell, 1980; Echeverry et al., 1991; Kähkönen 1993; Ballard and Howell; 1994; Aalami et al., 1998a) developed classification schemas that primarily classify rationale with respect to its “origin”. For example, Echeverry et al. (1991) classify sequencing rationale with respect to physical component relationships, trade interactions, and code regulations. However, they do not address how to classify sequencing rationale in a way that enables planners to distinguish disjointedly the different types of sequencing rationale with respect to their role.

Previous research studies on representing sequencing rationale (e.g., Kähkönen 1993; Aalami et al., 1998) in construction AI planning systems do not explicitly represent the role of the specific constraints and do not enable planers to reuse and customize the specific constraints. In addition, they formalized the representation for sequencing rationale with the goal of generating correct plans, rather than enabling a computer system to infer the role and status of activities in an existing plan.

My answer to the first research question defines an ontology that models a classification schema that meets the criteria described above. Chapter 2 describes how I developed and implemented the ontology. The chapter also presents the validations performed to demonstrate that the ontology meets the criteria described above. The conclusions chapter summarizes the contribution associated with the first research question.

RQ (2) *How can the process of classifying activities using the formal representation of RQ 1 be formalized to* ***automatically*** *classify activities in a CPM network?*

The second research question addresses the requirement for a "classification" mechanism that automatically infers the role and status of activities given a CPM network schedule, where the rationale for activity sequences has been explicitly described using the formal representation for sequencing rationale addressed in the first research question.

To classify a typical activity in a CPM network, planners need to identify unique "paths" that link an activity to the activity requiring earlier execution (i.e., the target activity), which I call "network chains". Subsequently, they need to infer the role and status of an activity based on the role and flexibility of the individual constraints in these network chains. Hence, automating the inference process requires designing an algorithm that correctly and automatically identifies relevant network chains in a CPM network, and formalizing a set of inference rules (i.e., axioms) that generalizes how planners infer the role and status of activities given multiple network chains.

Prior research efforts have also classified activities to perform construction analyses (Levitt and Kunz, 1985; Russell and Wong, 1993; Seibert et al., 1996; Aalami et al., 1998a; Bentley Systems, 1997). A few of these classifications (e.g., Aalami et al., 1998a) also attempt to provide "context" to activities, in particular to the "role" or "function" an activity has with respect to the overall project. However, these classification approaches place the burden of classifying activities on the planner.

My answer to the second research question defines a classification mechanism that consists of a network chain search algorithm and inference rules. The network chain search algorithm adopts Warshall's transitive closure algorithm (Warshall, 1962) to identify unique network chains between activities in a CPM network. The inference rules formalize the planner's inference process for inferring the role and status of activities based on the role and flexibility of constraints existing in an activity's relevant network chains. Chapter 3 describes how I formalized and implemented the classification mechanism.

RQ (3) *How can the identification and re-sequencing process be formalized to assist planners in developing sequencing alternatives correctly and rapidly?*

The third research question addresses the need to develop a formal process that utilizes the formal representation and classification mechanisms defined in the first and second research questions to enable planners to develop sequencing alternatives correctly and rapidly.

Planners identify activities to delay by classifying the role and status of activities on the critical path with respect to a target activity. To re-sequence activities, planners first use the role and status of activities that are in workspace or resource conflict to decide *which* needs to be delayed. Subsequently, planners "shift forward" the activity selected to delay either by using the activity's float or by relaxing flexible constraints in the selected activity's network chain. Hence, the research challenge was to model why and how planners re-sequence activities in a formal and general way, so that planners in turn could use the formalized processes to correctly and rapidly develop sequencing alternatives for different project schedules.

Prior research efforts have developed CPM-based techniques with the goal of accelerating the overall schedule duration. These techniques include time-cost trade-off analysis, (e.g., Fondahl, 1961; Meyer and Shaffer, 1963; Paulson, 1971) and resource-constrained scheduling (e.g., Crandall, 1985; Chang et al., 1989). These techniques assume that activity sequences in a CPM network are fixed and that resources can be added to accelerate the schedule duration.

These techniques do not allow planners to re-sequence activities with the goal of expediting a specific target activity. They also do not promote planners to use their judgment to make decisions when developing sequencing alternatives, as they are frequently reduced to rigidly coded computer algorithms (Paulson, 1973). I also investigated "replanning" techniques in general-purpose AI planning systems. However, these replanning techniques were not designed to meet the domain-specific requirements (i.e., a construction specific ontology and its utilization) needed to re-sequence activities correctly.

My answer to the third research question defines a formal process that uses the ontology and classification mechanism to assist planners in developing sequencing alternatives correctly and rapidly. The identification process identifies a set of "candidate" activities (i.e., activities to delay to expedite a target activity). Planners can delay one of these activities to expedite a target activity. The re-sequencing process subsequently assists planners by using a set of pre-defined priority rules to delay activities correctly. Planners can repeat the process to develop multiple sequencing alternatives. Chapter 4 describes how I formalized and implemented the formal process, and also the validation studies performed.

The next section overviews the system architecture of the CLCPM (i.e., Constraint-Loaded CPM) prototype system that implements the formalizations I developed to answer the three research questions. The implementation was required to demonstrate the power and generality of the three formalizations.

1.4 FORMAL APPROACH FOR DEVELOPING SEQUENCING ALTERNATIVES

To answer the three research questions, I formalized (1) an ontology that correctly models the classifications that planners make for different types of specific constraints with respect to their role and flexibility, (2) a classification mechanism that automatically classifies the role and status of activities in a CPM network, (3) and a formal process that guides planners throughout the development of sequencing alternatives. I developed prototype software, called CLCPM (i.e., Constraint-Loaded CPM), by implementing the three formalizations developed in this research.

Figure 1.10 shows an IDEF0 diagram for the CLCPM prototype software. As shown in the first module, CLCPM (Figure 1.10) takes an existing construction CPM schedule as input. It assumes that activity sequences are represented using exclusively FS precedence relationships. It also assumes that workspace and resources are defined and allocated.

In the first module, planners describe the rationale for precedence relationship using the ontology implemented in CLCPM. The ontology consists of four abstract types of constraints: enabling-flexible, enabling-inflexible, impeding-flexible and impeding-inflexible. Users can select one of these generic constraints to define "project-independent" types of constraints. The user-defined constraint inherits the same values for its role and flexibility from the abstract type. For example, a *damaged by* constraint is created by classifying it as a subtype of the abstract type impeding-inflexible. Correspondingly, the *damaged by* constraint has the values impeding and inflexible for its role and flexibility. Using this framework, I formalized project-independent constraint types that represent specific instances of sequencing rationale compiled from existing literature (Darwiche et al., 1988; Echeverry et al., 1991; Kähkönen, 1993; Aalami and Fischer, 1998). I focused in particular on the rationale of constraints that affect the installation operations of multiple trades working in common workspaces. The rationale for these constraints includes: physical relationships (e.g., supported by) between components, trade interactions (e.g., workspace competition) and code regulations (e.g., safety requirements).

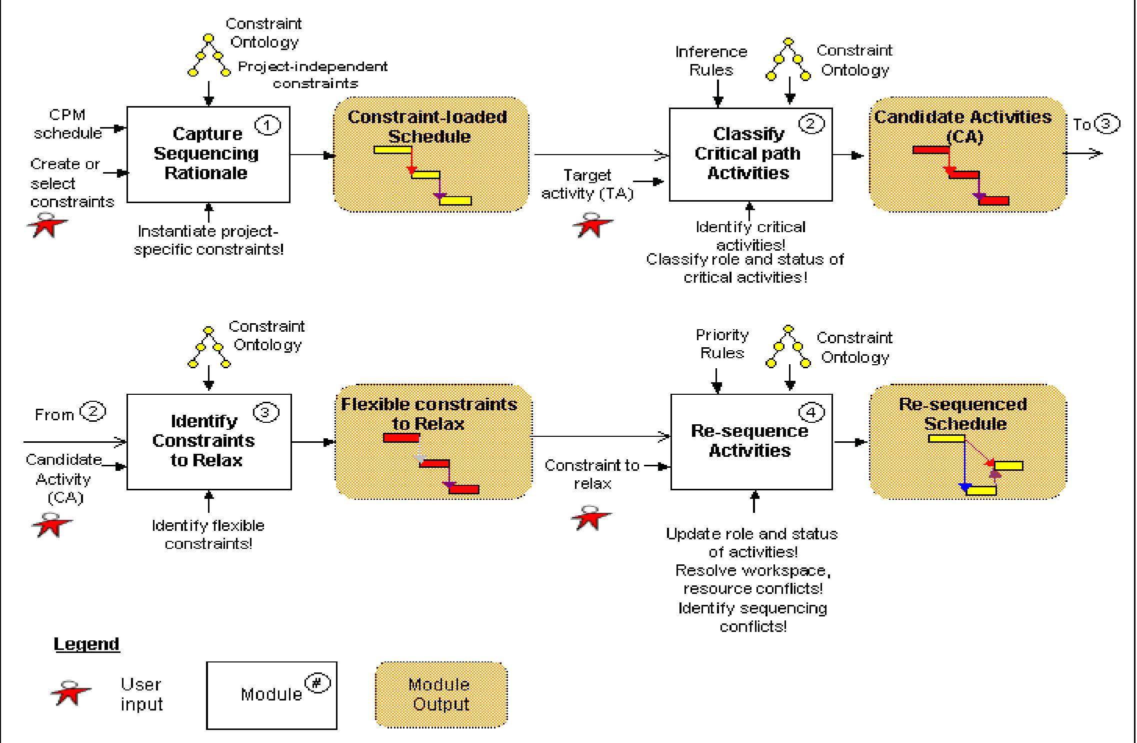

Constraint Ontology
Project-independent constraints
CPM schedule
Create or select constraints
Capture Sequencing Rationale
1
Instantiate project-specific constraints!
Constraint-loaded Schedule
Target activity (TA)
Inference Rules
Constraint Ontology
Classify Critical path Activities
2
Identify critical activities!
Classify role and status of critical activities!
Candidate Activities (CA)
To 3
Constraint Ontology
From 2
Candidate Activity (CA)
Identify Constraints to Relax
3
Identify flexible constraints!
Flexible constraints to Relax
Constraint to relax
Priority Rules
Constraint Ontology
Re-sequence Activities
4
Update role and status of activities!
Resolve workspace, resource conflicts!
Identify sequencing conflicts!
Re-sequenced Schedule
Legend
User input
Module #
Module Output

Figure 1.10: Detailed IDEF0 of CLCPM prototype software.

For a specific project schedule, users can either create a new type of project-independent constraint, or select one of the project-independent constraints. Subsequently, the system instantiates the selected constraint type as a project-specific constraint between the selected activities. Users can subsequently customize the default values for flexibility to reflect the circumstances of a specific project. The output of the first module is a CPM schedule where the rationale for every precedence relationship is explicitly represented, i.e., a constraint-loaded schedule.

In the second module, users select the activity requiring earlier execution, (i.e., the "target activity"). CLCPM identifies activities that are time-critical with respect to the target activity, that is, activities that have zero target float. CLCPM classifies the role and status for these critical activities using the formal classification mechanism. Specifically, CLCPM identifies unique paths or "routes" that link the individual activities to the target activity. I call such paths an activity's "network chain". (For example, Figure 1.7 in Section 1.1 shows an example of the network chain for the activity Erect Frame A.) CLCPM then uses predefined inference rules to classify each of these activities based on the role and flexibility of the constraints in each activity's network chains. CLCPM uses the classification to determine activities that are impeding and non-driving (i.e., can be delayed) with respect to the target activity, which I call candidate activities. Hence, the output of the second module is a list of candidate activities.

In the third module, users select one of the candidate activities to delay. CLCPM identifies flexible constraints between the candidate activity and the target activity, and outputs these constraints as a list for users to choose. If more than one flexible constraint exists, users can decide which constraint to relax by comparing the constraints' degree of flexibility (DOF).

In the fourth module, CLCPM relaxes the chosen constraint and subsequently shifts the target activity and its successors backward[6]. Consequently, a workspace or resource conflict can occur due to the shifted activities. CLCPM assumes that two overlapping activities requiring the same workspace (e.g., zone) or resource results in a workspace or resource conflict.

CLCPM uses the formal classification mechanism to update the role and status of activities and uses predefined priority rules (e.g., driving over non-driving) to determine which activity needs to be delayed. If the activity that needs to be delayed has positive target float, CLCPM tries to resolve the workspace or resource conflict by using the activity's target float to shift the activity forward. However, if the activity does not have target float, and is linked to the target activity by one or more network chains, a "sequencing conflict" can occur. In such cases, further delaying the activity will in turn delay the target activity. Hence, the only way to resolve the sequencing conflict is to relax flexible constraints in the activity's network chains. Hence, CLCPM identifies flexible constraints that need to be further relaxed. The user can subsequently select one of these constraints. The relaxation of these constraints enables the activity to be delayed further, resolving the sequencing conflict and correspondingly the workspace and resource conflict. Finally, CLCPM instantiates a *resource* or *workspace* constraint to ensure that the logic remains correct. The user can repeat the entire process to generate several sequencing alternatives.

Chapter 2 elaborates how I formalized the ontology to enable planners to describe sequencing rationale as shown in the first module. Chapter 3 describes how I formalized the classification mechanism used to classify and update the role and status of activities in modules 2

[6] Backward with respect to time (i.e., expedites).

and 4. Chapter 4 elaborates how I formalized the steps of modules 2, 3 and 4 as a formal identification and re-sequencing process.

1.5 RESEARCH METHODOLOGY

I based my methodology for this research primarily on three studies: Kunz and Fischer (2002), Russell and Norvig (1995), and Paulson (1971). Their research guided me in formulating and solving engineering problems, specifically through the development and implementation of knowledge-based systems using AI based representation and reasoning approaches.

Kunz and Fischer (2002) provide the overall framework for the research process. They state that research starts with an observation of a practical problem and the associated engineering problem, which is followed by an intuition to solve the engineering problem. The intuition must be accompanied by a thorough investigation of theoretical points of departure from which formal representations and reasoning mechanisms can be adopted or built upon. The formalizations subsequently need to be implemented so that validation studies can be performed. The validation needs to demonstrate that the formalizations address the engineering problem in a formal and general way and consequently resolve the practical problem.

Secondly, a review of AI literature on knowledge engineering and ontology development (e.g., Gruber, 1993; Davis et al., 1993; Russell and Norvig, 1995) also helped me in determining how the representation and reasoning mechanisms need to be formalized. In particular, Russell and Norvig (1995) state that for a specific domain, a vocabulary or ontology needs to be defined and general knowledge about the domain needs to be encoded using the ontology as axioms or rules. Russell and Norvig (1995) also emphasize the importance of a well-thought out ontology that enables specific objects of the domain to be instantiated and supports the correct inference of knowledge in that domain.

Hence, in addition to Kunz and Fischer's framework, the research process included performing numerous "Gedanken" experiments (Chaitin, 1965) to design an ontology for sequencing rationale that enables the instantiation of specific constraints and also supports the correct inference of the role and status of activities.

Thirdly, Paulson (1971) and a review of replanning techniques in AI planning systems (e.g., Smith, 1989; Zweben et al. 1993; Aarup et al., 1994) also helped me in determining the general philosophy for how knowledge based systems should be designed. These studies emphasize the need to develop planning techniques as a decision support tool, i.e., tools that support planners to focus their judgment and expertise in making planning decisions, in contrast to optimization techniques that rely too heavily on unrealistic assumptions. Hence, Paulson's "human judgment-oriented" philosophy also guided how the formalizations should be designed and implemented.

Based on the three studies, my methodology consisted of the following seven steps: (1) observation of the problem (2) intuition, (3) Gedanken experiments, (4) research background, (5) formalizations, (6) implementation and (7) validation.

The following section describes each step in detail.

(1) Observation of the problem

The primary motivation for this research stems from my experience working on the Intel FAB 22 project. As discussed in the test case example, the Intel Central Utility Building (CUB) project illustrated how the project manager modified initial activity sequences to expedite the installation of process pipes.

I observed another similar instance during the construction of the main FAB building, where activity sequences were again modified to expedite the activity Make up Air Handler units to meet a critical milestone activity, FAB Blow down.

I had also observed a similar problem on a previous project I was involved in, the McWhinney project. On this project (a two-story office building), activities for interior HVAC ducts needed to be expedited to enable the earlier testing and start up of HVAC systems.

On another project I visited, the carpet installation had to be expedited to meet an inspection deadline.

Hence, I observed that re-sequencing activities during the course of a project was a frequent occurrence and was one way for construction planners to address changing project conditions. However, while existing scheduling tools (e.g., Primavera P3, Microsoft Project) based on the CPM framework mechanically allow planners to edit activity precedence, they do not provide a principled framework to support planners in re-sequencing activities systematically. Consequently, I noticed that re-sequencing decisions were often made in an ad hoc way and without the aid of CPM-based schedules. I also observed that without a formal approach for developing sequencing alternatives, planners found it difficult to quickly and thoroughly explore potential sequencing alternatives.

Realizing the need for a formal approach for developing sequencing alternatives, I examined the conceptual process involved in re-sequencing activities to expedite the activity Install Process Pipes B for the Intel CUB schedule. Through discussions with the project manager and superintendent of the Intel CUB building, I identified the conceptual reasoning process they performed in their minds to identify and re-sequence activities.

As discussed in the test case example, the project manager implicitly distinguished precedence relationships with respect to the different types of sequencing rationale, and they classified a particular sequencing rationale with respect to its role and flexibility. He then used this classification to infer the role and status of activities. Subsequently, he used the role and status of activities to identify activities to delay and prioritize activities when re-sequencing activities. The observation showed that an inherent relationship exists between constraints and activities.

However, the current CPM framework does not enable planners to describe sequencing rationale explicitly, and only distinguishes the time-criticality of activities. Hence, the inference process can only be performed in their minds. Consequently, developing sequencing alternatives in a CPM schedule is a time-consuming and error-prone process, especially for complex CPM network schedules.

(2) Intuition

My intuition then was to formalize the inherent relationship between constraints and activities in a general and parsimonious way that was also computer-interpretable, so that a computer system could leverage the formalization to automatically classify the role and status of activities. The automatic classification of activities would enable planners to better understand the physical and technical dependencies between upstream activities in relation to specific downstream activities, and ultimately help planners to develop sequencing alternatives more accurately and quickly.

Given my intuition, I realized the need for a formal representation of sequencing rationale that enables construction planners to describe the rationale for constraints in a CPM network schedule. My intuition was that this formalization should be abstract but simple, for it to be realistically used by planners in practice. I also realized a need for a mechanism that leverages the representation to correctly classify the role and status of activities. Finally, I also realized the need for a formal process that guides planners in using the representation and mechanism to help them in developing sequencing alternatives correctly and quickly.

Correspondingly, I performed Gedanken experiments and investigated existing representation and reasoning approaches using CPM-based schedules in the construction and AI literature.

(3) Gedanken experiments

"Gedanken" experiments are a way to identify generic solutions in computer science by giving a system inputs and looking at the outputs and formulating an automation based on the outputs (Chaitin, 1965).

Using the Intel CUB schedule, I performed numerous "paper-based" Gedanken experiments to develop a formal representation, classification mechanism and process that generalize how planners conceptually develop sequencing alternatives in CPM schedules.

The Gedanken experiments enabled me to identify the main research challenges for this research, and also provided direction and insight as to how these challenges could be met.

For a formal representation of sequencing rationale, I realized that I needed to find a classification schema that enables planners to describe the unique role and flexibility for different types of specific constraints, and also use the classification schema to define specific types of constraints.

I also realized that there existed typical constraints that could be used repeatedly in a single project and from project to project (e.g., *supported by* or *protected by* constraints). Therefore, the representation needed to enable planners to define project-independent types of constraints that they could reuse to describe sequencing rationale in a correct and consistent way.

For the classification mechanism, I realized that the mechanism needs to correctly classify all activities in a CPM network, regardless of the specific types of constraints used to describe sequencing rationale in the schedule. I identified the need for a network chain search algorithm that identifies unique network chains between a typical activity in a CPM network schedule and a given target activity, and the need for inference rules that formalize how planners infer the role and status of activities based on the role and flexibility of constraints existing in an activity's relevant network chains.

For the formal process, I realized that planners identify activities to delay by identifying critical activities that are impeding and non-driving. When re-sequencing activities, I realized that planners use the role and status of activities to prioritize activities that are in workspace or resource conflict. I found that many of the steps involved in re-sequencing activities were repetitive in nature. Hence, I needed to formalize a generic set of steps that guide planners throughout the identification and re-sequencing process.

(4) Research background

While performing these Gedanken experiments, I also performed a literature review of existing research efforts in the areas of 1) construction sequencing rationale classification, 2) sequencing rationale representation in construction AI planning systems, 3) existing activity classification schemas, 4) graph algorithms, 5) CPM-based acceleration techniques, and 6) replanning techniques in AI planning systems.

The results of investigating existing sequencing rationale classifications (e.g., Wiest and Levy, 1969; Paulson, 1971; Echeverry et al., 1991; Ballard and Howell; 1994) and representations (e.g., Kähkönen 1993; Aalami et al., 1998a) were twofold. First, it enabled me to identify and compile specific constraints commonly encountered in construction schedules. Secondly, I found that previous research also identified the need for classifying sequencing rationale and explicitly representing rationale in CPM-based schedules. However, I also found that existing classification schemas do not classify constraints with respect to their role in a disjoint (i.e., exclusive) way. I existing representation approaches in construction AI planning systems do not explicitly represent the role of the specific constraints and do not enable planers to reuse and customize the specific constraints.

Investigation of existing activity classification approaches (e.g., Levitt and Kunz, 1985; Russell and Wong, 1993; Seibert et al., 1996) showed that previous research also identified the need for a richer and domain-specific classification of activities when performing construction schedule analysis. However, the main drawback of these approaches was a manual approach for classifying activities.

By investigating graph algorithms (Tamassia and Tollis, 2002), I was able to identify that network chains between activities in a CPM network could be automatically identified using transitive closure algorithms. Hence, I used Warshall's transitive closure algorithm (Warshall, 1962), as it was simple to encode.

I also performed an extensive review of time-cost trade-off (e.g., Fondahl, 1961; Meyer and Shaffer, 1963; Paulson, 1971) and resource allocation techniques (e.g., Crandall, 1985; Chang et al., 1989) in the construction literature. I found that these techniques fundamentally rely on the CPM framework. Thus, they do not differentiate between constraints that may or may not be relaxed and are limited to prioritizing activities based on their float values. However, the investigation of these approaches exposed me to several important papers, including Fondahl (1961), Wiest and Levy (1969), and Pauslon (1971). Paulson (1971) in particular provided guidance as to how the re-sequencing process should be developed and implemented.

I also investigated replanning techniques (e.g., Smith, 1989; Zweben et al. 1993; Aarup et al., 1994) in general purpose AI planning systems. I found that many of these systems rely on CPM networks as the basic data structure for representing schedules, and also rely on human judgment when making re-sequencing decisions. Most importantly, I realized that these systems also require a domain-specific representation of constraints and associated mechanisms to re-sequence activities.

Consequently, I adopted these approaches but focused on developing a construction domain-specific representation of constraints that is tailored to support mechanisms for developing sequencing alternatives for construction schedules.

Based on the Gedanken experiments and a review of previous research, I formalized an ontology, a classification mechanism and a formal process that together support construction planners in developing sequencing alternatives in CPM schedules.

(5) Formalizations

The three formalizations defined in this research are:

i) An ontology that formalizes a classification schema for sequencing rationale that is disjoint enough to represent correctly the unique role and flexibility of constraints, and also exhaustive enough to represent comprehensively the role and flexibility for different types of construction sequencing rationale. The ontology enables planners to define project-independent constraints that they can reuse and customize to describe sequencing rationale in construction schedules correctly and consistently.

ii) A classification mechanism that automatically infers the role and status of activities given a CPM network schedule where the rationale for activity sequences has been explicitly represented using the ontology developed.

iii) An identification and re-sequencing process that formalizes the planner's rationale for determining *which* activities to delay and *how* the selected activity needs to be delayed when re-sequencing activities. The process utilizes the classification mechanism to classify and prioritize activities, and utilizes the ontology to determine which constraints may be relaxed.

The three formalizations would be meaningless unless I demonstrated that the formalizations enable planners to correctly and quickly develop sequencing alternatives for different construction schedules. Therefore, I needed to implement the three formalizations and perform validations studies to provide evidence for their power and generality.

(6) Implementation

I used Microsoft Visual Basic to implement the formalizations. The reason for using Visual Basic was twofold. First, Visual Basic is an Object Oriented Programming (i.e., OOP) language. "OOP" is a language paradigm that encapsulates the common attributes of objects as variables of a class. Objects can then be created as instances of the class and inherit the values of the variables of the class. In addition, the inherited values of the variables can be overridden with local values.

The OOP language paradigm was a nice fit for implementing the formalizations, in particular for implementing the ontology. I implemented the ontology as a Constraint class in CLCPM, where the variables are the role and flexibility of constraints. The Constraints class also has four subclasses, which represent the four abstract types of the ontology.

Using these subclasses, I implemented the project-independent types of constraints and stored them in CLCPM. If necessary, planners can define a new project-independent constraint and also store the constraint in CLCPM.

Correspondingly, I also implemented the classification mechanism and formal process in Visual Basic.

The second reason for using Visual Basic was to take advantage of the Gantt chart representation of a CPM schedule implemented in Microsoft Project (Microsoft Corporation, 1998). Microsoft Visual Basic allowed me to reference the Microsoft Project library 9.0, which defines the basic classes for representing a Gantt chart, such as Tasks class and Precedence class. Hence, I implemented CLCPM to import task and precedence relationship objects, and subsequently enabled planners to describe the rationale for a precedence relationship by assigning one or more project-independent types of constraints to the precedence relationship.

In addition, I could also highlight the role and status of activities directly in the Gantt chart using the implementation of the classification mechanism. By implementing the formal process, CLCPM could guide users during the individual steps involved in identifying activities to delay and re-sequencing activities.

(7) Validation

To claim a contribution for the ontology developed, I performed three retrospective case studies and one charrette test (Clayton et al., 1998). The three retrospective cases involved classifying sequence rationale that exists in schedules for different phases of construction and for different types of work. The three retrospective cases provide evidence that the ontology is <u>disjoint and exhaustive,</u> since the ontology correctly represents the classification planners make for different types of specific constraints.

The charrette test involved using eight graduate students to "interpret" the rationale for activity sequences for two projects, in which one half of the students used a "constraint-loaded" schedule developed in CLCPM, and the other half of the students interpreted the rationale using a conventional CPM schedule. The test result provides evidence that the project-independent constraints can be used to describe sequencing rationale correctly and consistently, by showing that resultant constraint-loaded schedules make it more likely for planners and other project participants to correctly interpret sequencing rationale than conventional CPM schedules.

To create evidence for my claim of contributions for the classification mechanism and the formal process, I also performed three retrospective case studies and one charrette test. For the retrospective case studies, I used three different construction schedules that describe the sequences for different phases of construction projects. For each of these schedules, I used the formal identification process to identify the enabling activities and the candidate activities for a single target activity and subsequently used the re-sequencing process to expedite the target activity until all candidate activities were exhausted. Subsequently, I confirmed with an experienced project scheduler whether the candidate activities identified and the sequencing alternatives I developed were indeed correct. Therefore, the retrospective cases provide evidence that the process is <u>formal</u> and <u>general</u>, since it demonstrates that the process <u>correctly</u> identifies and re-sequences activities for different construction schedules.

For the charrette test, I used the same eight graduate students to identify and re-sequence activities for two construction schedules, where one half of the students used a "constraint-loaded" schedule using CLCPM, and the other half used a conventional scheduling tool, Microsoft Project (MSP). Then I compared how correctly and quickly the two groups could identify and re-sequence activities given a single target activity. Hence, the charrette test provides evidence that the process is <u>formal</u>, since it demonstrates that the process enables users to develop sequencing alternatives <u>more correctly and rapidly</u> than a conventional CPM scheduling tool.

1.6 READER'S GUIDE TO THE THESIS

This thesis consists of the Introduction and Conclusion chapters to provide an overview of my research and contributions, and of three chapters (Chapter 2, 3, 4) in between, focusing on the three specific contributions of my research, i.e., an ontology for sequencing rationale, a classification mechanism and a formal process for developing sequencing alternatives. Each of the intermediate chapters focus on the investigations, formalizations and validations I performed to address each of the three research questions of this Ph.D. research. However, the reader should keep in mind that there is a relationship between them; each subsequent chapter builds on the formalizations defined by the previous chapter.

Below is a quick description of each of the chapters of my thesis:

Chapter 2 describes a formal representation for describing sequencing rationale explicitly in CPM schedules. It explains how I built on existing classification and representation approaches to develop an ontology for representing sequencing rationale, and how I implemented and validated the ontology using the prototype software, CLCPM.

Chapter 3 describes a formal classification mechanism for automating the classification of activities in a CPM network. It explains how I used transitive closure algorithms to automatically identify network chains and how I developed inference rules that formalize the planners' inference process for classifying the role and status of activities. It also describes how I implemented the mechanism to be used in conjunction with the ontology implemented in CLCPM.

Chapter 4 describes a formal identification and re-sequencing process that formalizes the steps involved in developing sequencing alternatives in a CPM network. It explains how I integrated the ontology and the classification mechanism in a process to assist planners in developing multiple sequencing scenarios. It also describes how I implemented the process and validated the process using CLCPM.

Chapter 5 gives the Conclusions of this work. It summarizes the contributions I claim for this research in the areas of construction sequencing rationale classification and representation, and in the construction planning and scheduling domain. It states the limitations of this research and suggests possible future research areas.

CHAPTER 2. FORMAL REPRESENTATION OF CONSTRUCTION SEQUENCING RATIONALE

2.1 INTRODUCTION

As discussed in Chapter 1, planners need to understand the rationale for activity sequences when identifying and re-sequencing activities to develop sequencing alternatives. In particular, planners need to understand sequencing rationale to infer the role and status of activities, since they use the activity classification to identify activities to delay and to prioritize activities when re-sequencing activities. Currently, inferring the role and status of activities in a CPM schedule is a time-consuming and error-prone process. This is because the CPM framework only represents the temporal aspects of constraints (i.e., precedence relationships), and the rationale for activity sequences is implicit. Hence, a goal of this research was to formalize a representation that enables planners to explicitly describe sequencing rationale in a way that subsequently supports a computer-system to utilize the representation to automatically classify activities. In particular, I focused on formalizing a representation for sequencing rationale that typically exists in construction schedules.

Meeting this goal first required conducting "Gedanken" experiments (Chaitin, 1965) in which I investigated the different types of sequencing rationale (e.g., supported by, protected by, workspace constrained by etc.) that exist in construction schedules and studied how planners use sequencing rationale to classify the role and status of activities. I realized that planners conceptually classify a particular sequencing rationale (i.e., a "specific" constraint such as a supported by) with respect to its role and flexibility, and it is this classification that planners use to infer the role and status of activities.

Therefore, a formal representation of sequencing rationale needs to enable planners to not only describe the particular sequencing rationale using specific constraints, but also describe explicitly the classifications that planners make for a specific constraint with respect to its role and flexibility.

The requirements in turn requires formalizing a classification schema that is disjoint enough to represent correctly the unique role and flexibility that planners assign for a specific constraint, and also exhaustive enough to represent comprehensively the role and flexibility planners assign for the different types of specific constraints that exist in construction schedules. In addition, I also realized that a specific constraint could be used several times within a project, and on different projects. Hence, the specific constraints need to be reusable and "specializable" (i.e., customizable) so that planners can describe sequencing rationale consistently.

To develop a representation that meets such requirements, I investigated prior research in the areas of classifying and representing construction sequencing rationale. Researchers as early as the 1960's and 70's (Fondahl, 1961; Wiest and Levy, 1969; Pauslon, 1971; Antill and Woodhead, 1970; Davis, 1974) also recognized that the CPM framework does not adequately represent sequencing rationale. Hence, they have attempted to develop classification schemes to better understand how sequencing rationale affects construction activity sequences. However, these classification schemas do not distinguish specific constraints with respect to their role in a disjoint way. Researchers in the construction AI planning domain have formalized representations that can be used to automatically sequence activities (Kähkönen, 1993; Aalami et al., 1998a). However, these approaches do not explicitly represent the role of the specific constraints and do not enable planers to reuse and customize the specific constraints.

Consequently, I developed an ontology (i.e., a computer vocabulary) based on the Gedanken experiments and investigations of previous classification and representation approaches. The ontology consists of four "abstract" types of constraints that formalize the conceptual classifications planners make for specific constraints with respect to their role and flexibility. Planners can use the ontology to define specific types of constraint as instances of one of the abstract types, and subsequently reuse and customize the specific constraints to describe sequencing rationale in construction schedules. Consequently, the ontology enables planners to develop a "constraint-loaded" schedule, where the individual classifications for the specific constraints are retained. The constraint-loaded schedule in turn enables a computer system to use the individual classifications to automatically infer the role and status of activities. The mechanism for automating the inference process is described in Chapter 3.

This Chapter focuses on describing the investigations performed and the formalization of the ontology in detail. I also present three retrospective case studies and a charrette test I performed to validate the ontology. The retrospective cases provide evidence that the ontology is disjoint and exhaustive, as it demonstrates that the ontology correctly describes the unique role and flexibility of different types of specific constraints that exist in different types of construction schedules. The charrette test provides evidence that the specific types of constraints created using the ontology is reusable and "specializable," as it demonstrates that a "constraint-loaded" schedule enables a more correct and consistent interpretation of sequencing rationale than a conventional CPM schedule. I conclude with a discussion of the immediate implications of the formal ontology, and its limitations.

The following section introduces a use case to describe the requirements for an ontology to represent construction sequencing rationale.

2.1.1 Motivating Case Example

To illustrate the requirements of a formal representation of construction sequencing rationale, I revisit the Intel CUB test case example. Figure 2.1 shows how the project manager for the Intel CUB inferred the role and status of the activities Erect Frame B and Apply Fireproofing B with respect to the activity Install Process Pipes B. The project manager generically classified in his mind the *supported by* constraint as an "enabling" and "inflexible" type of constraint. He also classified the *damaged by* constraint as an "impeding" and "flexible" type of constraint. Based on the conceptual classification of the constraints, he in turn inferred the activity Erect Frame B to be an "enabling", type activity and an activity that could be delayed. He also inferred the activity Apply Fireproofing B to be an "impeding" activity that could be delayed.

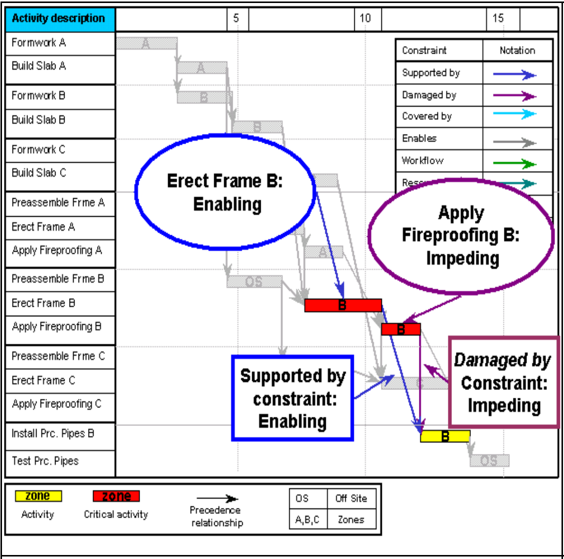


Figure 2.1a: Role of activities based on the role of constraints.

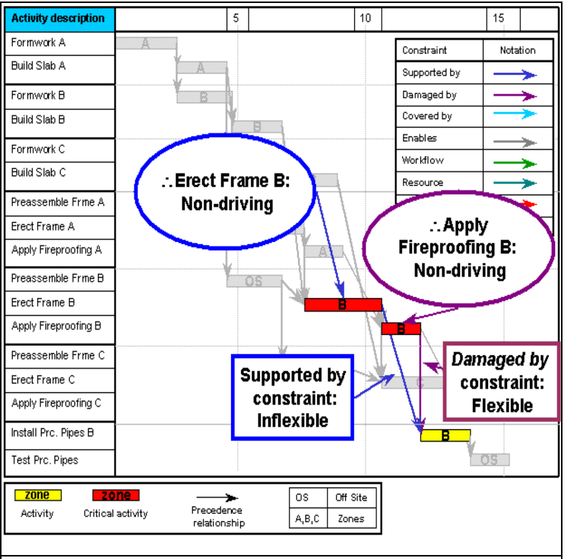


Figure 2.1b: Status of the activities based on the flexibility of constraints.

Figure 2.1: Example of planners' inference process.

Figure illustrates planners' inference process for the activities Erect Frame B and Apply Fireproofing B of the Intel CUB test case.

The example shows that an inherent relationship exists between (the rationale of) constraints and (the behavior of) activities that is today inferred in the planners' minds. Although CPM schedules do not explicitly represent the rationale for activity sequences, planners implicitly distinguish precedence relationships as different types of specific constraints (e.g., supported by and damaged by, etc.), and conceptually classify the specific constraints with respect to their role and flexibility. As discussed in Chapter 1, the role of a constraint is the functional property implicit in a particular sequencing rationale, and the flexibility of a constraint refers to whether a particular sequencing rationale may or may not be relaxed. As the example shows, it is the conceptual classification of sequencing rationale that planners use to subsequently infer the role and status of activities.

Currently, the inference of activities can only be performed in the planners' minds. Furthermore, the current CPM framework does not distinguish the specific constraints, and does not represent the classifications that planners make for the specific constraints. Hence, a goal of this research was to develop a representation that formalizes the planner's conceptual classification in a way that enables planners to instantiate specific constraints that they can use to describe sequencing rationale explicitly. The formalized classification can in turn be used by a computer system to infer the role and status of activities.

Meeting this goal can be met by formalizing the representation as an ontology. An ontology is a "domain-specific vocabulary" (Fikes, 1996) or a "set of agreements about a set of concepts" (Fox and Gruninger, 1994). Ontologies are often developed when objects or knowledge in a domain need to be organized into more general categories[7] based on the objects' common characteristics, so that reasoning can take place at the level of categories rather than at the level of individual objects (Russell and Norvig, 1995). Gruber (1993) also notes that a primary goal of an ontology is to enable the instantiation of more specific objects and to sanction inferences.

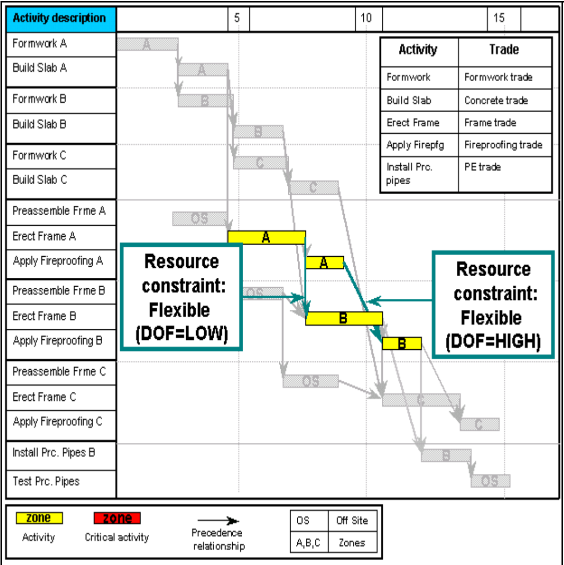
The purpose of developing an ontology concur with the requirements observed for a formal representation of sequencing rationale. That is, the formal representation needs to organize specific constraints into a general categorization (i.e., a classification schema) based on the specific constraints' common characteristics (i.e., their role and flexibility), so that a computer system can reason (i.e., infer the role and status of activities) using the categorization, rather than the individual specific constraints. The formal representation also needs to enable the instantiation of more specific constraints for planners to describe the rationale for constraints.

Furthermore, Russell and Norvig (1995) state that when designing an ontology to represent categories, it is critical for the categorization to be as "disjoint" and "exhaustive" as possible. That is, two or more categories in a categorization need to be disjoint enough so that the instantiated objects have distinct properties from other objects, which in turn avoids the

[7] Categories are also called classes, collections, kinds and concepts by other authors (Russell and Norvig, 1995).

occurrence of inference conflicts. The categories in a categorization also need to be exhaustive enough so that they collectively constitute a comprehensive decomposition of the domain concepts.

The design criteria also apply for representing the classification schema of sequencing rationale. It is critical that the representation formalize a classification schema that is disjoint enough to correctly distinguish the individual role and flexibility of specific constraints, and also exhaustive enough to instantiate the different types of specific constraints that may exist in construction schedules. For example, for the two specific constraints shown in Figure 2.1, the role of the two constraints can be distinguished correctly as enabling and impeding. For just two constraints then, a classification schema that distinguishes the role explicitly as either enabling and impeding, is disjoint enough to correctly describe the individual role of the constraints, and exhaustive enough to instantiate the two different types of specific constraints. Obviously, the goal is to formalize a classification schema that is disjoint and exhaustive enough not just for two constraints, but also for the many other types of specific constraints (e.g., protected by, workspace, resource, etc.) that may exist in construction schedules. Only then can planners use the formalized classification schema to describe the different types of sequencing rationale and their unique role and flexibility correctly in construction schedules.



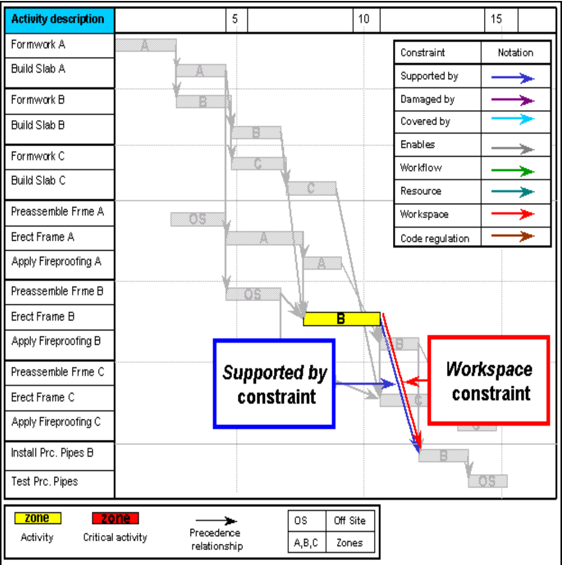


Figure 2.2a: Resource constraints used to describe two different activity sequences.

Figure 2.2b: Multiple constraints used to describe a single activity sequence.

Figure 2.2: Example showing different uses of specific constraints for rationale description.

Examples of specific constraints used for the Intel CUB schedule demonstrate the design requirements of the ontology for describing sequencing rationale.

Finally, Gruber (1993) notes that the specific objects instantiated using an ontology need to be reusable and "specializable" (i.e., customizable). Again, the design criteria apply for the specific constraints created using the formal representation.

For example in Figure 2.2a, the project manager understood that the rationale for sequencing the activity Erect Frame A before Erect Frame B was the limited availability of resources (i.e., single Frame trade). Thus, the project manager can describe the rationale for the activity sequence as a *resource constrained by* or *resource* constraint. However, his rationale for sequencing the activity Apply Fireproofing A before Apply Fireproofing B was also the limited number of Fireproofing trades available. Therefore, the project manager needed to use the same specific constraint to describe the rationale for two different pairs of activity sequences. Hence, the specific constraints need to be reusable for a single project. In addition, he felt that it was relatively easier to relax the sequence between the fireproofing activities than the frame erection activities. As shown in Figure 2.2a, the project manager needed to describe the same specific constraint to have different degrees of flexibility. Thus, the flexibility for a specific constraint needs to be customizable. Finally, planners may need more than one specific constraint to describe the sequencing rationale between two activities. For example, in Figure 2.2b, the project manager knew that not only was there a support dependency between the frames and the pipes, but also a potential workspace conflict between the trades which prevented the frame and pipe installation trades from working concurrently. Hence, the accurate description for the rationale of the activity sequence requires both the *supported by* and *workspace* constraint.

As the example shows, the formal representation needs to enable planners to <u>reuse and "specialize"</u> the constraints to describe sequencing rationale <u>consistently within a project, and also consistently from project to project</u>.

In summary, this section discussed the requirements of a formal representation for construction sequencing rationale and how those requirements could be met by formalizing the representation as an ontology. The next section discusses the design requirements of a domain-specific ontology for representing construction sequencing rationale.

2.1.2 Research Goals

The test case illustrated the requirements for developing a domain-specific ontology that is disjoint, exhaustive, reusable and specialiazable:

(1) "Disjoint" and "Exhaustive": As shown in the test case, the ontology (i.e., classification schema) that represents the planner's classification for specific constraints needs to be disjoint enough so that planners can use the ontology to correctly assign the role and flexibility for a specific constraint. In addition, the ontology needs to be exhaustive enough so that planners can instantiate (i.e., create) and use different types of specific constraints to describe the different types of sequencing rationale in construction schedules. Hence, an ontology that is disjoint and exhaustive enables planners to describe sequencing rationale explicitly, while also retaining the correct classification that planners associate with different types of sequencing rationale. The explicit and correct representation of the planners' classification for specific constraints in turn enables a computer system to infer the role and status of activities correctly.

(2) **Reusable and "Specializable":** As shown in the test case, the specific constraints created using the ontology need to be reusable and specializable. To reuse constraints, planners need to be able to create and store a specific constraint and use it within a single project and also from project to project. Hence, the specific constraints need to be project-independent. Planners also need to customize the flexibility of a specific constraint. By being reusable and specializable, the ontology enables planner to create a specific constraint once, and use it to describe sequencing rationale consistently and in a project-specific context.

The ontology formalized in this research to represent construction sequencing rationale meets these design criteria. The ontology results form combining and extending previous research efforts in the areas of sequencing rationale classification and representation. The next section describes the related research and background.

2.2 RESEARCH BACKGROUND AND POINTS OF DEPARTURE

As discussed above, I needed to develop a classification schema that is disjoint and exhaustive enough to represent the classification planners make for different types of specific constraints, and represent the classification schema in a way that enables planners to create specific constraints that are reusable and specializable. To meet this goal, I investigated previous research in the areas of sequencing rationale classification and representation. I investigated existing classification schemas for construction sequencing rationale to determine whether these schemas are applicable for classifying specific constraints in a disjoint and exhaustive way. I also investigated existing representation approaches in AI planning systems to determine whether these representations explicitly represent the role and flexibility of specific constraints, and whether the representations enable the reuse and specialization of specific constraints.

The following sections describe the previous classification schemas and representation for construction sequencing rationale in detail.

2.2.1 Classification of Sequencing Rationale

As early as the 60's through the 80's (Fondahl, 1961; Popescu and Borcherding, 1975; Birrell, 1980), researchers have addressed the need to improve upon how CPM schedules are used in the construction industry. One of the limitations identified was that the temporal classification used in CPM for precedence relationships is not sufficient in describing accurately the different types of constraints found in construction projects. In response to these needs, several researchers have focused on identifying and classifying the rationale for construction activity sequences. A common theme among the research was to classify sequencing rationale with respect to its "origin", or sequencing factors. For example, Antill and Woodhead (1970) identified factors to include physical, hazard, safety, equipment, resource management and project specified factors (Table 2.1). In addition, researchers and planners alike recognized that there exist "soft" and "hard" types of constraints, i.e., that some constraints can be relaxed and others can not (Barrie and Paulson, 1978; Tamimi and Diekmann, 1988).

Table 2.1 shows chronologically some of the previous research that has addressed these limitations of CPM by formalizing different classification schemas for the rationale of activity sequences. Most of the classification schemas have been developed to either support a heuristic process for schedule analysis (Wiest and Levy, 1969; Paulson, 1971; Tamimi and Diekmann, 1988; Russell and Wong, 1993) or automate plan generation (Echeverry et al., 1991; Kähkönen 1993; Aalami et al., 1998a). A few of the classification schemas are developed for better understanding of construction sequencing rationale (Birrell, 1980; Ballard and Howell; 1994). A more detailed discussion of how some of these classification schemas support heuristic schedule analyses is presented in Chapter 3.

Authors	Classification Schema		Criteria for Role Classification	
	Origin	Flexibility	Exhaustive	Disjunctive
Wiest and Levy (1969)	Technological, resource	No	Yes	No
Antill and Woodhead (1970)	Physical, hazard, safety, equipment, resource management and project specified factors	No	Yes	No
Paulson (1971), Barrie and Paulson (1978)	Technological, resource	Yes	Yes	No
Birrell (1980)	NA	Absolute, preference	NA	NA
Tamimi and Diekmann (1988) SOFTCPM	Technological, resource	Soft, hard	Yes	No
Echeverry (1991)	Physical component relationships, trade interaction, path interference, code regulations	Flexible, inflexible	Yes	No
Kähkönen (1993)	Structural, contractual, production technology, site conditions, safety, resource, work area and working practice.	Conditional, unconditional	Yes	No
Russell and Wong (1993)	Typical, non-typical	No	Yes	No
Ballard and Howell (1994) Last Planner	None	Yes (implicit)	Yes	No
Aalami et al. (1998a) CMM	Component/process	No	Yes	No

Table 2.1: Existing classification schemas for construction sequencing rationale.

As discussed, a goal of the research was to formalize a classification schema that classifies specific constraints with respect to their role and flexibility in a disjoint and exhaustive way. A "disjoint" classification requires the classification schema to exclusively classify a specific constraint with respect to its role and flexibility. An exhaustive classification requires the classification schema to collectively include as many specific constraints as necessary. In the test case, I illustrated that planners need at least a binary classification to distinguish the role and flexibility of specific constraints (i.e., enabling or impeding, and flexible or inflexible, respectively). Hence, I evaluated the classification schemas of Table 2.1 by determining whether the schemas are able to at least classify the role and flexibility of the different types of specific constraints exclusively as enabling or impeding, and flexible or inflexible.

Factors (Echeverry et al., 1991)	Specific Constraints	Role	Flexibility
Physical component relationships	Supported by	Enabling	Inflexible
	Connected to[8]	Impeding	Flexible
	Covered by	Enabling	Inflexible
	Enclosed by	Impeding	Inflexible
	Closer to	Enabling	Flexible
	Protected by	Enabling	Inflexible
Trade interaction	Workspace	Impeding	Flexible
	Resource	Impeding	Flexible
	Damaged by	Impeding	Inflexible
	Serviced by	Enabling	Inflexible
	Workflow	Impeding	Flexible
Path Interference	Obstructed by	Impeding	Inflexible
Code regulations	Safety	Impeding	Inflexible
	Inspection	Impeding	Inflexible
	Testing	Impeding	Inflexible

Table 2.2: Examples of specific constraints classified using Echeverry's four sequencing factors.

For example, Echeverry et al. (1991) focused in particular on factors that affect the sequence of an activity that "installs, removes, modifies or tests a particular component of a facility". Specifically, he defined these to be physical relationships between building components (e.g., *supported by* relationship), trade interaction (e.g., workspace competition), path interference (e.g., obstruction) and code regulations (e.g., inspection).

The factors identified by Echeverry et al. (1991) are most applicable to the proposed research, as the factors pertain to the actual installation and operation processes and associated interaction between multiple trades during the construction of a facility. Hence, using his classification schema, I classified the specific types of constraints (e.g., damaged by) I compiled from existing literature (Navinchandra et al., 1988; Darwiche et al., 1988; Echeverry et al., 1991; Kähkönen, 1993; Aalami et al., 1998a, Tommelein et al., 1998; Chua et al., 2003)[9]. Table 2.2 shows the classifications I made with respect to Echeverry's classification, and the classifications of the specific constraints with respect to their role and flexibility.

The classification results in Table 2.2 show that Echeverry's factors do not exclusively (i.e., in a disjoint way) classify the different types of constraints with respect to their role and

[8] The "connected to" constraint describes sequencing rationale based on a connection relationship between two components, but one that are not necessarily a supports relationship (Echeverry et al., 1991). Please refer to Appendix A for detailed descriptions for sequencing constraints identified in the literature.

[9] Please refer to Appendix A for specific descriptions of the various relationships, interactions and regulations identified in the literature.

flexibility. For example, the *supported by* constraint and *connected to* constraint are specific constraints that are classified as physical component relationships. The *supported by* constraint describes sequencing rationale between two activities where a component related to one of the activities provide physical support for the component related to the following activity. Hence, the *supported by* constraint is an enabling type of constraint as it describes a functional behavior (i.e., role) of an activity that is “conducive” to the following activity. Typically, the supported by constraint is also classified as inflexible (Echeverry et al., 1991; Kähkönen, 1993). Conversely, the *connected to* constraint describes sequencing rationale between two activities where a component related to one of the activities is physically connected to the component related to the following activity. The constraint does not necessarily describe a functional behavior of an activity that is conducive to the following activity. Hence, it is an impeding type of constraint. In addition, the *connected to* constraint is not necessarily an inflexible constraint (Echeverry et al., 1991).

Thus, Echeverry’s factors, which classifies sequencing rationale with respect to its origin, do not classify sequencing rationale disjointedly with respect to its role and flexibility.

The last two columns of Table 2.1 show the results for the other classification schemas in particular with respect to the role of constraints. The results show that whereas many of the classifications are exhaustive (i.e., comprehensive) they do not exclusively (i.e., in a disjoint way) classify constraints with respect to their role.

Consequently, for this research, I adopted the flexibility of constraints described in the previous research. However, I used a different classification for classifying the sequencing rationale with respect to its role.

Prior to the introduction of the formal ontology, the next section introduces existing approaches used to represent sequencing rationale.

2.2.2 Representation of Sequencing Rationale

As discussed, the motivation for investigating the existing representations of sequencing rationale in AI planning systems was to determine whether these representations explicitly represent the role and flexibility of specific constraints, and whether the representations enable the reuse and specialization of specific constraints.

Previous attempts to generate plans with artificial intelligence techniques have tended towards two approaches: General purpose planning systems and domain-specific AI planning systems (Darwiche et al. 1988, Dym and Levitt, 1991).

According to Darwiche et al. (1988), general-purpose planning systems or “classical” AI planning systems represent “actions” in terms of their preconditions and effects and contain a generic planning engine which conducts a search, typically aided by heuristics, to include and order correctly a set of actions that will change the initial state of the world of interest into the goal state.

In AI literature, planning and problem solving are considered different subjects because of the differences in representation of goals, states, actions and the differences in the representation of and construction of action sequences. Hence, solving the planning problem required developing a restricted language with which problems can be defined. The classical language that most general-purpose AI planners use is the STRIPS (Fikes and Nilsson, 1971) language, or extensions thereof (Russell and Norvig, 1995). This language requires no *a priori* assumptions about the domain to which it will be applied (Darwiche et al., 1988).

Conversely, many recent efforts of AI based planning systems have migrated toward domain-specific planning systems (Aalami et al. 1998b). In the construction planning domain, examples of such systems include CONSTRUCTION PLANEX (Hendrickson et al., 1987), GHOST (Navinchandra et al., 1988), OARPLAN (Darwiche et al., 1988), Builder (Cherneff et al., 1991), KNOW-PLAN (Morad and Beliveau, 1994) and CasePlan (Dzeng and Tommelein, 1997). These planning systems are built with a specific planning domain in mind. Advocates of these approaches argue that domain knowledge for planning construction projects is not naturally or commonly expressed in the form of action preconditions and effects, but rather as a set of more detailed constraints representing underlying causes of precedence (Darwiche et al., 1988).

Consequently, these approaches have focused on formalizing domain-specific languages (i.e., representations) tailored for construction specific purposes. For example, OARPLAN and CMM have developed a language tailored to describe activities in the construction domain, i.e., a Component <C>, Action <A>, and Resource <R> ontology. The domain-specific representation has been shown to support the rapid generation of hierarchical construction plans. CMM also has enabled the generation of alternate plans by using construction specific-method templates (i.e., CMMT’s). Furthermore, OARPLAN and CMM formalized how to sequence activities based on construction-specific constraints (e.g., *supported by*, *enclosed by* and *adjacent to,* etc.).

Therefore, I focused my investigation on existing construction domain-specific AI planners for developing a domain-specific ontology of construction sequencing rationale.

Generally speaking, the input to these systems is a computer-interpretable project description, and the output is a construction plan (Aalami et al., 1998c). I note, however, that the objective of my research is not the automatic generation of a construction plan. Rather, I assume that a plan already exists, and the goal is to infer the behavior (i.e., role and status) of activities based on a computer-interpretable description of sequencing rationale. Thus, my research differs from existing AI planning systems in the way the sequencing rationale is represented, and in the way sequencing rationale is used.

Nevertheless, I investigated the representation used in these systems as they provide guidance as to how to represent sequencing rationale in a project-independent way. Aalami et al. (1998c) noted that a gradual migration has occurred from knowledge-based planning systems (e.g., Stefik 1981; Kähkönen, 1993) to model-based planning systems (e.g., Darwiche et al., 1989). Hence, I investigated both of these approaches. I briefly describe each approach and describe aspects of previous research that I adopted and extended.

2.2.2.1 Sequencing rationale formalized as relationships in the product model

CONSTRUCTION PLANEX, (Hendrickson et al., 1987), OARPLAN (Darwiche et al., 1989), and CMM (Aalami et al., 1998c) are knowlede-based systems that derive activity sequences from a formal description of building components. For example, the most recent of these types of systems, CMM, can sequence two activities correctly if a *supported by* relationship exists between components related to the activities in a product model. However, the current implementation of CMM is limited to the *supported by* relationship. Other inter-component relationships (e.g., *protected by*, *connected to*, etc.) are not defined, and the effects on activity sequences are not formalized. In addition, activity sequences that are due to trade interactions or code regulations cannot be derived from relationships in the product model. Currently, CMM bypasses this problem by requiring users to define these constraints separately as "technical" constraints. Hence, whereas some activity sequences can be derived from the product model, other sequences need to be specified by the planner. In addition, the current implementation of CMM does not recognize the flexibility of constraints. Thus, users cannot specify or customize the flexibility of constraints. Finally, the output production model does not have constraints represented explicitly. That is, the initial rationale for the activity sequences is implicit in the code (i.e., in the product model). Consequently, users have to modify the product model or the technical constraints to modify sequences.

2.2.2.2 Sequencing rationale formalized as activity sequencing knowledge files

In contrast, Kähkönen (1993) developed a knowledge-based system where sequencing rationale is represented as pre-defined factors in sequencing knowledge files. Each factor is associated with a type (conditional, unconditional) specifying a factor's flexibility. For example, a "Structural" factor is unconditional. The factor and type is stored in an activity sequence knowledge file together with a generic "activity type pair" (e.g., Column, Beam). During the sequencing process, if the system identifies two activities whose type matches the activity type pairs, (e.g., activity Column and Beam) the system sequences the two activities and instantiates the factor and type between the two activities. Hence, Kähkönen's system retains the initial rationale for activity sequences.

However, the factors used by Kähkönen are too broad to distinguish between specific constraints. For example, *supported by* and *connected to* constraints are both classified as "Structural" factors. In addition, the value for the type of each factor is hard coded. That is, his system does not allow users to customize the values.

Table 2.3 summarizes the comparisons made for the two approaches discussed in this section. I adopted Kähkönen's approach of predefining rationale and flexibility as computer-interpretable constraints. I extended the representation to distinguish the rationale of constraints with respect to their role. In addition, I enable planners to customize the values for the flexibility of constraints.

Researched by	Rationale Represented as	Flexibility (customizable?)	Rationale for Constraints in Output Schedule?
Aalami et al. (1998)	Relationships in the product model	None (no)	No
Kähkönen (1993)	Sequencing knowledge file (factor and type)	Flexibility (no)	Yes

Table 2.3: Existing approaches to represent rationale and flexibility of constraints.

In summary, the sections above described the findings with respect to prior classification schemas and representation approaches for classifying and representing construction sequencing rationale. Existing classification schemas do not classify construction sequencing rationale in a way that allows constraints to be classified with respect to their role. Existing representation approaches do not explicitly represent the role of constraints, and also do not enable planners to reuse and customize specific constraints to describe sequencing rationale in construction schedules.

The following section introduces the ontology I formalized based on these findings and my observations from the test case.

2.3 ONTOLOGY FOR REPRESENTING SEQUENCING RATIONALE

This section describes the formalization and subsequent implementation of the ontology developed in this research.

2.3.1 Ontology for Sequencing Rationale

Based on the Gedanken experiments and a review of prior classifications and representation approaches, I formalized a constraint ontology that enables planners to assign the role and flexibility for specific constraints, and use the specific constraints to describe the different types of sequencing rationale in construction schedules.

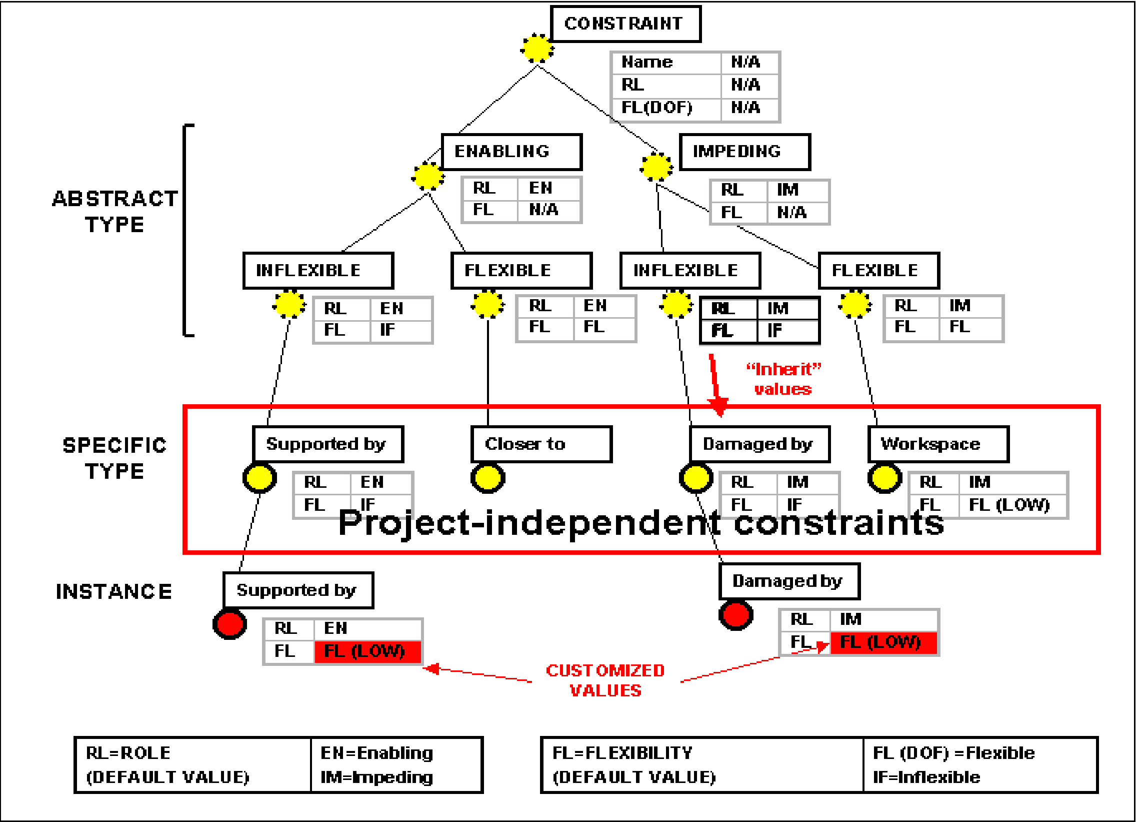


Figure 2.3: Constraint Ontology.

Figure illustrates the formalized ontology consisting of four abstract types. Specific types of constraints (e.g., *damaged by*) are subtypes of one of the abstract types. I call these specific constraints project-independent types of constraints, as they can be instantiated and used in a specific project schedule.

Figure 2.3 shows the formalized constraint ontology in the form of a decomposition hierarchy. A formalized constraint has the following attributes: name, role, flexibility and degree of flexibility. I formalized four generic or "abstract" types of constraints: *enabling-inflexible*, *enabling-flexible*, *impeding-inflexible* and *impeding-inflexible*. The abstract types have predefined values for role and flexibility. For example, the enabling-inflexible abstract type has the default value "enabling" for its role, and the default value "inflexible" for its flexibility. The default values are based on my observations and review of the construction literature. For example, as shown in the test case, planners understand that the role of a *supported by* constraint is conceptually "enabling". In addition, Echeverry et al. (1991) define a *supported by* constraint as inflexible. However, I do not fix (i.e., hard code) these values, as they may need to be customized for a specific project. For example, a *supported by* constraint still may be flexible if an alternate form of support can temporarily be provided. Hence, the flexibility of a constraint needs to be considered within the circumstances (e.g., availability of labor and materials) of a specific project. Hence, I defined the role and flexibility of a constraint as project-dependent variables. In addition, Echeverry et al. (1991) stated that constraints have varying degrees of flexibility (DOF). However, they do not specify how the varying levels should be qualified. I arbitrarily use a scale of *high*, *medium* and *low* to describe the degree of flexibility.

Based on one of these abstract types, planners can create a "project-independent" constraint type (e.g., *damaged by*) as a specific subtype of one of the abstract types. The subtype inherits the values for its role and flexibility from its abstract type. For example, planners can create a *damaged by* constraint by classifying it as a subtype of the abstract type impeding-inflexible (Figure 2.3). Thus, the *damaged by* constraint type has the values impeding and inflexible for its role and flexibility.

Using this framework, I formalized the specific constraints listed in Table 2.2 as project-independent types of constraints. The specific constraints are shown again in Table 2.4[10]. In addition, I defined additional project-independent constraints that are required to describe a trade's dependency for installed components, workspaces and shared resources. Specifically, I explicitly distinguish between a *component-supported by* and *resource-supported by* constraint. Trades or crews rely on installed components to perform their work. For example, trades installing second-floor HVAC ducts require second-floor slabs on which to work. In this case, the *resource-supported by* constraint is a more accurate description of the rationale than the *supported by* constraint. Similarly, I distinguish between a *component-protected by* and *resource-protected by* constraint. In Table 2.4, I distinguish these constraints using Aalami et al.'s (1998b) Component <C>, Action <A>, and Resource <R> typology. For example, Table 2.4 shows the

[10] I use Echeverry's factors to organize the specific constraints with respect to their origin. As discussed, however, Echeverry's factors do not classify specific constraints disjointedly with respect to their role and flexibility. Thus, Echeverry's factors are not an explicit part of the ontology.

supported by constraint further distinguished as *component-* and *resource- supported by*, or S_{cc} and S_{cr}.

I also distinguish the workspace constraint as workspace-mild, medium and severe. Akinci and Fischer (1998) identified varying levels of workspace conflicts, and qualified these conflicts as mild, medium and severe. When using workspace constraints to describe the conflict, planners may have preferences to sequence activities either sequentially or concurrently, depending on the severity of the conflict. Table 2.4 shows the constraints distinguished as WS_{Mild}, WS_{Medium} and WS_{Severe}.

Finally, I distinguish between a resource-shared and resource-trade constraint. The distinction is required as resource limitations can be due to resources that are shared (e.g., elevators or scaffolding) among multiple trades (i.e., resource-shared), or shared within a single trade (i.e., resource-trade).

Factor	Constraint	Specialization using <CAR> typology	Role (Default value)	Flexibility (DOF) (Default value)
Physical Component relationships	**Supported by**	Component supported by component	Enabling	Inflexible (N/A)
		Resource supported by component	Enabling	Inflexible (N/A)
	Connected to	None	Impeding	Flexible (LOW)
	Covered by	None	Enabling	Inflexible (N/A)
	Enclosed by	None	Impeding	Inflexible (N/A)
	Closer to	None	Enabling	Flexible (LOW)
	Protected by	Component protected by component	Enabling	Inflexible (N/A)
		Resource protected by component	Enabling	Inflexible (N/A)
Trade interaction	**Workspace**	Workspace-mild	Impeding	Flexible (LOW)
		Workspace-medium	Impeding	Flexible (LOW)
		Workspace-severe	Impeding	Flexible (LOW)
	Resource	Resource-trade	Impeding	Flexible (LOW)
		Resource-shared	Impeding	Flexible (LOW)
	Damaged by	Component damaged by activity	Impeding	Inflexible (N/A)
		Resource damaged by activity	Impeding	Inflexible (N/A)
	Serviced by	None	Enabling	Inflexible (N/A)
	Workflow	None	Impeding	Flexible (LOW)
Path Interference	**Obstructed by**	None	Impeding	Inflexible (N/A)
Code regulations	**Safety**	None	Impeding	Inflexible (N/A)
	Inspection	None	Impeding	Inflexible (N/A)
	Testing	None	Impeding	Inflexible (N/A)

Table 2.4: Formalized project-independent constraints.

In summary, the formalized ontology allows the creation of project-independent types of constraints as “specific” types of constraints. By defining “abstract” types, the ontology allows the project-independent constraints to be classified with respect to their role. It also allows the “specific” types to be used in a project-specific context (i.e., as instances), by allowing the flexibility of constraints to be customized for a specific project.

The ontology developed is a “domain-specific” ontology in contrast to a general-purpose ontology. AI researchers work to construct general-purpose ontologies, i.e., ontologies that are applicable more or less to any special-purpose domain. Arguably, general-purpose ontologies are more difficult to construct than special-purpose ontologies, as knowledge of different domains may need to be unified (Russell and Norvig, 1995).

Extensive research in building general-purpose ontologies has been performed in the AI community. Examples include ontology languages such as KIF[11] (Genesereth and Fikes, 1992), ontology development tools such as Ontonlingua (Gruber, 1991) and Protégé (Noy et al., 2001), and ontology knowledge bases such as CYC (Lenat and Guha, 1990). A general-purpose ontology for representing sequencing rationale is beyond the scope of this research, and consequently I did not implement my ontology using these implementations or languages.

The ontology developed would be meaningless unless I demonstrated that planners could use the ontology to correctly describe the role and flexibility of different types of specific constraints disjointedly and exhaustively, and subsequently reuse and customize the specific constraints to describe sequencing rationale consistently in a CPM schedule. Hence, demonstrating that the ontology meets these design requirements required implementing the ontology in a prototype tool. The implementation is described in the following section.

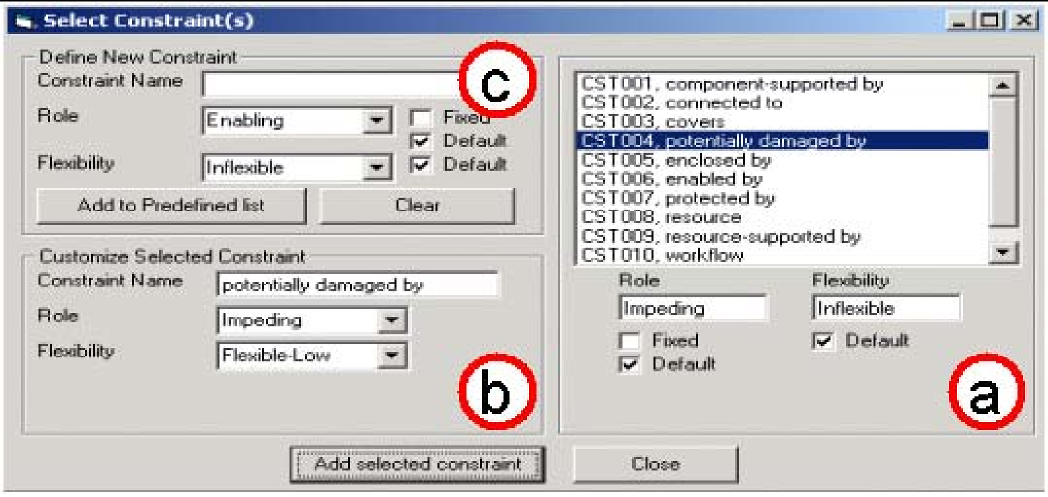
2.3.2 Implementation

I implemented the ontology in a prototype tool, which I call CLCPM (i.e., “Constraint-Loaded” CPM). I coded CLCPM using Microsoft Visual Basic (Microsoft Corporation, 1998). The rationale behind using Visual Basic was twofold. First, Visual Basic is an Object Oriented Programming (i.e., OOP) language. “OOP” is a language paradigm that encapsulates the common attributes of objects as variables of a class. Objects can then be created as instances of the class and inherit the values of the variables of the class. In addition, the inherited values of the variables can be overridden with local values.

[11] Knowledge Interchange Format.

Using the OOP framework, I implemented the abstract types of the ontology as a Constraint class in CLCPM, where the variables are the role and flexibility of constraints. The Constraints class also has four subclasses, which represent the four abstract types of the ontology.

Using these subclasses, I implemented the project-independent types of constraints and stored them in CLCPM. Each project-independent constraint also has the default values defined in Table 2.4 (Figure 2.4a). If necessary, planners can define a new project-independent constraint and also store the constraint in CLCPM (Figure 2.4b).


Select Constraint(s)

Define New Constraint
Constraint Name
Role: Enabling [ ] Fixed [✓] Default
Flexibility: Inflexible [✓] Default
Add to Predefined list | Clear

Customize Selected Constraint
Constraint Name: potentially damaged by
Role: Impeding
Flexibility: Flexible-Low

CST001, component-supported by
CST002, connected to
CST003, covers
CST004, potentially damaged by
CST005, enclosed by
CST006, enabled by
CST007, protected by
CST008, resource
CST009, resource-supported by
CST010, workflow

Role: Impeding
Flexibility: Inflexible
[ ] Fixed
[✓] Default
[✓] Default

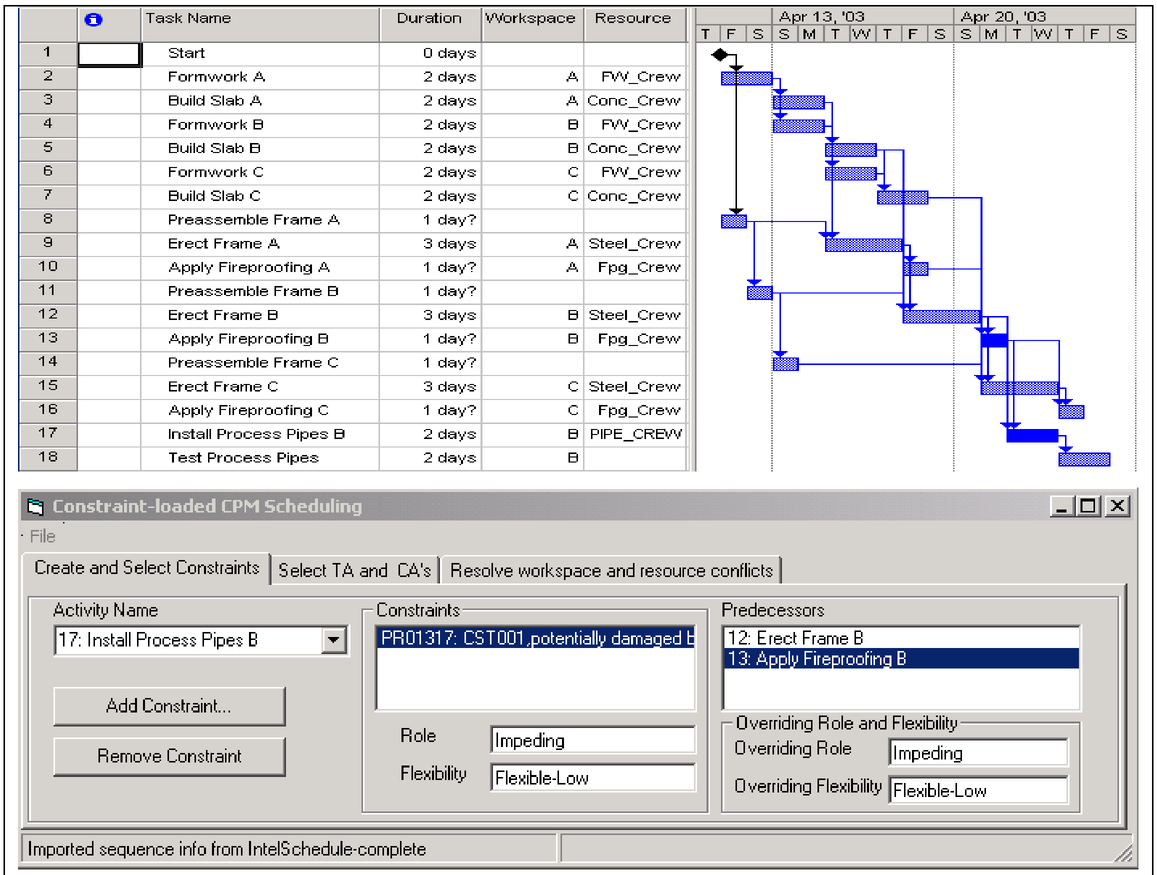
Add selected constraint | Close

Figure 2.4: CLCPM user interface for describing sequencing rationale.

Figure illustrates screenshot of CLCPM's user interface for selecting (a), customizing (b) and creating (c) specific constraints.

For a specific project schedule, planners can instantiate one of these project-independent types of constraints, customize their default values for role and flexibility, and use them to describe the particular rationale for an activity sequence of the schedule. For example, Figure 2.4c shows that users can select a *potentially damaged by* constraint and customize the flexibility of the constraint to "flexible-low".

The second reason for using Visual Basic was to take advantage of the Gantt chart representation of a CPM schedule implemented in Microsoft Project (Microsoft Corporation, 1998). Microsoft Visual Basic allowed me to reference the Microsoft Project library 9.0, which defines the basic classes for representing a Gantt chart, such as Tasks class and Precedence class. Thus, I implemented CLCPM to import task and precedence relationship objects, and subsequently enabled planners to describe the rationale for a precedence relationship by assigning one or more project-independent types of constraints to the precedence relationship. For example, Figure 2.5 shows the Intel CUB schedule in a Microsoft Project schedule and the precedence relationship between the activities Apply Fireproofing B and Install Process Pipes B highlighted in CLCPM. The figure shows the rationale for the sequence between the two activities represented in CLCPM as a *potentially damaged by* constraint.



		Task Name	Duration	Workspace	Resource
1		Start	0 days		
2		Formwork A	2 days	A	FW_Crew
3		Build Slab A	2 days	A	Conc_Crew
4		Formwork B	2 days	B	FW_Crew
5		Build Slab B	2 days	B	Conc_Crew
6		Formwork C	2 days	C	FW_Crew
7		Build Slab C	2 days	C	Conc_Crew
8		Preassemble Frame A	1 day?		
9		Erect Frame A	3 days	A	Steel_Crew
10		Apply Fireproofing A	1 day?	A	Fpg_Crew
11		Preassemble Frame B	1 day?		
12		Erect Frame B	3 days	B	Steel_Crew
13		Apply Fireproofing B	1 day?	B	Fpg_Crew
14		Preassemble Frame C	1 day?		
15		Erect Frame C	3 days	C	Steel_Crew
16		Apply Fireproofing C	1 day?	C	Fpg_Crew
17		Install Process Pipes B	2 days	B	PIPE_CREW
18		Test Process Pipes	2 days	B	

Constraint-loaded CPM Scheduling

File

Create and Select Constraints | Select TA and CA's | Resolve workspace and resource conflicts

Activity Name: 17: Install Process Pipes B

Add Constraint...

Remove Constraint

Constraints: PR01317: CST001,potentially damaged b

Role: Impeding

Flexibility: Flexible-Low

Predecessors:
12: Erect Frame B
13: Apply Fireproofing B

Overriding Role and Flexibility
Overriding Role: Impeding
Overriding Flexibility: Flexible-Low

Imported sequence info from IntelSchedule-complete

Figure 2.5: CLCPM user interface for describing sequencing rationale.

Figure illustrates screenshot of CLCPM showing a specific constraint (i.e., potentially damage by constraint) used to describe sequencing rationale for the Intel CUB schedule.

A limitation of the Microsoft Project libraries is that the libraries do not allow classes to be modified or extended. Correspondingly, the CLCPM implementation does not allow users to annotate the rationale for precedence relationships directly in the Gantt chart, but can only describe the rationale in CLCPM in the form of lists (Figure 2.5). Hence, I implemented CLCPM to highlight a precedence relationship in the MSP Gantt chart when users instantiate a specific constraint to describe the rationale for the precedence relationship.

In summary, this section described a formal ontology for representing construction sequencing rationale. I also described how the ontology was implemented using an OOP based framework that enables users to use the ontology to explicitly describe sequencing rationale in a CPM schedule. The following section describes the tests I performed using CLCPM to demonstrate that the ontology is disjoint, exhaustive, reusable and yet specializable.

2.4 VALIDATION

My research formalized an ontology that enables planners to describe sequencing rationale and also describe correctly the role and flexibility of specific constraints that planners associate for sequencing rationale. The design goals for the ontology were:

(1) **Disjoint and Exhaustive:** The ontology needs to disjointedly (i.e., exclusively) distinguish specific constraints with respect to their role and flexibility. The ontology also needs to exhaustively (i.e., comprehensively) cover as many constraints as necessary. A disjoint and exhaustive classification of constraints ensures that the ontology correctly represents the classification planners can make for different types of specific constraints.

(2) **Reusable and Specializable:** Planners need to be able use the project-indpendent constraints to describe sequencing rationale consistently. This need in turn requires the ontology to support the reuse and specialization of the project-independent constraints. The output of a "constraint-loaded" schedule ultimately needs to help construction planners to understand the rationale for activity sequences more correctly and consistently than a conventional CPM schedule.

To demonstrate that the ontology meets these design goals, I performed three retrospective case studies and one charrette test (Clayton et al., 1998). The three retrospective cases involved classifying sequence rationale that exists in schedules for different phases of construction and for different types of work. The three retrospective cases provide evidence that the ontology is disjoint and exhaustive, since the ontology correctly represents the classification planners make for different types of specific constraints. The charrette test involved using eight students to "interpret" the rationale for activity sequences for two projects, in which one half o f the students used a "constraint-loaded" schedule developed in CLCPM, and the other half of the students interpreted the rationale using a conventional CPM schedule. The test provides evidence that the project-independent constraints allows users to describe sequencing rationale correctly and consistently, by showing that resultant constraint-loaded schedules make it more likely for planners and other project participants to correctly interpret sequencing rationale than conventional CPM schedules. In other words, the charrette test demonstrates that a schedule where the rationale for activity sequences has been described using the project-independent constraints in a consistent and structured way promotes the correct and consistent interpretation of sequencing logic.

The following sections describe the tests in detail.

2.4.1 Retrospective Validation Studies

The primary goal of the retrospective case studies was to demonstrate that the ontology is disjoint and sufficiently exhaustive to correctly classify different types of sequencing rationale typically used in construction schedules. Hence, I evaluated how the ontology could be used to correctly classify different types of constraints that exist in three different construction projects. Table 2.5 shows the projects I used and the specific phases and types of work evaluated[12].

Project	**Number of Activities**	**Number of Precedence Relationships**	**Phase of Project**
Intel CUB	18	25	Concrete and structural frame.
McWhinney	27	37	Structural roof, interior MEP and finishes.
Bay Street	27	36	Concrete and structural frame, exterior closure, interior MEP and finishes.

Table 2.5: Overview of the three retrospective cases.
Number of activities, precedence relationships and types of work for the three retrospective cases.

For each of these projects, I measured the number of constraints I classified correctly using the formal ontology by confirming with Dean Reed[13], an experienced project scheduler on whether he agreed with the classifications. Specifically, I asked him to describe in his own words the rationale for activity sequences. Subsequently, I used the pre-defined constraints or created new constraints in CLCPM that matched his descriptions. Then, I classified the constraints using the ontology. Finally, we reviewed the descriptions and classification of the individual constraints together.

[12] Please refer to Appendix B.1 for a detailed description of the three projects.
[13] Dean Reed of DPR Construction, Inc. helped me with the validation. He was the main scheduler for the Bay Street project. Although he was not directly involved with the Intel project and the McWhinney project, he was still able to describe and classify the rationale for the activity sequences of these projects.

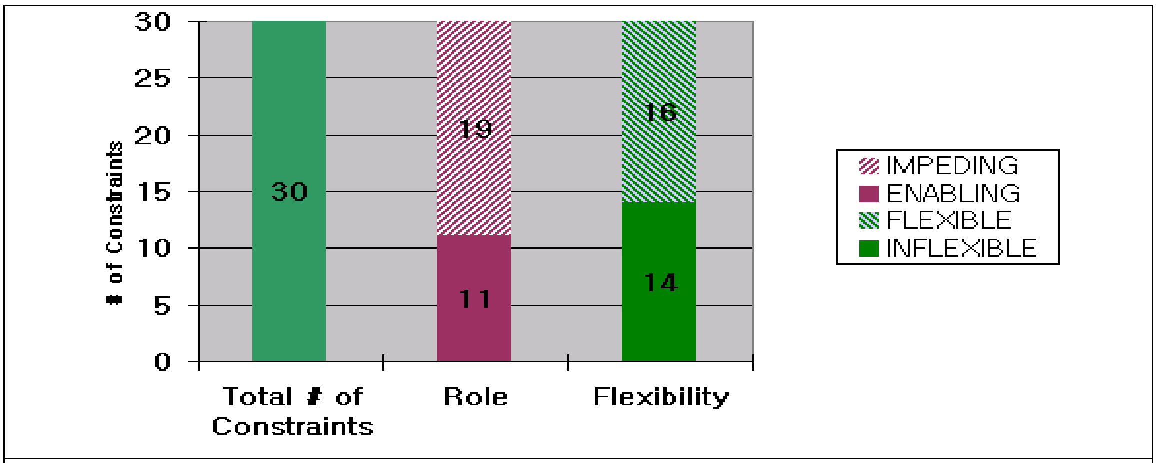


Figure 2.6: Classification results for the Intel CUB schedule.

The first bar shows the total number of constraints used to describe sequencing rationale for the Intel CUB schedule. Although there exist 25 precedence relationships (Table 2.5), I used multiple constraints for 5 of the precedence relationships. Hence, in total I used 30 constraints. The second and third bars show the specific constraints classified with respect to their role and flexibility. Dean Reed and I agreed with the classifications for all specific constraints. (Please refer to Appendix B.2 for the complete list of specific constraints used for the Intel CUB schedule).

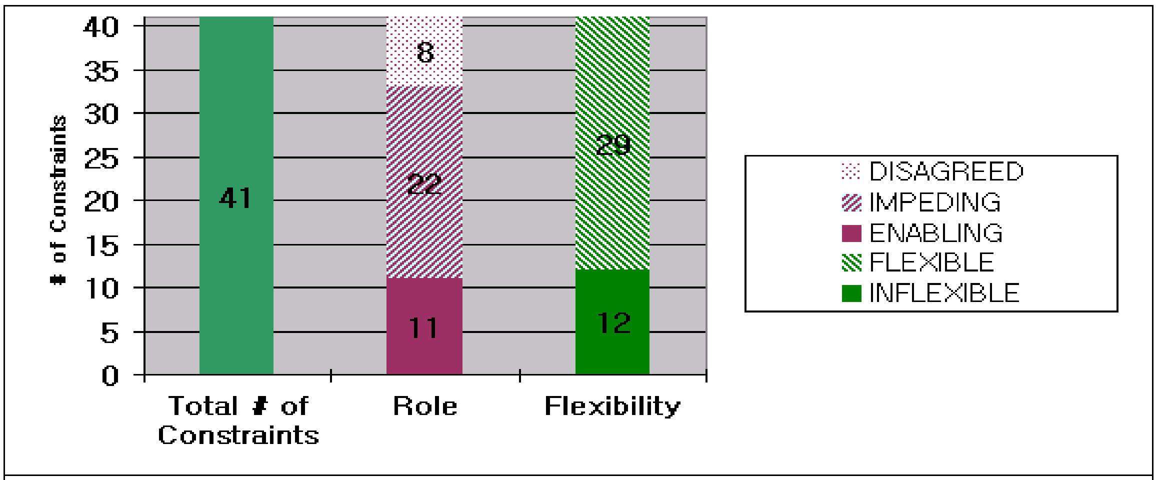


Figure 2.7: Classification results for the McWhinney project schedule.

The first bar shows the total number of constraints used to describe sequencing rationale for the Intel CUB schedule. Although there exist 37 precedence relationships (Table 2.5), I used multiple constraints for 4 of the precedence relationships. Hence, in total I used 41 specific constraints. The second bar shows that for 8 specific constraints (i.e., 4 “less bulky than” constraints, and 4 “potentially obstructed by” constraints), Dean Reed did not agree with my classifications. Dean Reed agreed with my classifications for flexibility of the specific constraints. (Please refer to Appendix B.2 for the complete list of specific constraints used for the McWhinney project schedule).

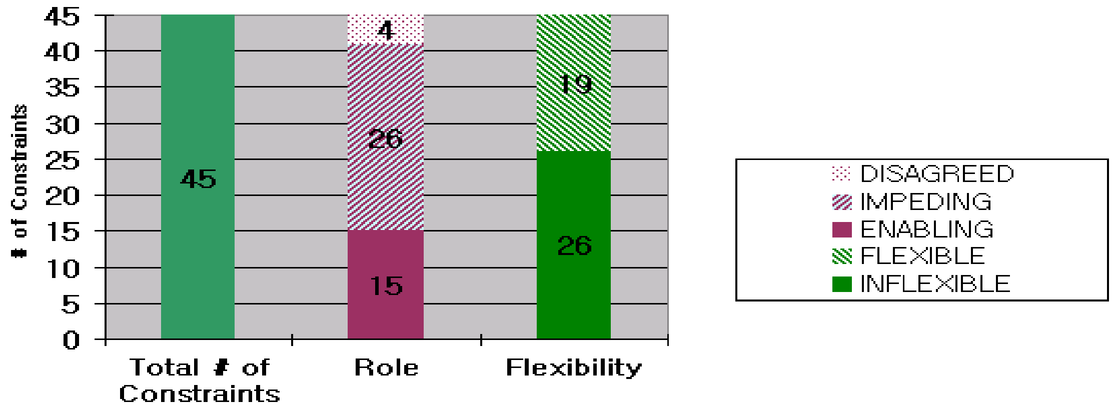


Figure 2.8: Classification results for the Bay Street project schedule.

The first bar shows the total number of constraints used to describe sequencing rationale for the Intel CUB schedule. Although there exist 36 precedence relationships (Table 2.5), I used multiple constraints for 9 of the precedence relationships. Hence, in total I used 45 specific constraints. The second bar shows that for 4 specific constraints (i.e., 2 "less bulky than constraints", and 2 "potentially obstructed by" constraints), Dean Reed did not agree with my classifications. Dean Reed agreed with my classifications for flexibility of the specific constraints. (Please refer to Appendix B.2 for the complete list of specific constraints used for the Bay Street project schedule).

Figures 2.6, 2.7 and 2.8 show the results of the three retrospective cases. For the Intel CUB schedule, he agreed with the classifications I made. However, he did not agree with a few of the classifications for the constraints in the McWhinney and Bay Street project schedule. In particular, we disagreed with the classifications for the constraints used to describe the sequence of work for MEP and interior finishes for the two projects.

Predecessor	Activity	Constraint used to describe rationale	My classification	Project scheduler's classification
Core Wall Framing	Overhead HVAC	Potentially obstructed by	Impeding/flexible	Enabling/flexible
Overhead HVAC	MEP Rough-ins	Workspace Less bulky than	Impeding/flexible	Enabling/flexible

Table 2.6: Classification differences for the Bay Street project schedule.

For example, Table 2.6 shows the specific constraints for which our classifications differed for the Bay Street project schedule. For the sequence between the activities Core Wall Framing and Overhead HVAC, I described the rationale for the activity sequence as a *potentially obstructed by* type of constraint and classified the constraint as an impeding and flexible type of constraint. However, Dean Reed noted that although the description was reasonable, the rationale for sequencing wall framing before overhead HVAC was to ensure that the HVAC components did not hamper the installation of wall framing. That is, if HVAC components were installed first, the wall framing crew would have difficulty in installing the frame walls. Hence, the rationale for the activity sequence also implies an enabling relationship between the activities. Similarly, I used a *workspace* constraint to describe the rationale between the activities Overhead HVAC and MEP Rough-ins (Table 2.6), and classified the constraint as an impeding and flexible type of constraint. However, he noted that the rationale for the sequence of the two activities was because the HVAC components were "bulkier" than the sprinklers. For lack of a better description, we added a "*less bulky than*" constraint between these activities. In addition, he noted that the *less bulky than* constraint also implies an enabling relationship between the activities. Thus, the project scheduler did not agree with my initial classifications for these specific constraints.

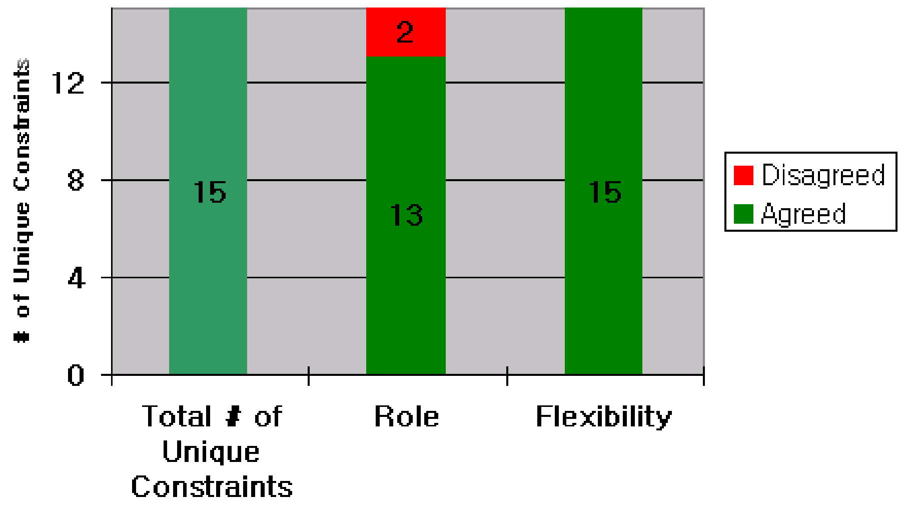


Figure 2.9: Aggregated classification results for the three retrospective cases.

Although there were these disagreements, the project scheduler still agreed with the binary classification for the role of specific constraints. That is, he did not require a different classification scheme to correctly assign the role for the specific constraints described in Table 2.6. Therefore, the results of the classification demonstrate that the initial disagreements do not compromise the correctness of the classification for the role of specific constraints.

Figure 2.9 shows the combined results of the three retrospective cases. The first bar shows the total number of unique constraints used in the three retrospective cases. The second bar in the figure shows that Dean Reed and I agreed on 13 of the 15 constraints, but disagreed on 2 specific constraints used in the McWhinney and Bay Street project schedules. The third bar shows we agreed on classifications made for the flexibility of the unique constraints.

For the three project schedules, we simply agreed to commit to one classification for the specific constraints in Table 2.6 and use the classification consistently for the specific constraint throughout the three project schedules. We used Dean's classification on account of his extensive experience and expertise. As discussed, we could have also added a new constraint to describe the sequencing rationale in more detail if necessary.

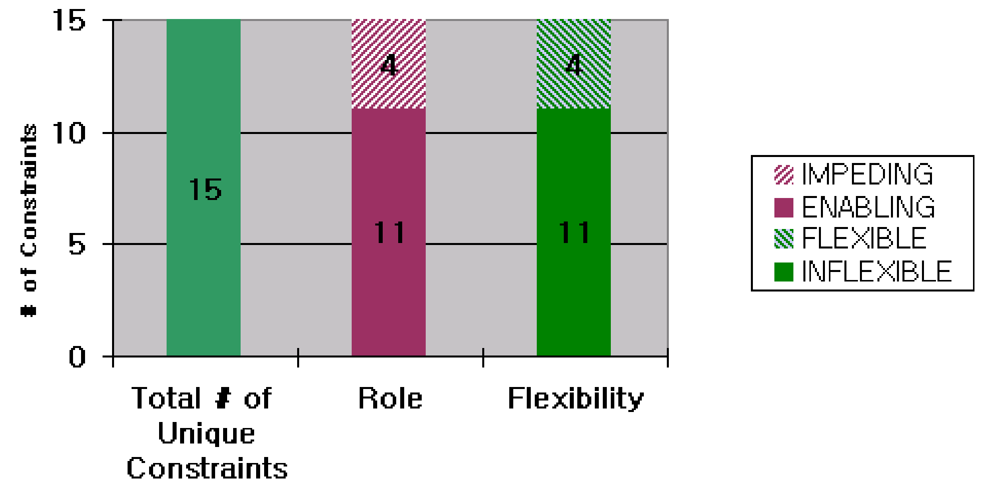


Figure 2.10: Classification results for the unique constraints of the three retrospective cases.

Figure 2.10 shows the resultant classifications for the unique constraints. The figure shows that we were able to agree on the role and flexibility for all of the specific constraints.

In summary, the ontology correctly classified all the different types of constraints for the three retrospective cases. The results indicate that the ontology is disjoint and sufficiently exhaustive enough to correctly represent the conceptual classifications planners make for sequencing rationale that typically exists in construction schedules.

The following section describes the charrette test that demonstrates that the ontology enables planners to interpret sequencing rationale correctly and consistently.

2.4.2 Charrette Test

As discussed, the goal of the charrette test was to test whether a "constraint-loaded" schedule developed using the ontology for describing sequencing rationale promotes the correct and consistent interpretation of sequencing rationale. Therefore, I evaluated whether students using CLCPM could interpret sequencing rationale more correctly and consistently than students using CPM schedule based on Microsoft Project (MSP).

Specifically, eight graduate students in the construction engineering and management program at Stanford University were asked questions concerning the rationale for activity sequences for two different schedules, the Intel CUB schedule and the Bay Street project schedule. For the first project, four of the students individually used CLCPM, while the latter four individually used MSP. For the second project, the eight students switched roles. That is, the first four students used MSP to answer the questions, and the latter four now used CLCPM. For each of the projects, each student was provided with a 3D model and a brief description of the overall sequence of the activities[14]. When using CLCPM, the students had access to the rationale for activity sequences that I predefined using the project-independent constraints. When using MSP, as is the case in CPM schedules, the students did not have access to the sequencing rationale. In addition, all eight students answered questions for the project schedule in which they used CLCPM first. This would allow them to be more aware of the tasks they need to perform when using the MSP. Hence, the procedure benefits the conventional CPM-based tool, and hence lends more credibility to the test results.

The eight students answered the following type of questions for the two project schedules:

i) **Rationale Type questions:** The questions asked students to describe the rationale for different types of precedence relationships. The questions test whether the specific constraint descriptions (e.g., *supported by*) help users to understand the different types of rationale for activity sequences.

ii) **Flexibility and Role questions:** The questions asked students to determine whether precedence relationships can or cannot be relaxed and which activities provide an "enabling" or "assisting" role to the activity's successors. The questions test whether CLCPM helps users to understand the functional properties of constraints originally implicit in CPM schedules.

iii) **Logic integrity questions:** The questions asked students to determine whether "missing" or "redundant" precedence relationships exist in the test schedules. The questions test whether CLCPM helps users to identify illogical precedence relationships.

[14] Please refer to Appendix C to view the questionnaires for the charrette test I used to instruct students to interpret sequencing rationale for the Intel CUB schedule and Bay Street project schedule.

I prepared the same number of questions for both project schedules (Table 2.7). I assigned a single point for each correct answer. Hence, the total points a student could score for a project was 11.

Project	Rationale Type	Flexibility/Role	Logic Integrity	Total
Intel CUB	5	4	2	11
Bay Street	5	4	2	11

Table 2.7: Number of questions for each type of question for the two project schedules.

For each project, I evaluated the "level of correctness" by comparing the average score of the four students using CLCPM with that of the average score of four students using MSP. I also evaluated the "level of consistency" for each project by comparing the variance for the scores of the four students using CLCPM with that of the variance for the scores of the four students using MSP.

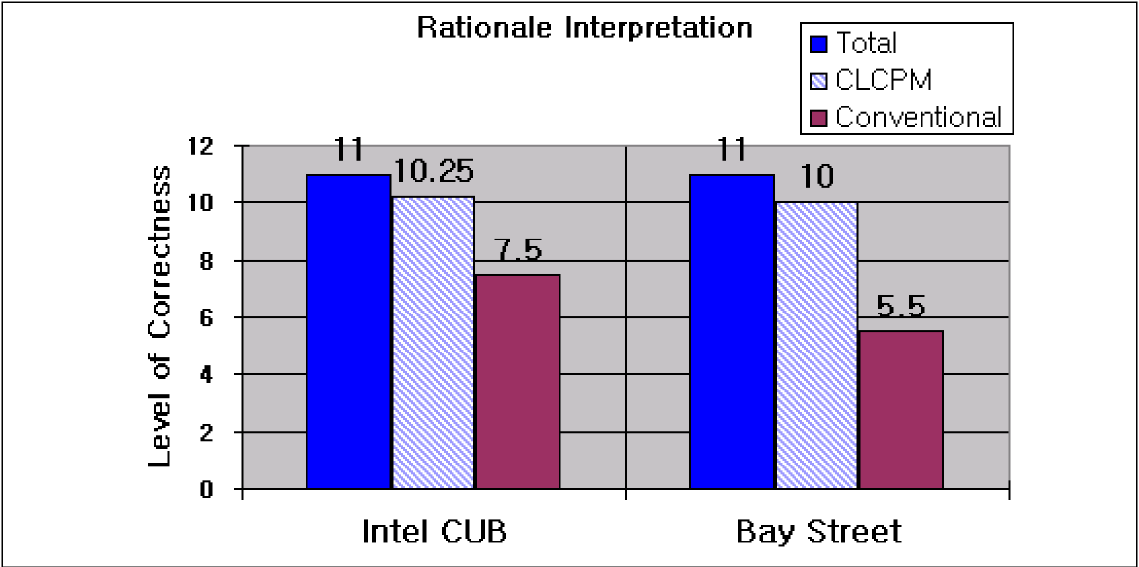


Figure 2.11: Charrette test results for the two projects.

The first bar is the total possible score for the rationale interpretation questions. The second and third bars are the average scores of four students for the Intel CUB schedule using CLCPM and Microsoft Project, respectively. For example, the average score of four students using CLCPM for the Intel CUB schedule is calculated as (10+11+10+10)/4=10.25. The second set of bars describes the corresponding scores for the Bay Street project schedule.

Project	CLCPM	Conventional
Intel CUB	0.18	1.25
Bay Street	1	4.25

Table 2.8: Variance for the scores of the two project schedules.

I calculated the variance using a standard variance equation. For example, the variance for scores of four students using CLCPM for the Intel CUB schedule is calculated as $((10-10.25)^2+(11-10.25)^2+(10-10.25)^2+(10-10.25)^2)/4=0.18$.

Figure 2.11 shows the aggregated results of the charrette test for the Intel CUB and Bay Street project schedules. The figure shows that for both projects, students using CLCPM answered on average over 90% (i.e., (10.25/11+10/11)/2) of the questions correctly, compared to 59% (i.e., (7.5/11+5.5/11)/2) for students using MSP. Hence, the students using CLCPM interpreted the schedule logic more correctly than students using MSP. Table 2.8 shows the variance for the students' scores for each project schedule. The table shows that the variance of the scores for students using CLCPM was 0.18 and 1 for the two project schedules, compared to 1.25 and 4.25 for students using MSP. The results indicate that the formal ontology enables users to interpret the rationale of the activity sequences more correctly and consistently than MSP.

In summary, the charrette test demonstrates that a constraint-loaded schedule based on my ontology enables multiple project participants to better understand the rationale for activity sequences.

This section described the procedures and results of three retrospective case studies and a charrette test. The retrospective cases demonstrate that the formal ontology was designed in a way that correctly represents the classification that planners make for different types of constraints. The charrette test demonstrates that a schedule where rationale for activity sequences is described using the ontology enables planners to interpret sequencing logic more correctly and consistently than conventional CPM schedules.

2.5 CONCLUSIONS

This chapter presented an ontology that formalizes a classification schema for sequencing rationale that is disjoint enough to represent correctly the unique role and flexibility of constraints, and also exhaustive enough to represent comprehensively the role and flexibility for different types of construction sequencing rationale. The ontology enables planners to define project-independent constraints that they can reuse and customize to describe sequencing rationale in construction schedules correctly and consistently. Thus, the ontology not only enables planners to describe a particular sequencing rationale using specific constraints, but also describes explicitly the classification that planners make for a specific constraint with respect to its role and flexibility.

The ontology is limited in many ways. Some of the limitations are due to the scope I defined for this research. I limited the scope of the ontology to primarily represent sequencing rationale of construction schedules, and hence did not consider rationale in design and procurement schedules. I also designed the ontology to be used primarily for Finish to Start relationships in a CPM schedule. I also did not explicitly associate the specific constraints with the individual costs that may be incurred for relaxing constraints. A practical limitation is that planners currently need to manually input the specific constraints to describe rationale for activity sequences, which may become impractical for large construction schedules. I discuss these limitations and possible extensions in more detail in Chapter 5.

Using the ontology, planners can describe the rationale for all precedence relationships in a CPM schedule, in effect creating a constraint-loaded schedule. As the individual specific constraints retain their individual classification for role and flexibility, the constraint-loaded schedule in turn enables a computer system to use the classification to automatically classify activities. As discussed, the automated classification of activities in turn supports planners in developing sequencing alternatives correctly and rapidly. The mechanism for automating the classification of activities is described in Chapter 3.

CHAPTER 3. FORMAL CLASSIFICATION MECHANISM

3.1 INTRODUCTION

In Chapter 1, I used a test case to illustrate that classifying the role and status of activities is an integral part of identifying and developing sequencing alternatives. Planners need to classify the role and status of activities to identify suitable activities that when delayed expedite an activity requiring earlier execution (i.e., target activity). In addition, planners need to prioritize the activities based on their role and status to re-sequence activities correctly. The test case also demonstrated that planners infer the role and status of activities based on the rationale and flexibility of their constraints. The CPM framework does not explicitly represent the rationale for activity sequences and only classifies the time-criticality (i.e., zero total float) of the activities. Therefore, the process of inferring the role and status of activities today can only be performed in the planners' minds. Practically, manually inferring the role and status of activities for complex CPM networks can become time-consuming and error-prone. Hence, a goal of this research was to develop a "classification" mechanism that automatically infers the role and status of activities, so that planners in turn can use the mechanism to identify and re-sequence activities correctly and rapidly. In Chapter 2, I presented an ontology that planners can use to describe the rationale for constraints in a way that enables a computer system to understand and utilize the planner's classification of the constraints' role and flexibility. Thus, the next step of the research was to develop a classification mechanism that utilizes the formal ontology to automatically classify the role and status of activities in a CPM network.

Achieving this goal required investigating how planners conceptually infer the role and status of a typical activity in a CPM network schedule. Hence, I performed Gedanken experiments (Chaitin, 1965) to understand the planner's logic for classifying activities. To classify a typical activity, planners need to identify unique "paths" that link an activity to the activity requiring earlier execution, which I call network chains. Subsequently, they need to infer the role and status of a typical activity in relation to the target activity based on the role and flexibility of the individual constraints in these network chains. Therefore, automating the inference process required designing an algorithm that automatically identifies relevant network chains in a CPM network, and formalizing a set of inference rules (i.e., axioms) that generalize how planners infer the role and status of activities given multiple network chains.

To develop a mechanism that meets such requirements, I investigated prior research efforts that have also classified activities to perform construction analyses (Levitt and Kunz, 1985; Russell and Wong, 1993; Seibert et al., 1996; Aalami et al., 1998a; Bentley Systems, 1997).

A few of these classifications (e.g., Aalami et al., 1998a) also attempt to provide "context" to activities, in particular to the "role" or "function" an activity has with respect to the overall project. However, these classification approaches place the burden of classifying activities on the planner. I also investigated graph algorithms and identified that Warshall's transitive closure algorithm (Warshall, 1962) could be adopted to correctly identify the unique network chains.

Based on these investigations, I developed a classification mechanism that consists of a network chain search algorithm and a set of generalized inference rules. The search algorithm identifies unique network chains between an activity and the target activity. The inference rules generalize the planners' inference process for classifying the role and status of an activity based on the role and flexibility of the specific constraints that exist in the activity's network chains. Using the mechanism, planners can automatically identify the role and status of activities regardless of the different types of specific constraints used to describe sequencing rationale in a CPM network schedule.

This chapter details the investigations performed and the resulting formal classification mechanism. I validated the power and generality of the classification mechanism by demonstrating the power and generality of the identification and re-sequencing process, which is described in the next Chapter. I conclude with a discussion of the immediate implications and limitations of the formal classification mechanism.

The following section describes a test case to illustrate the requirements of the classification mechanism.

3.1.1 Motivating Case Example

Figure 3.1 shows the rules or "axioms" I defined in Chapter 1 that formalize how planners infer the role and status of a target activity's immediate predecessors based on the role and flexibility of the constraint between the two activities.

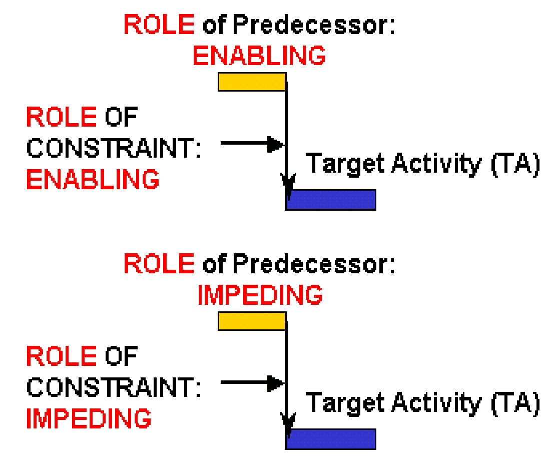


Figure 3.1a: Role (i.e., enabling or impeding) of a predecessor based on the role (i.e., enabling or impeding) of the constraint.

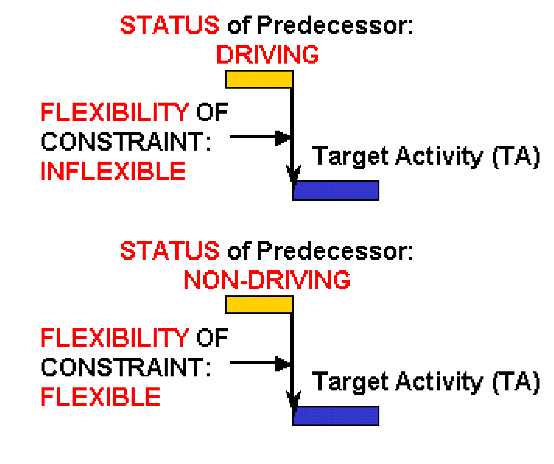


Figure 3.1b: Status (i.e., driving or non-driving) of the predecessor based on the flexibility (i.e., inflexible or flexible) of the constraint.

Figure 3.1: Axioms between activities and constraints.

Figure illustrates the basic axioms that formalize the relationship between the role and flexibility of a constraint and the role and status of an immediate predecessor.

The axioms accurately describe a planner's inference process. For example in the Intel CUB schedule, the project manager inferred the role of the activity Erect Frame B, an immediate predecessor to the target activity Install Process Pipes B, as an enabling type of activity as he knew that the *supported by* constraint was an enabling type of constraint (Figure 3.2a). He also inferred the status of the immediate predecessor Apply Fireproofing B as a "non-driving" type of activity, as he knew that the *damaged by* constraint was a flexible type of constraint (Figure 3.2b). As discussed, I defined non-driving activities as activities that when delayed either by using float or by relaxing constraints between activities and the target activity, enable the earlier start of the target activity. In other words, a non-driving activity is an activity that can be delayed <u>without in turn delaying</u> the target activity. Therefore, relaxing the *damaged by* constraint enables the non-driving activity Apply Fireproofing B to be delayed and in turn expedites the target activity Install Process Pipes B.

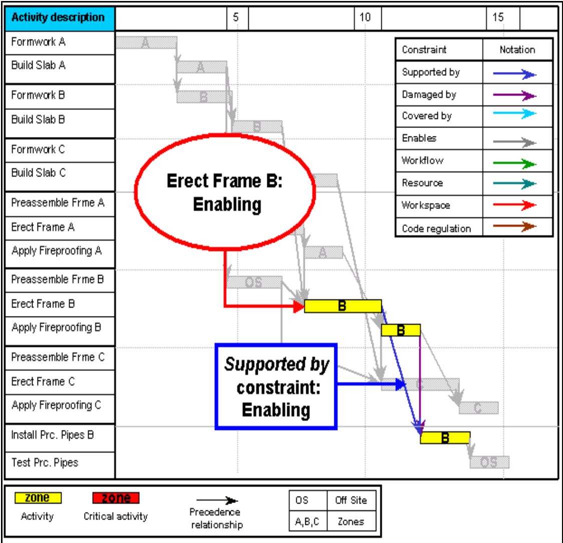


Figure 3.2a: Role of activity Erect Frame B based on role of the *supported by* constraint.

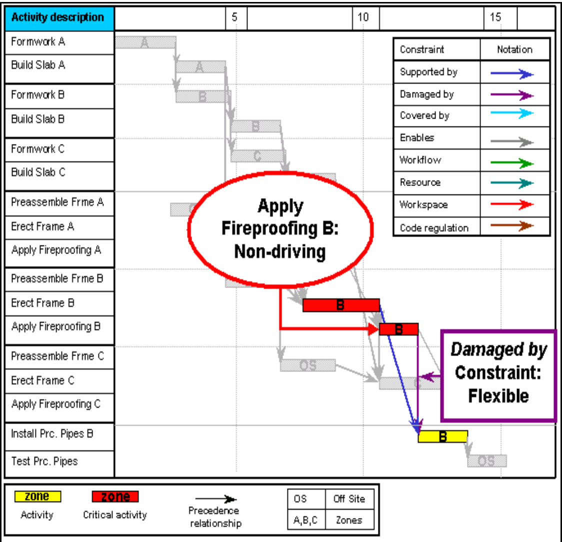


Figure 3.2b: Status of activity Apply Fireproofing B based on the flexibility of the *damaged by* constraint.

Figure 3.2: Example of planner's inference process for immediate predecessors of a target activity.

Figure illustrates the planner's inference process for inferring the role of the activity Erect Frame B and the status of the activity Apply Fireproofing B.

The example shows that the axioms in Figure 3.1 accurately represent the planner's inference process for immediate predecessors. However, planners also need to infer the role and status of activities that are not immediate predecessors, but are still related to the target activity by indirectly constraining the target activity. For example in Figure 3.3, the project manager also needs to determine the role and status of the activity Erect Frame A in relation to the Install Process Pipes B activity, since delaying this activity could also potentially expedite the Install Process Pipes B activity. I call such activities "related" activities. The immediate predecessors of a target activity are one type of related activities.

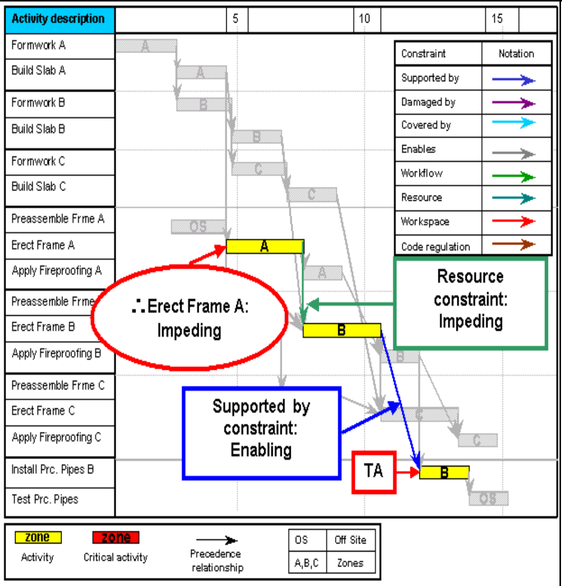


Figure 3.3a: Role of activity Erect Frame A based on one of the network chains.

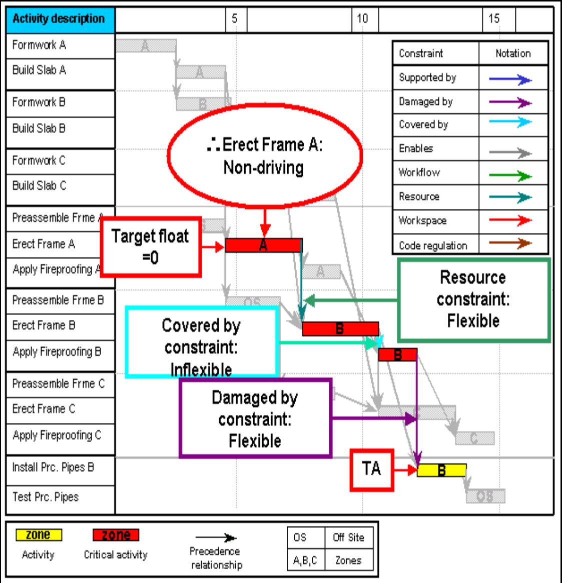


Figure 3.3b: Status of activity Erect Frame A based on a "critical" network chain.

Figure 3.3: Example of planner's inference process for related activities of a target activity.

Figure illustrates the planner's inference process for classifying the role and status of the activity Erect Frame A (i.e., a related activity).

To determine the role of the related activity Erect Frame A, the project manager observed that the activity Erect Frame A is related to the activity Install Process Pipes B by the network chain highlighted in Figure 3.3a. Within this network chain, he understood that the *resource* constraint between activities Erect Frame A and Erect Frame B is an impeding type of constraint. Hence, the activity Erect Frame A was impeding the start of the activity Erect Frame B. Hence, he concluded that because the activity Erect Frame A is impeding an activity (i.e., the activity Erect Frame B) which is enabling the target activity, the activity Erect Frame A was an impeding activity with respect to the target activity Install Process Pipes B.

Having determined the role of the related activity Erect Frame A as impeding, the project manager also needed to know whether delaying this activity would in turn expedite the target activity Install Process Pipes B. That is, he needed to know whether the status of the related activity Erect Frame A was driving or non-driving. The project manager first checked whether the activity was time-critical (i.e. zero total float) with respect to the target activity, as this indicates that the activity is preventing the earlier start of the target activity Install Process Pipes B (Figure 3.3b). I call such float the activity's "target" float. Subsequently, he identified the network chain that causes the activity Erect Frame A to be time-critical as shown in Figure 3.3b. I call such network chains "critical" network chains. Then, he observed that the *resource* constraint and the *damaged by* constraints in the “critical” network chain were flexible. Hence, he inferred that, although the activity Erect Frame A has zero target float, it could still be delayed by relaxing one of these constraints, and delaying the activity would in turn expedite the activity Install Process Pipes B to start earlier. That is, the status of the activity Erect Frame A is non-driving.

As the example shows, inferring the role and status of a related activity in a CPM network in today’s practice requires planners to identify unique network chains of the related activity manually, and derive the role and status of the activity based on the role and flexibility of the constraints in the network chain. In addition, the example shows that planners need to identify multiple network chains to determine the role and status of activities correctly. To infer the role and status of all the related activities for a given target activity, planners would need to repeat the process for each and every related activity.

The example illustrates that the process of inferring the role and status of activities in a CPM network schedule becomes quickly intractable, time-consuming and error-prone. As discussed, a goal of this research was to develop a classification mechanism to automate the inference process. Hence, a network chain search algorithm needed to be developed that correctly and rapidly identifies all network chains for a related activity with respect to a given target activity in a CPM network schedule. In addition, inference rules that extend the rules described in Figure 3.1 need to be developed that correctly formalize how planners infer the role and status of related activities based on the role and flexibility of constraints in the activity’s network chains. As discussed, the ontology formalized in Chapter 2 generalizes the role and flexibility of constraints as abstract types. Hence, the inference rules need to be formalized using the abstract types of the ontology.

In summary, planners need automated support for classifying the role and status of activities. The classification mechanism needs to correctly classify the role and status of activities given a CPM network schedule where the rationale for activity sequences has been explicitly represented using the formalized ontology.

The following section summarizes the design requirements of the classification mechanism.

3.1.2 Research Goals

The case example illustrates that the classification mechanism needs to be formal and general.

(1) Formal: As shown in the test case, the classification mechanism needs to correctly and rapidly distinguish the role and status of activities with respect to a target activity.

Meeting this goal required developing inference rules that correctly formalize how activities are classified based on the role and flexibility of constraints in the activity's network chain. In addition, meeting the goal required developing a network chain search algorithm that correctly and rapidly identifies network chains of activities with respect to a given target activity in a CPM network.

(2) **General:** The classification mechanism needs to be general enough to correctly classify activities in a CPM schedule where different types of sequencing rationale has been explicitly represented using the formalized ontology.

The classification mechanism formalized in this research meets these criteria. Developing the classification mechanism required investigating activity classification schemas developed in previous research and graph algorithms. The next section describes these investigations in detail.

3.2 RESEARCH BACKGROUND AND POINTS OF DEPARTURE

As discussed, formalizing the classification mechanism required developing a set of inference rules and a network chain search algorithm. To meet this goal, I investigated previous research efforts that have also classified activities to perform construction analyses. The motivation for investigating activity classification schemas was to determine whether the mechanisms used to classify activities could be used to classify the role and status of activities. A CPM network is a special type of graph. Hence, I also investigated graph algorithms to determine whether these algorithms could be used to identify network chains in a CPM network.

3.2.1 Existing Activity Classification Schemas

Several researchers and practitioners have defined activity classification schemas (Fondahl, 1961; Levitt and Kunz, 1985; Waugh, 1990; Russell and Wong, 1993; Ballard and Howell, 1994; Seibert et al., 1996; Bentley Systems, 1997; Aalami et al., 1998a). Activities in CPM schedules have been classified differently depending on the particular information required by planners. Table 3.1 shows the different classification schemas used and the related purpose for their particular classification schema. As the examples in the table show, many of the classifications allow a richer distinction than that provided by CPM schedules. The classifications are similar in that they attempt to provide "context" to activities, in particular to the "role" or "function" an activity has with respect to the overall project.

For example in the test case in Section 3.1.1, I classified the activity Erect Frame B as an enabling activity. Using the classifications in Table 3.1, the activity can be classified similarly as a core, value-adding and permanent activity. Conversely, the activity Preassemble Frame B can be classified as a non-core, contributory and constructive activity.

Researched by	Classification Schema	Purpose of Classification	Classification Method
Fondahl (1961) Wiest and Levy (1969) Paulson (1971, 1973)	Resource critical Non-resource critical	Identify "critical sequence" in a resource constrained CPM schedule to perform time-cost trade-off.	Unified approach, User-computer interaction.
Levitt and Kunz (1985)	Long/Short, Knights/Villains	Utilize task and project management knowledge to forecast the expected performance of remaining activity durations based on the project's performance to date.	Inferred by system using updated information of activity durations.
Waugh (1990)	ToDo, CanDo, Doing, Ended, Done	Ensure correct sequence of plans using updated project conditions.	Uses "status" knowledge to infer activity sequence.
Russell and Wong (1993)	Continuous, Ordered, Shadow, Cyclic	Enable planners to explicitly describe the location worked and the sequence in which they are to be worked, and requirements for work continuity.	User input.
Ballard and Howell (1994)	Will, Can, Should	Improve workflow reliability of plans.	User input.
Seibert et al. (1996)	Value-adding, Contributory, Ineffective	Assess the value-added by individual activities as well as assess the aggregate value-added effectiveness of an overall construction plan.	User input.
Bentley's Schedule Simulator Bentley Systems, 1997)	Permanent, Temporary, Constructive, Destructive	Visually distinguish activities during the 4D simulation process.	User input.
Aalami et al. (1998a)	Core, Non-Core	Prevent over-constrained plans.	User input.

Table 3.1: Examples of activity classification schemas.

As shown in Table 3.1, the majority of the approaches used in these studies place the burden of classifying activities on the user. Hence, these approaches quickly become impractical when classifying activities for full-scale projects, where typically a hundred or more activities may be managed and updated throughout the project life cycle. However, a few of the approaches, i.e., Paulson (1971), Levitt and Kunz, (1985) and Waugh (1990), developed heuristics to quickly classify activities and enable planners to leverage the classification to perform construction-specific schedule analyses. In particular, the "unified" approach formalized by Paulson (1971) is of particular interest as his heuristics for classifying activities were developed with an interactive man-computer system in mind. More specifically, Paulson adopted

Wiest and Levy's (1969) concept of "critical sequence" to develop a way to perform time-cost trade-off analysis on a resource-constrained schedule. The resource limitations in a resource-constrained schedule generally reduce the amount of the activities' float initially calculated using CPM's total float calculations. Wiest and Levy (1969) developed heuristics to recalculate the float, and defined several criteria to identify a "critical sequence" of activities whose recalculated total float values were still zero. Paulson used the critical sequence of activities to define "intervals" or groups of activities, which alleviated the planner from having to input volumes of non-essential data for intervals that experience and judgment tell him would be economically impractical to expedite.

Wiest and Levy's heuristic is not directly applicable for inferring the role and status of activities. The goal of their heuristics was to identify a sequence of activities that are "resource-critical," rather than identify the role and status of activities with respect to a specific target activity. In addition, my approach assumes that the resources are already loaded in the schedule, and that the resource limitations are explicitly represented as *resource* constraints. Hence, the "target" floats calculated are based on a resource-constrained schedule.

However, I do adopt Paulson's (1971) general philosophy, i.e., that CPM-based scheduling techniques such as time-cost trade-off need to be developed in a way that supports planners in utilizing their judgment and expertise in making planning decisions, rather then a stand alone optimization technique. Accordingly, the classification mechanism has been developed not just to automate the inference process, but so that the mechanism alleviates planners from manually having to classify the role and status of activities and hence focuses their expertise on making correct and quick decisions when developing sequencing alternatives. Chapter 4 describes how this is achieved in more detail.

In summary, many of the existing classification schemas classified activities that are conceptually similar to the role of activities defined in my research. However, many of these research efforts are impractical due to the manual classification they impose on planners. Paulson's unified approach serves as a guide for developing the classification mechanism in a way that can be used as a decision-support tool.

As discussed, formalizing the classification mechanism required developing a way to correctly and rapidly identify network chains in CPM network. A CPM network is a special type of graph (Tamassia and Tollis, 2002). Correspondingly, I investigated graph algorithms that could be used to automatically identify network chains. This is discussed in the next section.

3.2.2 Graph Algorithms

A graph is a set of vertices and edges or paths between vertices (Tamassia and Tollis, 2002). Graphs visually represent a particular data structure and to can protray algebraic equations. There are several types of graphs, which are broadly distinguished, as directed or undirected, cyclic or acyclic and weighted or not weighted. A CPM network is one type of graph classified as a directed acyclic and weighted graph (Tamassia and Tollis, 2002). That is, in a CPM network, each path can follow from one vertex to the next (i.e., directed), paths do not start and end at the same vertex (i.e., acyclic), and vertices have numeric values (e.g., ES, etc.) assigned (i.e., weighted).

As shown in the test case, inferring the role and status of an upstream activity with respect to a downstream target activity requires identifying the unique paths or network chains between the two activities in a CPM network. To support the automatic classification of activities, an algorithm was required that could identify these network chains.

Hence, I investigated graph algorithms that identify unique paths between vertices in directed acyclic graphs. In graph theory, these algorithms are called transitive closure algorithms. Several algorithms exist (Agrawal and Jagadish, 1987; Jiang, 1990; Jakobsson, 1991), and researchers continually try to improve the run time performance of these algorithms (Dar and Ramrkrishnan, 1994). However, I used one of the earlier algorithms developed, specifically Warshall's algorithm (Warshall, 1962) as its search process was simple and easy to code. Warshall's algorithm is a specialized but earlier version of Floyd's algorithm (Floyd, 1962) that identifies a path between any two vertices in a directed graph.

AI planning systems typically use transitive closure algorithms during the plan generation process for cycle detection and precedence relations. That is, the algorithms are used to ensure that no cycles exist in the plan, and to find the set of steps that could be ordered before a certain step (Yang, 1997). However, the requirement identified in the test case required identifying the unique paths for two activities *after* a plan has been generated. Hence, I adopted Warshall's algorithms so that a computer system can automatically identify unique network chains for two activities in a CPM network schedule.

In summary, previous research has shown the requirement for a richer classification that can be used to either evaluate or modify activity sequences. However, these approaches have not developed methods to automatically classify activities in a way that supports the rapid re-sequencing activities. Investigation of graph algorithms enabled me to identify a simple algorithm for automatically identifying network chains in a CPM network. The following section introduces the classification mechanism I formalized based on these findings and my observations from the test case.

3.3 FORMAL MECHNAISM FOR CLASSIFYING THE ROLE AND STATUS OF ACTIVITIES

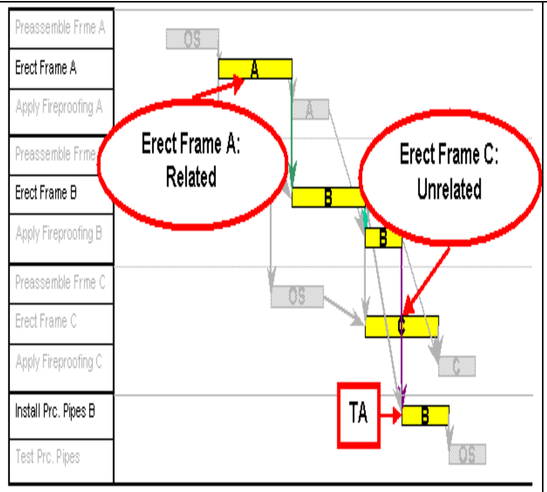
This section describes the formalization and subsequent implementation of the classification mechanism developed in this research.

3.3.1 Formal Classification Mechanism

Based on the "Gedanken" experiments, a review of prior activity classification schemas and graph algorithms, I formalized a classification mechanism that automatically classifies the role and status of activities given a CPM schedule where the rationale for activity sequences has been explicitly represented using the formal ontology described in Chapter 2. As shown in the test case, the classification mechanism needs a network chain search algorithm to automatically identify unique network chains between a related activity and a target activity. The mechanism also needs inference rules that formalize how planners infer the role and status of activities using the role and flexibility of the constraints in multiple network chains. The following sections describe the formalizations in detail.

3.3.1.1 Network Chain Search Algorithm

As discussed in the test case, a related activity in a CPM network can be linked to the target activity by one or more network chains. More formally, I define an activity's network chain as "a single succession of activities and paths (i.e., constraints) between an activity and the target activity". A target activity may be any activity in a CPM network, and not necessarily the last or end activity. When the target activity is not the end activity, activities in the CPM schedule may or may not be related to the target activity by one or more network chains. For example, the activity Erect Frame A is a related activity as the activity is part of three unique network chains (Figure 3.4) linking the activity Erect Frame A to the target activity Install Process Pipes B. Conversely, the activity Erect Frame C is not part of any network chains with respect to the target activity Install Process Pipes B, and hence is an "unrelated" activity (Figure 3.4a).



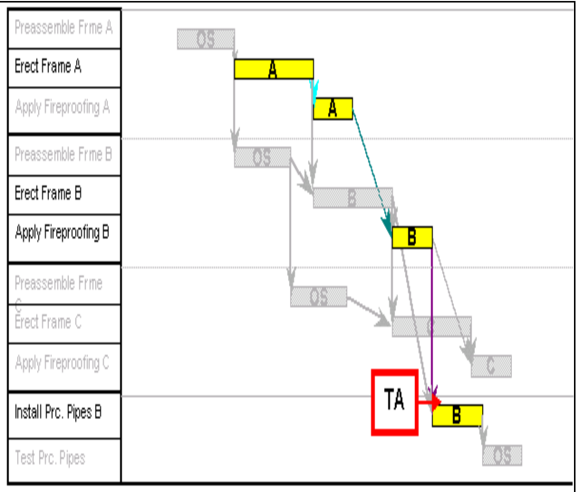


Figure 3.4a: Network chain 1. **Figure 3.4b:** Network chain 2.

Figure 3.4: Example network chains in the Intel CUB schedule.

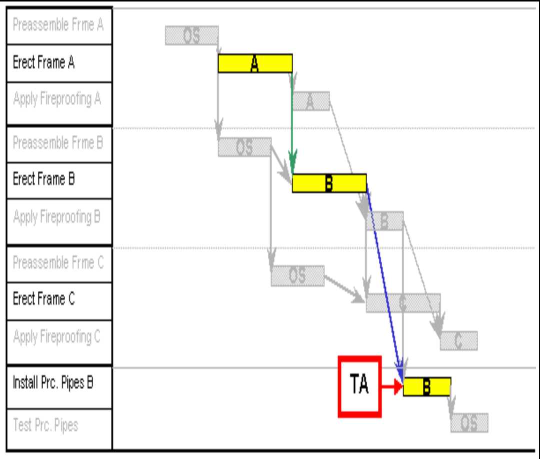


Figure 3.4c: Network chain 3.

Figure 3.4: Example network chains in the Intel CUB schedule.

Consequently, developing a network search chain algorithm first required enabling the algorithm to distinguish between related and unrelated activities. As discussed, Warshall's transitive closure algorithm (Warshall, 1962) finds the paths for any two vertices on a directed graph. Hence, I used his algorithm to perform a backward pass to iteratively identify related activities starting from a target activity to the first or start activity in a CPM network. For example, Figure 3.5 shows the backward pass performed for a sample CPM network. The following code shows the Warshall's algorithm implemented in CLCPM to identify the related activities using a backward pass.

```
'Add TA to currentTasks collection
currentTasks.Add TA, CStr(TA.ID)
'BACKWARD PASS
Do While Not currentTasks.Count = 0 'i.e., no more predecessors
For Each currentTask In currentTasks
 For Each predTask In currentTask.PredecessorTasks
  For Each succTask In predTask.SuccessorTasks
    If succTask.Text10 = "TRUE" Then
    predTask.Text10 = "TRUE"|
    End If
  Next succTask
  nextTasks.Add predTask
 Next predTask
Next currentTask
Set currentTasks = nextTasks
Set nextTasks = Nothing
Loop
```

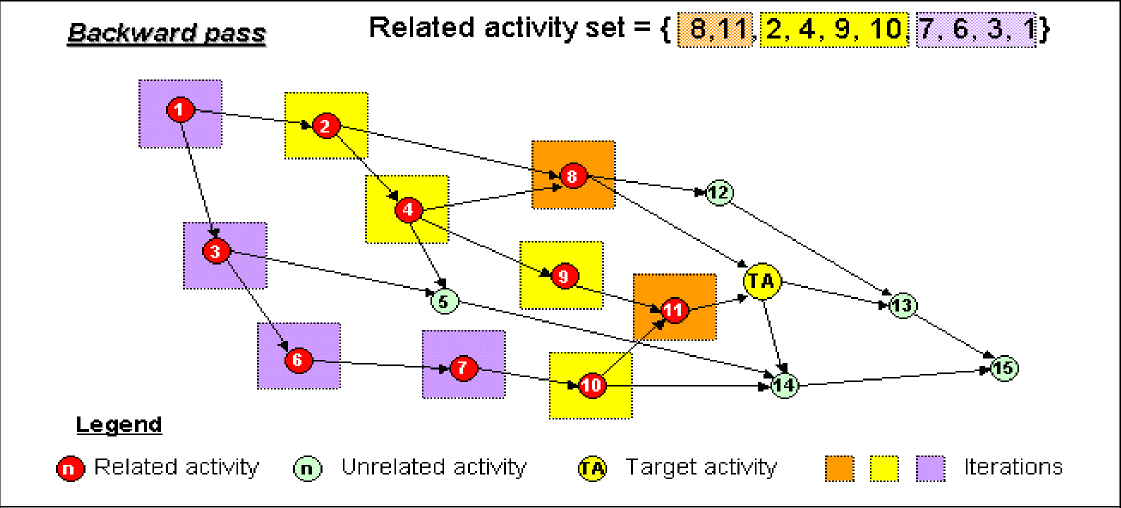


Figure 3.5: Backward pass using transitive closure algorithm to identify related activities for a sample CPM network.

Starting with the target activity (TA), the algorithm identifies the activity's immediate predecessors (i.e., activities 8 an 11). Hence, these are related activities. For each of these activities, the algorithm identifies its predecessors (i.e., 8: 2 and 4; 11: 9 and 10), and for each of these predecessors, queries whether at least one of the predecessor's successors is a related activity. If so, the predecessor is also a related activity. For example, one of the activity 8's predecessors is 4. Activity 4 in turn has two successors 8 and 9. Since activity 8 is a related activity, activity 4 is also a related activity. The algorithm repeats this process until in reaches activity 1. The related activity set shown in the figure describes the order in which the related activities are identified.

Once the related activities are identified, the algorithm also uses the distinctions (i.e., related versus unrelated) in a forward pass to incrementally construct the network chains between a single related activity and a given a target activity. As shown in Figure 3.6, the algorithm steps through each related activity and adds paths (i.e., constraints) until it reaches the target activity.

The process is repeated for each related activity in the CPM network. I once again used Warshall's transitive closure algorithm to perform the forward pass.

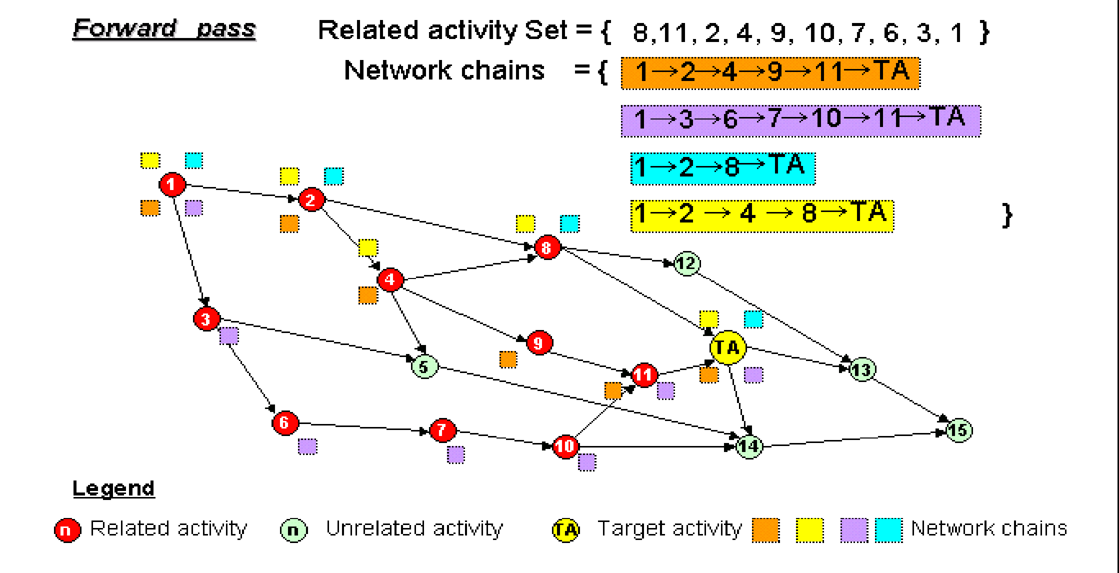


Figure 3.6: Forward pass using transitive closure algorithm to identify unique network chains for a sample CPM network.

Having identified the related activities, the algorithm starts from the first related activity (i.e., activity 1) and queries each of its successors on whether it is a related activity. For all successors that are related activities (i.e., activities 2 and 3), the algorithm stores the path between the related activity and the successors (i.e., 1→2. 1→3). Then, for each of the related activities identified, (i.e., activities 2 and 3), the algorithm queries its successors (i.e., 2: 4 and 8, 3: 5 and 6) on whether they are related activities. For all successors that are related activities (i.e., 4, 8 and 6), the algorithm adds the paths between the related activities and the successor (3->6) to the initially stored paths (1->2, 1->3). For example, the related activity 2 in the figure has two successors that are themselves related activities (i.e., activities 4 and 8). Hence, the algorithm adds the paths 2→4 and 2→8 to the initially stored path 1→2. The process is repeated until the target activity is reached. The figure shows that the algorithm identifies four unique network chains for the related activity 1.

This section described a formal network chain search algorithm that identifies unique network chains between a related activity and a target activity in CPM network. As discussed, the classification mechanism also needs inference rules defined to correctly classify the role and status of activities. The following section describes these rules in detail.

3.3.1.2 Inference Rules

As discussed, the role and status of activities are dependent on the role and flexibility of constraints that exist in a related activity's network chain. Hence, the inference rules need to correctly formalize how planners classify a related activity using the role and flexibility of constraints in a single network chain. As shown in Figure 3.7, a related activity can have multiple network chains. In addition, multiple constraints can exist in one of the activity's network chains. For example, Figure 3.7 shows two constraints, *workspace* and *supported by* constraints, between two activities in one of the network chains for the activity Erect Frame A.

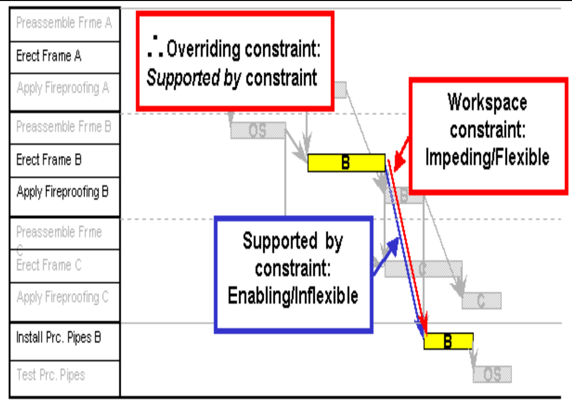


Figure 3.7: Example of multiple constraints.

Figure illustrates multiple constraints between two activities to describe sequencing rationale in the Intel CUB schedule.

The following two sections introduce inference rules that formalize the inference process for classifying the role and status of activities, respectively.

(1) Inference rules for classifying the role of activities

I define a related activity to be enabling if and only if the activity is enabling an activity that in turn is enabling the target activity. Enabling activities are evaluated based on the role of constraints. Formalizing this logic required defining the following set of inference rules.

Multiple constraint rule

In cases where multiple constraints exist and their values for role differ, I define the enabling constraint to override the impeding constraint. I call this constraint the "overriding" constraint. Hence, I define the following rule:

Rule 1: Enabling/Impeding-multiple constraint rule

If at least one of the constraints is enabling, the enabling constraint is the overriding constraint.

For example in Figure 3.7, the *workspace* constraint is impeding, while the *supported by* constraint is enabling and hence the *supported by* constraint becomes the overriding constraint.

Single network chain rule

As discussed, I define a related activity to be enabling if the activity enables an activity that in turn is enabling the target activity. This in effect requires all overriding constraints in a related activity's network chain to be enabling. More formally, I define the following rule:

Rule 2: Enabling/Impeding-single network chain

An activity is enabling if and only if all overriding constraints in the activity's network chain are enabling.

I demonstrate this rule using the test case example. Figure 3.8a shows one of the network chains of the activity Erect Frame A. The *resource* constraint within this network chain is impeding. Correspondingly, the activity Erect Frame A is impeding. This is logical as the activity Erect Frame A does not provide support for the target activity or its successor Erect Frame B. Hence, with respect to the target activity, it is impeding.

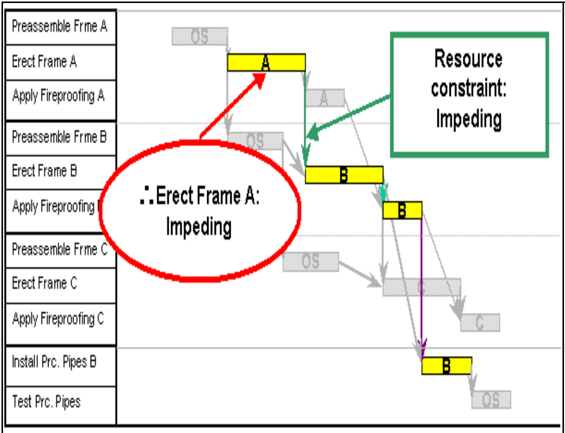


Figure 3.8a: Role classification (network chain 1)

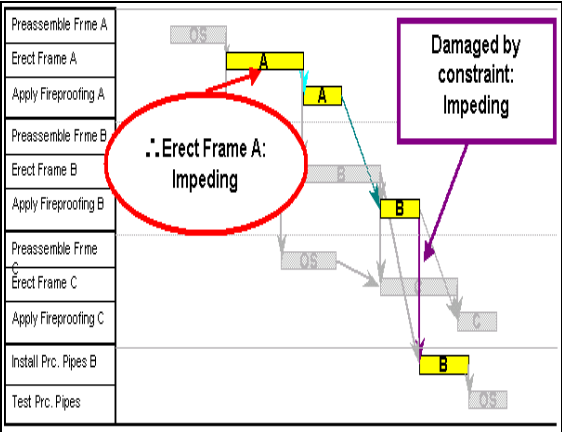


Figure 3.8b: Role classification (network chain 2)

Figure 3.8: Examples of inference rules used to infer the role of activities.

Figure illustrates inference rules used to infer the role of the activity Erect Frame A in the Intel CUB schedule.

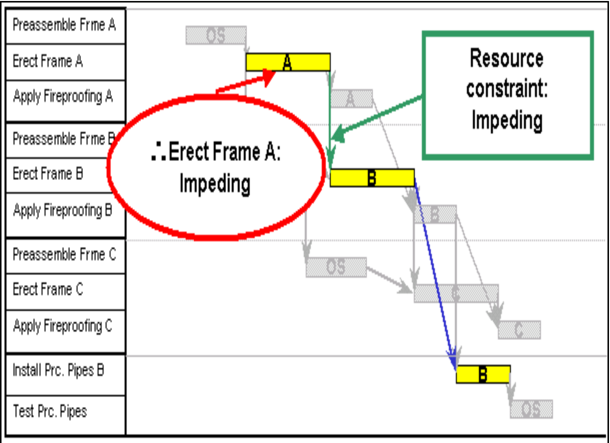


Figure 3.8c: Role classification (network chain 3).

Figure 3.8: Examples of inference rules used to infer the role of activities.

Multiple network chain rule

As discussed, a related activity can have multiple network chains. Applying the single network chain rule to each of the network chains can potentially return conflicting values (i.e., enabling versus impeding) for a related activity. I call such conflicts a "classification" conflict.

In such cases, I define the enabling classification to override the impeding classification. More formally, I define the following rule:

Rule 2.1: Enabling/Impeding-multiple network chain

An activity is enabling if at least one of the activity's network chains returns the activity as enabling.

I illustrate the rule using the test case example. Activity Erect Frame A is part of three network chains that are highlighted in Figures 3.8a, 3.8b and 3.8c. As the figures show, each of the network chains has at least one impeding constraint. Hence, each network chain returns the activity to be impeding. Hence, in this case, no classification conflict exists, and the activity is impeding.

(2) Inference rules for classifying the status of activities

As discussed in the test case example, I define an activity to be driving if the activity cannot be delayed by either using float or relaxing constraints between the activity and the target activity. Hence, the status of an activity needs to be evaluated with respect to its float value as well as the flexibility of its constraints. Formalizing this logic requires defining the following set of inference rules.

Multiple constraint rule

In cases where multiple constraints exist and their values for flexibility differ, I defined the inflexible constraint to override the flexible constraint. This is logical as relaxing one of the flexible constraints still prevents the activity from being delayed due to the inflexible constraint. Hence, I define the following rule:

Rule 3: Driving/non-driving-multiple constraint rule

If at least one of the constraints is inflexible, the inflexible constraint is the overriding constraint.

For example, in Figure 3.7, although the *workspace* constraint is flexible, the *supported* by constraint is inflexible and hence becomes the overriding constraint.

Single network chain rule

A related activity with a single network chain can be shifted forward if one or more overriding constraints in its network chain are flexible. However, even if the overriding constraints of a related activity's network chain are inflexible, the activity can still be delayed if the activity has positive target float. As discussed in the test case, the target float is the activity's total float calculated with respect to the target activity. Hence, only the network chains populated by activities with zero target float are applicable for determining whether an activity is driving or non-driving. I call such network chains "critical" network chains. More formally, I define the following rule:

Rule 4: Driving/non-driving-single network chain

An activity is driving if and only if all overriding constraints on the "critical" network chain are inflexible.

I demonstrate this rule using the test case example. As shown in Figures 3.9a, 3.9b and 3.9c, the activity Erect Frame A has three network chains. Of these network chains, only the first network chain is a critical network chain (Figure 3.9a) and is the cause for the activity Erect Frame A having zero target float. Within this critical network chain, the *resource* constraint is flexible. Hence, the activity is non-driving.

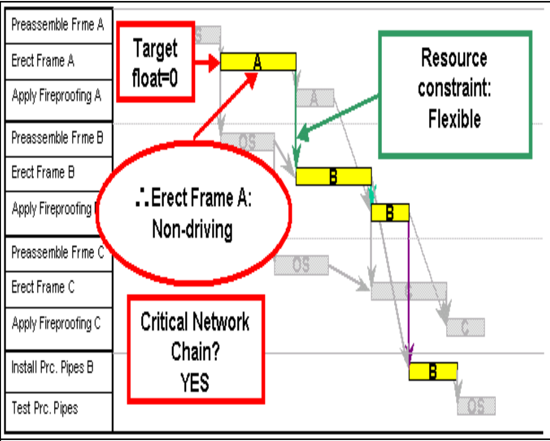


Figure 3.9a: Status classification (network chain 1)

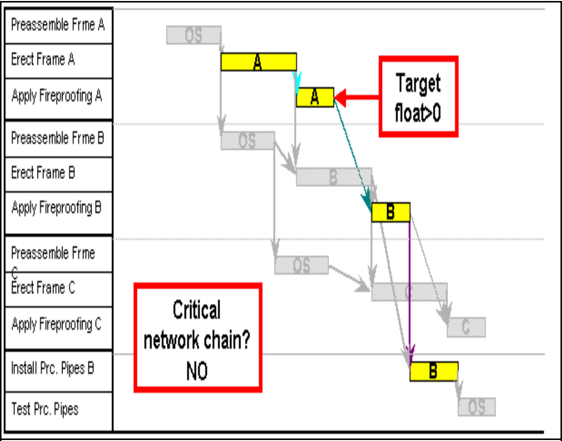


Figure 3.9b: Status classification (network chain 2)

Figure 3.9: Examples of inference rules used to infer the status of activities.

Figure illustrates inference rules used to infer the status of the activity Erect Frame A in the Intel CUB schedule.

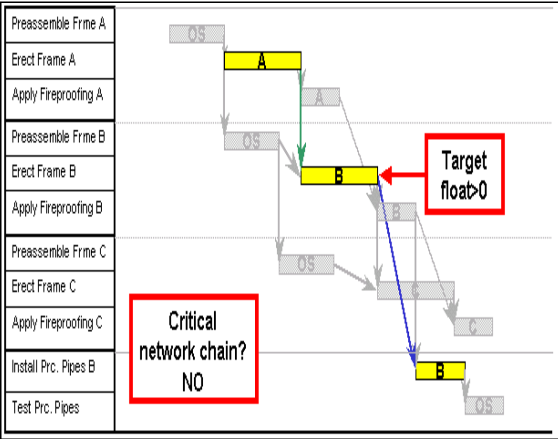


Figure 3.9c: Status classification (network chain 3).

Figure 3.9: Examples of inference rules used to infer the status of activities.

Multiple network chain rule

As discussed, a related activity can have multiple network chains. Of these network chains, one or more can be critical network chains. Hence, a classification conflict can occur when two or more critical network chains return conflicting values (e.g., driving versus non-driving) for a related activity. In such cases, a related activity is driving if at least one of the critical network chains returns the activity to be driving. More formally, I define the following rule:

Rule 4.1: Driving/non-driving-multiple network chain

An activity is driving if at least one of the activity's network chains returns the activity as driving.

I demonstrate this rule using the test case example. As discussed, only the network chain in Figure 3.9a is a critical network chain. Hence, in this case no classification conflict occurs and the activity is non-driving.

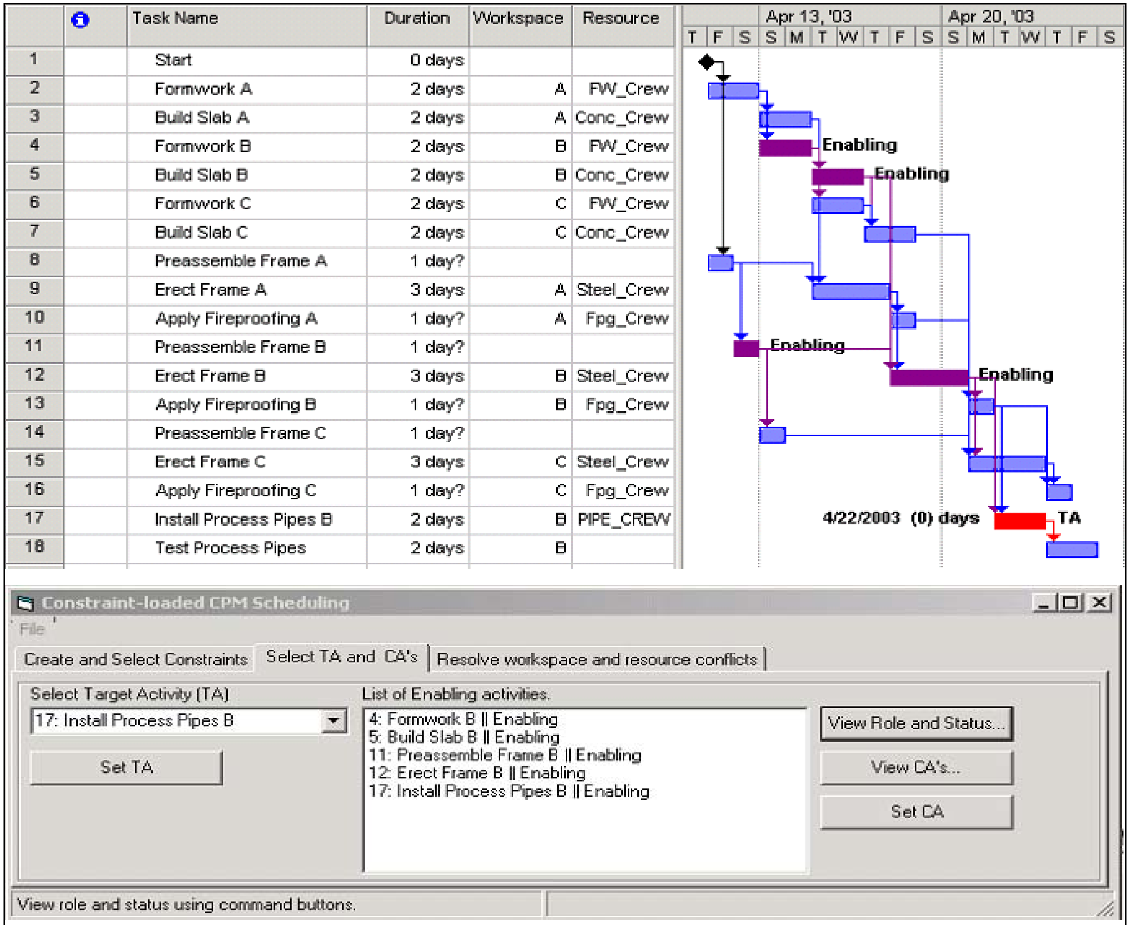
In summary, the sections above presented a network search algorithm that automatically identifies unique network chains for related activities in a CPM network. I also presented inference rules that formalize the planners' inference process for classifying the role and status of activities given multiple network chains in a CPM network.

As discussed, the purpose of developing the classification mechanism (i.e., the network search chain algorithm and inference rules) was to enable a computer system to automatically classify activities with respect to their role and status rapidly and correctly. As discussed in Chapter 2, the ontology is designed in a way to support the classification mechanism. That is, the ontology formalizes the role and flexibility of specific types of constraint rationale as abstract types. The ontology was implemented in CLCPM. Hence, I also implemented the classification mechanism in CLCPM.

The following section describes the implementation.

3.3.2 Implementation

I implemented the network chain search algorithm and inference rules in CLCPM. Figures 3.10 and 3.11 show the classification mechanism used to classify the role and status of activities for the Intel CUB schedule. The figures show how CLCPM classifies the role and status of activities with respect to the target activity Install Process Pipes B accurately and thoroughly.



		Task Name	Duration	Workspace	Resource
1		Start	0 days		
2		Formwork A	2 days	A	FW_Crew
3		Build Slab A	2 days	A	Conc_Crew
4		Formwork B	2 days	B	FW_Crew
5		Build Slab B	2 days	B	Conc_Crew
6		Formwork C	2 days	C	FW_Crew
7		Build Slab C	2 days	C	Conc_Crew
8		Preassemble Frame A	1 day?		
9		Erect Frame A	3 days	A	Steel_Crew
10		Apply Fireproofing A	1 day?	A	Fpg_Crew
11		Preassemble Frame B	1 day?		
12		Erect Frame B	3 days	B	Steel_Crew
13		Apply Fireproofing B	1 day?	B	Fpg_Crew
14		Preassemble Frame C	1 day?		
15		Erect Frame C	3 days	C	Steel_Crew
16		Apply Fireproofing C	1 day?	C	Fpg_Crew
17		Install Process Pipes B	2 days	B	PIPE_CREW
18		Test Process Pipes	2 days	B	

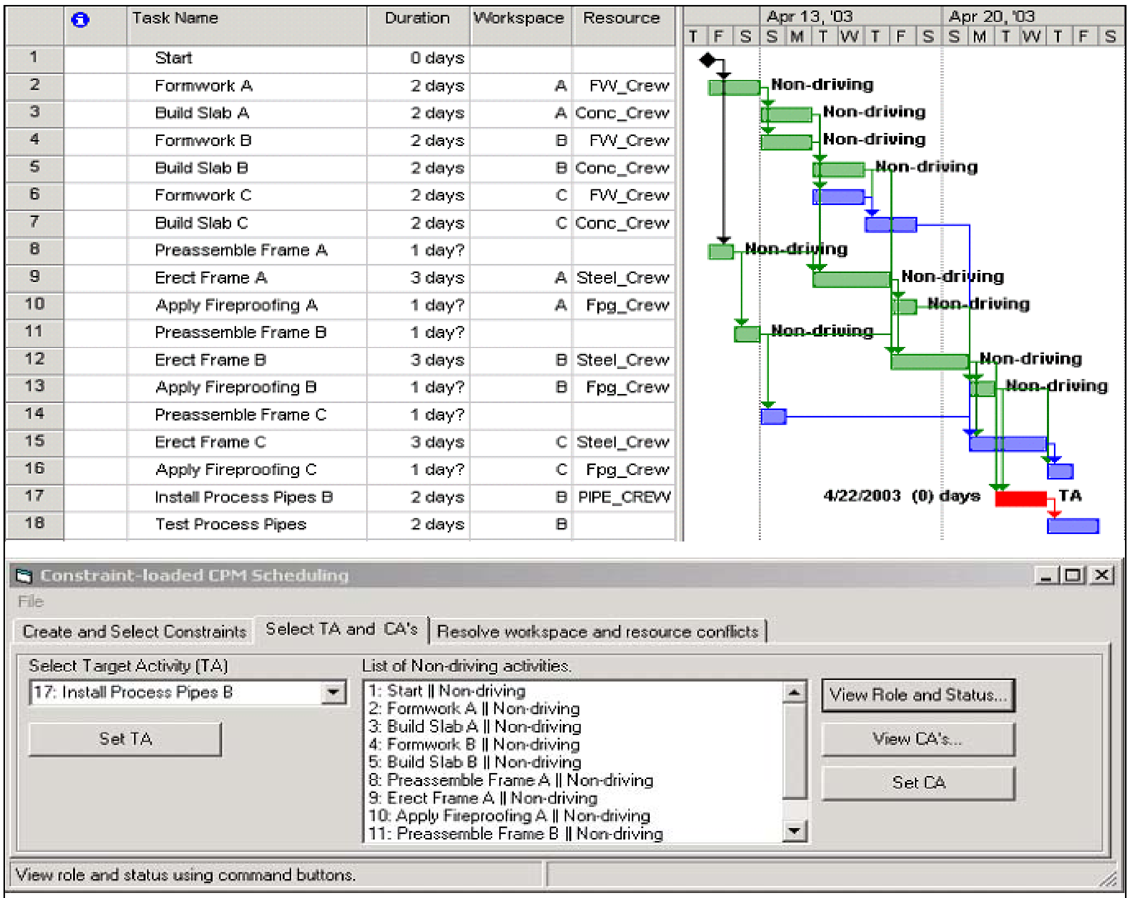
Select Target Activity (TA)
17: Install Process Pipes B

List of Enabling activities.
4: Formwork B || Enabling
5: Build Slab B || Enabling
11: Preassemble Frame B || Enabling
12: Erect Frame B || Enabling
17: Install Process Pipes B || Enabling

View role and status using command buttons.

Figure 3.10: CLCPM user interface used to identify the role of activities in the Intel CUB schedule.

Figure shows the enabling activities for the related activities of the target activity Install Process Pipes B only. As shown in the figure, the activities CLCPM classified as enabling are: Formwork B, Build Slab B, Preassemble B, Erect Frame B and Install Process Pipes B.



	Task Name	Duration	Workspace	Resource
1	Start	0 days		
2	Formwork A	2 days	A	FW_Crew
3	Build Slab A	2 days	A	Conc_Crew
4	Formwork B	2 days	B	FW_Crew
5	Build Slab B	2 days	B	Conc_Crew
6	Formwork C	2 days	C	FW_Crew
7	Build Slab C	2 days	C	Conc_Crew
8	Preassemble Frame A	1 day?		
9	Erect Frame A	3 days	A	Steel_Crew
10	Apply Fireproofing A	1 day?	A	Fpg_Crew
11	Preassemble Frame B	1 day?		
12	Erect Frame B	3 days	B	Steel_Crew
13	Apply Fireproofing B	1 day?	B	Fpg_Crew
14	Preassemble Frame C	1 day?		
15	Erect Frame C	3 days	C	Steel_Crew
16	Apply Fireproofing C	1 day?	C	Fpg_Crew
17	Install Process Pipes B	2 days	B	PIPE_CREW
18	Test Process Pipes	2 days	B	

Constraint-loaded CPM Scheduling

File

Create and Select Constraints | Select TA and CA's | Resolve workspace and resource conflicts

Select Target Activity (TA): 17: Install Process Pipes B

List of Non-driving activities.
1: Start || Non-driving
2: Formwork A || Non-driving
3: Build Slab A || Non-driving
4: Formwork B || Non-driving
5: Build Slab B || Non-driving
8: Preassemble Frame A || Non-driving
9: Erect Frame A || Non-driving
10: Apply Fireproofing A || Non-driving
11: Preassemble Frame B || Non-driving

View role and status using command buttons.

Figure 3.11: CLCPM user interface used to identify the status of activities in the Intel CUB schedule.

The figure shows the non-driving activities for the related activities of the target activity Install Process Pipes B only. As shown in the figure, the activities CLCPM classified as non-driving are: Formwork A, Build Slab A, Formwork B, Build Slab B, Preassemble Frame A, Preassemble Frame B, Erect Frame A and Erect Frame B and Apply Fireproofing A and Apply Fireproofing B.

This section described a formal classification mechanism for automatically classifying activities with respect to their role and status. The next section describes how the inferences rules and network search algorithms were validated using CLCPM.

3.4 VALIDATION

As discussed in Section 3.1, the primary motivation for formalizing a classification mechanism was to alleviate planners from having to manually classify the role and status of activities with respect to the target activity when developing sequencing alternatives. Planners need to classify the role and status of activities to identify activities suitable for delay, and also use the classifications to resolve workspace and resource conflicts during the re-sequencing process accurately and rapidly. Consequently, I demonstrated that the classification mechanism is formal and general by testing the identification and re-sequencing process. I discuss the validation methods and results in the next chapter.

3.5 CONCLUSIONS

This chapter presented a formal classification mechanism that automatically infers the role and status of activities given a CPM network schedule where the rationale for activity sequences has been explicitly represented using the formal ontology described in Chapter 2.

I formalized a network search algorithm that identifies unique network chains between an activity and a target activity in a CPM network. I also formalized inference rules that generalize the planners' inference process for classifying the role and status of an activity based on the role and flexibility of the specific constraints that exist in the activity's network chains. Using the formalization, a prototype tool, CLCPM can automatically identify the role and status of activities regardless of the different types of specific constraints used to describe sequencing rationale in a CPM network schedule. Hence, the formal mechanism obviates the need for planners to manually classify activities when trying to identify and re-sequence activities to develop sequencing alternatives.

The classification mechanism is limited in many ways. Many of these limitations are due to the assumptions and design of the ontology. Specifically, the ontology was limited to describing sequencing rationale between construction activities, as to design and procurement activities. The extension of the ontology (e.g., a ternary classification as to the current binary classification of constraints) will correspondingly require extending the inference rules to meet these changes. I also designed the ontology to be used for FS precedence relationships. The extension of the ontology to incorporate other precedence relationships would correspondingly require extending the network search chain algorithm to correctly identify unique network chains.

Finally, the run time performance of the transitive closure algorithm I used (i.e., Warshall's algorithm) is $O(n^3)$. As discussed, I did not evaluate other algorithms, but chose Warshall's algorithm, as it was simple to code. For large-scale networks, the run-time may limit the effectiveness of the classification mechanism. Thus, alternative algorithms with faster performance may need to be explored.

The purpose of developing a classification mechanism was to support planners in identifying activities to delay and prioritizing activities when developing sequencing alternatives. Hence, the classification mechanism needs to be integrated into a formal process that supports planners in developing sequencing alternatives correctly and rapidly. This formal process is introduced in the following chapter.

CHAPTER 4. FORMAL IDENTIFICATION AND RE-SEQUENCING PROCESS

4.1 INTRODUCTION

In Chapter 1, I used a test case to illustrate that planners frequently re-sequence activities to expedite particular activities of a project. To make well-informed re-sequencing decisions, planners need to identify activities to delay and re-sequence activities correctly and quickly to experiment with different sequencing scenarios. However, the current CPM-based scheduling tools do not support the easy or accurate re-sequencing of activities. Consequently, CPM-based scheduling tools limit planners' ability to thoroughly identify potential activities to delay, and require planners to re-sequence activities manually and in an ad hoc manner. Hence, a goal of this research was to develop a formal identification and re-sequencing process that assists planners in systematically identifying and developing sequencing alternatives correctly and quickly. In Chapter 2, I presented a formal ontology that enables planners to describe sequencing rationale accurately and in a computer-interpretable form. In Chapter 3, I presented a classification mechanism that automatically classifies the role and status of activities with respect to the activity requiring earlier execution. These formalizations are the key components in representing the planner's inference process for classifying activities to delay using the rationale of constraints. Therefore, the next step of the research required formalizing an identification and re-sequencing process that helps planners utilize the ontology and classification mechanism to develop multiple sequencing alternatives.

Meeting this goal required understanding the step-by-step process planners manually take and how they conceptually use the role and status of activities to re-sequence activities. Hence, I performed paper-based "Gedanken" experiments (Chaitin, 1965) to step through the individual tasks. I identified that planners identify activities to delay by classifying the role and status of activities on the critical path with respect to a target activity. To re-sequence activities, planners first use the role and status of activities that are in workspace or resource conflict to decide *which* needs to be delayed. Subsequently, planners "shift forward" the activity selected to delay either by using the activity's float or by relaxing flexible constraints in the selected activity's network chain. Thus, the research challenge was to model why and how planners re-sequence activities in a formal and general way, so that planners in turn could use the formalized processes to correctly and rapidly develop sequencing alternatives for different project schedules.

To develop a process that meets such requirements, I investigated prior research efforts that have developed CPM-based techniques with the goal of accelerating the overall schedule duration. The techniques include time-cost trade-off analysis, (e.g., Fondahl, 1961; Meyer and Shaffer, 1963; Paulson, 1971) and resource-constrained scheduling (e.g., Crandall, 1985; Chang et al., 1989). These techniques assume that activity sequences in a CPM network are fixed and that resources can be added to accelerate the schedule duration. These techniques do not allow planners to re-sequence activities with the goal of expediting a specific target activity. They also do not promote planners to use their judgment to make decisions when developing sequencing alternatives, as they are frequently reduced to rigidly coded computer algorithms (Paulson, 1973). I also investigated "replanning" techniques in general-purpose AI planning systems. However, these replanning techniques were not designed to meet the domain-specific requirements (i.e., a construction specific ontology and its utilization) needed to re-sequence activities correctly.

Based on the "Gedanken" experiments and investigation of CPM-based acceleration techniques and AI replanning techniques, I formalized an identification and re-sequencing process. The formal process uses the ontology and classification mechanism to assist planners in developing sequencing alternatives correctly and rapidly. The identification process identifies a set of "candidate" activities (i.e., activities that expedite a target activity if delayed). Planners can delay one of these activities to expedite a target activity. The re-sequencing process subsequently assists planners by using a set of pre-defined priority rules to delay selected activities correctly. Planners can repeat the process to develop multiple sequencing alternatives.

This chapter presents in detail the investigations performed and a formal identification and re-sequencing process. I also present three retrospective case studies and a charrette test I performed to validate the formal process. The retrospective cases provide evidence that the process is formal and general, as it demonstrates that the process correctly identifies and re-sequences activities for three different types of construction schedules. The charrette test provides evidence that the process is formal, as it demonstrates that the process enables users to develop sequencing alternatives more correctly and rapidly than a conventional CPM scheduling tool. I conclude with a discussion of the immediate implications of the formal process and its limitations.

The following section describes a Gedanken experiment I performed for the Intel CUB schedule. The test case illustrates the design requirements of the identification and re-sequencing process.

4.1.1 Motivating Case Example

To demonstrate the step-by-step process planners take to re-sequence activities, I revisit the Intel CUB schedule. As discussed in Chapter 1, the project manager identified the activity Apply Fireproofing B to be an activity that, when delayed, expedites the target activity Install Process Pipes B. He understood that the activity could be delayed as the activity was an impeding and non-driving type of activity (Figure 4.1a). In addition, he understood that the activity when delayed expedites the target activity as it is on the critical path (i.e., target float=0). As discussed, I call such activities "candidate" activities.

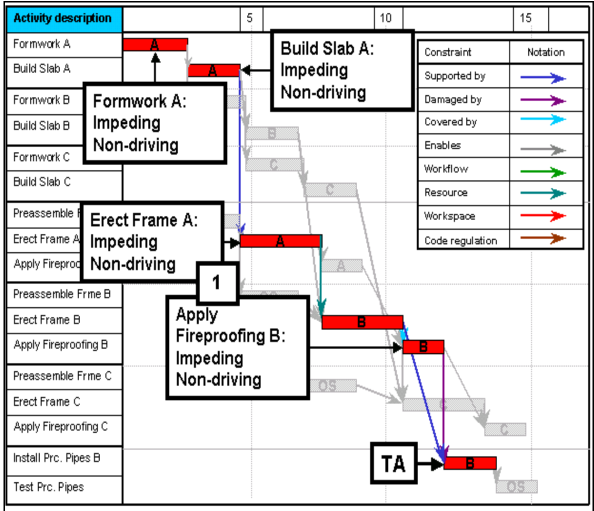


Figure 4.1a: Step 1.

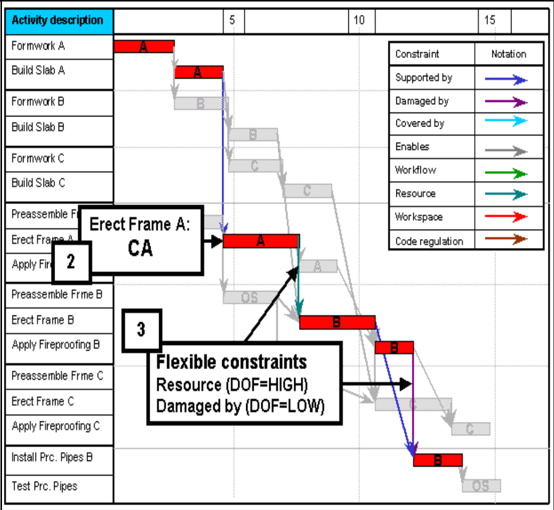


Figure 4.1b: Step 2 and 3.

Figure 4.1: Example of steps for identifying candidate activities.

Figure shows the steps the project manager of Intel CUB took to identify candidate activities.

Hence, to identify candidate activities, the project manager needs to classify the role and status of activities on the critical path with respect to the target activity. As shown in Figure 4.1a (Step 1), the candidate activities for the target activity Install Process Pipes B were the activities Erect Frame A, Build Slab A and Formwork A. Thus, the project manager can delay any one of these candidate activities to expedite the target activity.

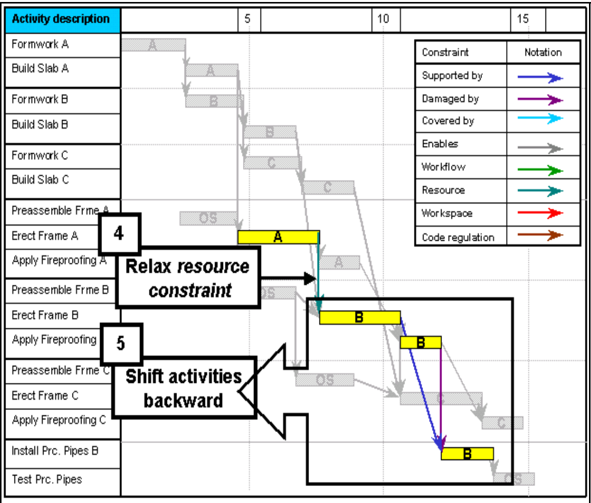


Figure 4.2a: Steps 4 and 5.

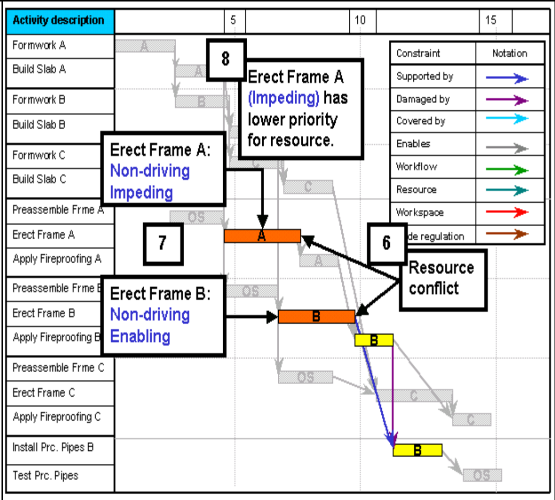


Figure 4.2b: Steps 6, 7 and 8.

Figure 4.2: Example of steps for re-sequencing activities.

Figure shows the steps the project manager of Intel CUB took to re-sequence the candidate activity Erect Frame A.

Assuming the project manager selects the activity Erect Frame A as the candidate to delay, he then needs to relax one of the flexible constraints in the activity's critical network chain. There are two flexible constraints: the *damaged by* constraint and the *resource* constraint (Figure 4.1b, steps 2 and 3). Assuming he chooses the relatively more flexible *resource* constraint to relax, he subsequently needs to shift the target activity and its predecessors backward[15] (Figure 4.2a, steps 4 and 5). The activity Install Process Pipes B can now start a day earlier. However, a resource conflict exists between the activities Erect Frame A and Erect Frame B (Figure 4.2b, step 6). The project manager now needs to decide which of the two activities he should delay to resolve the conflict. The project manager first updates the role and status of the two activities. He understands that both activities can be delayed as they are both non-driving. However, Erect Frame A is an impeding activity, whereas the activity Erect Frame B is an enabling activity. Hence, he gives priority for resources to the enabling activity (Figure 4.2b, steps 7 and 8) and determines the activity Erect Frame A as the activity to delay (i.e., activity of lower priority).

[15] Backward with respect to time (i.e., expedites).

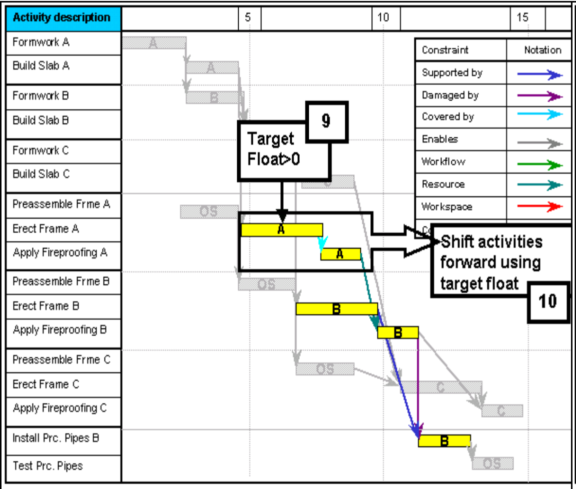


Figure 4.3a: Steps 9 and 10.

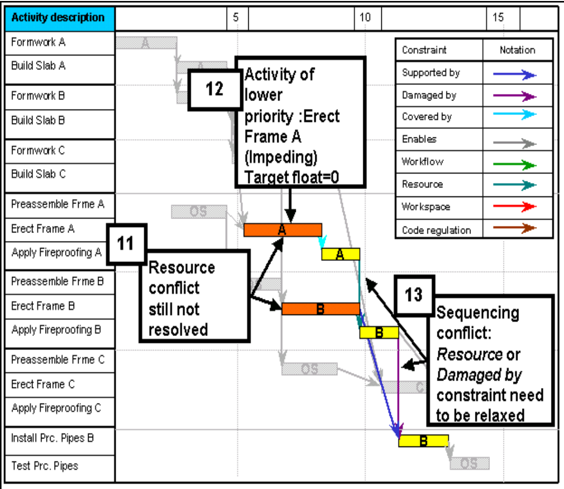

Activity description
12
Activity of lower priority :Erect Frame A (Impeding) Target float=0
11
Resource conflict still not resolved
13
Sequencing conflict: Resource or Damaged by constraint need to be relaxed
Code regulation

Figure 4.3b: Steps 11,12 and 13.

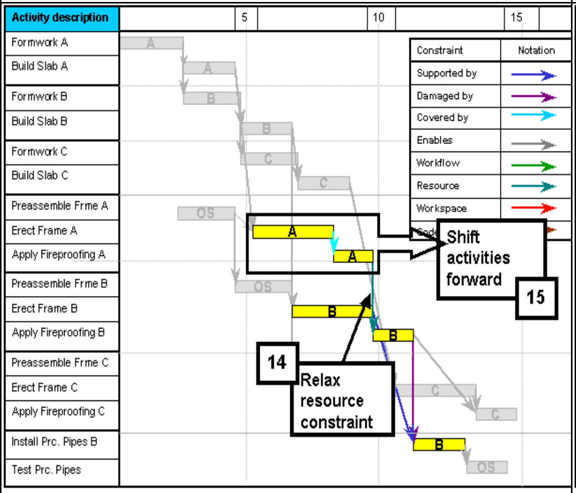


Figure 4.3c: Steps 14 and15.

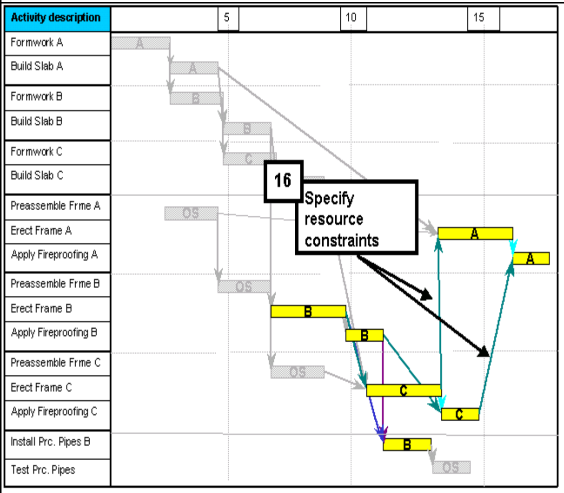


Figure 4.3d: Step 16.

Figure 4.3: Example of steps for re-sequencing activities.

Figure shows the steps the project manager of Intel CUB took to re-sequence the candidate activity Erect Frame A.

As the activity Erect Frame A has positive target float, he tries to resolve the workspace or resource conflict by delaying the activity (Figure 4.3a, steps 9 and 10). However, the conflict is still not resolved. The activity Erect Frame A is still the lower priority activity. However, the activity now has zero target float, and is linked to the target activity by a critical network chain. Therefore, a "sequencing conflict" has occurred (Figure 4.3b, steps 11, 12). Further delaying the activity will in turn delay the target activity. Hence, the only way to resolve the sequencing conflict is to relax flexible constraints in the activity's critical network chain. The activity can only be delayed by relaxing either the *resource* constraint or the *damaged by* constraint (Figure 4.3b, step 13). Again, assuming the project manager decides to relax the *resource* constraint, the relaxation of the constraint enables the activity to be delayed further, resolving the sequencing conflict and correspondingly the resource conflict (Figure 4.3c, steps 14 and 15). Finally, the planner specifies a *resource* constraint to ensure that the logic remains correct (Figure 4.3d, step 16).

Table 4.1 summarizes the steps the planner needs to take. The table shows that several of the steps are repetitive. Planners need to continually expedite or delay activities until no further workspace or resource conflicts exist. When a resource or workspace conflict is identified, planners need to first decide which activity to delay. As the example shows, the planners' rationale for determining the activity to delay is based on the role and status of the activity pair in conflict. Subsequently, they need to either delay the activity selected to delay (i.e., lower priority activity) using available target float, or relax a flexible constraint in the selected activity's network chain.

Processes	Required Steps	Steps in Figures 4.1, 4.2 and 4.3.	Required Representation or Mechanism
Identification process	Identify CA's	Step 1	• Classification mechanism
	Select CA	Step 2	• NA
Re-sequencing process	Identify constraints to relax in activity's network chain	Step 3, Step 13	• Network chain search algorithm • Ontology
	Relax constraint	Step 4, Step 14	• Ontology
	Shift activities backward or forward	Step 5, Step 10, Step 15	• NA
	Identify activities in resource or workspace conflicts	Step 6, Step 11	• NA
	Update role and status of activities	Step 7, Step 12	• Classification mechanism
	Decide which activity to delay	Step 8, Step 12	• Classification results
	Determine whether float exists	Step 9, Step 12	• NA

Table 4.1: Summary of the required steps described in the test case example.

The example case demonstrated the delay of only one of the candidate activities. Planners need to repeat the identification and re-sequencing processes to delay additional candidate activities. Furthermore, each time a candidate activity has been delayed, a new set of candidate activities needs to be identified as the critical path of the schedule may have changed.

The example shows that developing multiple sequencing scenarios can be time-consuming and error-prone. Hence, the identification and re-sequencing process needs to be formalized so that planners can utilize the ontology and classification mechanism to make correct and quick decisions while developing sequencing alternatives. The classification mechanism is required to identify CA's and update the classification of activities in resource or workspace conflict. The ontology is required to support the classification mechanism, but is also required for planners to determine which constraints can be relaxed.

In summary, planners need a formal process that guides and supports them while developing sequencing alternatives. The formal process needs to enable planners to correctly and rapidly identify and re-sequence activities for different construction schedules that may have different types of sequencing rationale. The following section summarizes the design requirements of the formal process.

4.1.2 Research Goals

The case example illustrates that the identification and re-sequencing process needs to be designed in a formal and general way:

(1) Formal: A formal identification and re-sequencing process must ensure that planners identify and re-sequence activities correctly, rapidly and predictably, consistently. The identification process needs to identify the correct candidate activities with respect to a target activity. As shown in the test case, the identification process utilizes the classification mechanism to identify candidate activities. Thus, the accuracy of the identification process is dependent on the correctness of the classification mechanism. The re-sequencing process needs to ensure that planners re-sequence activities correctly so that the target activity is expedited in the sequencing alternative. Both processes need to enable planners to develop multiple sequencing scenarios rapidly.

(2) General: The identification and re-sequencing process needs to enable planners to develop sequencing alternatives for construction schedules that may have different types of specific constraints describing sequencing rationale in these schedules.

The identification and re-sequencing process formalized in this research meets these criteria. Developing a formal process required investigating existing CPM-based techniques used for accelerating schedule durations and "replanning" techniques in AI planning systems. The next section describes the related research background.

4.2 RESEARCH BACKGROUND AND POINTS OF DEPARTURE

As discussed, the goal of the identification and re-sequencing process is to assist planners in correctly re-sequencing activities in a CPM network so that a target activity can be expedited. In the construction literature, several researchers have developed CPM-based techniques for accelerating schedule durations while optimizing the use of resources. In Artificial Intelligence, researchers developed "replanning" techniques in general purpose AI planning systems to repair plans when a plan fails during its execution. The motivation for investigating these techniques was to determine whether these techniques could be used or adopted to re-sequence activities by selectively relaxing constraints in a CPM schedule. The following section introduces both of these approaches.

4.2.1 CPM-based Schedule Acceleration Techniques

Currently, the primary techniques for accelerating schedule durations are time-cost trade-off (TCTP) analysis and resource constrained scheduling (i.e., resource allocation). In this section, I provide a brief overview of the fundamental concepts and advances made to date. I also describe how I adopt approaches similar to those used in these techniques and explain that a different heuristic is required to formalize the identification and re-sequencing process. In addition, I also describe the practical limitations of these techniques originally identified and addressed by researchers as early as the 60's and 70's (Fondahl, 1961; Wiest and Levy, 1969; Paulson 1971; Crandall, 1970; Fondahl, 1991). These limitations are still relevant today for formalizing the identification and re-sequencing process.

4.2.1.1 Time-cost Trade-off Analysis

Kelley and Walker (1959) first introduced time-cost trade-off as an optimization feature in CPM. The goal of time-cost trade-off (TCTP) analysis or "network compression" is to expedite project delivery time while minimizing the associated increase in cost. Various techniques have been developed over the years to solve the time-cost trade-off problem, including heuristic methods (e.g., Fondahl, 1961; Prager, 1963; Paulson, 1971; Crandall, 1970; Siemens, 1971; Moselhi, 1993), optimization techniques in operations research (Meyer and Shaffer, 1963; Patterson and Huber, 1974; Burns et al., 1996; Mosehli and Lorterapong, 1993) and more recently genetic algorithms (Feng and Liu, 1997; Hegazy, 1999) and machine learning (Li et al., 1999).

The fundamental approach used by these techniques is to shorten the duration of activities on the critical path, starting with the activity with the least cost slope. The process can be repeated until the schedule duration meets a specific acceleration date that planners deem satisfactory, or alternatively until all possible solutions are exhausted.

These techniques assume that activity sequences are fixed in a CPM network. In addition, the method of accelerating the duration of individual activities is by incorporating overtime, resource augmentation or the use of alternative technologies (Pauslon, 1973).

However, as shown in the test case, situations occur during the course of a project when existing constraints between upstream activities need to be relaxed to expedite a specific downstream activity (i.e., a target activity). In addition, the test case shows that planners do not need to add more resources but rather relax constraints to expedite the activity.

Arguably, TCTP could be used to identify upstream activities to accelerate that will in turn expedite a particular downstream activity (and not necessarily the last or end activity). However, TCTP would still require accelerating the upstream activities by adding more resources, while retaining the original sequence of activities. Thus, the heuristics and algorithms developed for solving TCTP problems are not directly applicable for re-sequencing activities.

I do adopt the fundamental approach used to solve the TCTP problem, i.e., enabling planners the freedom to use the heuristics to either exhaust all potential solutions or alternatively until a satisfactory solution is met.

However, a formal process for identifying and re-sequencing activities needs to identify constraints to relax (as to activities that can be shortened) as potential solutions for expediting a target activity. Correspondingly, the degree of flexibility of each constraint (as to the activity's cost slope) needs to be the criteria for determining which constraints to relax first.

4.2.1.2 Resource and Workspace Allocation Techniques

Resource allocation assumes constraining limits on the availability of resources, then allocates the available resources to the project activities while trying to minimize the project duration (Paulson, 1971). Various techniques have been developed to solve the resource allocation problem using heuristic methods (Crandall, 1985; Chang et al., 1989; Halpin and Riggs, 1992, Abeyasinghe et al., 2001), or optimization techniques such as integer and dynamic programming (Davis, 1973, Lee and Gatton, 1994), genetic algorithms (Hegazy, 1999, Leu and Yang, 1999) and genetic algorithms combined with fuzzy logic (Leu et al., 1999). Recent attempts have focused on incorporating limited space availability as a nontangible resource (Zouein et al., 1993; Thabet and Beliveau, 1997). For example, Thabet and Beliveau (1997) developed ScarC[16], a knowledge based system that accounts for space limitations during the sequencing of activities.

The fundamental approach used by these techniques is to prioritize activities based on their time-criticality. That is, resource and workspace are allocated to activities with less float. However, as shown in the test case, situations occur when activities need to be re-sequenced with the goal of expediting a target activity. In such cases, activities also need to be prioritized with respect to their role and status in relation to the target activity.

Hence, the existing approaches used in resource and workspace allocation techniques (i.e., prioritizing activities exclusively with respect to their float) cannot be used to correctly re-sequence activities. Rather, a formal identification and re-sequencing process needs to enable planners to correctly prioritize activities with respect to their role and status.

4.2.1.3 Practical Limitations of CPM-based Techniques to Support Re-sequencing Decisions

Despite the advances made in the areas of time-cost trade-off analysis and resource constrained scheduling, many of the researchers do not address the practical problems originally identified by researchers in the early 60's and 70's (Fondahl, 1961, Paulson 1971, Crandall, 1970; Wiest and Levy, 1969; Fondahl, 1991). For example, Paulson (1971) notes that a primary reason for the lack of adoption of CPM-based techniques in the construction industry is both theoretical and practical. With respect to TCTP, Paulson states that the problems are (1) the mathematical definitions of the "critical path" and "activity float" lose their exact meaning when resource constraints are introduced, (2) TCTP requires impractical amounts of data to feed grossly over-simplified models, and (3) TCTP does not promote humans (e.g., planners) to use their judgment, expertise and insight during the analysis. As discussed in Chapter 3, Paulson (1971) improved the theoretical limitation of TCTP by adopting Wiest and Levy's (1969) "critical sequence" calculations, which enabled TCTP to be performed correctly in a resource-constrained schedule. However, the "unified" approach was designed not only to resolve the theoretical limitation, but designed in a way to support planners in using their judgment to focus on the critical information and input only the relevant information. Paulson (1971) notes that such an approach, i.e., techniques that encourage the input of human judgment, is the critical factor for CPM techniques to be successfully adopted in the construction industry. The so-called "human-judgment-oriented" philosophy (Fondahl, 1961) has been recognized as the correct way to develop decision-support

[16] Space-Constrained Resource-Constrained Scheduling System

tools and advocated by Kunz et al. (1994) as “Desktop engineering”. The approach has been successfully incorporated in construction domain-specific planning systems such as ITRMM (Kunz et al., 1995) and CMM (Aalami et al. 1998a) where the input of data is large and the role of a tool is to assist in helping planners make informed decisions.

Correspondingly, I formalized the identification and re-sequencing process as a decision-support tool, and not as an optimization tool. The identification and re-sequencing process has been designed to enable planners to make decisions during the identification and re-sequencing process, and hence rely on the planner’s judgment and expertise to develop sequencing alternatives.

The sections above described the theoretical and practical limitations of existing CPM-based acceleration techniques for re-sequencing activities to develop sequencing alternatives. The following section describes “replanning” techniques used in general purpose AI planning systems I investigated to explore whether these techniques provide an appropriate framework for the identification and re-sequencing process required by construction planners.

4.2.2 “Replanning” Techniques in AI Planning Systems

“Replanning” is a term formally used in AI literature to describe the techniques used in general purpose AI planning systems to repair plans when a plan fails during the execution. In the blocks world, replanning is achieved by execution monitoring agents that identify changes in their environment and automatically repair plans by using planning agents to develop new planning alternatives. For example, PLANEX (Fikes et al., 1972) used execution monitoring together with the STRIPS planner to control the robot Shakey. However, partial order planners (POP) based on the STRIPS language could only be used for small, simple problems, mainly due to the lack of expressiveness in the STRIPS language representation, and the search process for finding alternative plans was unguided (Russell and Norvig, 1995).

In complex and realistic domains, replanning is not straightforward because there could be different levels or types of failures. In addition, in realistic domains, planning systems need to replan while considering time and resources limitations (Zweben and Fox, 1994).

Practical AI planning systems have been developed that recognize time and resources when replanning. For example, Nonlin (Tate and Whiter, 1984) and SIPE (Wilkens, 1988, Wilkens, 1990) could reason about the allocation of the limited resources to various plan steps and were some of the first planning systems to attempt replanning. ISIS (Fox and Smith, 1984) was developed specifically for scheduling and rescheduling and defined an opportunistic heuristic for rescheduling, which was improved in OPIS (Smith, 1989). GERRY (Zweben et al. 1993) uses an iterative repair method to repair violated constraints. Optimum AIV (Aarup et al. 1994), a successor of O-Plan (Currie and Tate, 1991), developed heuristics that assist planners in developing planning alternatives. SPA[17] (Hanks and Weld, 1992) and CABINS (Sycara and Miyashita, 1993) use a case based reasoning approach for rescheduling.[18]

These planning systems and their heuristics have evolved to meet different requirements of replanning (i.e., different types of failures) and different approaches to improve computational performance, and yet they incorporate similar methodologies for replanning. First, they depend on CPM or PERT networks to represent the schedule to manage the large number of activities in large-scale projects. Secondly, they define additional types of constraints to complement the precedence relationships to describe sequencing logic. Thirdly, the heuristics develop modified plans (i.e., re-plans) using the initial plan (rather than developing an entire new plan from scratch), to minimize the impact to existing resources and parities already committed to the project. Fourthly, the heuristics also avoid the full automation of plan repair, and rely on planners' input during the replanning process. Finally and most importantly, the representations and the heuristics are <u>specialized</u> to meet the replanning demands of a particular industry.

For example, OPIS (Smith, 1989) was used for scheduling in the semiconductor manufacturing industry. It distinguishes constraints into physical constraints, causal constraints, precedence and resource constraints. It uses an opportunistic heuristic for rescheduling referred to as "reactive" scheduling, which handles uncertainties that exist in semiconductor manufacturing.

GERRY (Zweben et al. 1993) was used to repair and refurbish the NASA Space Shuttle fleet. An important requirement was to provide a language for describing the physical "state" of the Space Shuttle. Hence, GERRY distinguishes constraints into temporal and milestone constraints, resource constraints and state constraints. A different repair heuristic exists for each constraint type. GERRY uses an iterative repair approach, in which violated constraints are repaired locally in an iterative repair loop.

Optimum AIV (Aarup et al., 1994) has also been used in the space industry. It distinguishes between precedence, precondition, temporal, resource usage and global activity constraints. Optimum AIV does not define a particular heuristic but assists the user in schedule and plan repair in an interactive way rather than performing the repair itself.

As these examples show, general-purpose AI planning systems in practice have tailored representations (e.g., physical constraints in OPIS versus state constraints in GERRY) and

[17] Systematic Plan Adaptor.
[18] A more detailed discussion of existing replanning tehcniques can be found in Russell and Norvig (1995) and Zweben and Fox (1994).

heuristics (e.g., reactive scheduling in OPIS versus iterative repair in GERRY) to meet domain-specific replanning requirements of a particular industry.

The test case in Section 4.1.1 showed a domain-specific requirement for "replanning" in the construction industry. That is, construction planners needed to re-sequence upstream activities in a CPM network with the goal of expediting a specific downstream activity (i.e., a target activity). The test case also showed that re-sequencing activities correctly and rapidly requires developing a domain-specific heuristic (i.e., identification and re-sequencing process) that in turn requires a tailored representation of construction sequencing rationale.

Thus, I do not use the various replanning algorithms discussed above. Rather, I developed a formal identification and re-sequencing process that utilizes the domain-specific ontology and classification mechanism discussed in the previous Chapters. Developing a domain-specific heuristic not only ensures that the domain-specific problem is solved correctly, but also enables planners using the process to understand the logic behind the heuristic process.

I adopt, however, the common methodologies used in general purpose AI planning systems. For example, the identification and re-sequencing process also depends on a CPM network to represent the temporal relationships between activity sequences. Using the CPM network enables construction planners to manage activities of a large project, and use a technique with which they are familiar. The identification and re-sequencing process also use sequencing rationale in the existing plan from which to develop alternative plans. Using the existing plan makes sense as it re-sequences activities whose resources are already committed and hence available. Finally, the identification and re-sequencing process also requires planners' input and decision throughout the development process. As discussed in the previous section, developing a process that involves the planner was also a critical factor for adoption of CPM-based techniques in the construction industry.

In summary, the sections above described acceleration techniques in CPM-based schedules and replanning techniques in AI planning systems. CPM-based acceleration techniques assume that activity sequences in a CPM network are fixed and that resources can be added to accelerate the schedule duration and hence do not allow planners to re-sequence activities with the goal of expediting a specific target activity. Replanning techniques were not designed in a way to meet the domain-specific requirements needed to re-sequence activities correctly. The following section describes a formal process that allows planners to re-sequence activities while using domain-specific sequencing rationale.

4.3 FORMAL IDENTIFICATION AND RE-SEQUENCING PROCESS

This section describes the formalization and subsequent implementation of the identification and re-sequencing process for developing sequencing alternatives developed in this research.

4.3.1 Formal Process

Based on "Gedanken" experiments similar to those of the test case, I derived generalized processes that utilize the ontology and classification mechanism to assist planners in identifying and developing alternative sequences correctly and rapidly.

Figure 4.4 shows a flow chart for the formal identification and re-sequencing process. The identification process is from steps 1 to 4 in the flowchart. The rest of the steps are required for the re-sequencing process. Table 4.2 describes the formalizations used and functionalities developed for automating several of the steps for both of the processes.

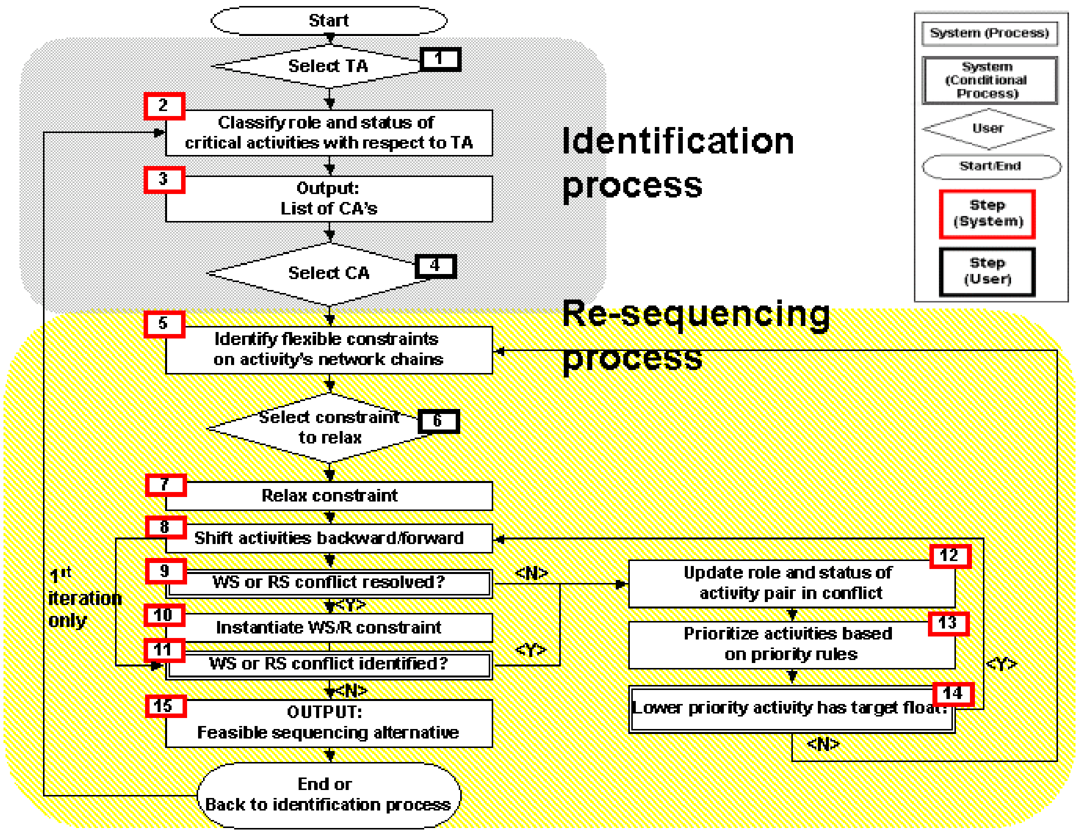

Start
Select TA
1
2
Classify role and status of critical activities with respect to TA
3
Output: List of CA's
Select CA
4
Identification process
Re-sequencing process
5
Identify flexible constraints on activity's network chains
Select constraint to relax
6
7
Relax constraint
8
Shift activities backward/forward
9
WS or RS conflict resolved?
<N>
<Y>
10
Instantiate WS/R constraint
11
WS or RS conflict identified?
<Y>
<N>
15
OUTPUT: Feasible sequencing alternative
End or Back to identification process
1st iteration only
12
Update role and status of activity pair in conflict
13
Prioritize activities based on priority rules
14
Lower priority activity has target float?
<Y>
<N>
System (Process)
System (Conditional Process)
User
Start/End
Step (System)
Step (User)

Figure 4.4: Formal identification and re-sequencing process.

The formal steps required for a system to correctly identify and re-sequence activities using the ontology and classification mechanism are steps 2, 5, 6, 12 and 13. Please view Table 4.2 for detailed descriptions of the steps.

Step (System/User)	Description	Required Functionality or Formalization
Step 1 (user)	• Select TA.	• None
Step 2 (system)	• Classify role and status of critical activities.	• Classification mechanism.
Step 3 (system)	• Output: List of CA's.	• NA.
Step 4 (user)	• Select CA.	• None.
Step 5 (system)	• Identify flexible constraints on activity's network chain.	• Network Chain search algorithm.
Step 6 (user)	• Select constraint to relax	• Ontology: Flexibility of constraints. • Query flexibility (FL) value of constraints in network chain.
Step 7 (system)	• Relax constraint.	• Delete constraint instance.
Step 8 (system)	• Shift activities forward, backward.	• Update ES dates of activities.
Step 9 (system)	• Workspace or resource conflict resolved?	• Identify overlapping activities requiring the same workspace or resource.
Step 10 (system)	• Instantiate workspace or resource constraint.	• If workspace conflict: workspace constraint • If resource conflict: resource constraint
Step 11 (system)	• Workspace or resource conflict identified?	• Identify overlapping activities requiring the same workspace or resource.
Step 12 (system)	• Update role and status of activity pair in conflict.	• Classification mechanism.
Step 13 (system)	• Prioritize activities based on priority rules.	• Formalization of planner's rationale for prioritizing activities.
Step 14 (system)	• Lower priority activity has target float?	• Calculate total float with respect to target activity.
Step 15 (system)	• Output: feasible sequencing alternative.	• None.

Table 4.2: Summary of the formal steps developed for the identification and re-sequencing process.

The first column distinguishes between the steps performed by a user versus a system. The second column describes the steps. The third column describes the required functionalities and formalizations. The formalizations required for a system to correctly identify and re-sequence activities using the ontology or the classification mechanism are shaded.

In the identification process, users select a target activity and the system identifies candidate activities using the classification mechanism. The system outputs a list of candidate activities and users can select one of these candidate activities to delay (Figure 4.4, steps 3 and 4).

In the re-sequencing process, the system first identifies flexible constraints and users select a constraint to relax to delay the candidate activity (Figure 4.4, steps 5, 6 and 7). The system subsequently shifts activities backward (i.e., expedites) (Figure 4.4, step 8), and checks whether a resource or workspace conflict exists. The system identifies a workspace or resource conflict by identifying activities that overlap in time (i.e., are concurrent), and whether these activities require the same type of workspace or resource. Workspaces and resources are described as text-based attributes for each activity of a CPM schedule. For example, workspaces can be denoted as zone A, B or C. The system assumes that two concurrent activities requiring the same workspace are in workspace conflict. If a workspace or resource conflict occurs due to the relaxed constraint (Figure 4.4, step 11), the system first determines *which* activity to delay, and subsequently determines *how* to delay the activity.

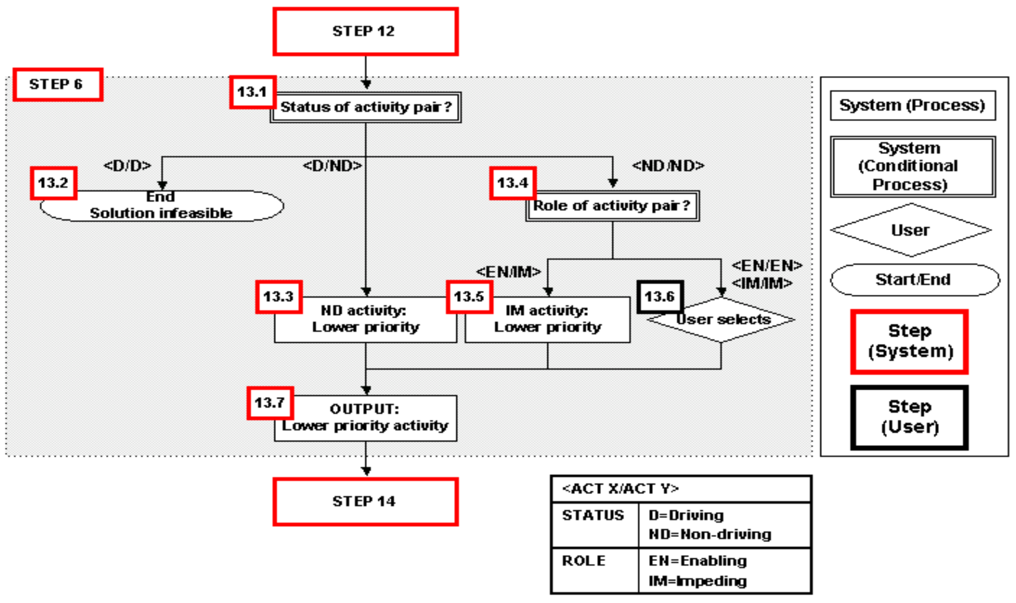
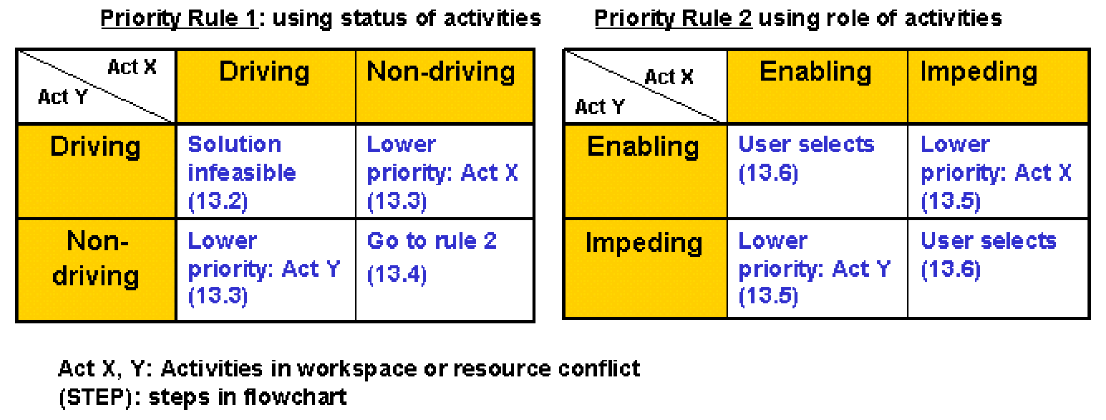


Figure 4.5: Formalized priority rules (flowchart view).

Figure shows Step 13 of the formal process: priority rules using role and status of activities.



<u>Priority Rule 1</u>: using status of activities

Act X / Act Y	Driving	Non-driving
Driving	Solution infeasible (13.2)	Lower priority: Act X (13.3)
Non-driving	Lower priority: Act Y (13.3)	Go to rule 2 (13.4)

<u>Priority Rule 2</u> using role of activities

Act X / Act Y	Enabling	Impeding
Enabling	User selects (13.6)	Lower priority: Act X (13.5)
Impeding	Lower priority: Act Y (13.5)	User selects (13.6)

Act X, Y: Activities in workspace or resource conflict
(STEP): steps in flowchart

<u>Figure 4.6:</u> Formalized priority rules (Step 13).

<u>Priority rule 1</u>

Driving activities have priority for workspace and resources over non-driving activities. If tied, resolve using priority rule 2.

<u>Priority rule 2</u>

Enabling activities have priority for workspace and resources over impeding activities. If tied, user decides.

To determine *which* activity to delay, the system uses pre-defined priority rules that formalize the planner's rationale for prioritizing activities based on the role and status of activities. As shown in Figures 4.5 and 4.6, the order of the steps first prioritizes activities with respect to their status and then prioritizes them with respect to their role. The order reflects the planner's rationale, or more specifically the planner's intent, which is to expedite the target activity. By definition, driving activities are activities that cannot be delayed. Hence, if both activities are driving then no solution exists (Figures 4.5 and 4.6, step 13.2). If one of the activities is non-driving and the other is driving, then logically the non-driving activity is the lower priority activity (i.e., the activity to be delayed) (Figures 4.5 and 4.6, step 13.3). If both activities are non-driving, then the priority cannot be decided based on the activities' status (Figures 4.5 and 4.6, step 13.4). As shown in the test case, planners will try to re-sequence activities so that enabling activities are prioritized over impeding activities. Hence, as shown in Figures 4.5 and 4.6, enabling activities have priority for workspace or resources over impeding activities (step 13.5). If both activities have the same role, the user has to decide which activity has lower priority (Figures 4.5 and 4.6, step 13.6).

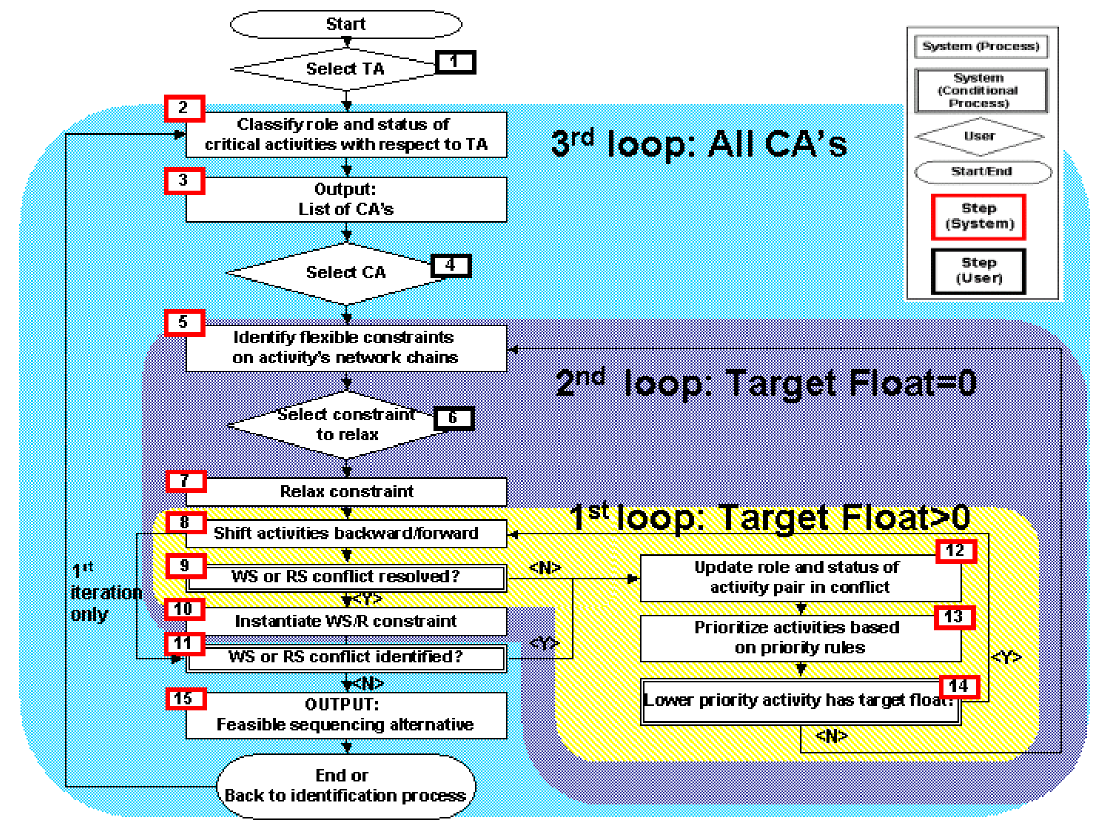

Start
Select TA
1
2
Classify role and status of critical activities with respect to TA
3
Output: List of CA's
Select CA
4
5
Identify flexible constraints on activity's network chains
Select constraint to relax
6
7
Relax constraint
8
Shift activities backward/forward
9
WS or RS conflict resolved?
<Y>
10
Instantiate WS/R constraint
11
WS or RS conflict identified?
<N>
15
OUTPUT: Feasible sequencing alternative
End or Back to identification process
1st iteration only
<N>
<Y>
12
Update role and status of activity pair in conflict
13
Prioritize activities based on priority rules
14
Lower priority activity has target float?
<N>
<Y>
3rd loop: All CA's
2nd loop: Target Float=0
1st loop: Target Float>0
System (Process)
System (Conditional Process)
User
Start/End
Step (System)
Step (User)

Figure 4.7: Individual loops used to identify and re-sequence activities.

Figure shows the three loops used in the identification and re-sequencing process. The first loop uses an activity's target float to delay activities. The second loop identifies constraints to relax to delay activities, and the third loop identifies a new set of candidate activities in a re-sequenced schedule.

Once the system has identified which activity to delay, the system determines *how* that activity needs to be delayed. As shown in the case example, an activity can be delayed by either using the target float or relaxing additional constraints to resolve workspace constraints. As shown in the first loop of Figure 4.7, the system tries to resolve the conflict by using the target float. If the conflict is not resolved, the system uses the second loop in Figure 4.7 to identify constraints that users can select to relax. Once the workspace or resource conflict is resolved, the system instantiates a *workspace* or *resource* constraint accordingly. Using these loops, the system re-sequences the candidate activity until no further workspace or resource conflicts exist. Once a candidate activity has been re-sequenced properly, users have the option of repeating the process to further expedite the target activity. As shown in Figure 4.7, the third loop enables the system to identify a new set of candidate activities based on the modified CPM network. The system will identify new sets of candidate activities until all activities on the critical path are driving. As driving activities are activities that cannot be delayed, no additional activities exist that when delayed expedite the target activity.

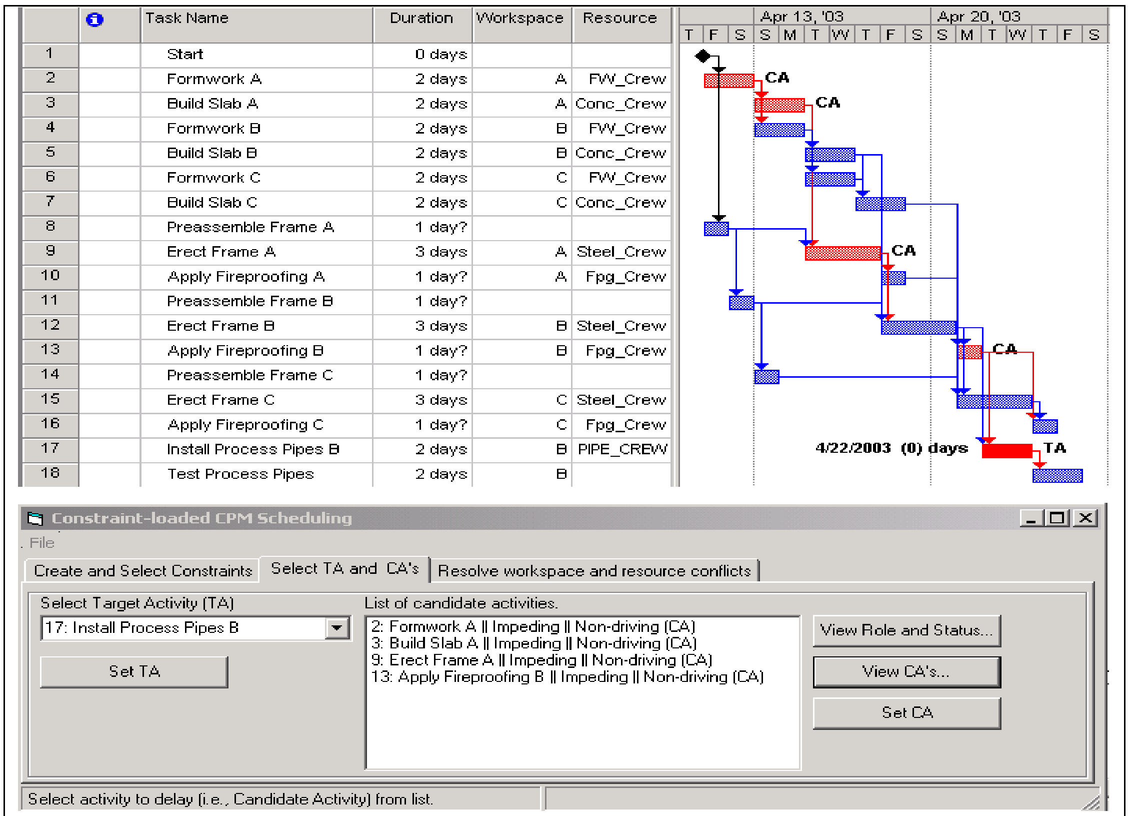
The flowchart in Figure 4.7 shows that while most of the steps have been automated, the system still relies on the planner's judgment to make decisions during the identification and re-sequencing process. In particular, they need to determine which flexible constraints to relax, and which candidate activities to delay. Therefore, the system is not a "black box", but rather an interactive decision support tool.

The following section describes the implementation of this formal identification and re-sequencing process in the prototype tool, CLCPM.

4.3.2 Implementation

As discussed in Chapters 2 and 3, I implemented the ontology and classification mechanism in CLCPM. Correspondingly, I also implemented the formal identification and re-sequencing process in CLCPM.

Figures 4.8 through 4.11 show the formal processes implemented in CLCPM used for the Intel CUB schedule. Figure 4.8 shows that CLCPM identified the candidate activities as the activities Formwork A, Build Slab A, Erect Frame A and Apply Fireproofing B. The activities correspond to the activities identified in the test case example (Figure 4.1 in Section 4.1.1). Figure 4.9 shows a resource conflict identified in CLCPM between the activities Erect Frame A and Erect Frame B (i.e., Steel_Crew). The figure also shows the *damaged by* and *resource* constraints identified in CLCPM that can be relaxed to resolve the resource conflict. The identified constraints are identical to those identified in the test case example (Figure 4.3b in Section 4.1.1). Figure 4.10 shows a sequencing alternative developed in CLCPM. The alternative is identical to the sequencing alternative described in the test case example (Figure 4.3d in Section 4.1.1). Finally, Figures 4.11 and 4.12 show all candidate activities re-sequenced using CLCPM. Consequently, the target activity Install Process Pipes B has been expedited by 4 days. Figure 4.11 shows that all enabling activities have been expedited. In addition, Figure 4.12 shows that all activities on the critical path with respect to the target activity are now driving activities. By definition, candidate activities are activities with zero target float that are impeding and non-driving activities. Hence, no more candidate activities exist.



		Task Name	Duration	Workspace	Resource
1		Start	0 days		
2		Formwork A	2 days	A	FW_Crew
3		Build Slab A	2 days	A	Conc_Crew
4		Formwork B	2 days	B	FW_Crew
5		Build Slab B	2 days	B	Conc_Crew
6		Formwork C	2 days	C	FW_Crew
7		Build Slab C	2 days	C	Conc_Crew
8		Preassemble Frame A	1 day?		
9		Erect Frame A	3 days	A	Steel_Crew
10		Apply Fireproofing A	1 day?	A	Fpg_Crew
11		Preassemble Frame B	1 day?		
12		Erect Frame B	3 days	B	Steel_Crew
13		Apply Fireproofing B	1 day?	B	Fpg_Crew
14		Preassemble Frame C	1 day?		
15		Erect Frame C	3 days	C	Steel_Crew
16		Apply Fireproofing C	1 day?	C	Fpg_Crew
17		Install Process Pipes B	2 days	B	PIPE_CREW
18		Test Process Pipes	2 days	B	

Apr 13, '03
Apr 20, '03
T F S S M T W T F S S M T W T F S
CA
CA
CA
CA
4/22/2003 (0) days
TA
Constraint-loaded CPM Scheduling
File
Create and Select Constraints
Select TA and CA's
Resolve workspace and resource conflicts
Set TA
View Role and Status...
View CA's...
Set CA

Select Target Activity (TA)

17: Install Process Pipes B

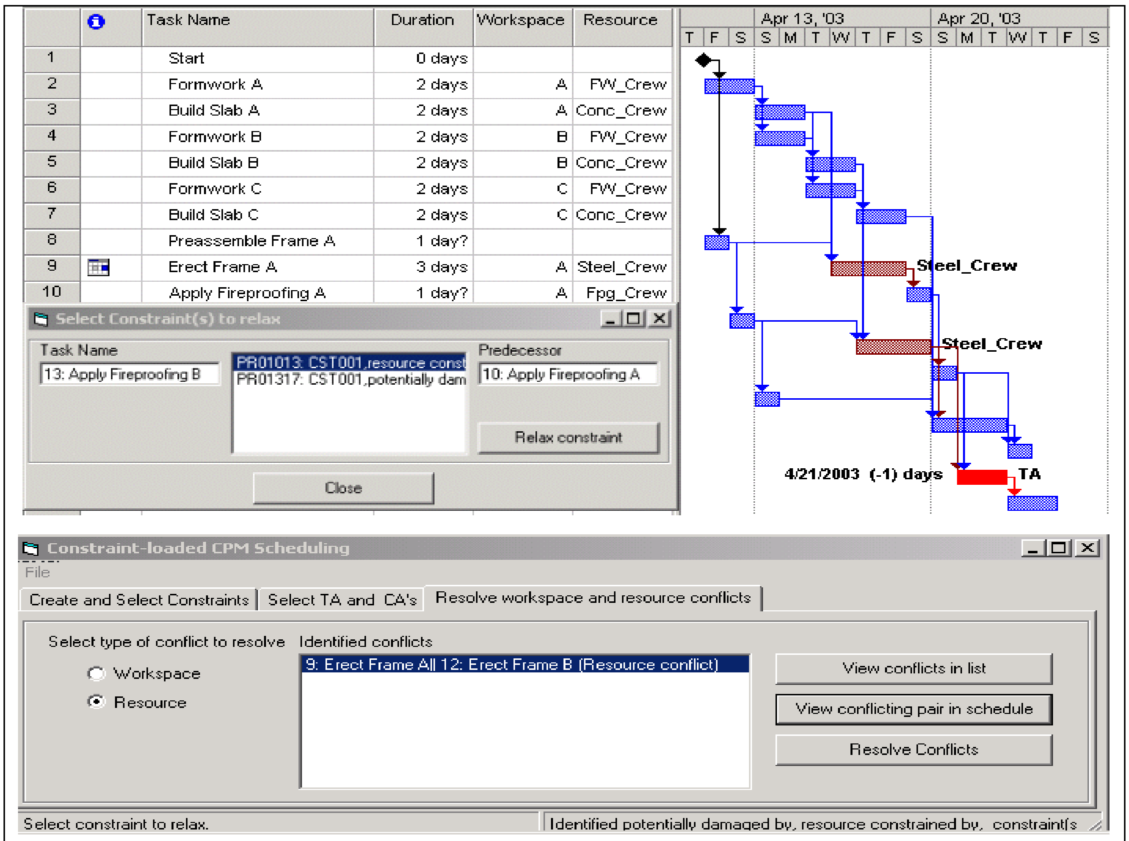
List of candidate activities.

2: Formwork A || Impeding || Non-driving (CA)
3: Build Slab A || Impeding || Non-driving (CA)
9: Erect Frame A || Impeding || Non-driving (CA)
13: Apply Fireproofing B || Impeding || Non-driving (CA)

Select activity to delay (i.e., Candidate Activity) from list.

Figure 4.8: CLCPM user interface to identify candidate activities.

Figure shows the candidate activities (i.e., CA's) identified by CLCPM in the Intel CUB schedule with respect to the target activity (TA).



		Task Name	Duration	Workspace	Resource
1		Start	0 days		
2		Formwork A	2 days	A	FW_Crew
3		Build Slab A	2 days	A	Conc_Crew
4		Formwork B	2 days	B	FW_Crew
5		Build Slab B	2 days	B	Conc_Crew
6		Formwork C	2 days	C	FW_Crew
7		Build Slab C	2 days	C	Conc_Crew
8		Preassemble Frame A	1 day?		
9		Erect Frame A	3 days	A	Steel_Crew
10		Apply Fireproofing A	1 day?	A	Fpg_Crew

Select Constraint(s) to relax

Task Name: 13: Apply Fireproofing B

PR01013: CST001,resource const
PR01317: CST001,potentially dam

Predecessor: 10: Apply Fireproofing A

Constraint-loaded CPM Scheduling

Select type of conflict to resolve: Workspace [✓ Resource]

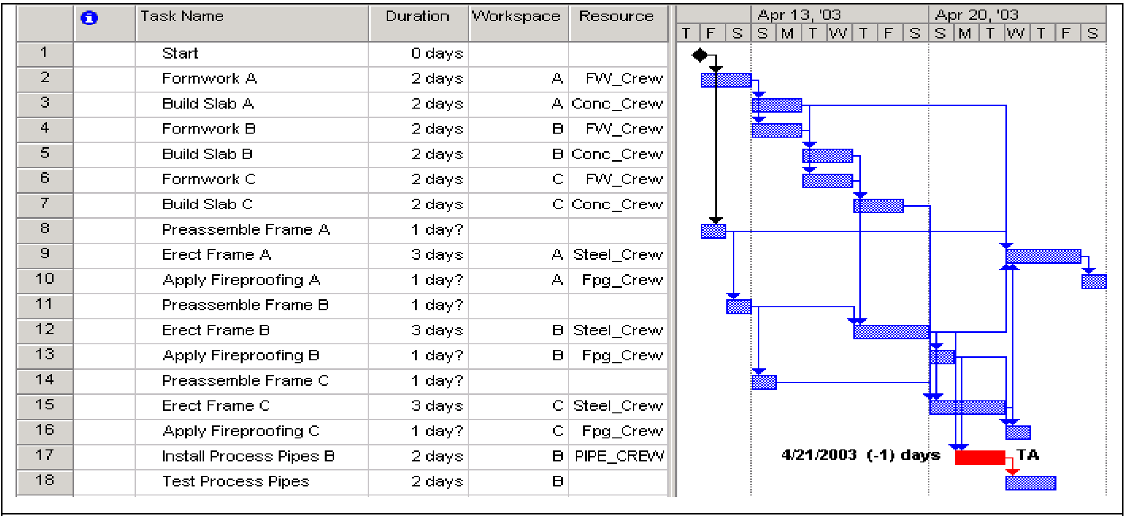
Identified conflicts: 9: Erect Frame A|| 12: Erect Frame B (Resource conflict)

Select constraint to relax.

Identified potentially damaged by, resource constrained by, constraint(s

Figure 4.9: CLCPM user interface to resolve resource and workspace conflicts.

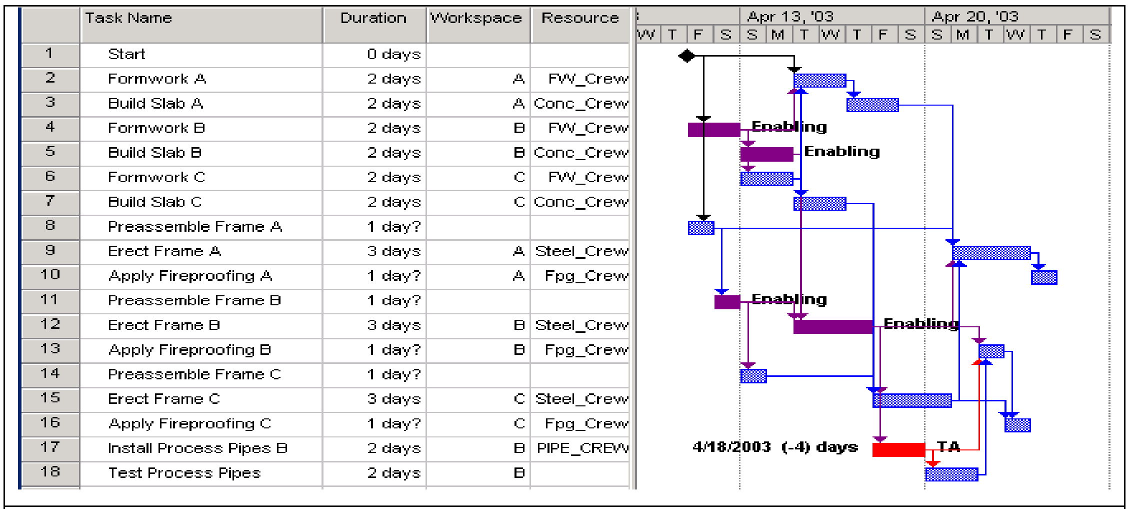
Figure shows a resource conflict between the activities Erect Frame A and Erect Frame B (i.e., Steel_Crew). Figure also shows the constraints identified by CLCPM that can be relaxed to resolve the conflict.



	Task Name	Duration	Workspace	Resource
1	Start	0 days		
2	Formwork A	2 days	A	FW_Crew
3	Build Slab A	2 days	A	Conc_Crew
4	Formwork B	2 days	B	FW_Crew
5	Build Slab B	2 days	B	Conc_Crew
6	Formwork C	2 days	C	FW_Crew
7	Build Slab C	2 days	C	Conc_Crew
8	Preassemble Frame A	1 day?		
9	Erect Frame A	3 days	A	Steel_Crew
10	Apply Fireproofing A	1 day?	A	Fpg_Crew
11	Preassemble Frame B	1 day?		
12	Erect Frame B	3 days	B	Steel_Crew
13	Apply Fireproofing B	1 day?	B	Fpg_Crew
14	Preassemble Frame C	1 day?		
15	Erect Frame C	3 days	C	Steel_Crew
16	Apply Fireproofing C	1 day?	C	Fpg_Crew
17	Install Process Pipes B	2 days	B	PIPE_CREW
18	Test Process Pipes	2 days	B	

Figure 4.10: Example of CLCPM used to re-sequence a single candidate activity.

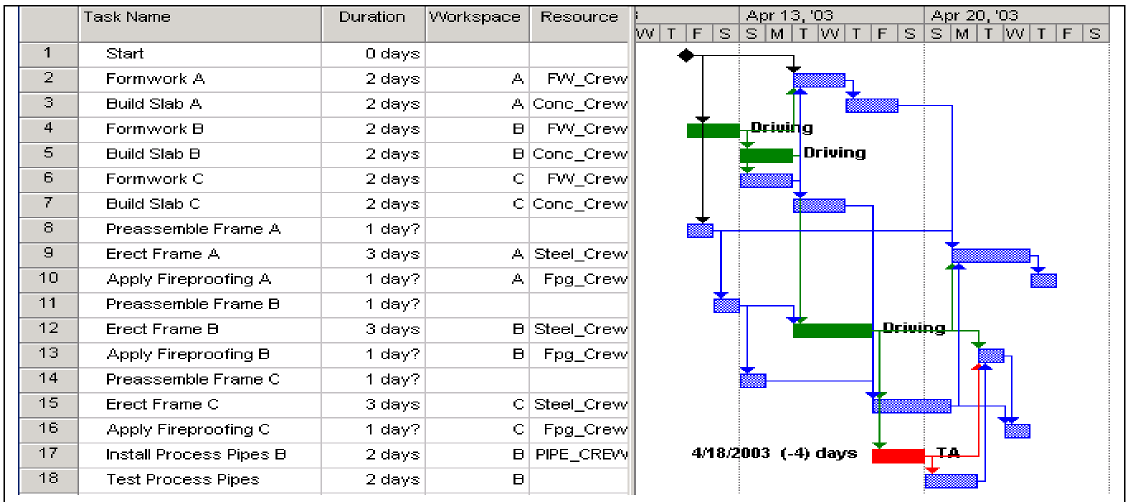
Figure shows the activity Erect Frame A delayed using CLCPM. The target activity Install Process Pipes B has been expedited by 1 day.



	Task Name	Duration	Workspace	Resource
1	Start	0 days		
2	Formwork A	2 days	A	FW_Crew
3	Build Slab A	2 days	A	Conc_Crew
4	Formwork B	2 days	B	FW_Crew
5	Build Slab B	2 days	B	Conc_Crew
6	Formwork C	2 days	C	FW_Crew
7	Build Slab C	2 days	C	Conc_Crew
8	Preassemble Frame A	1 day?		
9	Erect Frame A	3 days	A	Steel_Crew
10	Apply Fireproofing A	1 day?	A	Fpg_Crew
11	Preassemble Frame B	1 day?		
12	Erect Frame B	3 days	B	Steel_Crew
13	Apply Fireproofing B	1 day?	B	Fpg_Crew
14	Preassemble Frame C	1 day?		
15	Erect Frame C	3 days	C	Steel_Crew
16	Apply Fireproofing C	1 day?	C	Fpg_Crew
17	Install Process Pipes B	2 days	B	PIPE_CREW
18	Test Process Pipes	2 days	B	

Figure 4.11: Example of CLCPM used to re-sequence all candidate activities.

Figure shows that all enabling activities with respect to the target activity have been expedited for the Intel CUB schedule.



	Task Name	Duration	Workspace	Resource
1	Start	0 days		
2	Formwork A	2 days	A	FW_Crew
3	Build Slab A	2 days	A	Conc_Crew
4	Formwork B	2 days	B	FW_Crew
5	Build Slab B	2 days	B	Conc_Crew
6	Formwork C	2 days	C	FW_Crew
7	Build Slab C	2 days	C	Conc_Crew
8	Preassemble Frame A	1 day?		
9	Erect Frame A	3 days	A	Steel_Crew
10	Apply Fireproofing A	1 day?	A	Fpg_Crew
11	Preassemble Frame B	1 day?		
12	Erect Frame B	3 days	B	Steel_Crew
13	Apply Fireproofing B	1 day?	B	Fpg_Crew
14	Preassemble Frame C	1 day?		
15	Erect Frame C	3 days	C	Steel_Crew
16	Apply Fireproofing C	1 day?	C	Fpg_Crew
17	Install Process Pipes B	2 days	B	PIPE_CREW
18	Test Process Pipes	2 days	B	

Figure 4.12: Example of CLCPM used to re-sequence all candidate activities.

Figure shows that all activities on the critical path with respect to the target activity is driving for the Intel CUB schedule.

In summary, the sections above presented a formal process for developing sequencing alternatives in a CPM network schedule where the rationale for activity sequences has been described using the ontology defined in Chapter 2. In particular, I introduced how to use the classification mechanism during the identification and re-sequencing process to enable the correct and rapid generation of sequencing alternatives. The next section describes the tests I performed to validate the identification and re-sequencing process.

4.4 VALIDATION

My research formalized an identification and re-sequencing process that utilizes the ontology and classification mechanism to enable planners to develop sequencing alternatives correctly and rapidly. As discussed, the design requirements for the process were:

(1) Formal: The identification process needs to identify candidate activities correctly given a target activity in a CPM schedule. The re-sequencing process needs to enable planners to re-sequence activities correctly. This includes correctly prioritizing activities in conflict and correctly identifying constraints to relax and add during the re-sequencing process.

(2) General: The identification and re-sequencing process needs to enable planners to develop sequencing alternatives for construction schedules that may have different types of specific constraints describing sequencing rationale in these schedules.

To demonstrate that the identification and re-sequencing process meets these criteria, I performed three retrospective case studies and one charrette test (Clayton et al., 1998). For the retrospective case studies, I used three different construction schedules that describe the sequences for different phases of construction. For each of these schedules, I used the formal identification process to identify the enabling activities and the candidate activities for a single target activity and subsequently used the re-sequencing process to expedite the target activity until all candidate activities were exhausted. Subsequently, I confirmed with an experienced project scheduler on whether the candidate activities identified and the sequencing alternatives I developed were indeed correct. Thus, the retrospective cases provide evidence that the process is formal and general, since it demonstrates that the process correctly identifies and re-sequences activities for different construction schedules.

For the charrette test, I used eight graduate students to identify and re-sequence activities for two construction schedules, where one half of the students used a "constraint-loaded" schedule using CLCPM, and the other half used a conventional scheduling tool, Microsoft Project (MSP). Then I compared how correctly and quickly the two groups could identify and re-sequence activities given a single target activity. Hence, the charrette test provides evidence that the process is formal, since it demonstrates that the process enables users to develop sequencing alternatives more correctly and rapidly than a conventional CPM scheduling tool.

The following tests describe the test procedure and results in detail.

4.4.1 Retrospective Validation Studies

I performed the retrospective case studies to show that the identification and re-sequencing process can correctly develop sequencing alternatives for different types of construction projects. I demonstrated that the processes are formal by evaluating the level of correctness of the activities identified and sequencing alternatives developed using CLCPM for three different construction projects.

Table 4.3 describes the number of activities, precedence relationships, target activity and the phase of construction and types of work for the three projects[19].

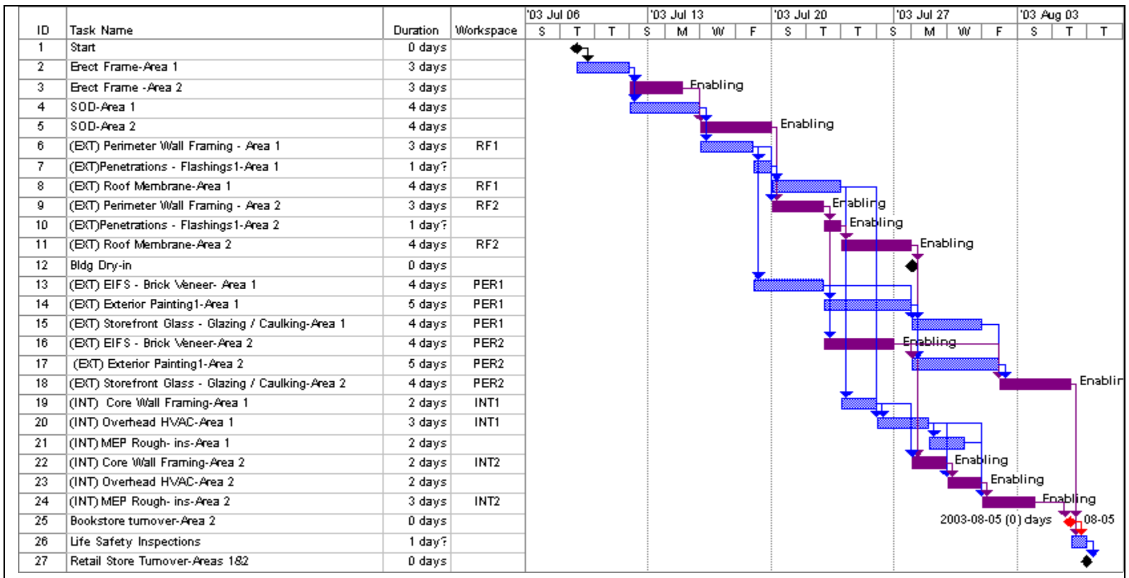
[19] Please refer to Appendix B.1 for a detailed description of the three projects.

Project	Number of Activities	Number of Precedence Relationships	Target Activity (TA)	Phase of Project
Intel CUB	18	25	Install Process Pipes B (ID: 17)	Concrete and structural frame.
McWhinney	27	37	HVAC Balance (ID: 25)	Structural roof, interior MEP and finishes.
Bay Street	27	36	Book Store Turnover (ID: 25)	Concrete and structural frame, exterior closure, interior MEP and finishes.

Table 4.3: Overview of the three retrospective cases.

I evaluated the level of correctness by measuring the number of activities correctly identified, and the number of re-sequencing alternatives correctly developed using CLCPM. In addition, I used CLCPM to expedite the target activity until all candidate activities identified were exhausted. Subsequently, I confirmed with an experienced project scheduler whether the candidate activities I identified and the sequencing alternatives I developed were correct.

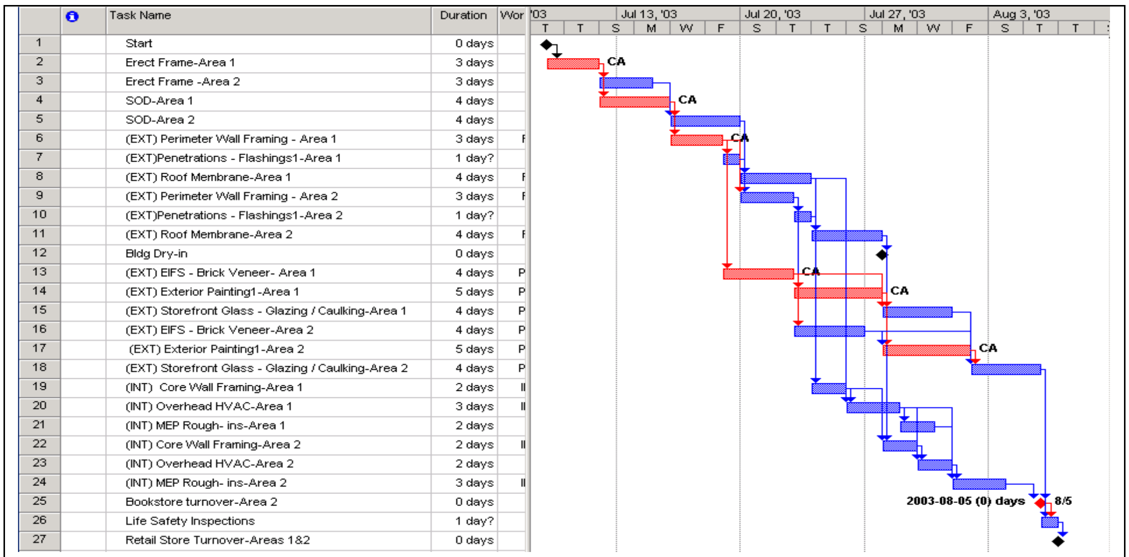
I use the Bay Street project schedule to illustrate the test method and results. Figures 4.13 and 4.14 show respectively the activities identified by CLCPM to be enabling types of activities and the candidate activities with respect to the target activity Bookstore Turnover (activity ID: 23). The project scheduler concurred that the activities classified by CLCPM were correct. He also concurred that he did not find any other enabling types of activities or candidate activities.



ID	Task Name	Duration	Workspace
1	Start	0 days	
2	Erect Frame-Area 1	3 days	
3	Erect Frame -Area 2	3 days	
4	SOD-Area 1	4 days	
5	SOD-Area 2	4 days	
6	(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7	(EXT)Penetrations - Flashings1-Area 1	1 day?	
8	(EXT) Roof Membrane-Area 1	4 days	RF1
9	(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10	(EXT)Penetrations - Flashings1-Area 2	1 day?	
11	(EXT) Roof Membrane-Area 2	4 days	RF2
12	Bldg Dry-in	0 days	
13	(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14	(EXT) Exterior Painting1-Area 1	5 days	PER1
15	(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16	(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17	(EXT) Exterior Painting1-Area 2	5 days	PER2
18	(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19	(INT) Core Wall Framing-Area 1	2 days	INT1
20	(INT) Overhead HVAC-Area 1	3 days	INT1
21	(INT) MEP Rough- ins-Area 1	2 days	
22	(INT) Core Wall Framing-Area 2	2 days	INT2
23	(INT) Overhead HVAC-Area 2	2 days	
24	(INT) MEP Rough- ins-Area 2	3 days	INT2
25	Bookstore turnover-Area 2	0 days	
26	Life Safety Inspections	1 day?	
27	Retail Store Turnover-Areas 1&2	0 days	

Figure 4.13: CLCPM used to identify the role of activities for retrospective case.

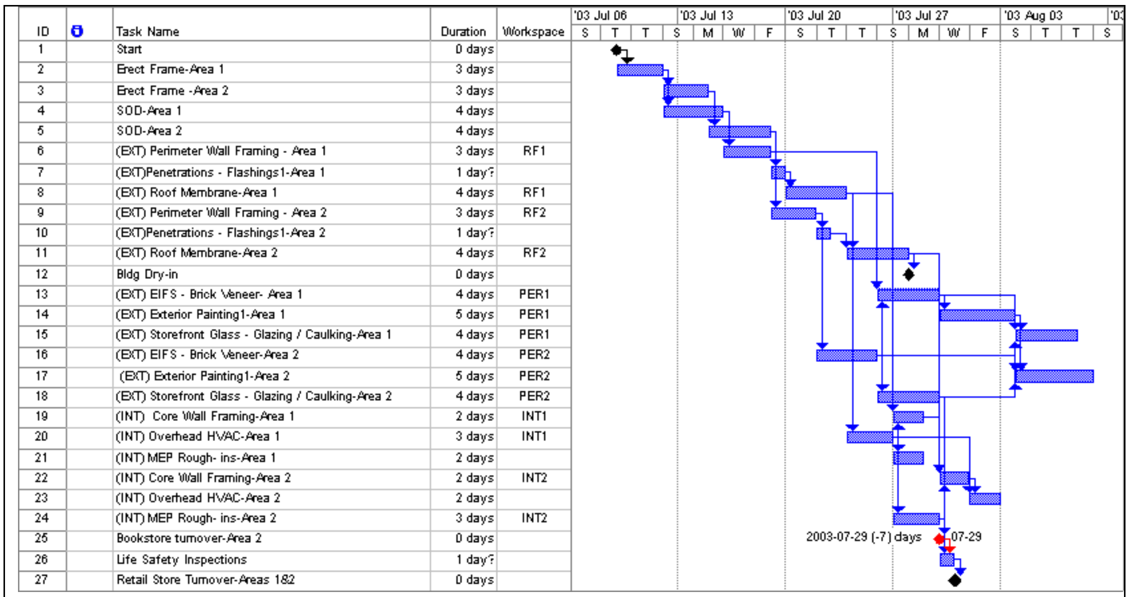
Figure shows the enabling activities identified by CLCPM shown in purple for the Bay street project schedule with respect to the target activity Bookstore Turnover-Area 2 (activity ID: 23).



	Task Name	Duration	Wor
1	Start	0 days	
2	Erect Frame-Area 1	3 days	
3	Erect Frame -Area 2	3 days	
4	SOD-Area 1	4 days	
5	SOD-Area 2	4 days	
6	(EXT) Perimeter Wall Framing - Area 1	3 days	F
7	(EXT)Penetrations - Flashings1-Area 1	1 day?	
8	(EXT) Roof Membrane-Area 1	4 days	F
9	(EXT) Perimeter Wall Framing - Area 2	3 days	F
10	(EXT)Penetrations - Flashings1-Area 2	1 day?	
11	(EXT) Roof Membrane-Area 2	4 days	F
12	Bldg Dry-in	0 days	
13	(EXT) EIFS - Brick Veneer- Area 1	4 days	P
14	(EXT) Exterior Painting1-Area 1	5 days	P
15	(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	P
16	(EXT) EIFS - Brick Veneer-Area 2	4 days	P
17	(EXT) Exterior Painting1-Area 2	5 days	P
18	(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	P
19	(INT) Core Wall Framing-Area 1	2 days	I
20	(INT) Overhead HVAC-Area 1	3 days	I
21	(INT) MEP Rough- ins-Area 1	2 days	
22	(INT) Core Wall Framing-Area 2	2 days	I
23	(INT) Overhead HVAC-Area 2	2 days	
24	(INT) MEP Rough- ins-Area 2	3 days	I
25	Bookstore turnover-Area 2	0 days	
26	Life Safety Inspections	1 day?	
27	Retail Store Turnover-Areas 1&2	0 days	

Figure 4.14: CLCPM used to identify candidate activities for retrospective case.

Figure shows the candidate activities identified by CLCPM shown in red in the Bay Street project schedule with respect to the target activity Bookstore Turnover-Area 2 (activity ID: 23).



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure 4.15: CLCPM used to re-sequence activities for retrospective case.

Figure shows the final sequencing alternative generated in CLCPM. All candidate activities have been re-sequenced. Consequently, the activity Bookstore Turnover-Area 2 (activity ID: 23) has been expedited by 7 days.

Figure 4.15 shows the sequencing alternative where all the candidate activities have been delayed using CLCPM. Consequently, I created six different sequencing alternatives[20]. The project scheduler concurred that each alternative was logical.

Tables 4.4 and 4.5 show the results of the three retrospective cases with respect to the identification and re-sequencing process, respectively. For each project, the project scheduler concurred with the activities identified by CLCPM and the sequencing alternatives I generated using CLCPM.

Project	TA initial date	Number of Enabling activities	Confirmed	Number of CAs	Confirmed
Intel CUB	4/22	4	4	4	4
McWhinney	7/21	11	11	3	3
Bay Street	8/05	10	10	6	6

Table 4.4: Validation results of the identification process for the three retrospective cases.[21]

Project	TA initial date	TA final expedited date	Number of sequencing alternatives	Confirmed
Intel CUB	4/22	4/18(-4 days)	3	3
McWhinney	7/21	7/15(-6 days)	2	2
Bay Street	8/05	7/29(-7 days)	6	6

Table 4.5: Validation results of the re-sequencing process for the three retrospective cases.[22]

The results demonstrate that for the three project schedules used, the formal identification and re-sequencing processes was 100% correct in identifying and re-sequencing activities. The tests demonstrate the generality of the formal process for a wide range of construction activities. The test also demonstrates the power of the formal process in generating multiple sequencing alternatives. However, the tests do not show whether the formal process is scalable, as the three project schedules used were limited to about 24 activities. As shown in Table 4.4, there were only 3 to 6 candidate activities in the three schedules. In a project schedule of 100 or more activities, it is foreseeable that there will many more candidate activities. In such cases, planners may need a

[20] Please refer to Appendix B.4.3 to view screenshots of the sequencing alternatives for the Bay Street project schedule.
[21] Please refer to Appendix B.3 to view screenshots of the activities identified for the three retrospective cases.
[22] Please refer to Appendix B.4 to view screenshots of the sequencing alternatives developed for the three retrospective cases.

way to rank candidate activities they wish to focus on and delay first. A more detailed discussion of this issue is presented in Chapter 5.

The following section describes the charrette test.

4.4.2 Charrette Test

As discussed, the goal of the charrette test was to demonstrate that a formal identification and re-sequencing process enables planners to develop sequencing alternatives more correctly and rapidly than a conventional CPM scheduling tool. Hence, I compared how correctly eight graduate students in the Construction Engineering and Management program at Stanford University could identify and re-sequence activities to develop a sequencing alternative for two project schedules, i.e., Intel CUB schedule and the Bay Street project schedule. The first four students individually used CLCPM to identify activities and develop a sequencing alternative for the Intel CUB schedule. The latter four students individually used Microsoft Project (MSP) to identify the same set of activities and develop the same sequencing alternative. Then, for the Bay Street project, the first four students used MSP, while the latter four students used CLCPM to once again identify activities and develop the same sequencing alternative. In addition to the schedule, each student was provided with a 3D model and a brief description of the overall sequence of the activities for the two projects[23]. I also limited the time for identifying activities (10 minutes) and for developing a sequencing alternative (15 minutes) for both projects. All students used CLCPM first to develop the sequencing alternative for each project. The procedure benefits the CPM scheduling tool, as students were more aware of the tasks to be performed on the second try.

For the identification process, I evaluated the *level of correctness* of the students' solutions by measuring the number of correct activities that students identified as enabling and activities that could be delayed (i.e., candidate activities). Table 4.6 shows the number of enabling activities and the number of candidate activities for the two projects.

Project	Number of Enabling activities	Number of CA's	Total
Intel CUB	4	4	8
Bay Street	7	4	11

Table 4.6: Metrics used to measure the level of correctness for the identification process.

[23] Please refer to Appendix C to view the questionnaires for the charrette test I used to instruct students to identify and re-sequence activities for the Intel CUB schedule and Bay Street project schedule.

Figure 4.16 shows the number of activities correctly identified and the time taken for both projects. As shown in Figure 4.16a, students using CLCPM identified on average 93% (i.e., (7.25/8+10.25/11)/2)) of the enabling and candidate activities correctly for the two projects, compared to 50% (i.e., (2.75/8+7.25/11)/2) for students using MSP. The students using CLCPM also took on average about half the time to identify the activities than students using MSP (Figure 4.16b). The reason behind students using CLCPM not being able to identify *all* activities correctly may be attributed to the specified time limit for identifying the activities. But more likely, it may be attributed to the difficulty in understanding the new terminology (e.g., enabling, candidate, etc.) and the crudeness of the CLCPM interface.

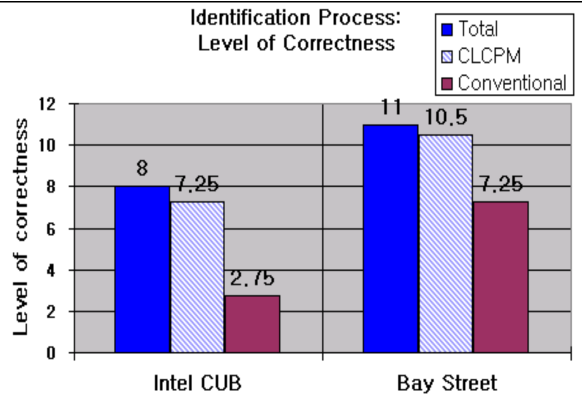


Figure 4.16a: Level of correctness for the two projects.

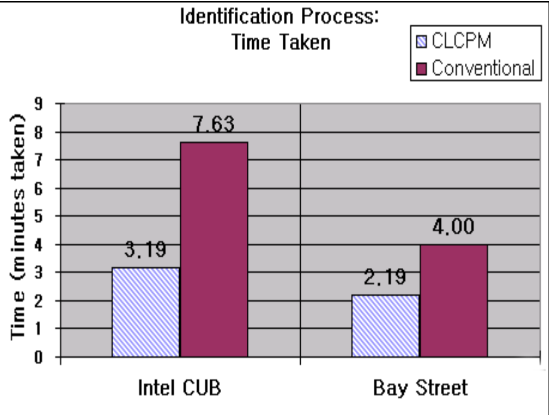


Figure 4.16b: Time taken to classify activities for the two projects.

Figure 4.16: Charrette test results for the identification process.

Figure 4.16a: The first bar describes the total number of enabling and candidate activities that need to be identified correctly for the Intel CUB schedule. The second and third bars are the average numbers of correct activities identified by four students for the Intel CUB schedule using CLCPM and a conventional scheduling tool, respectively. The second set of bars describes the level of correctness for the Bay Street project schedule.

Figure 4.16b: The first and second bars describe the average time taken by the four students to re-sequence an activity in the Intel CUB schedule using CLCPM and a conventional scheduling tool, respectively. The second set of bars describes the average time taken for four students for the Bay Street project schedule.

For the re-sequencing process, I evaluated the *level of correctness* of the students' solutions by measuring how correctly students could generate a sequencing alternative. Specifically, I assigned a single point for each of the following criteria:

i) **TA date:** The sequencing alternative needs to match the expedited date. As shown in Table 4.7, I gave a single point for matching the expected target date. I assigned a single point for the correct TA date.

ii) **Number of conflicts to resolve:** For each sequencing alternative, there were a finite number of workspace or resource conflicts students needed to resolve. I assigned a single point for each conflict resolved correctly.

iii) **Number of constraints to add and relax:** For each sequencing alternative, there were a finite number of constraints that needed to be relaxed and added to re-sequence the candidate activity correctly. I assigned a single point for each constraint added or relaxed correctly.

Table 4.7 shows the corresponding points for the Intel CUB and Bay Street project schedule.

Project	Expedited TA date	CA to delay	Number of conflicts	Number of constraints to relax	Number of constraints to add	Total
Intel CUB	4/22 (1)	Erect Frame A (ID: 9)	2	2	2	7
Bay Street	8/04 (1)	Exterior Painting-Area 1 (ID: 14)	1	2	1	5

Table 4.7: Metrics used to measure the level of correctness for the re-sequencing process.

Figure 4.17 shows the average scores and time taken for the students for both projects. As shown in Figure 4.17a, students using CLCPM scored on average 92% (i.e., (6.25/7+4.75/5)/2) compared to 52% (i.e., (3.75/7+ 2.5/5)/2) in developing the sequencing alternative for the two project schedules. More prominent is the disparity in the time taken for developing the sequencing alternatives (Figure 4.17b). Students using MSP took on average twice as long in developing sequencing alternatives of poorer quality. The reason behind students using CLCPM not being able to re-sequence activities perfectly may again be attributed to the specified time limit and crudeness of the CLCPM interface.

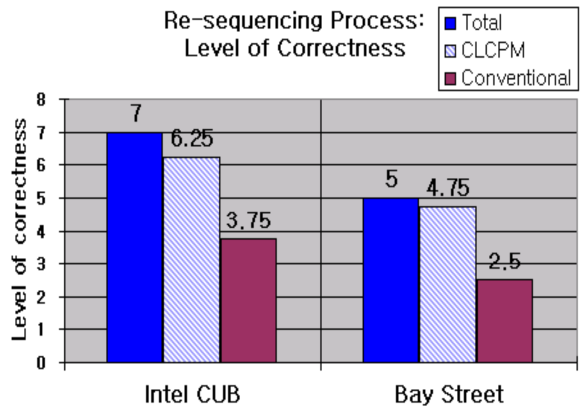
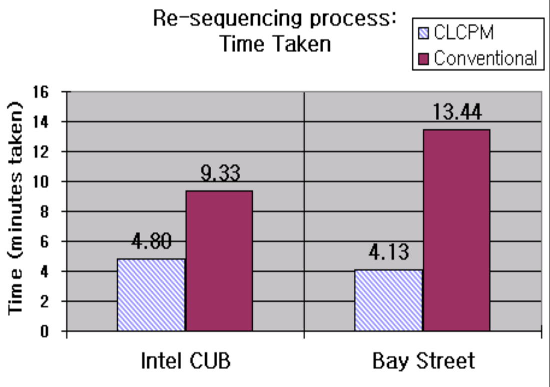
	

Figure 4.17a: Level of correctness for the two projects.	**Figure 4.17b:** Time taken to re-sequence candidate activities for the two projects.

Figure 4.17: Charrette test results for the re-sequencing process.

Figure 4.17a: The first bar describes the total score possible for re-sequencing a candidate activity correctly for the Intel CUB schedule. The second and third bars are the average scores of four students for the Intel CUB schedule using CLCPM and MSP, respectively. The second set of bars describes the level of completeness for the Bay Street project schedule.

Figure 4.17b: The first and second bars describe the average time taken by the four students to re-sequence an activity in the Intel CUB schedule using CLCPM and MSP, respectively. The second set of bars describes the average time taken for four students for the Bay Street project schedule.

In summary, the results of the charrette test demonstrates that the formal identification and re-sequencing process can help planners in identifying and developing sequencing activities more correctly and rapidly than a conventional CPM scheduling tool. The test is limited in the small number of the students used, and hence may require further validation with a larger sample size and the participation of industry planners. The test also indicated the requirement for a more intuitive implementation, to enable users to interact directly with the Gantt chart, rather than the current CLCPM interface.

In this section, the retrospective validation studies and the charrette test demonstrated the power and generality of the formal processes for developing sequencing alternatives. The retrospective validation studies demonstrated that using CLCPM, planners can create multiple sequencing scenarios correctly for different project schedules that have different types of constraint rationale. The charrette test demonstrated that CLCPM helps users to step through the formal process correctly and rapidly, while still enabling them to make decisions throughout the process.

4.5 CONCLUSIONS

This Chapter presented a formal identification and re-sequencing process that formalizes the necessary steps required to identify and re-sequence activities with the goal of expediting particular activities in a CPM schedule. The process formalizes the planner's rationale for determining *which* activities to delay and *how* the selected activity needs to be delayed. The process achieves these goals by utilizing the classification mechanism of Chapter 3 to classify and prioritize activities, and by utilizing the ontology of Chapter 2 to determine which constraints maybe relaxed.

The formal process implemented in CLCPM enables planners to correctly and rapidly develop multiple sequencing alternatives for construction schedules describing different phases and types of work. In addition, I designed the formal process as a decision support tool, as to an optimization technique. Hence, the formal process implemented in CLCPM also enables planners to follow the logical steps involved in identifying and re-sequencing activities, while providing the freedom for planners to make informed decisions during the development of sequencing alternatives. In addition, the formal process allows planners to develop sequencing alternatives that they deem practically feasible in contrast to an optimum solution that may rely heavily on impractical assumptions.

The formal process is limited in many ways. I did not perform validation tests for large-scale project schedules that may consist of 1,000 or more activities. I identified that a foreseeable problem for large-scale project schedules was the large number of candidate activities that may be identified in CLCPM and hence need to be ranked or prioritized. One approach to resolve the problem may be to rank candidate activities based on the number of successors a candidate activity impedes. Alternatively, the candidate activities may be ranked with respect to the cost of delaying a candidate activity. The approach in turn will require formalizing the cost of relaxing individual constraints. The charrette test demonstrated the need for a more intuitive graphical interface that facilitates users' understanding of the role and status of activities in a Gantt chart. One approach may be to visually show the components related to enabling or impeding activities in a 4D model. The 4D model may also be useful in visually showing the sequencing alternatives developed.

Other limitations are due to the scope defined for the ontology. The process cannot identify and re-sequence schedules where design and procurement activities exist together with construction-specific activities, as the ontology has only been formalized to represent construction-specific sequencing rationale. The formal process has been formalized for CPM schedules where rationale is described using FS precedence relationships only. Finally, the formal process also does not select or prioritize specific constraints in relation to costs that may be incurred when constraints are relaxed. A more detailed discussion of these limitations and possible extensions is in Chapter 5.

I speculate that the formal process can be used most effectively in construction trade coordination meetings, where individual trades of a project can use the identification process to understand their role in the overall scheme of the project progress, and subsequently use the re-sequencing process to experiment and evaluate various sequencing alternatives to make correct sequencing decisions.

CHAPTER 5. CONCLUSIONS

This Ph.D. research developed and implemented three formalizations: (1) an ontology to represent construction sequencing rationale in CPM schedules, (2) a classification mechanism to automatically classify the role and status of activities using the ontology, and (3) a formal identification and re-sequencing process that uses the ontology and classification mechanism to assist planners in developing sequencing alternatives.

The ontology enables planners to describe correctly and consistently the different types of sequencing rationale in construction schedules. The classification mechanism correctly and rapidly infers the role and status of activities in constraint-loaded schedules. The formal process assists planners during the identification and re-sequencing process, enabling them to develop sequencing alternatives for different types and phases of construction schedules in a correct and timely manner.

I provided evidence for the power and generality of the three formalizations by performing three retrospective case studies and one charrette test. The three retrospective cases demonstrate that the formalizations apply for a wide range of construction activities and enable the <u>correct</u> development of sequencing alternatives. The charrette test demonstrates that the use of the formalizations results in developing sequencing alternatives more <u>correctly and rapidly</u> than the use of conventional CPM scheduling tools.

Consequently, I claim the formalizations developed contribute to the current knowledge of construction sequencing rationale representation and construction planning and scheduling.

Chapters 2, 3 and 4 discussed each of the three formalizations, and their validation in detail. Sections 5.1.1 through 5.1.3 of this Chapter summarize these contributions and their validations, and state the limitations of each contribution.

This Ph.D. research presents a formal approach for developing sequencing alternatives for construction projects. The validation studies suggest that the research provides a practically feasible approach for multiple project members to participate and develop sequencing alternatives. Section 5.2 of this Chapter discusses the practical implications of this research.

To make solid contributions and validate those contributions within a certain period of time, I limited the focus of my study in many ways. For example, I focused primarily on sequencing rationale in construction schedules, and did not include rationale in design and procurement schedules. I also did not formalize costs associated with relaxing constraints. The validation studies also suggest the need for developing methods to manage the volume of constraints that need to be input manually, and for methods to handle the large number of activities in large-scale projects. Section 5.3 discusses future extensions to this research that address these limitations identified and other possible applications of the formalizations.

5.1 SUMMARY OF CONTRIBUTIONS AND VALIDATIONS

The following sections summarize the three contributions I claim for this research, the validations that provide evidence for each contribution, and the limitations of each contribution in its current state.

5.1.1 Contribution 1: Ontology for Construction Sequencing Rationale

The first contribution I claim is for an ontology developed for formally representing construction sequencing rationale. The ontology models a classification schema that is disjoint enough to correctly distinguish the unique role and flexibility of a specific constraint, and also sufficiently exhaustive to define as many different types of specific constraints as necessary. The ontology also enables planners to define specific constraints in a project-independent way, that they can reuse and specialize to describe sequencing rationale correctly and consistently.

As discussed in Section 2.4 of Chapter 2, I tested the ontology using three retrospective cases and one charrette test. The three retrospective cases demonstrate that the ontology correctly represents the unique role and flexibility of different types of sequencing rationale in construction schedules. The charrette test shows that planners and multiple participants can interpret sequencing rationale more correctly and consistently using a "constraint-loaded" schedule compared to a conventional CPM schedule.

Based on the evidence provided from the validation tests, I claim that the ontology contributes to the current state of knowledge in the area of construction sequencing rationale classification and representation.

The ontology presents an alternative approach for classifying sequencing rationale from existing classification approaches, as it classifies rationale with respect to its role. As discussed in Section 2.2 of Chapter 2, existing classification schemas identified in previous research have mainly classified sequencing rationale with respect to their origin (e.g., Echeverry et al., 1991).

The ontology is also unique from existing representation approaches as the ontology enables planners to describe both the specific constraint and the individual classifications of specific constraints in a computer-interpretable form. As discussed in Section 2.2 of Chapter 2, existing representations of sequencing rationale in construction AI planning systems have been used to develop plans (i.e., sequence activities), and hence do not enable the automated classification of activities. In contrast, the ontology allows a computer-system to use the specific constraints to infer the behavior (i.e., role and status) of activities.

The following section describes the limitations of the ontology.

5.1.1.1 Limitations of the Ontology

The ontology is limited in several ways. Some of the limitations are due to the scope I defined for this research. I limited the scope of the ontology to primarily represent sequencing rationale of construction schedules, and hence did not consider sequencing rationale in design and procurement schedules. I also designed the ontology to be used for FS relationships in a CPM schedule. I also did not explicitly associate the specific constraints with the individual costs that may be incurred for relaxing constraints. A practical limitation is that planners currently need to manually input the specific constraints to describe rationale for activity sequences, which may become impractical for large construction schedules. The following section describes these limitations and possible extensions of the ontology to address these limitations.

(1) Sequencing rationale limited to activities in construction schedules

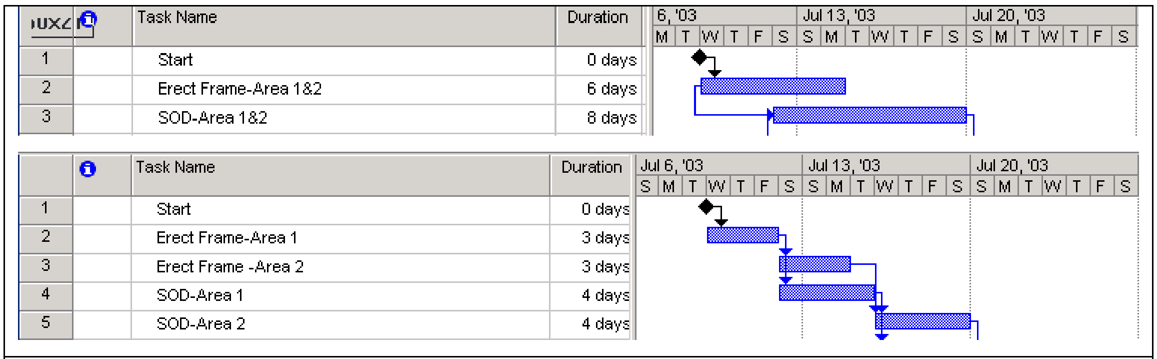
I limited the scope of the ontology to represent sequencing rationale between activities in construction schedules, (i.e., construction-specific activities). The sequencing rationale I identified in actual construction schedules and the literature are predominantly based on the physical dependency relationships between components or physical interactions between trades (e.g., *physically* supports, *physically* protects, *physically* requires resource or workspace etc.). In practice, design and procurement activities constrain the construction progress as well (Paulson 1976). The rationale for design and procurement activity sequences do not necessarily constrain progress physically, but constrains activity sequences "technically" (e.g., *technically* requires shop drawing approval, *technically* requires material delivery). Currently, the ontology has not been formalized to represent sequencing rationale in schedules where design and procurement activities may exist together with construction-specific activities. However, I speculate that the ontology in its current form can to a certain extent be used to correctly classify the rationale for design and procurement activity sequences. I base my speculation on the observation that basically the functional properties of the rationale for design and procurement constraints are similar to those of the constraint rationale for construction activities; i.e., that they either can or cannot be relaxed (i.e., flexible or inflexible) to re-sequence activities, and that they either enable or impede the start of following activities.

Nevertheless, planners may still wish to explicitly distinguish between sequencing rationale based on physical relationships as to technical or contractual relationships. In particular, planners may wish to describe the role for these types of sequencing rationale in a more disjoint way than what the ontology currently allows. For example, they may want to explicitly distinguish between *physically* enabling constraints versus *technically* enabling constraints. A more detailed discussion of how the ontology may be extended to incorporate sequences based on design and procurement activities is introduced in Section 5.3.

(2) Precedence relationships limited to Finish to Start relationships

I formalized the ontology and correspondingly the project-independent constraints so that they could primarily be used to describe the rationale for FS precedence relationships. I scoped out the other precedence relationships based on the assumption that activities whose sequences are represented as SS, SF or FF precedence relationships in a CPM schedule can be described using FS relationships by breaking down these activities into further detail. I further assumed that the elaboration of these activities into more detailed activities would not compromise the initial rationale for the activity sequences, and also not alter the duration of the entire schedule.

For example, Figure 5.1 shows two activities of the Bay Street project schedule where the structural frame and concrete work for the two areas are represented using a start-to-start precedence relationship. I broke down these activities into two more activities to represent the sequence using finish-to-start precedence relationships. The modifications do not change the duration of the original activities. More importantly, they do not alter the rationale for the sequence of activities.



	Task Name	Duration
1	Start	0 days
2	Erect Frame-Area 1&2	6 days
3	SOD-Area 1&2	8 days

	Task Name	Duration
1	Start	0 days
2	Erect Frame-Area 1	3 days
3	Erect Frame -Area 2	3 days
4	SOD-Area 1	4 days
5	SOD-Area 2	4 days

Figure 5.1: Example of activities in the Bay Street Project elaborated to describe sequences as FS relationships.

Nevertheless, I did not address in detail whether the ontology would have to be designed differently for SS, SF or FF precedence relationships. Additional investigations and validations are required to determine whether the ontology is sufficient in describing the rationale for these precedence relationships.

(3) Costs of relaxing constraints not formalized

In practice, cost is a major factor in determining which constraints to relax. Planners will avoid relaxing a constraint that incurs high costs. For this research, I did not formalize the flexibility of constraints with respect to precise, quantitative costs. I enabled planners to distinguish the level of flexibility using a simple qualitative scale (i.e., low, medium and high), to represent the relative physical level of difficulty that may be involved in shifting resources or project components. Hence, the flexibility of the constraint I defined does enable planners to describe how "difficult" relaxing a constraint may be, which may implicitly include the associated cost of relaxing a constraint. However, the cost is not explicitly formalized. I describe how "shadow prices" may be used to incorporate costs in Section 5.3.

(4) Manual input of constraints

Currently, the formal ontology requires planners to describe the rationale for constraints manually. Although, I formalized the ontology to enable the reuse and customization of project-independent types of constraints, the manual input of these constraints can still become tedious for large project schedules.

A practical approach would be to use CLCPM to focus on particular phases of the project where the need arises to re-sequence activities. Planners could also use the 3D model as a reference when selecting the project-independent types of constraints to describe sequencing rationale more effectively.

More formal approaches in this area include incorporating "subnetworks" (i.e., grouping activities) and automatically inferring constraints from construction product models and specification tools such as Regnet (Kerrigan and Law, 2003). Section 5.3 describes in detail the possible research directions in this area.

This section described the first contribution I claim and its implications. I also discussed the limitations of the ontology. The following section describes the second contribution.

5.1.2 Contribution 2: Classification Mechanism to Automatically Classify the Role and Status of Activities

The second contribution I claim is for a formal classification mechanism that automatically infers the role and status of activities given a CPM network schedule where the rationale for activity sequences has been described using the formal ontology. The classification mechanism consists of a network chain search algorithm and inferences rules. The network chain search algorithm uses Warshall's transitive closure algorithm (Warshall, 1962) to identify unique network chains between a related activity and a given target activity. The inference rules generalize the relationships between the role and flexibility of specific constraints in an activity's network chains with the behavior (i.e., role and status) of an activity in relation to a given target activity.

The classification mechanism is an integral part of the formal identification and re-sequencing process. Hence, the tests performed and discussed in Section 4.4 of Chapter 4 provide the evidence for the power and generality of the classification mechanism. Specifically, the three retrospective cases demonstrate that the classification mechanism correctly classifies the role and status of activities for construction schedules where sequencing rationale is described using different types of specific constraints. The charrette test also demonstrates that the classification mechanism implemented in CLCPM enables planners to classify activities more correctly and rapidly than using a conventional CPM scheduling tool.

Based on the evidence provided from the validation tests, I claim that the formal classification mechanism contributes to the current state of knowledge in the area of construction planning and scheduling domain.

I distinguish my research from existing construction AI planning systems as the formal mechanism uses constraints to infer the role and status of activities, rather than use constraints to generate an initial sequence of activities. Hence, the mechanism is useful for evaluating and modifying a plan, as to generating plans. In addition, my approach is unique in presenting an alternate approach for utilizing graph algorithms to solve a construction-specific problem. As discussed, I applied transitive algorithms not to check for schedule integrity such as cycle detection, but to identify unique network chains in a CPM network. The following section describes the limitations of the classification mechanism.

5.1.2.1 Limitations of the Classification Mechanism

The classification mechanism is limited in several ways. Specifically, I assumed that the activity sequences in a CPM network are only represented using finish-to-start (FS) precedence relationships. In addition, I did not evaluate the different types of transitive closure algorithms with respect to their run time performance. The following section describes these limitations and possible extensions to the classification mechanism to address these limitations.

(1) Precedence relationships limited to Finish to Start relationships

I assumed that the activity sequences in a CPM network were only represented using finish-to-start (FS) precedence relationships. As discussed in Section 5.1.1.1, using a FS precedence relationship seems acceptable as it does not fundamentally change the rationale behind activity sequences and does not change the duration of the individual activities. However, additional extensions also would need to be made for the classification mechanism to correctly classify activities in schedules where SS, SF and FF relationships are required and cannot be substituted using the FS relationship. This would in turn require modifying the inference rules to correctly infer the role and status of activities based on the rationale of activity sequences of SS, SF and FF relationships. It would also require modifying the network chain search algorithm to correctly identify network chains in these schedules.

An alternate approach may be to develop mechanisms that automatically modify SS, SF and FF relationships into FS relationships, or break down activities to enable the relationships to be replaced with a FS relationship.

(2) Run time performance of network search algorithm

Warshall's algorithm run time performance is $O(n^3)$. Other transitive closure algorithms (e.g., Agrawal and Jagadish, 1987; Jiang, 1990; Jakobsson, 1991) have been developed to increase the performance run time of identifying paths between vertices in a graph. I did not evaluate the efficiency of the different algorithms, but simply chose Warshall's algorithm, as it was easy and simple to code. For practical large-scale networks, Warshall's algorithm may practically take too much time. Hence, additional research is required to determine whether other algorithms will enable the network chain search algorithm to run faster.

This section described the second contribution I claim for this research, which is a formal classification mechanism for automatically inferring the role and status of activities. I also discussed the limitations of the classification mechanism. The following section describes the third contribution.

5.1.3 Contribution 3: Formal Process for Identifying and Re-sequencing Activities

The third contribution I claim is for a formal identification and re-sequencing process that formalizes the necessary steps required to identify and re-sequence activities with the goal of expediting specific target activities in a CPM schedule. The process formalizes the planner's rationale for determining *which* activities to delay and *how* the selected activity needs to be delayed. The process achieves these goals by utilizing the classification mechanism to classify and prioritize activities, and by utilizing the ontology to determine which constraints may be relaxed.

As discussed in Section 4.4 of Chapter 4, the three retrospective cases demonstrate that the process correctly identifies and re-sequences activities for different types of construction schedules. The charrette test demonstrates that the process enables planners to develop sequencing alternatives more correctly and rapidly than a conventional CPM scheduling tool.

Based on the evidence provided from these tests, I claim that the formal process contributes to the current state of knowledge in the area of construction planning and scheduling domain.

The formal process differs from existing CPM-based acceleration techniques (i.e., TCTP and resource allocation techniques), as the process allows existing sequences to be relaxed and prioritizes activities with respect to their role and status. In addition, the formal process does not accelerate individual activities, but enables the correct re-sequencing of activities with the goal of expediting specific target activities in a CPM network.

I also distinguish the formal process from existing AI replanning techniques as the process uses a domain-specific representation (i.e., ontology) and mechanisms to meet a domain-specific planning requirement. As discussed in Section 4.2 of Chapter 4, investigation of existing AI planning techniques showed that replanning techniques use customized representation and reasoning mechanisms to meet domain-specific requirements.

The following section describes the limitations of the formal process.

5.1.3.1 <u>Limitations of the Formal Process</u>

Several limitations exist for the formal process. In particular, I did not perform the validation tests for large-scale project schedules. In addition, the current graphical interface using Gantt charts needs to be improved in visually describing the role and status of activities identified in CLCPM.

The following section describes these limitations and possible extensions to address them.

(1) Scalability

The schedules I used to validate the formal process were limited to 27 activities. In practice, project schedules can range from 100 to even 1000 or more activities. Hence, additional validation is required to test the formal process for larger project schedules. As discussed in Section 4.4 of Chapter 4, I identified that a foreseeable problem for large-scale project schedules was the large number of candidate activities that may be identified and hence need to be ranked or prioritized in some way. Section 5.3 suggests possible ways for ranking candidate activities.

(2) Gantt chart representation

I implemented the formal identification and re-sequencing process in a CPM schedule represented as a Gantt chart. Although CLCPM visually highlights the role and status of activities, it may still be difficult for planners to comprehend or conceptually visualize how an upstream activity affects a downstream activity. One approach would be to integrate CLCPM with 4D models. Section 5.3 suggests possible research directions in this area.

This section described the third contribution I claim for this research, which is a formal process for developing sequencing alternatives. I also discussed the limitations of the formal process. The following section describes the practical implications of the research.

5.2 PRACTICAL IMPLICATIONS

The goal of the formalizations developed and implemented in this research was to enable planners to develop sequencing alternatives in CPM schedules correctly and rapidly.

Practically, I anticipate that the formalizations implemented in a system such as CLCPM add the most value during the course of a construction project where multiple project members need to evaluate the sequence of the project and modify initial activity sequences to meet changing project demands.

As demonstrated in the charrette test of Section 2.4.2 of Chapter 2, the project-independent constraints implemented in CLCPM increase the likelihood that multiple planners interpret the sequencing rationale more consistently and correctly. Hence, the project-independent constraints provide a common terminology for planners to describe and communicate their logic with other project members.

In addition, CLCPM enables planners to retain sequencing rationale for constraints in CPM schedules. Hence, CLCPM obviates the need of having to figure out and explain to multiple project participants the initial rationale for the activity sequences. Multiple project participants can easily interpret the rationale for constraints and whether particular constraints may or may not be relaxed. Hence, CLCPM can help project participants to save time by enabling them to focus on evaluating and determining potential sequencing alternatives.

As shown in the retrospective cases of Section 4.4.1 of Chapter 4, CLCPM correctly and visually shows the inter-dependent role between the trades involved in a project. Hence, CLCPM enables trades to quickly understand their role in the overall scheme of construction. That is, they can see how their work impacts following trades' work. Hence, CLCPM may encourage discussion and reduce the adversarial relationships sometimes experienced between trades in construction projects (O'Brien, 1994).

In addition, the retrospective cases show that planners can develop multiple sequencing alternatives correctly and quickly. Hence, multiple project participants can evaluate several sequencing alternatives and decide on a sequencing alternative that all members can agree on.

As shown in the charrette test of Section 4.4.2 of Chapter 4, CLCPM enables planners to develop sequencing alternatives in an interactive way. I anticipate that the hands-on approach enables multiple participants to make sequencing alternatives that are practically feasible, rather than rely on results from an optimization tool that may heavily rely on impractical assumptions.

This section discussed how the formalizations help multiple project participants in developing realistic scheduling alternatives to meet changing project requirements in a practical setting. The following section presents possible extensions to the current formalizations for future exploration.

5.3 SUGGESTED FUTURE WORK

This section addresses how several of the limitations identified in each of the contributions may be addressed by extending the formalizations of the current research. I also discuss how my research may be adopted or integrated with existing scheduling and planning frameworks. Several of the suggestions relate to addressing the scope I defined for this research, including the incorporation of design and procurement activities, incorporating costs, and developing ways to automate the current manual input of constraints. I also suggest improvements to the current user interface by integrating CLCPM with 4D models. I also discuss possible improvements required when using CLCPM for large-scale projects. Finally, I speculate how the formalizations may be adopted in a distributed coordination-planning framework. The following section discusses these approaches for future exploration.

(1) Incorporating sequencing rationale between design and procurement activities

As discussed in Section 5.1.1.1, I limited the ontology to primarily represent sequencing rationale of construction-specific activities, and hence the ontology does not formalize sequencing rationale between design/procurement activities, or between design/procurement activities and construction-specific activities.

One way to incorporate design and procurement activities may be to extend the current ontology to distinguish between physical and technical constraints with respect to their role (i.e., physically enabling/impeding, and technically enabling/impeding). The extensions to the ontology would in turn require extending the identification and re-sequencing process accordingly.

In the identification process, the process of identifying candidate activities would need to be extended to enable planners not only to differentiate between enabling and impeding activities, but also between physically enabling/impeding activities and technically enabling/impeding activities. In the re-sequencing process, the priority rules would need to be extended to prioritize activities based on the extended classification for the role of activities. Again, planners could prioritize between technically enabling/impeding activities with physically enabling/impeding activities.

The extended distinction for the role of activities may actually enable planners to identify and re-sequence activities in finer granularity. That is, the distinction may enable planners to distinguish between physically impeding and technically impeding activities when identifying activities to delay, and prioritize activities using the distinction when re-sequencing activities. Hence, the extended distinction may be particularly useful for managing the large number of candidate activities that may exist in a large project schedule.

(2) Incorporating Costs

As discussed in Section 5.1.1.1, cost is predictably a major factor for planners in determining which constraints to relax. Hence, costs will have to be incorporated so that planners can make re-sequencing decisions based on the cost implications. However, like time-cost trade-off, a foreseeable issue is where to get accurate cost data in the first place. An obvious source may be historical data, or another approach may be to enable planners to describe the flexibility of constraints qualitatively with respect to cost.

Once cost data for the constraints are obtained, an interesting future research is to formalize ways to use shadow prices. Shadow prices measure the marginal value or economic contribution of a resource to a particular performance measure (Hillier and Leiberman, 1995). In the context of my research, shadow prices may be used to perform sensitivity analysis, i.e., the impact or contribution a constraint has to the overall expedition of a target activity when it is relaxed. The analysis could be performed before actually relaxing any constraints, so that planners can *a priori* get a sense of the impact relaxing a constraint will have to the scheduling alternative developed.

(3) Developing methods to automate input of constraints

As discussed in Section 5.1.1.1, manually entering constraints may become impractical for large-scale projects. I foresee two possible approaches for relieving planners from having to input constraints manually:

i) Automatic inference of constraints from product models and specification tools: One possible approach would be to incorporate the methods used in domain-specific AI planning systems such as CMM (Aalami et al., 1998a) that use a product model that has sequencing relationships defined between components to automatically generate activity sequences. Formal methods could be developed to link the ontology with these relationships in the product model, so that activity sequences generated would retain their rationale. More recently, Haymaker et al. (2003) formalized “Perspectors” that automatically infers physical component relationships as “Perspectives” (i.e., views) using geometric features of components. Their approach could be used to automatically populate relationships in the product model, which can then be linked to the ontology.

As discussed in Section 2.2.2 of Chapter 2, however, it may not be possible to represent certain types of constraints that are not based on relationships between components in a product model (e.g., *damaged by*). One approach may be to integrate the ontology with existing internet-based regulation compliance tools such as “Regnet” (Kerrigan and Law, 2003) to import sequencing rationale dependent on regulatory information and compliance requirements.

ii) Use of Subnetworks to group activities: Another approach would be to group activities into subnetworks to reduce the number of activities and hence the number of precedence relationships. For example, one approach may be to use an approach similar to that used by Paulson (1973) to group activities. Paulson used the critical sequence of activities to group activities as “intervals.” Aalami et al. (1998b) also used construction method templates (i.e., CMMT’s) to group activities that constitute a unique construction method. Similarly, formal methods could be developed where planners group activities that represent repetitive work (e.g., formwork, rebar, and concrete activities) and that are linked by inflexible constraints, and the same types of constraints are instantiated for the same type of activities throughout the schedule.

(4) Integrating CLCPM with 4D models

As discussed in Section 5.1.1.3, I implemented the formal identification and re-sequencing process in a CPM schedule represented as a Gantt chart that limits planners' comprehension of how an upstream activity affects a downstream activity. One approach would be to integrate CLCPM with 4D models. For example, planners could view the 3D components in a 4D model that correspond to the activities identified by CLCPM as enabling or candidate activities. Planners could also use the 4D models to review the sequencing alternatives developed in CLCPM. Alternatively, 4D models could be used to identify sequence problems in an existing schedule. For example, Koo and Fischer (2000) describe how 4D models can help planners identify potential sequencing mistakes overlooked in a CPM-based Gantt chart. Subsequently, planners could use CLCPM to develop sequencing alternatives that correct these mistakes.

(5) Improving identification and re-sequencing process to be used for large-scale projects

As discussed in 5.1.1.3, I identified that a foreseeable problem for large-scale project schedules was the large number of candidate activities that may be identified and hence need to be ranked or prioritized. Possible approaches for resolving this limitation are:

(i) Use of Subnetworks to group activities: One approach discussed for relieving the manual input of constraints was to group activities of repetitive work. Correspondingly, the candidate activities could be identified at the group level and not at the individual activity level.

ii) Potential criteria for ranking candidate activities:

- *Cost:* Rank candidate activities based on the cost incurred for delaying a specific candidate the activity.
- *Number of impeding activities:* Rank candidate activities based on the number of successors the candidate activity impedes.
- *Technically versus physically impeding activities:* As discussed, extending the ontology to distinguish between technical versus physical constraints can further distinguish the role of activities into physical enabling/impeding versus technically enabling/impeding activities. The classification may in turn be used to rank candidate activities in finer granularity.

I anticipate that all three criteria may be used separately or collectively to rank candidate activities.

(6) Using the ontology and formal process in a distributed coordination framework

The current framework for re-sequencing activities is a centralized coordination planning approach. Recent research has focused on a distributed coordination framework for meeting project schedule changes. For example, Kim and Paulson (2003) use an agent-based compensatory negotiation (ABCN) methodology to allocate resources between multiple trades in a distributed planning framework. One drawback of this approach is that Kim and Paulson do not distinguish whether activity sequences may or may not be relaxed. Hence, integrating the ontology and formal process with distributed agents may also be a promising extension to the research.

In this section, I described six possible research directions I believe to be promising next steps for extending the formalizations developed to support planners in developing sequencing alternatives more comprehensively and effectively. Research in these areas would significantly increase construction planners' ability to evaluate different sequencing scenarios, enabling them to respond to changes during construction projects in a timely and efficient way.

APPENDIX A. COMMON TYPES OF CONSTRAINTS IN CONSTRUCTION SCHEDULES

Table A.1 is an excerpt from Echeverry et al. (1991) describing the governing factors that pertain to the actual installation processes and associated interaction between multiple trades during the construction of a facility.

Tables A.1, A.2 and A.3 describe specific component relationships, interactions and regulations identified in the literature.

Governing factor	Description
Physical relationships among building components	Building components are spatially restricted, weather protected, or gravity supported by other components. Activity sequencing has to respond to these inter-component relationships.
Trade interaction	Activity sequencing also responds to different ways in which trades affect each other during the construction phase.
Path interference	Building components have to be moved around the jobsite in order to be installed. An activity sequence has to guarantee an interference-free path for installation of any component.
Code regulation	Activity sequencing is also responsive to construction-phase safety considerations.

Table A.1: Governing factors.

Physical component relationship	Identified by	Description	Example
Supported by	Darwiche et al. (1988), Navinchandra et al. (1998), Echeverry et al. (1991), Kähkönen (1993)	Component provides physical support for another component.	Column supports Beam.
Connected to	Darwiche et al. (1988), Navinchandra et al. (1998), Echeverry et al. (1991)	Relating to physical connection, not necessarily a support.	Wall lamps connected to electrical fittings.
Covered by	Darwiche et al. (1988), Echeverry et al. (1991)	Component covering another component.	Wall covered by paint.
Embedded in (Structural)	Echeverry et al. (1991)	Component embedded in another component, combining to serve a structural function.	Reinforcement embedded in cast in place concrete.
Embedded in (non-structural) or Enclosed by	Darwiche et al. (1988), Echeverry et al. (1991)	Component embedded in another component without a structural function.	Electrical conduit embedded into masonry wall.
Relative distance to support, with flexibility of installation	Echeverry et al. (1991)	Components relying on third component for support is installed based on distance to support and flexibility of their installation.	Cast iron waste pipe is less flexible than air handling duct.
Relative distance to access	Echeverry et al. (1991)	Several identical components having to be installed in a work area with limited access.	Pile driving with single access (start from farthest to closest from access).
Weather protected by	Darwiche et al. (1988), Echeverry et al. (1991)	Component requires weather protection prior to installation.	Drywall needs dry enclosure prior to installation.

Table A.2: Physical relationships between components

Interaction type	Identified by	Description	Example
Space competition	Echeverry et al. (1991), Kähkönen (1993)	Two crews or trades need to work concurrently and need the same workspace.	Concrete slab shoring occupies space required by interior wall layout crew.
Resource limitations	Echeverry et al. (1991), Kähkönen (1993)	Two crews or trade need the same resource.	Two crews compete for the same crane.
Unsafe environment effects (Hazardous to)	Antill and Woodhead (1970) Echeverry et al. (1991), Kähkönen (1993)	Environmental effects that are hazardous to workers on site.	Fireproofing may be hazardous to rough plumbing crew.
Damaging of installed components (Damaged by)	Darwiche et al. (1988), Echeverry et al. (1991) Kähkönen (1993)	If an activity damages the finished work of another activity, then the damageable work should be performed afterwards.	Carpeting may be damaged by paint.
Requirement of service	Echeverry et al. (1991), Kähkönen (1993)	If a crew requires a service like water or power supply, then the system providing the service needs to be available as a requisite for that crew's work.	Power supply required for elevator installation.
Workflow	Kähkönen (1993), Tommelein (1998)	Crews prefer moving from one area to the next which is situated as near as possible.	Concrete and reinforcement crew must follow dictated workflow.

Table A.3: Interactions between trades

Factor	Identified by	Description	Example
Path interference	Echeverry et al. (1991)	Building components have to be moved around the jobsite in order to be installed. Activity sequence has to guarantee an interference-free path for installation of any component.	Permanent units (e.g., boiler) needs to placed prior to enclosure installation.
Code regulations	Echeverry et al. (1991), Kähkönen (1993)	Activity sequencing is also responsive to construction-phase safety considerations.	Codes enforce safety, testing and inspection by specifying activity sequence.

Table A.4: Path interference and code regulations

APPENDIX B. RESULTS OF THE RETROSPECTIVE VALIDATION STUDIES

In this section, I present the results for the three retrospective validation studies performed to demonstrate the power and generality of the ontology, classification mechanism and formal process. Specifically, I performed the retrospective validation studies using three project schedules:

(1) Intel CUB schedule

(2) McWhinney project schedule

(3) Bay Street project schedule

Section B.1 provides a brief overview for each of the three project schedules and discusses the specific need of each project for expediting particular activities.

As discussed in Section 2.4.1 of Chapter 2, I used the three project schedules to demonstrate that the ontology classifies sequencing rationale in a disjoint and exhaustive way. Section B.2 presents the results for using the ontology to describe and classify the sequencing rationale for the three projects.

As discussed in Section 4.4.1 of Chapter 4, I used the three project schedules to demonstrate that the formal process identifies and re-sequences activities correctly. Section B.3 and B.4 present the results in detail for the identification and re-sequencing process, respectively.

B.1 PROJECT OVERVIEW FOR THE THREE RETROSPECTIVE CASES

As discussed in Chapters 2 and 4, I used three different retrospective cases to demonstrate the power and generality of the formalizations developed in this research. Table B.1 provides an overview of the three retrospective cases.

Project	Number of Activities	Number of Precedence Relationships	Target Activity (TA)	Phase of Project
Intel CUB	18	25	Install Process Pipes B (ID: 17).	Concrete and structural frame.
McWhinney	27	37	HVAC Balance (ID: 25)	Structural Roof, Interior MEP and finishes.
Bay Street	27	36	Book Store Turnover (ID: 25)	Concrete and structural frame, Exterior closure, interior MEP and finishes.

Table B.1: Overview of the three retrospective cases.

In the following sections, I briefly discuss the project characteristics and discuss why activities in the initial schedule needed to be re-sequenced to expedite specific target activities.

B.1.1 Intel CUB Schedule

Figure B.1 shows a 3D model view of the Intel Central Utility Building (CUB). The schedule provided describes the sequences of trades performing the foundation, structural frame and process pipes of the building. The project manager has sectioned the CUB building into three major zones (i.e., A, B and C) for planning and coordination purposes.

On viewing the schedule, the project manager determined that process pipes (i.e., activity Install Process Pipes B (ID: 17) in Figure B.2) needed to be installed earlier than the planned start date (4/22/03[24]). The process pipes are a major component of the project and needs to be installed as early as possible to enable work in adjacent building to start.

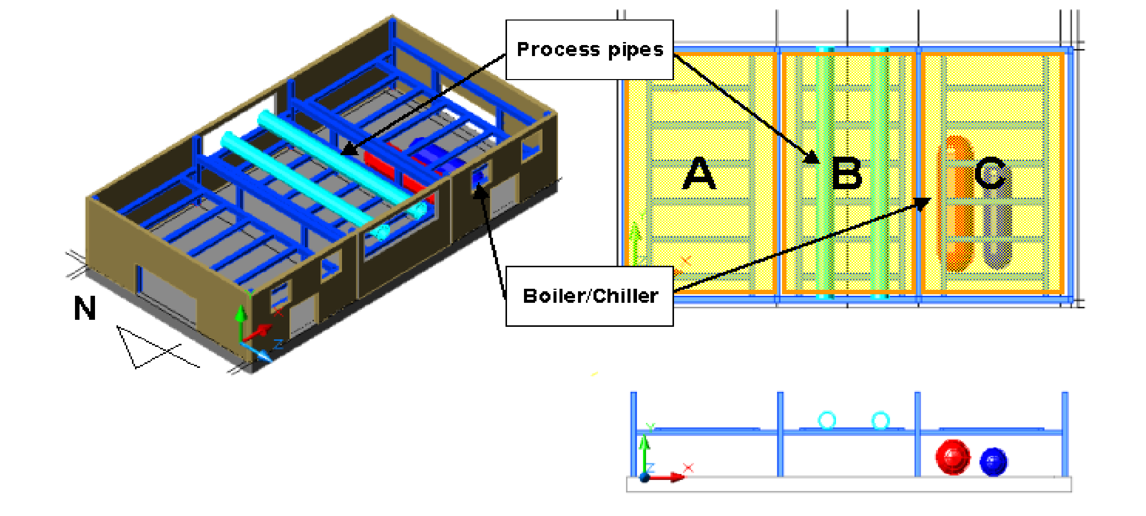
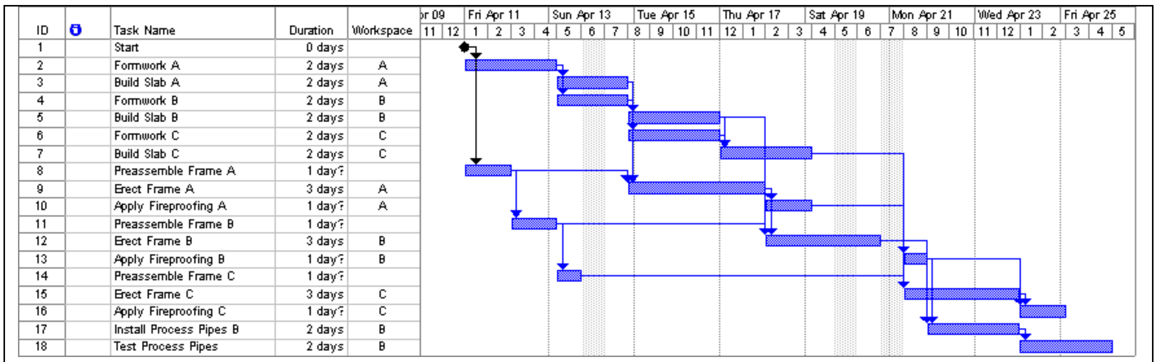


Figure B.1: Overview diagram of Central Utility Building (CUB).
A, B and C denote the zones sectioned for planning and coordination purposes.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Formwork A	2 days	A
3		Build Slab A	2 days	A
4		Formwork B	2 days	B
5		Build Slab B	2 days	B
6		Formwork C	2 days	C
7		Build Slab C	2 days	C
8		Preassemble Frame A	1 day?	
9		Erect Frame A	3 days	A
10		Apply Fireproofing A	1 day?	A
11		Preassemble Frame B	1 day?	
12		Erect Frame B	3 days	B
13		Apply Fireproofing B	1 day?	B
14		Preassemble Frame C	1 day?	
15		Erect Frame C	3 days	C
16		Apply Fireproofing C	1 day?	C
17		Install Process Pipes B	2 days	B
18		Test Process Pipes	2 days	B

Figure B.2: Intel CUB schedule.

[24] Date is fictitious.

B.1.2 McWhinney Project Schedule

Figure B.3 shows a 3D model of the McWhinney project, a two story office building. The schedule shown in Figure B.4 describes the sequence of trades' work for the gable roof, interior MEP and finishes for four phases of the 2nd floor of the building. All trades are sequenced to work from Phase 1 to 2 to 3 and 4. Roofing includes the Gable Roof, Roof Membrane and Roof Top Units (i.e., RTU's). Interior MEP and finishes in each phase can start after Roof Membrane is installed. Each phase of interior MEP and finishes includes Wall Framing, Wall Electrical, Overhead HVAC, Sprinklers and Prime Paint.

On viewing the schedule, the project manager of the project determined that the testing of HVAC systems (i.e., activity HVAC balance (ID: 25) in Figure B.4) needed to be performed earlier than the planned start date (7/21/03[25]).

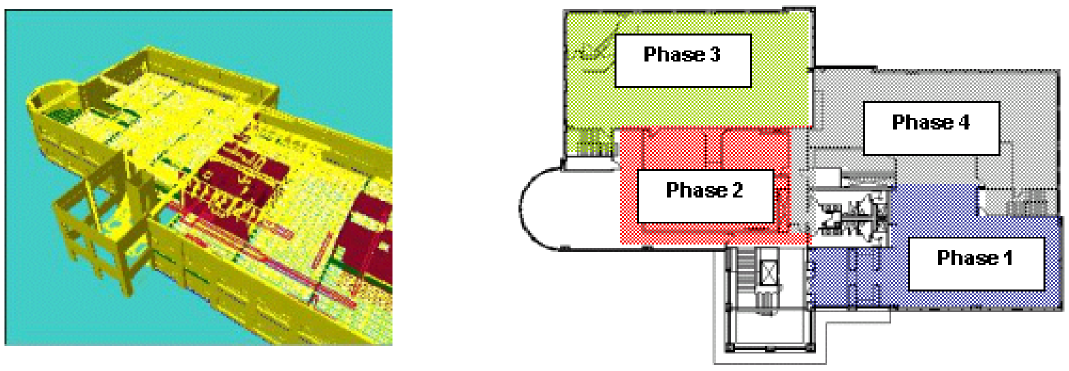
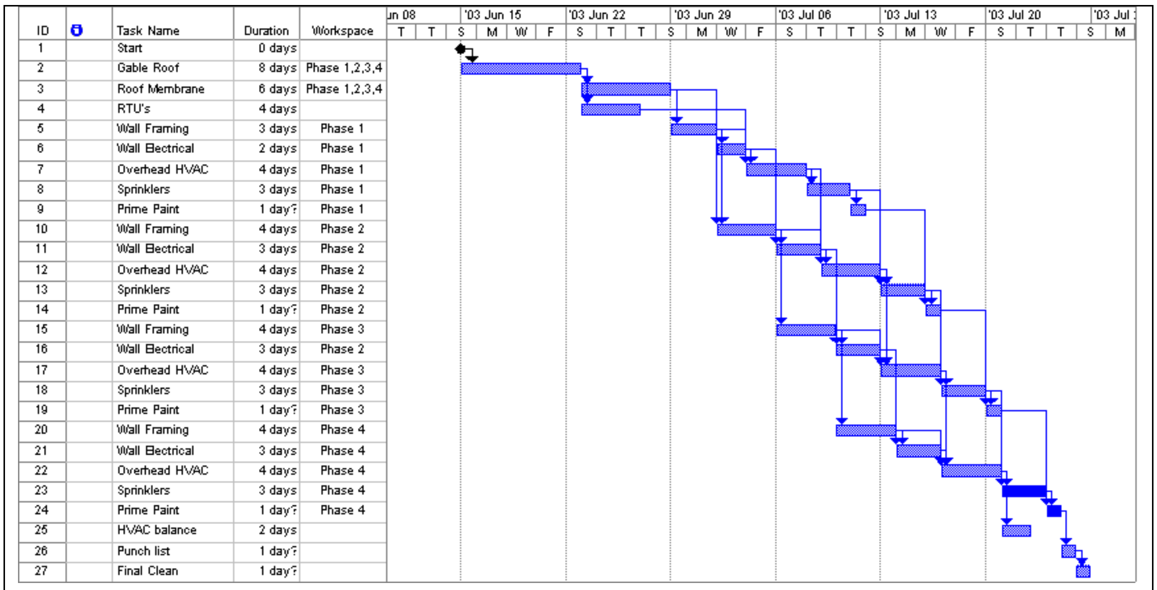


Figure B.3: Overview diagram of McWhinney project schedule.
Phases 1, 2, 3 and 4 denote the zones sectioned for planning and coordination purposes.

[25] Date is fictitious.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Gable Roof	8 days	Phase 1,2,3,4
3		Roof Membrane	6 days	Phase 1,2,3,4
4		RTU's	4 days	
5		Wall Framing	3 days	Phase 1
6		Wall Electrical	2 days	Phase 1
7		Overhead HVAC	4 days	Phase 1
8		Sprinklers	3 days	Phase 1
9		Prime Paint	1 day?	Phase 1
10		Wall Framing	4 days	Phase 2
11		Wall Electrical	3 days	Phase 2
12		Overhead HVAC	4 days	Phase 2
13		Sprinklers	3 days	Phase 2
14		Prime Paint	1 day?	Phase 2
15		Wall Framing	4 days	Phase 3
16		Wall Electrical	3 days	Phase 2
17		Overhead HVAC	4 days	Phase 3
18		Sprinklers	3 days	Phase 3
19		Prime Paint	1 day?	Phase 3
20		Wall Framing	4 days	Phase 4
21		Wall Electrical	3 days	Phase 4
22		Overhead HVAC	4 days	Phase 4
23		Sprinklers	3 days	Phase 4
24		Prime Paint	1 day?	Phase 4
25		HVAC balance	2 days	
26		Punch list	1 day?	
27		Final Clean	1 day?	

Figure B.4: McWhinney project schedule.

B.1.3 Bay Street Project Schedule

Figure B.5 shows a 3D model of the Retail store of the Bay Street Project. The schedule shown in Figure B.6 shows the sequence of trades performing the exterior closure (EXT) and interior work (INT) for Areas 1 and 2 of the retail store. All trades are sequenced to work in Area 1 then Area 2. The exterior work involves sealing the building (i.e., Building Dry-in), placing the brick veneer, painting and installing storefront glass. Interior work includes wall framing and MEP-rough-ins. Interior work can begin as soon as roofing is completed in each area.

The project has been delayed overall due to hazardous material found on-site. Based on the current sequence, the date (8/05/03[26]) for the bookstore turnover (i.e., activity Bookstore turnover-Area 2 (ID: 25)) needs to be expedited. Contractually, all interior and exterior work must be completed for the area (i.e., area 2) where the bookstore is to be placed, except exterior painting.

[26] Date is fictitious.

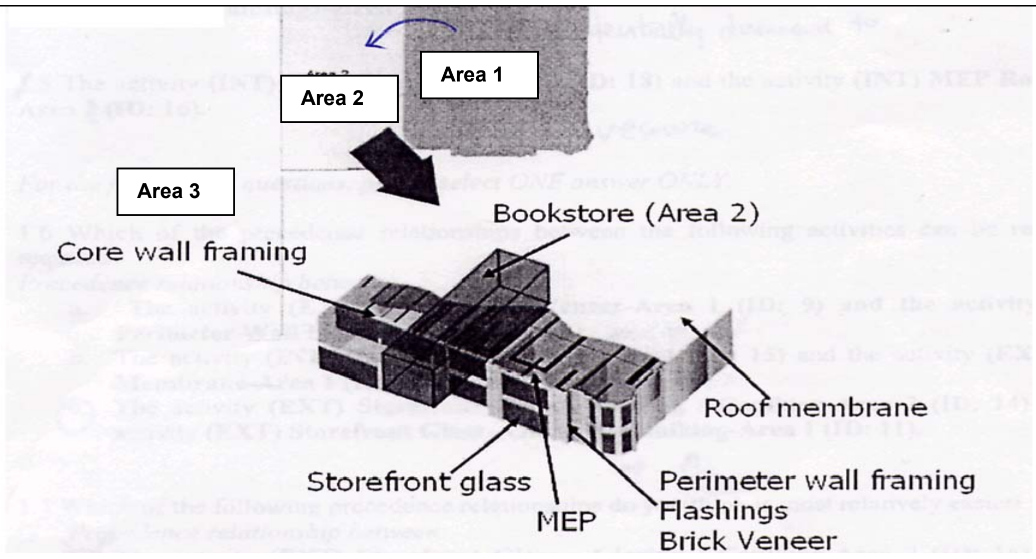
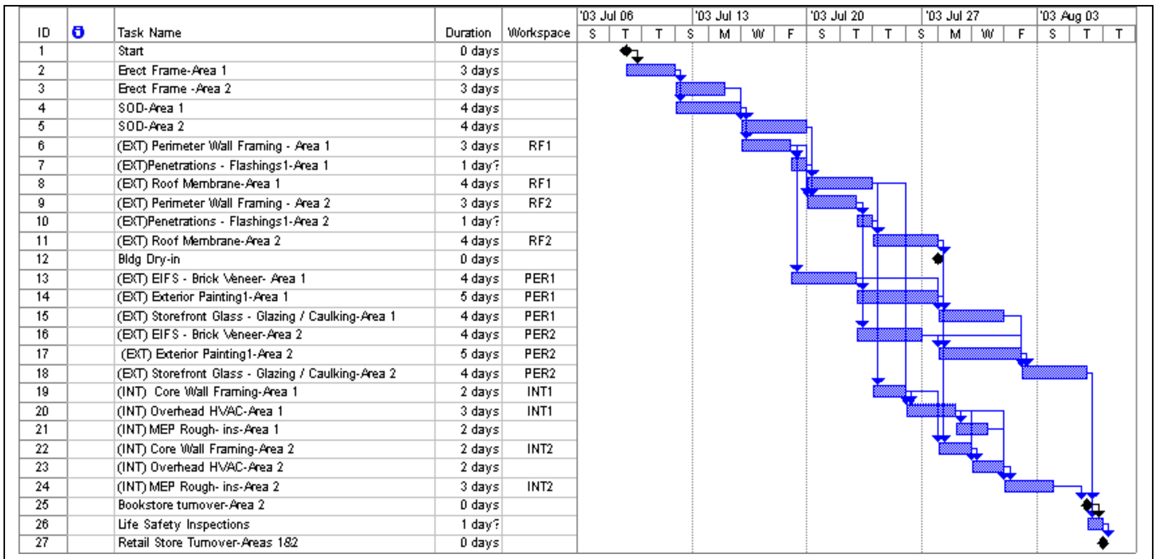


Figure B.5: Overview diagram of Bay Street project schedule.

Areas 1, 2, and 3 denote the zones sectioned for planning and coordination purposes.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.6: Bay Street project schedule.

This section provided an overview of the three project schedules. The following section describes the results of using the ontology to describe and classify the sequencing rationale for the three projects.

B.2 SEQUENCING RATIONALE CLASSIFICATION RESULTS

As discussed in Section 2.4.1 of Chapter 2, I used CLCPM to describe the sequencing rationale for the three project schedules, classified the specific constraints with respect to their role and flexibility, and confirmed the classifications with an experienced project scheduler.

Sections B.2.1 through B.2.3 present the specific constraints described and classified for the three project schedules.

B.2.1 Intel CUB Schedule

Tables B.2 and B.3 show the specific constraints used to describe sequencing rationale for the Intel CUB schedule. Table B.4 shows the unique constraints and their classifications. Table B.5 shows the names and corresponding ID's for each activity.

	Predecessor-Activity	Constraint Name	Role	Flexibility	Confirmed
1	2-3	Physically required by	ENABLING	INFLEXIBLE	Yes
2	2-3	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
3	2-4	Resource constrained by	IMPEDING	FLEXIBLE	Yes
4	4-5	Physically required by	ENABLING	INFLEXIBLE	Yes
5	3-5	Resource constrained by	IMPEDING	FLEXIBLE	Yes
6	4-6	Resource constrained by	IMPEDING	FLEXIBLE	Yes
7	6-7	Physically required by	ENABLING	INFLEXIBLE	Yes
8	5-7	Resource constrained by	IMPEDING	FLEXIBLE	Yes
9	8-9	Physically required by	ENABLING	INFLEXIBLE	Yes
10	3-9	Component-supported by	ENABLING	INFLEXIBLE	Yes
11	3-9	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
12	9-10	Covers	IMPEDING	INFLEXIBLE	Yes
13	8-11	Resource constrained by	IMPEDING	FLEXIBLE	Yes
14	11-12	Physically required by	ENABLING	INFLEXIBLE	Yes
15	5-12	Component-supported by	ENABLING	INFLEXIBLE	Yes

Table B.2: List of specific constraints used to describe sequencing rationale for the Intel CUB schedule.

	Predecessor -Activity	Constraint Name	Role	Flexibility	Confirmed
16	5-12	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
17	9-12	Resource constrained by	IMPEDING	FLEXIBLE	Yes
18	12-13	Covers	IMPEDING	INFLEXIBLE	Yes
19	10-13	Resource constrained by	IMPEDING	FLEXIBLE	Yes
20	11-14	Resource constrained by	IMPEDING	FLEXIBLE	Yes
21	14-15	Physically required by	ENABLING	INFLEXIBLE	Yes
22	12-15	Resource constrained by	IMPEDING	FLEXIBLE	Yes
23	7-15	Component-supported by	ENABLING	INFLEXIBLE	Yes
24	7-15	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
25	15-16	Covers	IMPEDING	INFLEXIBLE	Yes
26	13-16	Resource constrained by	IMPEDING	FLEXIBLE	Yes
27	12-17	Component-supported by	ENABLING	INFLEXIBLE	Yes
28	12-17	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
29	13-17	Potentially damaged by	IMPEDING	FLEXIBLE	Yes
30	17-18	Enables testing of	ENABLING	INFLEXIBLE	Yes

Table B.3: List of specific constraints used to describe sequencing rationale for the Intel CUB schedule (Continued from Table B.2).

Unique Constraints	Role	Flexibility	Confirmed
Physically required by	ENABLING	INFLEXIBLE	Yes
Resource constrained by	IMPEDING	FLEXIBLE	Yes
Component-supported by	ENABLING	INFLEXIBLE	Yes
Covers	IMPEDING	INFLEXIBLE	Yes
Potentially damaged by	IMPEDING	FLEXIBLE	Yes
Enables testing of	ENABLING	INFLEXIBLE	Yes

Table B.4: List of unique constraints classified using the abstract types of the ontology.

Activity ID	Task Name	Activity ID	Task Name
1	Start	10	Apply Fireproofing A
2	Formwork A	11	Preassemble Frame B
3	Build Slab A	12	Erect Frame B
4	Formwork B	13	Apply Fireproofing B
5	Build Slab B	14	Preassemble Frame C
6	Formwork C	15	Erect Frame C
7	Build Slab C	16	Apply Fireproofing C
8	Preassemble Frame A	17	Install Process Pipes B
9	Erect Frame A	18	Test Process Pipes

Table B.5: List of activity ID and names for the Intel CUB schedule.

B.2.2 McWhinney Project Schedule

Tables B.6 and B.7 show the specific constraints used to describe sequencing rationale for the Intel CUB schedule. Table B.8 shows the unique constraints and their classifications. Table B.9 shows the names and corresponding ID's for each activity.

	Predecessor - Activity	Constraint Name	Role	Flexibility	Confirmed
1	2-3	Attached to	ENABLING	INFLEXIBLE	Yes
2	2-4	Component-supported by	ENABLING	INFLEXIBLE	Yes
3	3-5	Protected by	ENABLING	INFLEXIBLE	Yes
4	5-6	Enclosed by	ENABLING	INFLEXIBLE	Yes
5	6-7	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
6	4-7	Technologically required by	ENABLING	INFLEXIBLE	Yes
7	5-7	Potentially obstructed by	IMPEDING	FLEXIBLE	No (Role)
8	7-8	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
9	7-8	Less bulky than	IMPEDING	FLEXIBLE	No (Role)
10	8-9	Covers	IMPEDING	INFLEXIBLE	Yes
11	5-10	Resource constrained by	IMPEDING	FLEXIBLE	Yes
12	3-10	Protected by	ENABLING	INFLEXIBLE	Yes
13	10-11	Enclosed by	ENABLING	INFLEXIBLE	Yes
14	7-12	Resource constrained by	IMPEDING	FLEXIBLE	Yes
15	7-12	Technologically required by	ENABLING	FLEXIBLE	Yes
16	10-12	Potentially obstructed by	IMPEDING	FLEXIBLE	No (Role)
17	12-13	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
18	12-13	Less bulky than	IMPEDING	FLEXIBLE	No (Role)
19	8-13	Resource constrained by	IMPEDING	FLEXIBLE	Yes
20	9-14	Resource constrained by	IMPEDING	FLEXIBLE	Yes
21	10-15	Resource constrained by	IMPEDING	FLEXIBLE	Yes

Table B.6: List of specific constraints used to describe sequencing rationale for the McWhinney project schedule.

	Predecessor - Activity	Constraint Name	Role	Flexibility	Confirmed
22	15-16	**Enclosed by**	ENABLING	INFLEXIBLE	Yes
23	11-16	**Resource constrained by**	IMPEDING	FLEXIBLE	Yes
24	16-17	**Workspace constrained by**	IMPEDING	FLEXIBLE	Yes
25	12-17	**Resource constrained by**	IMPEDING	FLEXIBLE	Yes
26	12-17	**Contractually required by**	ENABLING	FLEXIBLE	Yes
27	15-17	**Potentially obstructed by**	IMPEDING	FLEXIBLE	No (Role)
28	17-18	**Less bulky than**	ENABLING	FLEXIBLE	No (Role)
29	17-18	**Workspace constrained by**	IMPEDING	FLEXIBLE	Yes
30	18-19	**Covers**	IMPEDING	INFLEXIBLE	Yes
31	14-19	**Resource constrained by**	IMPEDING	FLEXIBLE	Yes
32	21-22	**Workspace constrained by**	IMPEDING	FLEXIBLE	Yes
33	17-22	**Resource constrained by**	IMPEDING	FLEXIBLE	Yes
34	17-22	**Physically required by**	ENABLING	FLEXIBLE	Yes
35	20-22	**Potentially obstructed by**	IMPEDING	FLEXIBLE	No (Role)
36	22-23	**Workspace constrained by**	IMPEDING	FLEXIBLE	Yes
37	22-23	**Less bulky than**	IMPEDING	FLEXIBLE	No (Role)
38	18-23	**Resource constrained by**	IMPEDING	FLEXIBLE	Yes
39	23-24	**Covers**	IMPEDING	INFLEXIBLE	Yes
40	19-24	**Resource constrained by**	IMPEDING	FLEXIBLE	Yes
41	22-25	**Contractually required by**	ENABLING	INFLEXIBLE	Yes

Table B.7: List of specific constraints used to describe sequencing rationale for the McWhinney project schedule (continued from Table B.6).

Unique Constraints	Role	Flexibility	Confirmed
Attached to	ENABLING	INFLEXIBLE	Yes
Component-supported by	ENABLING	INFLEXIBLE	Yes
Protected by	ENABLING	INFLEXIBLE	Yes
Enclosed by	ENABLING	INFLEXIBLE	Yes
Workspace constrained by	IMPEDING	FLEXIBLE	Yes
Physically required by	ENABLING	INFLEXIBLE	Yes
Potentially obstructed by	ENABLING	FLEXIBLE	No (Role)
Less bulky than	ENABLING	FLEXIBLE	No (Role)
Covers	IMPEDING	INFLEXIBLE	Yes
Resource constrained by	IMPEDING	FLEXIBLE	Yes

Table B.8: List of unique constraints classified using the abstract types of the ontology.

Activity ID	Task Name	Activity ID	Task Name
1	Start	15	Wall Framing – Phase 3
2	Gable Roof – Phase 1,2,3,4	16	Wall Electrical – Phase 2
3	Roof Membrane – Phase 1,2,3,4	17	Overhead HVAC – Phase 3
4	RTU's	18	Sprinklers – Phase 3
5	Wall Framing – Phase 1	19	Prime Paint – Phase 3
6	Wall Electrical – Phase 1	20	Wall Framing – Phase 4
7	Overhead HVAC – Phase 1	21	Wall Electrical – Phase 4
8	Sprinklers - Phase 1	22	Overhead HVAC – Phase 4
9	Prime Paint - Phase 1	23	Sprinklers – Phase 4
10	Wall Framing – Phase 2	24	Prime Paint – Phase 4
11	Wall Electrical – Phase 2	25	HVAC balance
12	Overhead HVAC – Phase 2	26	Punch list
13	Sprinklers – Phase 2	27	Final Clean
14	Prime Paint – Phase 2		

Table B.9: List of activity ID and names for the McWhinney Project schedule.

B.2.3 Bay Street Project Schedule

Tables B.10 and B.11 show the specific constraints used to describe sequencing rationale for the Intel CUB schedule. Table B.12 shows the unique constraints and their classifications. Table B.13 shows the names and corresponding ID's for each activity

#	Predecessor-Activity	Constraint Name	Role	Flexibility	Confirmed
1	2-3	Resource constrained by	IMPEDING	FLEXIBLE	Yes
2	2-4	Component-supported by	ENABLING	INFLEXIBLE	Yes
3	2-4	Workspace-constrained by	IMPEDING	FLEXIBLE	Yes
4	3-5	Component-supported by	ENABLING	INFLEXIBLE	Yes
5	3-5	Workspace-constrained by	IMPEDING	FLEXIBLE	Yes
6	4-5	Resource constrained by	IMPEDING	FLEXIBLE	Yes
7	4-6	Workspace constrained by	IMPEDING	INFLEXIBLE	Yes
8	4-6	Resource-supported by	ENABLING	INFLEXIBLE	Yes
9	6-7	Covers	IMPEDING	INFLEXIBLE	Yes
10	7-8	Component-supported by	ENABLING	INFLEXIBLE	Yes
11	7-8	Workspace-constrained by	IMPEDING	FLEXIBLE	Yes
12	6-9	Resource constrained by	IMPEDING	FLEXIBLE	Yes
13	5-9	Workspace Constrained by	IMPEDING	INFLEXIBLE	Yes
14	5-9	Resource-supported by	ENABLING	INFLEXIBLE	Yes
15	9-10	Covers	IMPEDING	INFLEXIBLE	Yes
16	8-11	Resource constrained by	IMPEDING	FLEXIBLE	Yes
17	10-11	Component-supported by	ENABLING	INFLEXIBLE	Yes
18	11-12	Technically required by	ENABLING	INFLEXIBLE	Yes

Table B.10: List of specific constraints used to describe sequencing rationale for the Bay Street project schedule.

#	Predecessor-Activity	Constraint Name	Role	Flexibility	Confirmed
19	06-13	Enclosed by	ENABLING	INFLEXIBLE	Yes
20	13-14	Covers	IMPEDING	INFLEXIBLE	Yes
21	14-15	Potentially damaged by	IMPEDING	FLEXIBLE	Yes
22	13-15	Attached to	ENABLING	INFLEXIBLE	Yes
23	13-16	Resource constrained by	IMPEDING	FLEXIBLE	Yes
24	09-16	Enclosed by	ENABLING	INFLEXIBLE	Yes
25	16-17	Covers	IMPEDING	INFLEXIBLE	Yes
26	14-17	Resource constrained by	IMPEDING	FLEXIBLE	Yes
27	15-18	Resource constrained by	IMPEDING	FLEXIBLE	Yes
28	16-18	Attached to	ENABLING	INFLEXIBLE	Yes
29	17-18	Potentially damaged by	IMPEDING	FLEXIBLE	Yes
30	17-18	Workspace-constrained by	IMPEDING	FLEXIBLE	Yes
31	18-19	Protected by	ENABLING	INFLEXIBLE	Yes
32	18-19	Workspace-constrained by	IMPEDING	FLEXIBLE	Yes
33	19-20	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
34	19-20	Potentially obstructed by	ENABLING	FLEXIBLE	No (Role)
35	20-21	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
36	20-21	Less bulky than	ENABLING	FLEXIBLE	No (Role)
37	11-22	Protected by	ENABLING	INFLEXIBLE	Yes
38	19-22	Resource constrained by	IMPEDING	FLEXIBLE	Yes
39	22-23	Potentially obstructed by	ENABLING	FLEXIBLE	No (Role)
40	22-23	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
41	23-24	Less bulky than	ENABLING	FLEXIBLE	No (Role)
42	23-24	Workspace constrained by	IMPEDING	FLEXIBLE	Yes
43	21-24	Resource constrained by	IMPEDING	FLEXIBLE	Yes
44	24-25	Technically required by	ENABLING	INFLEXIBLE	Yes
45	18-25	Technically required by	ENABLING	INFLEXIBLE	Yes

Table B.11: List of specific constraints used to describe sequencing rationale for the Bay Street project schedule (continued from Table B.10).

Unique constraint name	Role	Flexibility	Confirmed
Resource constrained by	IMPEDING	FLEXIBLE	Yes
Component-supported by	ENABLING	INFLEXIBLE	Yes
Workspace constrained by	IMPEDING	INFLEXIBLE	Yes
Resource-supported by	ENABLING	INFLEXIBLE	Yes
Covers	ENABLING	INFLEXIBLE	Yes
Resource constrained by	IMPEDING	FLEXIBLE	Yes
Technically required by	ENABLING	INFLEXIBLE	Yes
Enclosed by	ENABLING	INFLEXIBLE	Yes
Potentially damaged by	IMPEDING	FLEXIBLE	Yes
Attached to	ENABLING	INFLEXIBLE	Yes
Protected by	ENABLING	INFLEXIBLE	Yes
Potentially obstructed by	ENABLING	FLEXIBLE	No (Role)
Less bulky than	ENABLING	FLEXIBLE	No (Role)

Table B.12: List of unique constraints classified using the abstract types of the ontology.

Activity ID	Activity Name	Activity ID	Activity Name
1	Start	15	(EXT) Storefront Glass - Glazing / Caulking-Area 1
2	Erect Frame-Area 1	16	(EXT) EIFS - Brick Veneer-Area 2
3	Erect Frame -Area 2	17	(EXT) Exterior Painting1-Area 2
4	SOD-Area 1	18	(EXT) Storefront Glass - Glazing / Caulking-Area 2
5	SOD-Area 2	19	(INT) Core Wall Framing-Area 1
6	(EXT) Perimeter Wall Framing - Area 1	20	(INT) Overhead HVAC-Area 1
7	(EXT)Penetrations - Flashings1-Area 1	21	(INT) MEP Rough- ins-Area 1
8	(EXT) Roof Membrane-Area 1	22	(INT) Core Wall Framing-Area 2
9	(EXT) Perimeter Wall Framing - Area 2	23	(INT) Overhead HVAC-Area 2
10	(EXT)Penetrations - Flashings1-Area 2	24	(INT) MEP Rough- ins-Area 2
11	(EXT) Roof Membrane-Area 2	25	Bookstore turnover-Area 2
12	Bldg Dry-in	26	Life Safety Inspections
13	(EXT) EIFS - Brick Veneer- Area 1	27	Retail Store Turnover-Areas 1&2
14	(EXT) Exterior Painting1-Area 1		

Table B.13: List of activity ID and names for the Bay Street project schedule.

This section described the detailed results of using the ontology to describe and classify sequencing rationale for the three project schedules. The next section describes the results for using the identification process to identify enabling and candidate activities for the three project schedules.

B.3 ACTIVITY IDENTIFICATION RESULTS

As discussed in Section 4.4.1 of Chapter 4, I used CLCPM to identify the enabling activities and candidate activities of the three project schedules. Table B.14 summarizes the validation results of the identification process for the three retrospective cases.

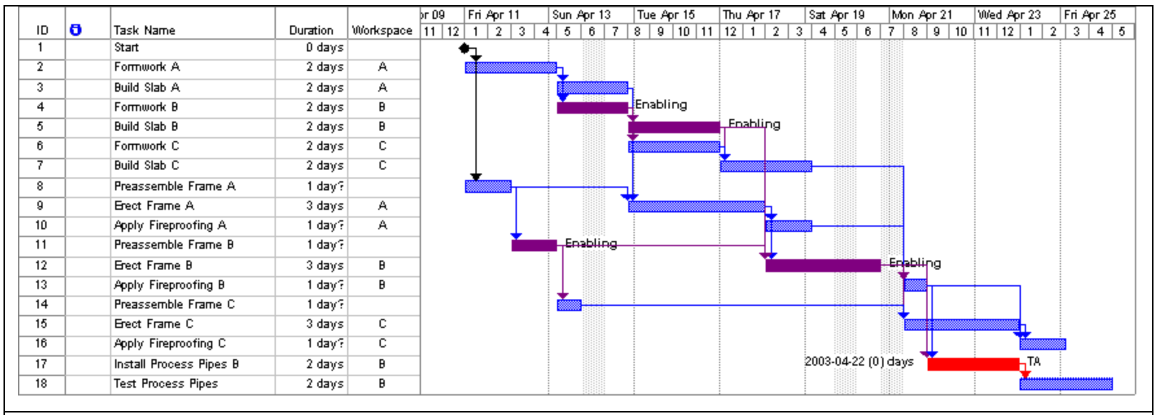
Project	TA initial date	Number of Enabling activities	Confirmed	Number of CA's	Confirmed
Intel CUB	4/22	4	4	4	4
McWhinney	7/21	11	11	3	3
Bay Street	8/05	10	10	6	6

Table B.14: Validation results of the identification process for the three retrospective cases.

The following section presents the detailed activities identified and the confirmation results for each project schedule.

B.3.1 Intel CUB Schedule

Figure B.7 and Table B.15 show the enabling activities identified for the Intel CUB schedule. Figure B.8 and Table B.16 show the candidate activities identified for the Intel CUB schedule.

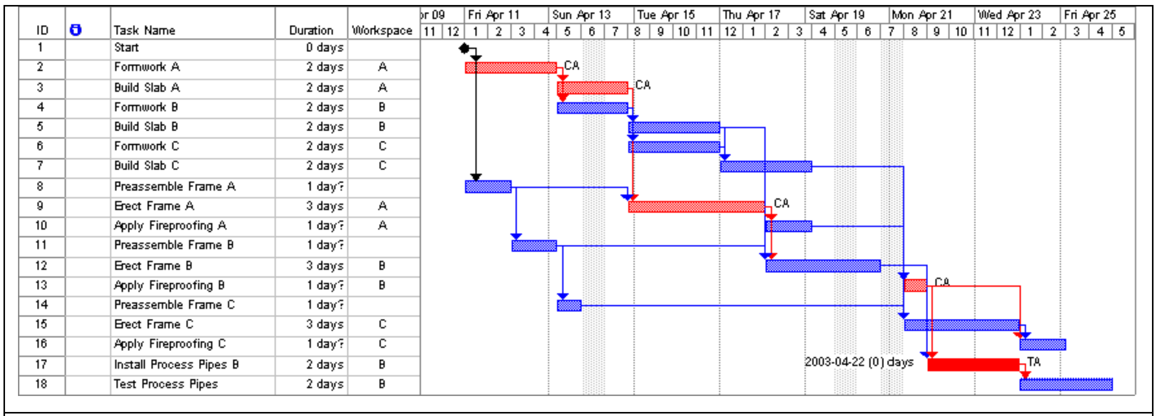


ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Formwork A	2 days	A
3		Build Slab A	2 days	A
4		Formwork B	2 days	B
5		Build Slab B	2 days	B
6		Formwork C	2 days	C
7		Build Slab C	2 days	C
8		Preassemble Frame A	1 day?	
9		Erect Frame A	3 days	A
10		Apply Fireproofing A	1 day?	A
11		Preassemble Frame B	1 day?	
12		Erect Frame B	3 days	B
13		Apply Fireproofing B	1 day?	B
14		Preassemble Frame C	1 day?	
15		Erect Frame C	3 days	C
16		Apply Fireproofing C	1 day?	C
17		Install Process Pipes B	2 days	B
18		Test Process Pipes	2 days	B

Figure B.7: Enabling activities for the Intel CUB schedule with respect to the target activity Install Process Pipes B.

Project	Activity ID	Role (Enabling)	Confirmed
Intel CUB	**4**	Formwork B	Yes
	5	Build Slab B	Yes
	11	Preassemble Frame B	Yes
	12	Erect Frame B	Yes
	17	Install Process Pipes B	Yes

Table B.15: List of enabling activities for the Intel CUB schedule.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Formwork A	2 days	A
3		Build Slab A	2 days	A
4		Formwork B	2 days	B
5		Build Slab B	2 days	B
6		Formwork C	2 days	C
7		Build Slab C	2 days	C
8		Preassemble Frame A	1 day?	
9		Erect Frame A	3 days	A
10		Apply Fireproofing A	1 day?	A
11		Preassemble Frame B	1 day?	
12		Erect Frame B	3 days	B
13		Apply Fireproofing B	1 day?	B
14		Preassemble Frame C	1 day?	
15		Erect Frame C	3 days	C
16		Apply Fireproofing C	1 day?	C
17		Install Process Pipes B	2 days	B
18		Test Process Pipes	2 days	B

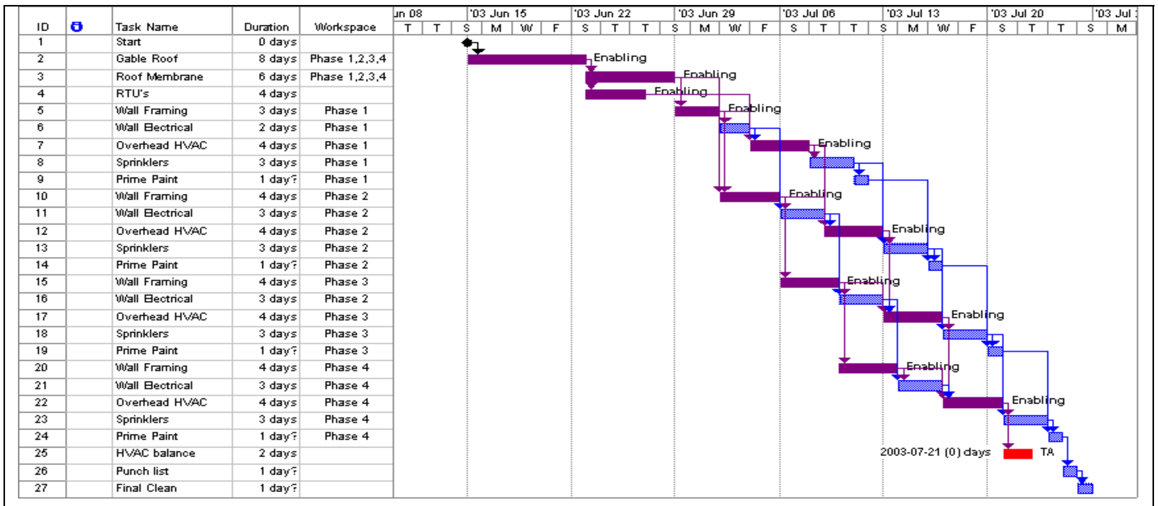
Figure B.8: Candidate activities (CA) for the Intel CUB schedule with respect to the target activity Install Process Pipes B.

Project	Activity ID	CA's (Candidate Activities)	Confirmed
Intel CUB	**2**	Formwork A	Yes
	3	Build Slab A	Yes
	9	Erect Frame A	Yes
	13	Apply Fireproofing B	Yes

Table B.16: List of candidate activities for the Intel CUB schedule.

B.3.2 McWhinney Project Schedule

Figure B.9 and Table B.17 show the enabling activities identified for the McWhinney schedule. Figure B.10 and Table B.18 show the candidate activities identified for the McWhinney schedule.

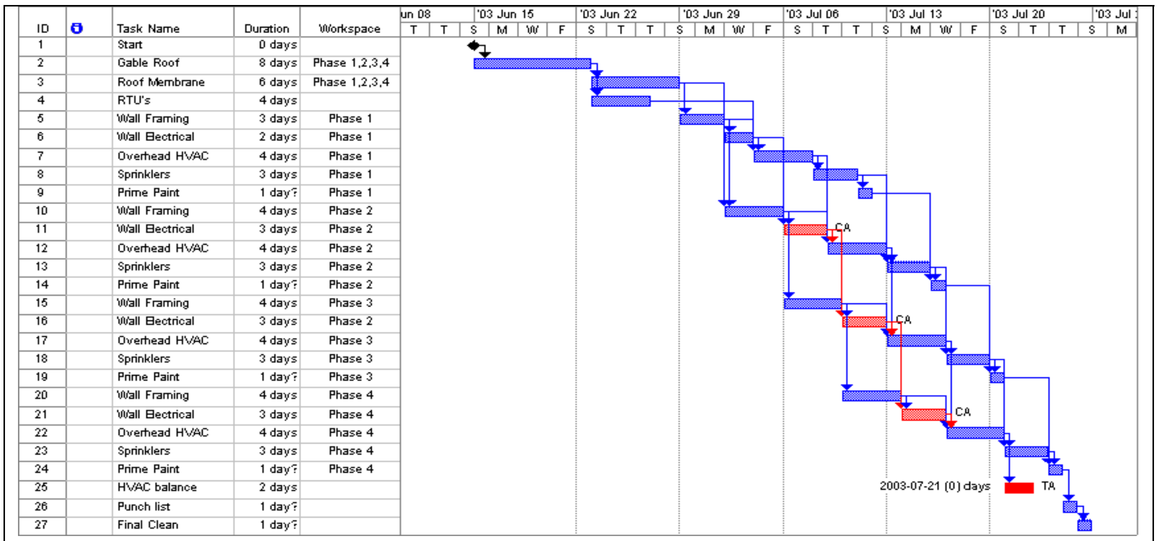


ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Gable Roof	8 days	Phase 1,2,3,4
3		Roof Membrane	6 days	Phase 1,2,3,4
4		RTU's	4 days	
5		Wall Framing	3 days	Phase 1
6		Wall Electrical	2 days	Phase 1
7		Overhead HVAC	4 days	Phase 1
8		Sprinklers	3 days	Phase 1
9		Prime Paint	1 day?	Phase 1
10		Wall Framing	4 days	Phase 2
11		Wall Electrical	3 days	Phase 2
12		Overhead HVAC	4 days	Phase 2
13		Sprinklers	3 days	Phase 2
14		Prime Paint	1 day?	Phase 2
15		Wall Framing	4 days	Phase 3
16		Wall Electrical	3 days	Phase 2
17		Overhead HVAC	4 days	Phase 3
18		Sprinklers	3 days	Phase 3
19		Prime Paint	1 day?	Phase 3
20		Wall Framing	4 days	Phase 4
21		Wall Electrical	3 days	Phase 4
22		Overhead HVAC	4 days	Phase 4
23		Sprinklers	3 days	Phase 4
24		Prime Paint	1 day?	Phase 4
25		HVAC balance	2 days	
26		Punch list	1 day?	
27		Final Clean	1 day?	

Figure B.9: Enabling activities for the McWhinney schedule with respect to the target activity HVAC Balance.

Project	Activity ID	Role (Enabling)	Confirmed
McWhinney	**2**	Gable Roof – Phase 1,2,3,4	Yes
	3	Roof Membrane – Phase 1,2,3,4	Yes
	4	RTU's	Yes
	5	Wall Framing – Phase 1	Yes
	7	Overhead HVAC – Phase 1	Yes
	10	Wall Framing – Phase 2	Yes
	12	Overhead HVAC – Phase 2	Yes
	15	Wall Framing – Phase 3	Yes
	17	Overhead HVAC – Phase 3	Yes
	20	Wall Framing – Phase 4	Yes
	22	Overhead HVAC – Phase 4	Yes

Table B.17: List of enabling activities for the McWhinney project schedule.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Gable Roof	8 days	Phase 1,2,3,4
3		Roof Membrane	6 days	Phase 1,2,3,4
4		RTU's	4 days	
5		Wall Framing	3 days	Phase 1
6		Wall Electrical	2 days	Phase 1
7		Overhead HVAC	4 days	Phase 1
8		Sprinklers	3 days	Phase 1
9		Prime Paint	1 day?	Phase 1
10		Wall Framing	4 days	Phase 2
11		Wall Electrical	3 days	Phase 2
12		Overhead HVAC	4 days	Phase 2
13		Sprinklers	3 days	Phase 2
14		Prime Paint	1 day?	Phase 2
15		Wall Framing	4 days	Phase 3
16		Wall Electrical	3 days	Phase 2
17		Overhead HVAC	4 days	Phase 3
18		Sprinklers	3 days	Phase 3
19		Prime Paint	1 day?	Phase 3
20		Wall Framing	4 days	Phase 4
21		Wall Electrical	3 days	Phase 4
22		Overhead HVAC	4 days	Phase 4
23		Sprinklers	3 days	Phase 4
24		Prime Paint	1 day?	Phase 4
25		HVAC balance	2 days	
26		Punch list	1 day?	
27		Final Clean	1 day?	

Figure B.10: Candidate activities (CA) for the McWhinney schedule with respect to the target activity HVAC Balance.

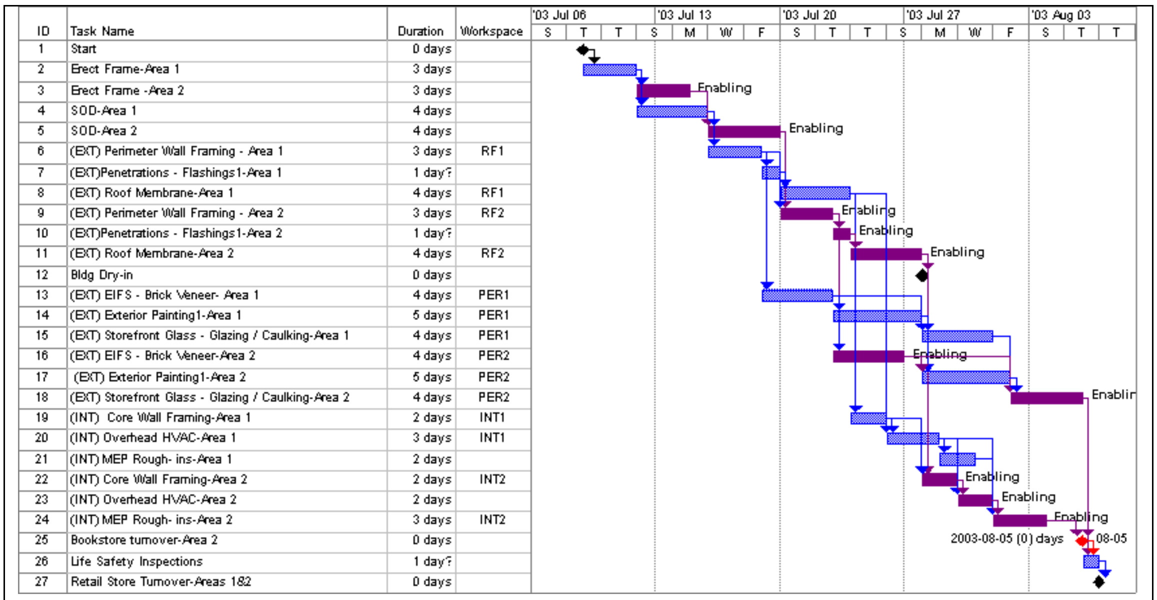
Results show that the earlier execution of the activity HVAC Balance requires expediting the activities Overhead HVAC in each phase, and expediting these activities in turn requires delaying the activities Wall Electrical in phases 2, 3 and 4. (Table B.18)

Project	Activity ID	CA (Candidate Activity)	Confirmed
McWhinney	**11**	Wall Electrical – Phase 2	Yes
	16	Wall Electrical – Phase 3	Yes
	21	Wall Electrical – Phase 4	Yes

Table B.18: List of candidate activities for the McWhinney project schedule.

B.3.3 Bay Street Project Schedule

Figure B.11 and Table B.19 show the enabling activities identified for the Bay Street project schedule. Figure B.12 and Table B.20 show the candidate activities identified for the Bay Street project schedule.

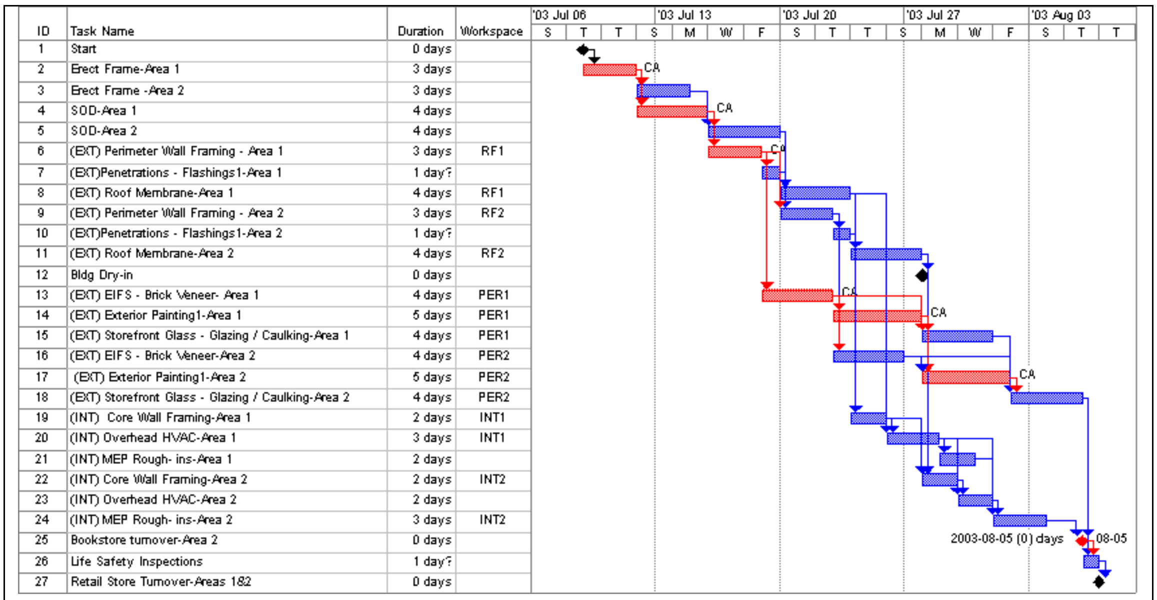


ID	Task Name	Duration	Workspace
1	Start	0 days	
2	Erect Frame-Area 1	3 days	
3	Erect Frame -Area 2	3 days	
4	SOD-Area 1	4 days	
5	SOD-Area 2	4 days	
6	(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7	(EXT)Penetrations - Flashings1-Area 1	1 day?	
8	(EXT) Roof Membrane-Area 1	4 days	RF1
9	(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10	(EXT)Penetrations - Flashings1-Area 2	1 day?	
11	(EXT) Roof Membrane-Area 2	4 days	RF2
12	Bldg Dry-in	0 days	
13	(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14	(EXT) Exterior Painting1-Area 1	5 days	PER1
15	(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16	(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17	(EXT) Exterior Painting1-Area 2	5 days	PER2
18	(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19	(INT) Core Wall Framing-Area 1	2 days	INT1
20	(INT) Overhead HVAC-Area 1	3 days	INT1
21	(INT) MEP Rough- ins-Area 1	2 days	
22	(INT) Core Wall Framing-Area 2	2 days	INT2
23	(INT) Overhead HVAC-Area 2	2 days	
24	(INT) MEP Rough- ins-Area 2	3 days	INT2
25	Bookstore turnover-Area 2	0 days	
26	Life Safety Inspections	1 day?	
27	Retail Store Turnover-Areas 1&2	0 days	

Figure B.11: Enabling activities for the Bay Street project schedule with respect to the target activity Bookstore turnover-Area 2.

Project	Activity ID	Role (Enabling)	Confirmed
Bay Street	3	Erect Frame -Area 2	Yes
	5	SOD-Area 2	Yes
	9	(EXT) Perimeter Wall Framing - Area 2	Yes
	10	(EXT)Penetrations - Flashings1-Area 2	Yes
	11	(EXT) Roof Membrane-Area 2	Yes
	16	(EXT) EIFS - Brick Veneer-Area 2	Yes
	18	(EXT) Storefront Glass - Glazing / Caulking-Area 2	Yes
	22	(INT) Core Wall Framing-Area 2	Yes
	23	(INT) Overhead HVAC-Area 2	Yes
	24	(INT) MEP Rough- ins-Area 2	Yes

Table B.19: List of enabling activities for the Bay Street project schedule.



ID	Task Name	Duration	Workspace
1	Start	0 days	
2	Erect Frame-Area 1	3 days	
3	Erect Frame -Area 2	3 days	
4	SOD-Area 1	4 days	
5	SOD-Area 2	4 days	
6	(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7	(EXT)Penetrations - Flashings1-Area 1	1 day?	
8	(EXT) Roof Membrane-Area 1	4 days	RF1
9	(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10	(EXT)Penetrations - Flashings1-Area 2	1 day?	
11	(EXT) Roof Membrane-Area 2	4 days	RF2
12	Bldg Dry-in	0 days	
13	(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14	(EXT) Exterior Painting1-Area 1	5 days	PER1
15	(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16	(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17	(EXT) Exterior Painting1-Area 2	5 days	PER2
18	(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19	(INT) Core Wall Framing-Area 1	2 days	INT1
20	(INT) Overhead HVAC-Area 1	3 days	INT1
21	(INT) MEP Rough- ins-Area 1	2 days	
22	(INT) Core Wall Framing-Area 2	2 days	INT2
23	(INT) Overhead HVAC-Area 2	2 days	
24	(INT) MEP Rough- ins-Area 2	3 days	INT2
25	Bookstore turnover-Area 2	0 days	
26	Life Safety Inspections	1 day?	
27	Retail Store Turnover-Areas 1&2	0 days	

Figure B.12: Candidate activities (CA) for the Bay Street project schedule with respect to the target activity Bookstore turnover-Area 2.

Results show that the earlier execution of the activity Bookstore Turnover-Area 2 can be achieved by delaying the activities Erect Frame, SOD, Perimeter Wall Framing, EIFS Brick Veneer, Exterior Painting1 in Area 1 and Exterior Panting1 in Area 2 (Table B.17).

Project	Activity ID	CA (Candidate Activity)	Confirmed
Bay Street	2	Erect Frame-Area 1	Yes
	4	SOD-Area 1	Yes
	6	(EXT) Perimeter Wall Framing - Area 1	Yes
	13	(EXT) EIFS - Brick Veneer- Area 1	Yes
	14	(EXT) Exterior Painting1-Area 1	Yes
	17	(EXT) Exterior Painting1-Area 2	Yes

Table B.20: List of candidate activities for the Bay Street project schedule.

This section described the results for using the identification process to identify enabling and candidate activities for the three project schedules. The following section describes the results of delaying the candidate activities to expedite the target activities for the three project schedules.

B.4 ACTIVITY RE-SEQUENCING RESULTS

As discussed in Section 4.4.1 of Chapter 4, I used CLCPM to re-sequence the candidate activities of the three project schedules. Table B.21 shows the final TA date, and the number of sequencing alternatives created by expediting all candidate activities.

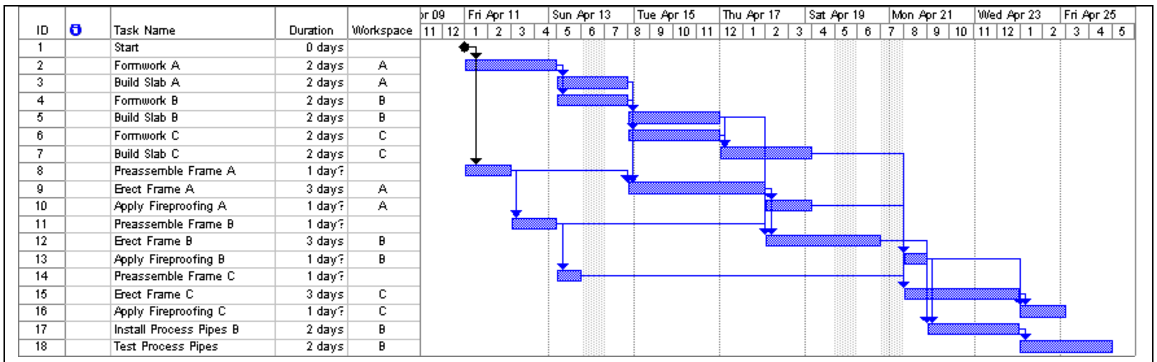
Project	TA initial date	TA final expedited date	Number of sequencing alternatives	Confirmed
Intel CUB	4/22	4/18(-4 days)	3	3
McWhinney	7/21	7/15(-6 days)	2	2
Bay Street	8/05	7/29(-7 days)	6	6

Table B.21: Validation results of the re-sequencing process for the three retrospective cases.

The following section presents the sequencing alternatives developed and confirmed for each project schedule.

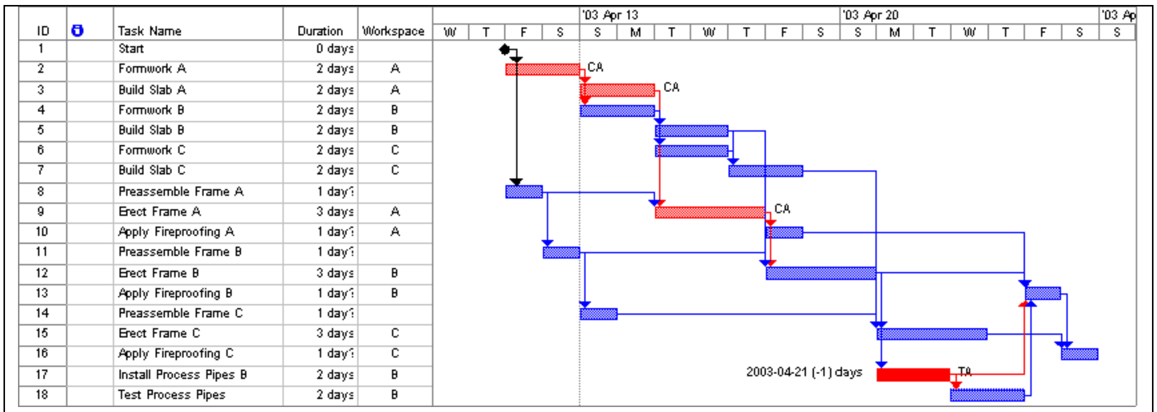
B.4.1 Intel CUB Schedule

Figures B.13 shows the initial Intel CUB schedule. Figures B.14 through B.16 show the 4 alternatives developed using CLCPM.



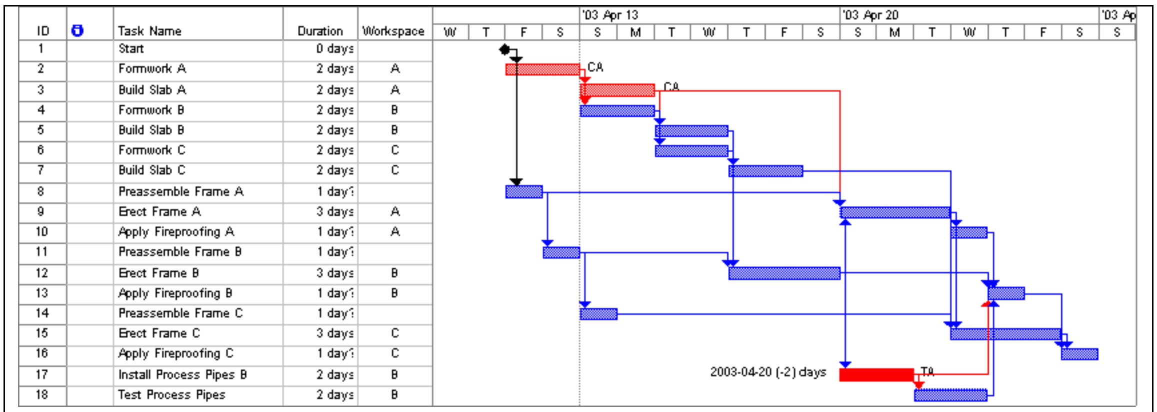
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Formwork A	2 days	A
3		Build Slab A	2 days	A
4		Formwork B	2 days	B
5		Build Slab B	2 days	B
6		Formwork C	2 days	C
7		Build Slab C	2 days	C
8		Preassemble Frame A	1 day?	
9		Erect Frame A	3 days	A
10		Apply Fireproofing A	1 day?	A
11		Preassemble Frame B	1 day?	
12		Erect Frame B	3 days	B
13		Apply Fireproofing B	1 day?	B
14		Preassemble Frame C	1 day?	
15		Erect Frame C	3 days	C
16		Apply Fireproofing C	1 day?	C
17		Install Process Pipes B	2 days	B
18		Test Process Pipes	2 days	B

Figure B.13: Intel CUB schedule.



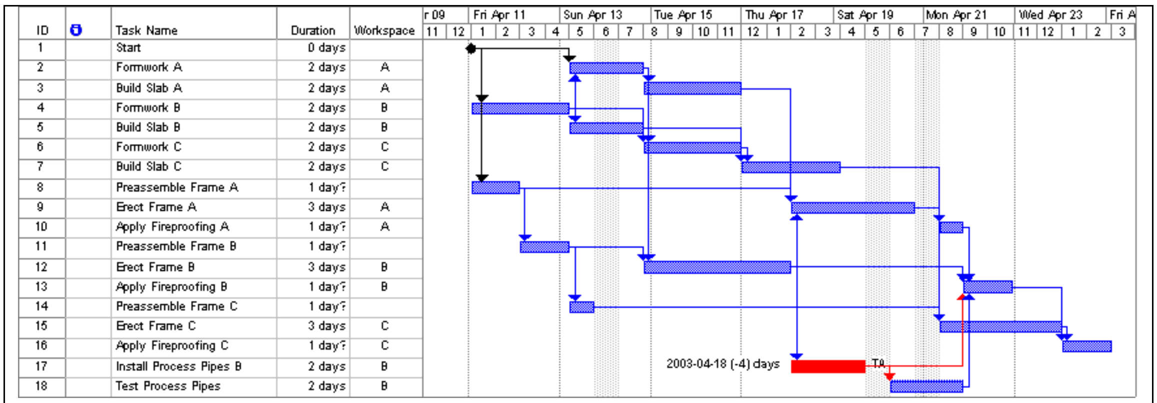
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Formwork A	2 days	A
3		Build Slab A	2 days	A
4		Formwork B	2 days	B
5		Build Slab B	2 days	B
6		Formwork C	2 days	C
7		Build Slab C	2 days	C
8		Preassemble Frame A	1 day?	
9		Erect Frame A	3 days	A
10		Apply Fireproofing A	1 day?	A
11		Preassemble Frame B	1 day?	
12		Erect Frame B	3 days	B
13		Apply Fireproofing B	1 day?	B
14		Preassemble Frame C	1 day?	
15		Erect Frame C	3 days	C
16		Apply Fireproofing C	1 day?	C
17		Install Process Pipes B	2 days	B
18		Test Process Pipes	2 days	B

Figure B.14: Alternative 1.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Formwork A	2 days	A
3		Build Slab A	2 days	A
4		Formwork B	2 days	B
5		Build Slab B	2 days	B
6		Formwork C	2 days	C
7		Build Slab C	2 days	C
8		Preassemble Frame A	1 day?	
9		Erect Frame A	3 days	A
10		Apply Fireproofing A	1 day?	A
11		Preassemble Frame B	1 day?	
12		Erect Frame B	3 days	B
13		Apply Fireproofing B	1 day?	B
14		Preassemble Frame C	1 day?	
15		Erect Frame C	3 days	C
16		Apply Fireproofing C	1 day?	C
17		Install Process Pipes B	2 days	B
18		Test Process Pipes	2 days	B

Figure B.15: Alternative 2.

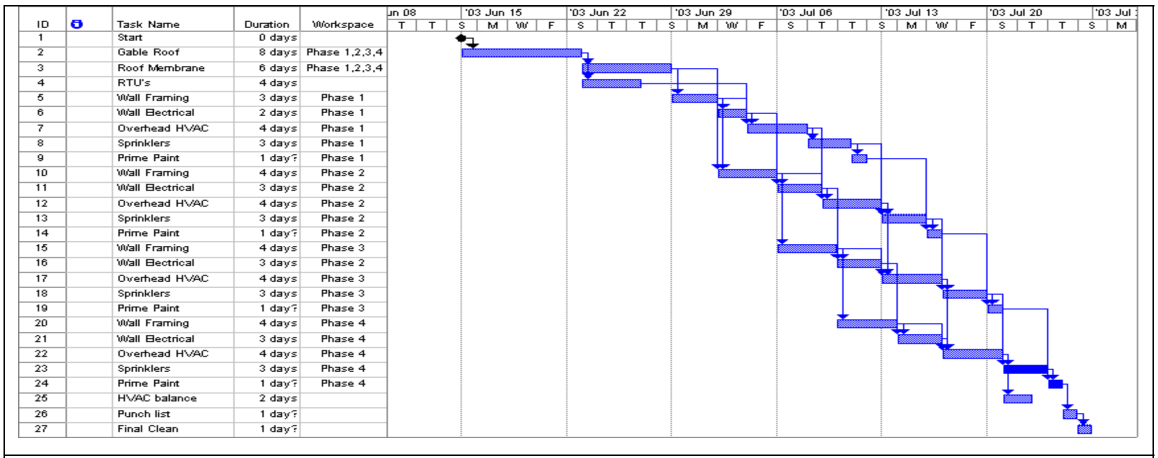


ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Formwork A	2 days	A
3		Build Slab A	2 days	A
4		Formwork B	2 days	B
5		Build Slab B	2 days	B
6		Formwork C	2 days	C
7		Build Slab C	2 days	C
8		Preassemble Frame A	1 day?	
9		Erect Frame A	3 days	A
10		Apply Fireproofing A	1 day?	A
11		Preassemble Frame B	1 day?	
12		Erect Frame B	3 days	B
13		Apply Fireproofing B	1 day?	B
14		Preassemble Frame C	1 day?	
15		Erect Frame C	3 days	C
16		Apply Fireproofing C	1 day?	C
17		Install Process Pipes B	2 days	B
18		Test Process Pipes	2 days	B

Figure B.16: Alternative 3.

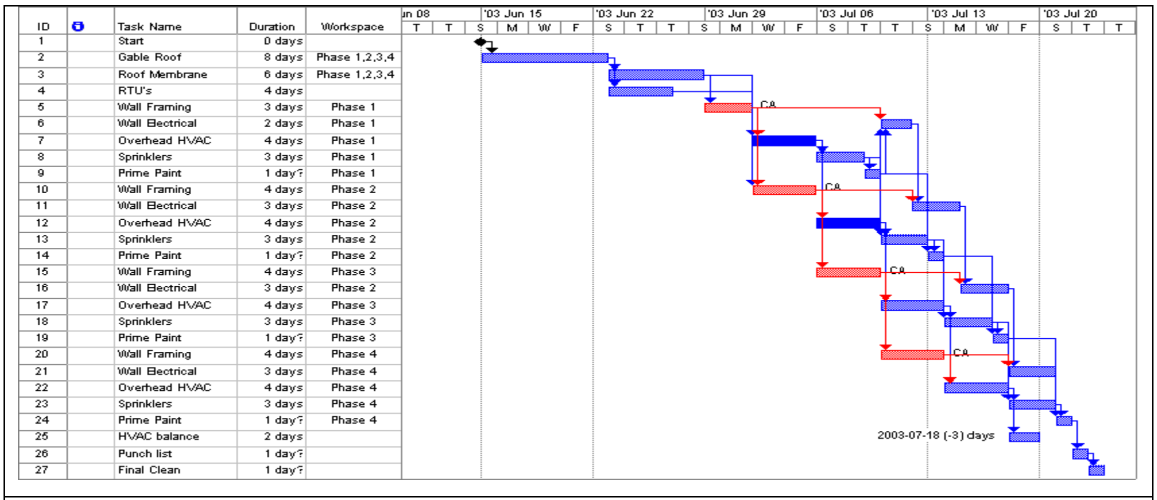
B.4.2 McWhinney Project Schedule

Figures B.17 shows the initial McWhinney project schedule. Figures B.18 through B.19 show the 2 alternatives developed using CLCPM.



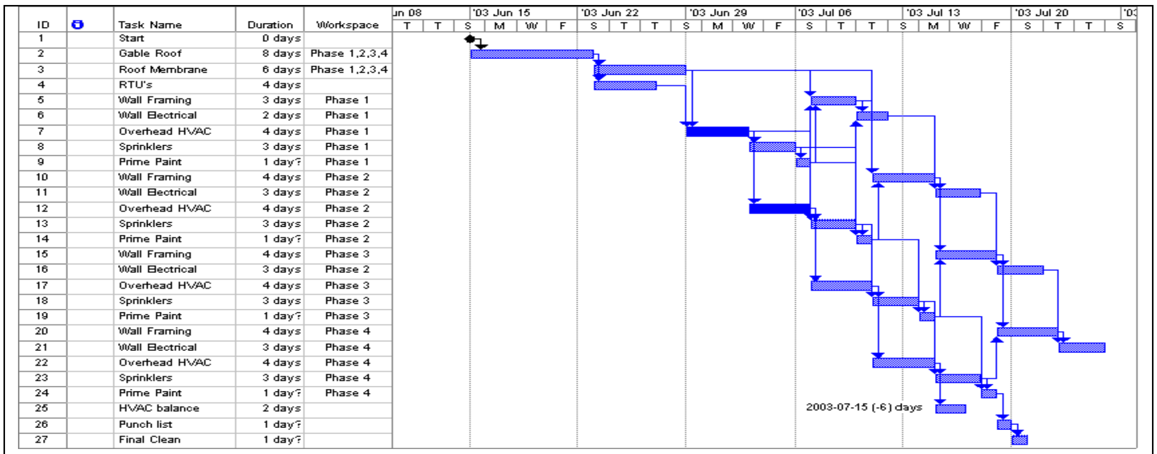
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Gable Roof	8 days	Phase 1,2,3,4
3		Roof Membrane	6 days	Phase 1,2,3,4
4		RTU's	4 days	
5		Wall Framing	3 days	Phase 1
6		Wall Electrical	2 days	Phase 1
7		Overhead HVAC	4 days	Phase 1
8		Sprinklers	3 days	Phase 1
9		Prime Paint	1 day?	Phase 1
10		Wall Framing	4 days	Phase 2
11		Wall Electrical	3 days	Phase 2
12		Overhead HVAC	4 days	Phase 2
13		Sprinklers	3 days	Phase 2
14		Prime Paint	1 day?	Phase 2
15		Wall Framing	4 days	Phase 3
16		Wall Electrical	3 days	Phase 2
17		Overhead HVAC	4 days	Phase 3
18		Sprinklers	3 days	Phase 3
19		Prime Paint	1 day?	Phase 3
20		Wall Framing	4 days	Phase 4
21		Wall Electrical	3 days	Phase 4
22		Overhead HVAC	4 days	Phase 4
23		Sprinklers	3 days	Phase 4
24		Prime Paint	1 day?	Phase 4
25		HVAC balance	2 days	
26		Punch list	1 day?	
27		Final Clean	1 day?	

Figure B.17: McWhinney project schedule.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Gable Roof	8 days	Phase 1,2,3,4
3		Roof Membrane	6 days	Phase 1,2,3,4
4		RTU's	4 days	
5		Wall Framing	3 days	Phase 1
6		Wall Electrical	2 days	Phase 1
7		Overhead HVAC	4 days	Phase 1
8		Sprinklers	3 days	Phase 1
9		Prime Paint	1 day?	Phase 1
10		Wall Framing	4 days	Phase 2
11		Wall Electrical	3 days	Phase 2
12		Overhead HVAC	4 days	Phase 2
13		Sprinklers	3 days	Phase 2
14		Prime Paint	1 day?	Phase 2
15		Wall Framing	4 days	Phase 3
16		Wall Electrical	3 days	Phase 2
17		Overhead HVAC	4 days	Phase 3
18		Sprinklers	3 days	Phase 3
19		Prime Paint	1 day?	Phase 3
20		Wall Framing	4 days	Phase 4
21		Wall Electrical	3 days	Phase 4
22		Overhead HVAC	4 days	Phase 4
23		Sprinklers	3 days	Phase 4
24		Prime Paint	1 day?	Phase 4
25		HVAC balance	2 days	
26		Punch list	1 day?	
27		Final Clean	1 day?	

Figure B.18: Alternative 1.

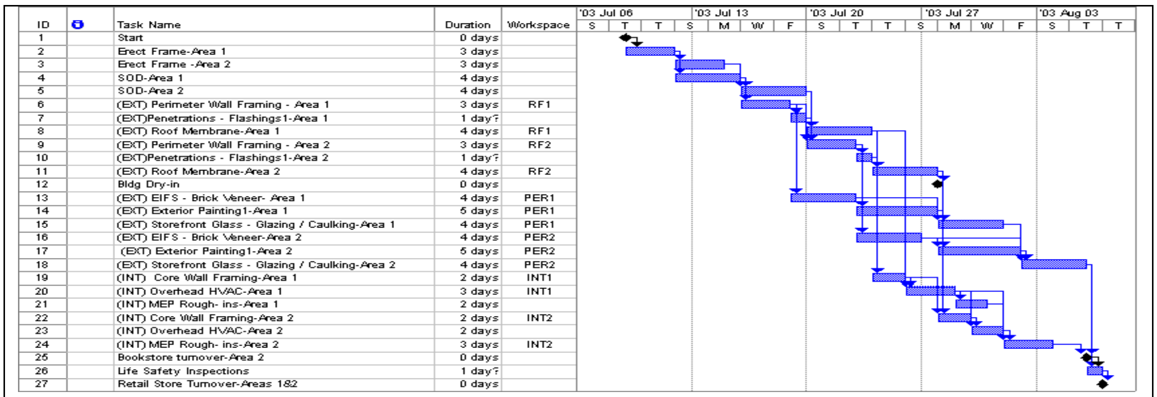


ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Gable Roof	8 days	Phase 1,2,3,4
3		Roof Membrane	6 days	Phase 1,2,3,4
4		RTU's	4 days	
5		Wall Framing	3 days	Phase 1
6		Wall Electrical	2 days	Phase 1
7		Overhead HVAC	4 days	Phase 1
8		Sprinklers	3 days	Phase 1
9		Prime Paint	1 day?	Phase 1
10		Wall Framing	4 days	Phase 2
11		Wall Electrical	3 days	Phase 2
12		Overhead HVAC	4 days	Phase 2
13		Sprinklers	3 days	Phase 2
14		Prime Paint	1 day?	Phase 2
15		Wall Framing	4 days	Phase 3
16		Wall Electrical	3 days	Phase 2
17		Overhead HVAC	4 days	Phase 3
18		Sprinklers	3 days	Phase 3
19		Prime Paint	1 day?	Phase 3
20		Wall Framing	4 days	Phase 4
21		Wall Electrical	3 days	Phase 4
22		Overhead HVAC	4 days	Phase 4
23		Sprinklers	3 days	Phase 4
24		Prime Paint	1 day?	Phase 4
25		HVAC balance	2 days	
26		Punch list	1 day?	
27		Final Clean	1 day?	

Figure B.19: Alternative 2.

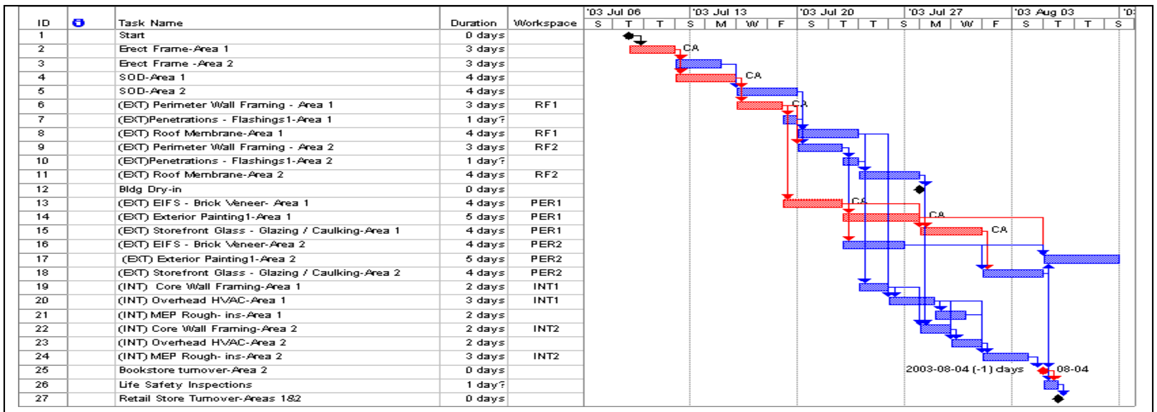
B.4.3 Bay Street Project Schedule

Figures B.20 shows the initial Bay Street project schedule. Figures B.21 through B.26 show the 6 alternatives developed using CLCPM.



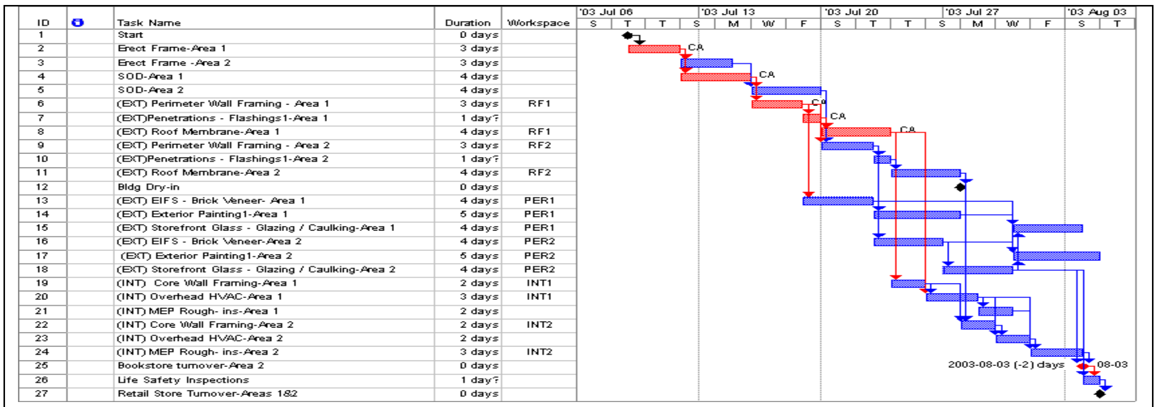
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.20: Bay Street project schedule.



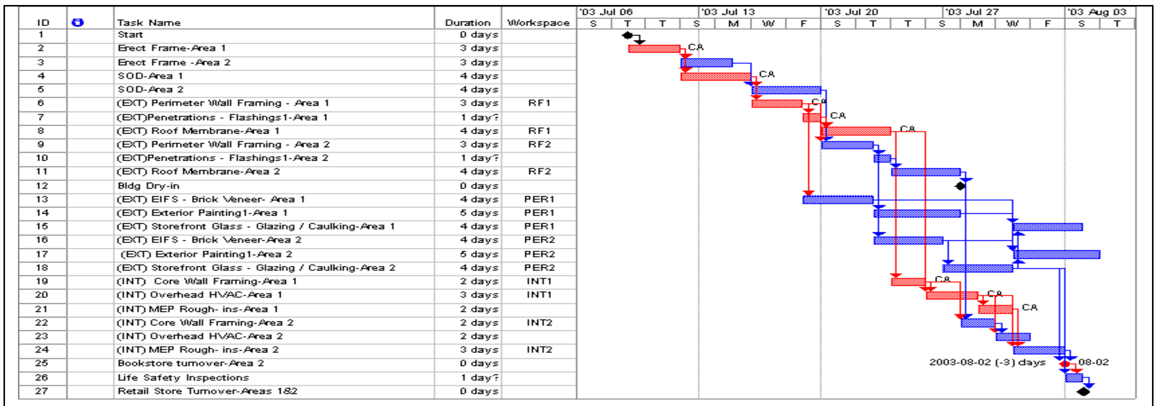
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.21: Alternative 1.



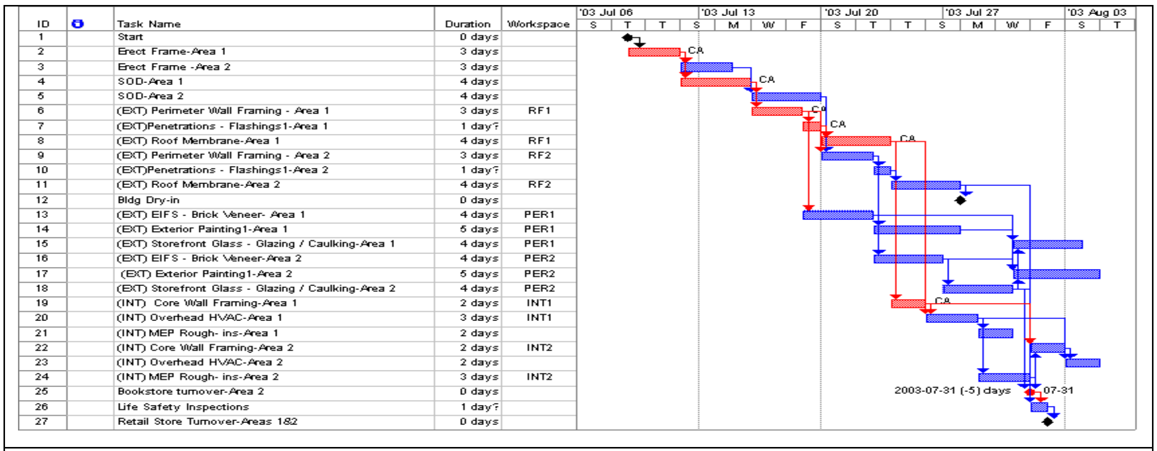
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.22: Alternative 2.



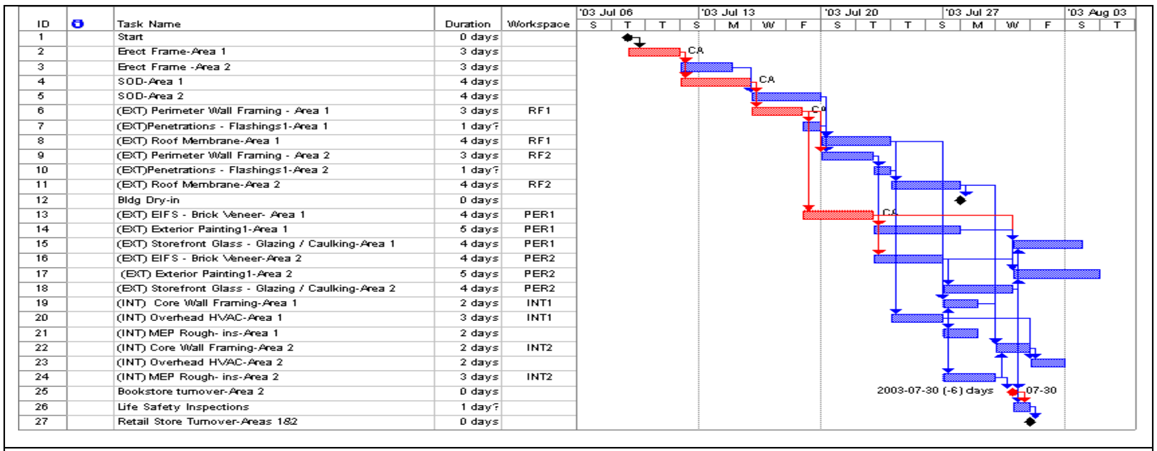
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.23: Alternative 3.



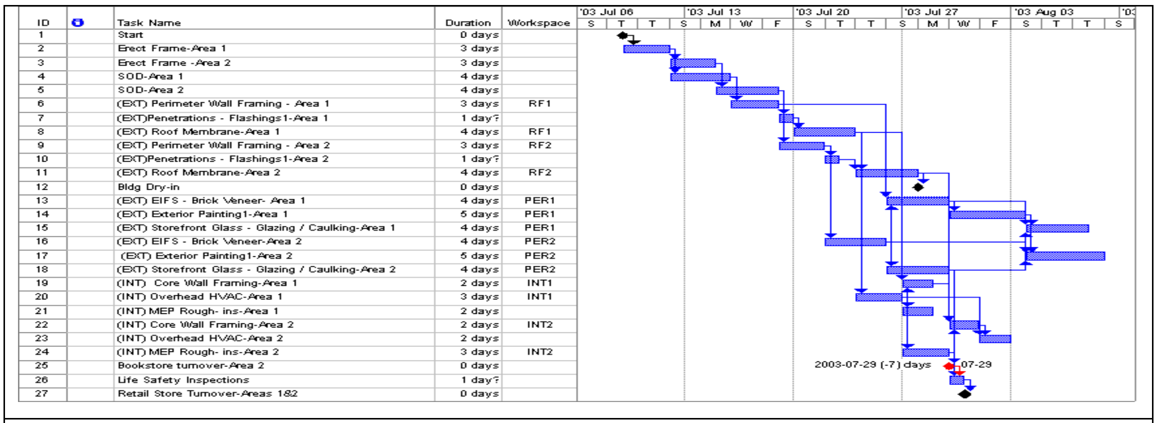
ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.24: Alternative 4.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.25: Alternative 5.



ID		Task Name	Duration	Workspace
1		Start	0 days	
2		Erect Frame-Area 1	3 days	
3		Erect Frame -Area 2	3 days	
4		SOD-Area 1	4 days	
5		SOD-Area 2	4 days	
6		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1
7		(EXT)Penetrations - Flashings1-Area 1	1 day?	
8		(EXT) Roof Membrane-Area 1	4 days	RF1
9		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2
10		(EXT)Penetrations - Flashings1-Area 2	1 day?	
11		(EXT) Roof Membrane-Area 2	4 days	RF2
12		Bldg Dry-in	0 days	
13		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1
14		(EXT) Exterior Painting1-Area 1	5 days	PER1
15		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1
16		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2
17		(EXT) Exterior Painting1-Area 2	5 days	PER2
18		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2
19		(INT) Core Wall Framing-Area 1	2 days	INT1
20		(INT) Overhead HVAC-Area 1	3 days	INT1
21		(INT) MEP Rough- ins-Area 1	2 days	
22		(INT) Core Wall Framing-Area 2	2 days	INT2
23		(INT) Overhead HVAC-Area 2	2 days	
24		(INT) MEP Rough- ins-Area 2	3 days	INT2
25		Bookstore turnover-Area 2	0 days	
26		Life Safety Inspections	1 day?	
27		Retail Store Turnover-Areas 1&2	0 days	

Figure B.26: Alternative 6.

APPENDIX C. CHARRETTE TEST QUESTIONNAIRE

Appendix C.1 and C.2 shows the questions used for the charrette test using the Intel CUB schedule and the Bay Street project schedule, respectively.

C.1 CHARRETTE QUESTIONNAIRE FOR INTEL CUB PROJECT

NAME:

TOOL: **<u>CLCPM</u>** **<u>Conventional</u>**

Project Overview

Figure 1 shows a 3D model view of the Intel Central Utility Building (CUB). The schedule provided (Figure 2) describes the sequence of trades performing the work for foundation, structural frame and process pipes of the building. The project manager has sectioned the CUB building into three major zones (i.e., A, B and C) for planning and coordination purposes.

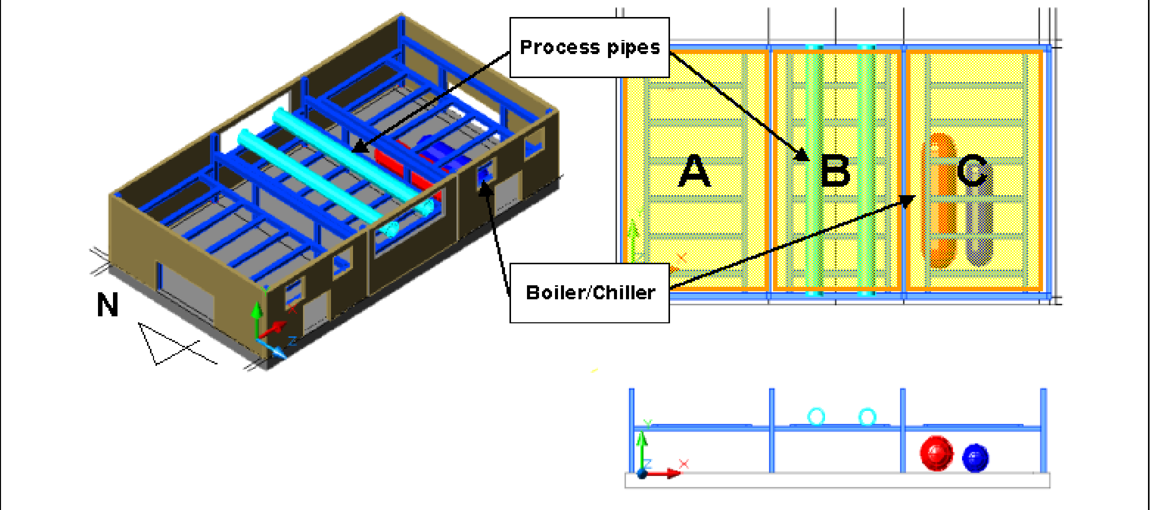
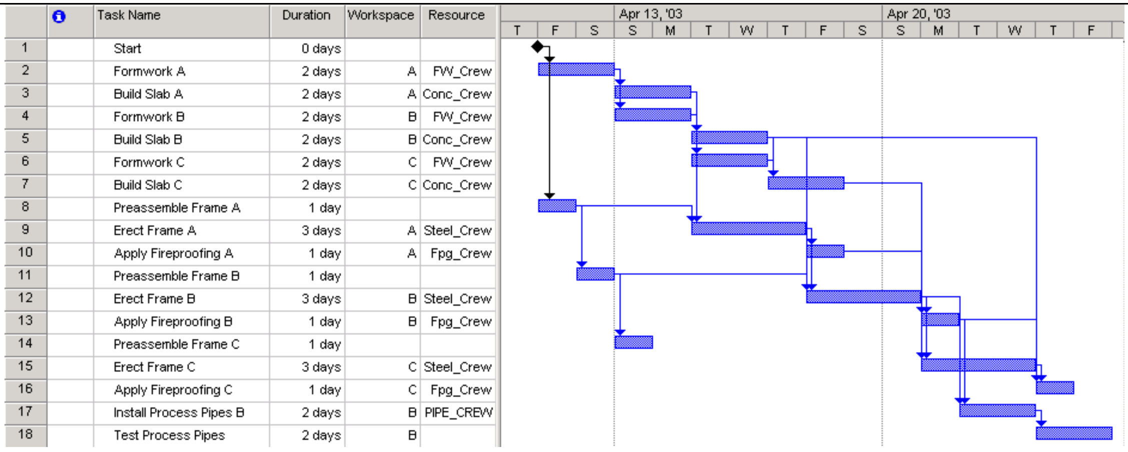


Figure 1: Overview diagram of Central Utility Building (CUB): A, B and C denote the zones sectioned for planning and coordination purposes.



	Task Name	Duration	Workspace	Resource
1	Start	0 days		
2	Formwork A	2 days	A	FW_Crew
3	Build Slab A	2 days	A	Conc_Crew
4	Formwork B	2 days	B	FW_Crew
5	Build Slab B	2 days	B	Conc_Crew
6	Formwork C	2 days	C	FW_Crew
7	Build Slab C	2 days	C	Conc_Crew
8	Preassemble Frame A	1 day		
9	Erect Frame A	3 days	A	Steel_Crew
10	Apply Fireproofing A	1 day	A	Fpg_Crew
11	Preassemble Frame B	1 day		
12	Erect Frame B	3 days	B	Steel_Crew
13	Apply Fireproofing B	1 day	B	Fpg_Crew
14	Preassemble Frame C	1 day		
15	Erect Frame C	3 days	C	Steel_Crew
16	Apply Fireproofing C	1 day	C	Fpg_Crew
17	Install Process Pipes B	2 days	B	PIPE_CREW
18	Test Process Pipes	2 days	B	

Figure 2: Intel CUB schedule.

Based on this information please answer the following questions:

1. Rationale Questions

For the following 5 questions, please describe, in your own words, the rationale for the precedence relationship between (e.g., activity Erect Beam is "supported by" activity Erect Columns, activity Interior Wall framing is "protected by" activity Erect Roof):

1.1 The activity Install Process Pipes B (ID: 17) and activity Erect Frame B (ID: 12).

1.2 The activity Install Process Pipes B (ID: 17) and the activity Apply Fireproofing B (ID: 13).

1.3 The activity Erect Frame B (ID: 12) and the activity Erect Frame A (ID: 9).

1.4 The activity Erect Frame B (ID: 12) and the activity Build Slab B (ID: 5).

1.5 The activity Apply Fireproofing B (ID: 13) and the activity Apply Fireproofing A (ID: 10).

For the following 4 questions, please select ONE answer ONLY:

1.6. Which of the precedence relationships between the following activities can be relaxed if required?

Precedence relationship between:

a. The activity Install Process Pipes B (ID: 17) and the activity Erect Frame B (ID: 12).
b. The activity Install Process Pipes B (ID: 17) and the activity Apply Fireproofing B (ID: 13).
c. The activity Erect Frame B (ID: 12) and the activity Build Slab B (ID: 5).

1.7. Which of the precedence relationships between the following activities do you think is relatively easiest to relax?

Precedence relationship between:

a. The activity Erect Frame B (ID: 12) and the activity Erect Frame A (ID: 9).
b. The activity Apply Fireproofing B (ID: 13) and the activity Apply Fireproofing A (ID: 10).
c. The activity Install Process Pipes B (ID: 17) and the activity Apply Fireproofing B (ID: 13).

1.8 Which of the following precedence relationships can be considered to be physically “assisting” or “enabling”?

Between the activity Erect Frame B (ID: 12) and:

a. The activity Install Process Pipes B (ID: 17).
b. The activity Erect Frame C (ID: 15).
c. The activity Apply Fireproofing B (ID: 13).

1.9 Which of the following precedence relationships can be considered NOT to be physically “assisting” or “enabling”?

Between the activity Build Slab B (ID: 12) and:

a. The activity Build Slab B (ID: 5).
b. The activity Erect Frame A (ID: 9).
c. The activity Preassemble Frame B (ID: 11).

2. Logical integrity questions

For the following 2 questions, please select ONE answer ONLY:

2.1 In the schedule provided, there exists a redundant precedence relationship. Which of the following relationships do you think is redundant?

Precedence relationship between:

a. The activity Apply Fireproofing C (ID: 16) and the activity Build Slab (ID: 5).

b. The activity Apply Fireproofing C (ID: 16) and the activity Erect Frame C (ID: 15).

c. The activity Apply Fireproofing C (ID: 16) and the activity Apply Fireproofing B (ID: 13).

2.2 In the schedule provided, a precedence relationship is missing. Which of the following two activities do you think are missing a relationship?

a. The activity Erect Frame C (ID: 15) and the activity Formwork A (ID: 2).

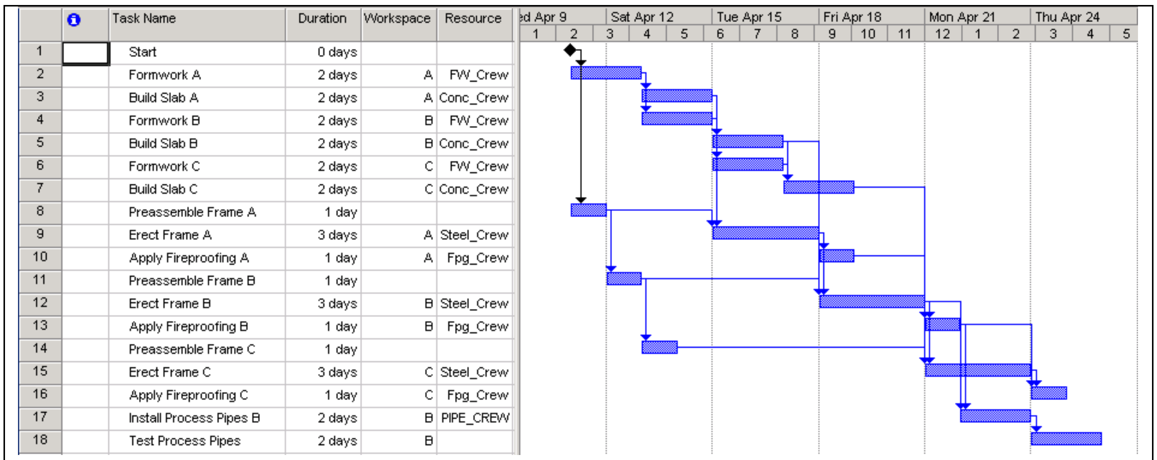
b. The activity Install Process Pipes B (ID: 17) and the activity Erect Frame (ID: 15).

c. The activity Erect Frame C (ID: 15) and the activity Preassemble Frame C (ID: 14).

3. Questions related to the identification process

STOP!

- **Please ask for schedule with correct sequences and precedence relationships (Figure 3).**
- **Please document start and finish times for answering each question.**
- **Please do not take more time than specified for each question.**



	ℹ	Task Name	Duration	Workspace	Resource
1		Start	0 days		
2		Formwork A	2 days	A	FW_Crew
3		Build Slab A	2 days	A	Conc_Crew
4		Formwork B	2 days	B	FW_Crew
5		Build Slab B	2 days	B	Conc_Crew
6		Formwork C	2 days	C	FW_Crew
7		Build Slab C	2 days	C	Conc_Crew
8		Preassemble Frame A	1 day		
9		Erect Frame A	3 days	A	Steel_Crew
10		Apply Fireproofing A	1 day	A	Fpg_Crew
11		Preassemble Frame B	1 day		
12		Erect Frame B	3 days	B	Steel_Crew
13		Apply Fireproofing B	1 day	B	Fpg_Crew
14		Preassemble Frame C	1 day		
15		Erect Frame C	3 days	C	Steel_Crew
16		Apply Fireproofing C	1 day	C	Fpg_Crew
17		Install Process Pipes B	2 days	B	PIPE_CREW
18		Test Process Pipes	2 days	B	

Figure 3: Intel CUB schedule.

Situation Overview

On viewing the schedule, the project manager determined that the process pipes (i.e., activity Install Process Pipes B (ID: 17)) needed to be installed earlier that the planned start date (4/22/03). The process pipes are a major component of the project and need to be installed as early as possible.

Given this situation, please answer the following questions:

(Please answer using activity ID's only.)

3.1 Which activities are either physically or contractually required for the activity Install Process Pipes B (ID: 17)? Limit your answer to 4 activities excluding activity ID: 17. **MAX: 10 minutes**

START TIME:___________ FINISH TIME:____________

3.2 Which activities would you delay to expedite the activity Install Process Pipes B (ID: 17)? Limit your answer to 4 activities excluding activity ID: 17. **MAX: 10 minutes**

START TIME:___________ FINISH TIME:____________

<u>4. Questions related to the re-sequencing process</u>

4.1 The project manager decided to delay the activity Erect Frame A (ID: 9). Please re-sequence the activity with the goal of maximizing the early start date of the activity Install Process Pipes B (ID: 17). **MAX: 15 minutes**

Please be aware of the following:

- **You may ask whether precedence relationships may or may not be relaxed.**
- **You must resolve all workspace and resource conflicts that may occur due to shifting of activities.**
- **The resulting alternative sequence must be logically sound, i.e., precedence relationships must be correctly specified where required.**

START TIME:___________ FINISH TIME:____________

<u>PLEASE SAVE YOUR MODIFIED .MPP FILE IN YOUR DIRECTORY!</u>

<u>THANK YOU!</u>

C.2 CHARRETTE QUESTIONNAIRE FOR BAY STREET PROJECT

NAME:

TOOL: **CLCPM** **Conventional**

Project Overview

Figure 1 shows a 3D model of the retail store of the Bay Street Project. The schedule provided (Figure 2) describes the sequence of trades performing the exterior closure (EXT) and interior work (INT) for Area 1 and 2 of the retail store. All trades are sequenced to work in Area 1 then Area 2. The exterior work includes sealing the building (i.e., Building Dry-in), placing the brick veneer, painting and installing storefront glass. Interior work includes wall framing and MEP-rough-ins. Interior work can begin as soon as roofing is complete in each area.

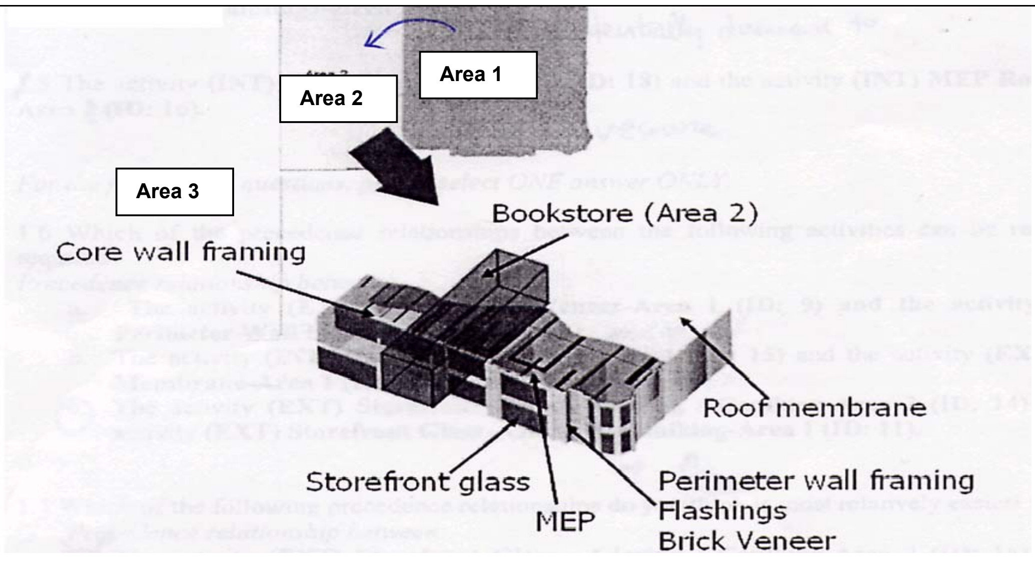
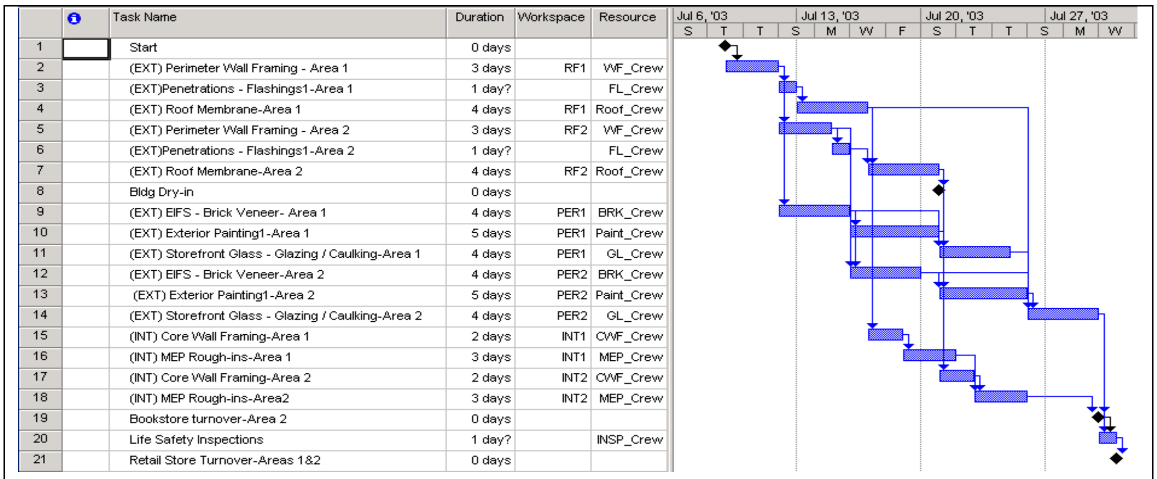


Figure 1: Overview diagram of Bay Street Project Schedule: Areas 1, 2, and 3 denote the zones sectioned for planning and coordination purposes.



	ℹ	Task Name	Duration	Workspace	Resource
1		Start	0 days		
2		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1	WF_Crew
3		(EXT)Penetrations - Flashings1-Area 1	1 day?		FL_Crew
4		(EXT) Roof Membrane-Area 1	4 days	RF1	Roof_Crew
5		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2	WF_Crew
6		(EXT)Penetrations - Flashings1-Area 2	1 day?		FL_Crew
7		(EXT) Roof Membrane-Area 2	4 days	RF2	Roof_Crew
8		Bldg Dry-in	0 days		
9		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1	BRK_Crew
10		(EXT) Exterior Painting1-Area 1	5 days	PER1	Paint_Crew
11		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1	GL_Crew
12		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2	BRK_Crew
13		(EXT) Exterior Painting1-Area 2	5 days	PER2	Paint_Crew
14		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2	GL_Crew
15		(INT) Core Wall Framing-Area 1	2 days	INT1	CWF_Crew
16		(INT) MEP Rough-ins-Area 1	3 days	INT1	MEP_Crew
17		(INT) Core Wall Framing-Area 2	2 days	INT2	CWF_Crew
18		(INT) MEP Rough-ins-Area2	3 days	INT2	MEP_Crew
19		Bookstore turnover-Area 2	0 days		
20		Life Safety Inspections	1 day?		INSP_Crew
21		Retail Store Turnover-Areas 1&2	0 days		

Figure 2: Bay Street project schedule.

Based on this information please answer the following questions:

1. Rationale Questions

For the following 5 questions, please describe, in your own words, the rationale for the precedence relationship between (e.g., activity Erect Beam is "supported by" activity Erect Columns, activity Interior Wall framing is "protected by" activity Erect Roof):

1.1 The activity (EXT) EIFS Brick Veneer-Area 1 (ID: 9) and the activity (EXT) Perimeter Wall Framing-Area 1 (ID: 2).

1.2 The activity (INT) Core Wall Framing-Area 1 (ID: 15) and the activity (EXT) Roof Membrane-Area 1 (ID: 4).

1.3 The activity (EXT) Storefront Glass – glazing / Caulking-Area 1 (ID: 11) and the activity (EXT) EIFS –Brick Veneer-Area 1 (ID: 9).

1.4 The activity (EXT) Storefront Glass – Glazing / Caulking-Area 1 (ID: 11) and the activity (EXT) Exterior Painting-Area 1 (ID: 10).

1.5 The activity (INT) MEP Rough-ins-Area 2 (ID: 18) and the activity (INT) MEP Rough-ins-Area 1 (ID: 16).

For the following 4 questions, please select ONE answer ONLY:

1.6. Which of the precedence relationships between the following activities can be relaxed, if required?

a. The activity (EXT) EIFS Brick Veneer-Area 1 (ID: 9) and the activity (EXT) Perimeter Wall Framing-Area 1 (ID: 2).
b. The activity (INT) Core Wall Framing-Area 1 (ID: 15) and the activity (EXT) Roof Membrane-Area 1 (ID: 4).
c. The activity (EXT) Storefront Glass – Glazing / Caulking-Area 2 (ID: 14) and the activity (EXT) Storefront Glass – Glazing /Caulking-Area 1 (ID: 11).

1.7. Which of the precedence relationships between the following activities do you think is relatively easiest to relax?

Precedence relationship between:

a. The activity (EXT) Storefront Glass –Glazing / Caulking-Area 2 (ID: 14) and the activity (EXT) Storefront Class – Glazing /Caulking –Area 1(ID: 11).
b. The activity (EXT) Storefront Glass –Glazing / Caulking-Area 2 (ID: 14) and the activity (EXT) Exterior Painting1-Area2 (ID: 13).
c. The activity (EXT) Exterior Painting1-Area2 (ID: 13) and the activity (EXT) Exterior Paintinf1-Area 1 (ID: 10).

1.8 Which of the following precedence relationships can be considered to be physically “assisting” or “enabling”?

Between the activity (EXT) EIFS Brick Veneer-Area 1 (ID: 9) and:

a. The activity (EXT) Storefront Glass – Glazing /Caulking-Area 1 (ID: 11).
b. The activity (EXT) Exterior Painting1 – Area 1 (ID: 10).
c. The activity (EXT) EIFS Brick Veneer-Area 2 (ID: 12).

1.9 Which of the following precedence relationships can be considered NOT to be physically “assisting” or “enabling”?

Between the activity (EXT) Perimeter Wall Framing-Area 1 (ID: 2) and:

a. The activity (EXT) Perimeter Wall Framing-Area 2 (ID: 5).

b. The activity (EXT) Penetrations-Flashings-Area 1 (ID: 3).
c. The activity (EXT) Roof Membrane-Area 1(ID: 4).

2. Logical integrity questions

For the following 2 questions, please select ONE answer ONLY:

2.1 In the schedule provided, there exists a redundant precedence relationship. Which of the following relationships do you think is redundant?

a. (EXT) Penetrations-Flashings-Area 1 (ID: 3) and (EXT) Perimeter Wall Framing-Area 1 (ID: 2).
b. (EXT) Roof Membrane-Area 1 (ID: 4) and (EXT) Penetrations-Flashings-Area 1(ID: 3).
c. (EXT) Storefront Class- Glazing / Caulking-Area 2 (ID: 14) and (EXT) Roof Membrane-Area 1 (ID: 4).

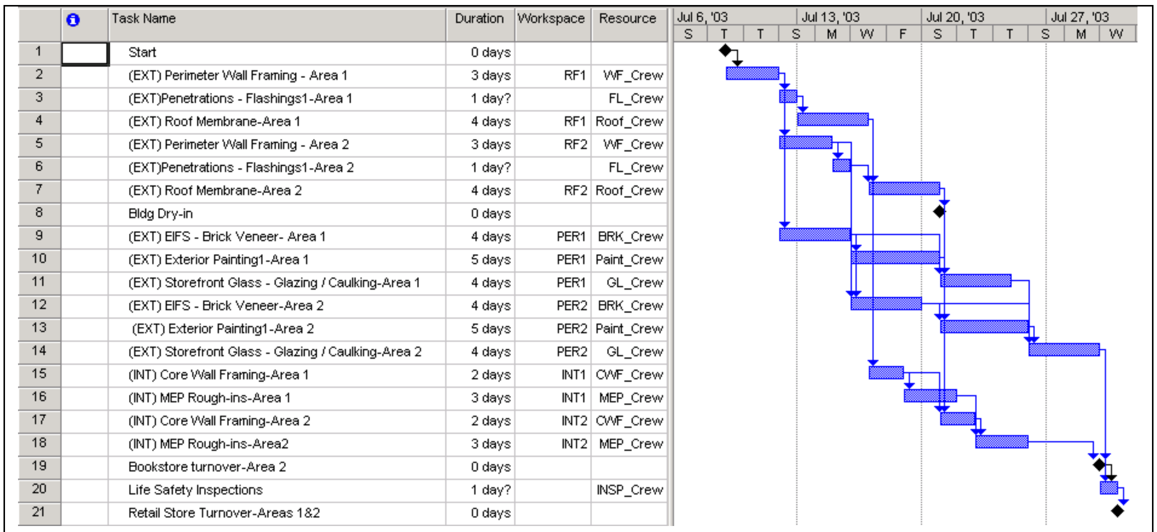
2.2 In the schedule provided, a precedence relationship is missing. Which of the following relationships do you think is missing?

a. (EXT) Core Wall Framing-Area 2 (ID: 17) and (EXT) Core Wall Framing-Area 1 (ID: 15).
b. (EXT) Core Wall Framing –Area 1 (ID: 15) and (EXT) EIFS-Brick Veneer-Area 1 (ID: 9).
c. (INT) MEP Rough-ins-Area 2 (ID: 18) and (EXT) Exterior Painting-Area 2 (ID: 13).

3. Questions related to the identification process

STOP!

- **Please ask for the schedule with the correct sequence and precedence relationships (Figure 3).**
- **Please document start and finish times for answering each question.**
- **Please do not take more time than specified for each question.**



		Task Name	Duration	Workspace	Resource
1		Start	0 days		
2		(EXT) Perimeter Wall Framing - Area 1	3 days	RF1	WF_Crew
3		(EXT)Penetrations - Flashings1-Area 1	1 day?		FL_Crew
4		(EXT) Roof Membrane-Area 1	4 days	RF1	Roof_Crew
5		(EXT) Perimeter Wall Framing - Area 2	3 days	RF2	WF_Crew
6		(EXT)Penetrations - Flashings1-Area 2	1 day?		FL_Crew
7		(EXT) Roof Membrane-Area 2	4 days	RF2	Roof_Crew
8		Bldg Dry-in	0 days		
9		(EXT) EIFS - Brick Veneer- Area 1	4 days	PER1	BRK_Crew
10		(EXT) Exterior Painting1-Area 1	5 days	PER1	Paint_Crew
11		(EXT) Storefront Glass - Glazing / Caulking-Area 1	4 days	PER1	GL_Crew
12		(EXT) EIFS - Brick Veneer-Area 2	4 days	PER2	BRK_Crew
13		(EXT) Exterior Painting1-Area 2	5 days	PER2	Paint_Crew
14		(EXT) Storefront Glass - Glazing / Caulking-Area 2	4 days	PER2	GL_Crew
15		(INT) Core Wall Framing-Area 1	2 days	INT1	CWF_Crew
16		(INT) MEP Rough-ins-Area 1	3 days	INT1	MEP_Crew
17		(INT) Core Wall Framing-Area 2	2 days	INT2	CWF_Crew
18		(INT) MEP Rough-ins-Area2	3 days	INT2	MEP_Crew
19		Bookstore turnover-Area 2	0 days		
20		Life Safety Inspections	1 day?		INSP_Crew
21		Retail Store Turnover-Areas 1&2	0 days		

Figure 3: Bay Street project schedule.

Situation Overview

The project has been delayed due to hazardous material found on-site. Based on the current sequence, the date for the bookstore turnover (7/29/03), i.e., activity Bookstore turnover-Area 2 (ID: 19), needs to be expedited. Contractually, all interior and exterior work must be completed for the area (i.e., area 2) where the bookstore is to be placed, EXCEPT exterior painting. Given this information, please answer the following two questions:

(Please answer using activity ID names only.)

3.1 Which activities are either physically or contractually "assisting" or "enabling" the activity Bookstore turnover-Area 2 (ID: 19)? Limit your answer to 7 activities excluding activity ID: 19. **MAX: 10 minutes**

START TIME:___________ FINISH TIME:____________

3.2 Which activities would you delay to expedite the activity Bookstore turnover-Area 2 (ID: 19)? Limit answer to 4 activities excluding activity ID: 19. **MAX: 10 minutes**

START TIME:___________ FINISH TIME:____________

4. Questions related to the re-sequencing process

4.1 The project manager decided to delay the activity (EXT) Exterior Painting1-Area 1 (ID: 10). Using MSP or CLCPM, please re-sequence the activity with the goal of maximizing the early start date of the activity Bookstore turnover-Area 2 (ID: 19). **MAX: 15 minutes**

Please be aware of the following:

- **You may ask whether precedence relationships may or may not be relaxed.**
- **You must resolve all workspace and resource conflicts that may occur due to shifting of activities.**
- **The resultant alternative sequence must be logically sound, i.e., precedence relationships must be correctly specified where required.**

START TIME:___________ FINISH TIME:____________

<u>PLEASE SAVE YOUR MODIFIED .MPP FILE IN YOUR DIRECTORY!</u>

<u>THANK YOU!</u>

BIBLIOGRAPHY

Aalami, F., Kunz, J., and Fischer, M. (1998a). "Model-Based Sequencing Mechanisms Used to Automate Activity Sequencing." Working Paper Nr. 50, CIFE, Stanford University, Stanford, CA.

Aalami, F., Levitt, R., and Fischer, M. (1998b). "A Customizable Representation for Construction Methods." Working Paper Nr. 51, CIFE, Stanford University, Stanford, CA.

Aalami, F., Kunz, J., and Fischer, M. (1998c). "AEC 4D Production Model: Definition and Automated Generation." Working Paper Nr. 52, CIFE, Stanford University, Stanford, CA.

Aarup, M., Arentoft, M. M., Parrod, Y., Stader, J., and Stokes, I. (1994). "OPTIMUM-AIV: A Knowledge based Planning and Scheduling System for Spacecraft AIV." In Fox, M. and Zweben, M. editors, Knowledge Bases Scheduling, Morgan Kaufmann, San Mateo, CA, 451-469.

Abeyasinghe, M. C., Greenwood, D. J. and Johansen, D. E. (2001). "An Efficient Method for Scheduling Construction Projects with Resource Constraints." Int'l Proj. Mgmt. 19(1), 29-45.

Agrawal, R., and Jagadish, H. V. (1987). "Direct Algorithms for Computing the Transitive Closure of Database Relations." Proc. 13th Int'l Conf. Very Large Data Bases, Brighton, England, 255-266.

Akinci, B., and Fischer, M. (1998). "Time-Space Conflict Analysis based on 4D Production Models." International Computing Congress, ASCE, Boston, October 18-21, Kelvin C.P. Wang ASCE, 342-353.

Antill, J., Woodhead, R. (1970). "Critical Path Methods in Construction Practice." Second Edition, John Wiley & Sons, Inc., New York, N.Y.

Ballard G., and Howell, G. (1994). "Implementing Lean Construction: Stabilizing Work Flow." Proceedings of the 2nd Annual Meeting of the International Group for Lean Construction, Pontificia Universidad Catolica de Chile, Santiago, September, http://www.vtt.fi/rte/lean/santiago.htm, reprinted in Lean Construction, 101-110.

Ballard, G., and Howell, G. (1997). “Shielding Production: An Essential Step in Production Control.” Journal of Construction Engineering and Management, ASCE, 124(1), 11-17.

Barrie, Donald S., and Boyd, C. Paulson, Jr. (1978). “Professional Construction Management”, McGraw-Hill Book Co., New York, first edition, 234-235.

Bentley Systems (1997). “Plantspace Enterprise Navigator: User’s Guide.” Bentley Systems, Inc., Gainsberg, MD, www.bentley.com.

Birrell, G. (1980). “Construction Planning-Beyond the Critical Path.” Journal of Construction Engineering and Management, ASCE, 106(CO3), 389-407.

Burns, S., Liu, L., and Feng, C. (1996). “The LP/IP Hybrid Method for Construction Time-Cost Trade-Off Analysis.” Constr. Mgmt. and Economics, 14, 265-276.

Chaitin, G. (1965). “An Improvement on a Theorem of E.F. Moore," IEEE Transactions on Electronic Computers EC-14, 466-467.

Chang, T., C., Ibbs, W., and Crandall, K., C. (1989) “Network Resource Allocation with Support of a Fuzzy Expert System.” Journal of Construction Engineering and Management, ASCE, 116(2), 239-260.

Cherneff, J., Logcher, R., and Sriram, D. (1991) “Integrating CAD with Construction-Schedule Generation.” Journal of Computing in Civil Engineering, ASCE, 5(1), 64-84.

Choo, H. J. and Tommelein, I. D. (1999). “Parade of Trades: A Computer Game for Understanding Variability and Dependence.” Technical Report 99-1, Construction Engineering and Management Program, Civil and Environmental Engineering Department, University of California, Berkeley, CA.

Chua, D. K. H., Shen L. J., and Bok, S. H. (2003). “Contraint-Based Planning with Integrated Production Scheduler over Internet.” Journal of Construction Engineering and Management, ASCE, 129(3), 293-301.

Clayton, M., Kunz, J., and Fischer, M. (1998). “The Charrette Test Method.” Technical Report Nr. 120, CIFE, Stanford University, Stanford, CA.

Crandall, K. C. (1970). “The Design of a Limited Interactive Management System for Construction Project Control.” Technical Report Nr. 131. Stanford: The Construction Institute, Dept. of Civil Engineering, Stanford University.

Crandall, K. C. (1985). “Resource Allocation with Project Manager Control”, Construction Research Applied to Practice, C. W. Ibbs, ed., ASCE, 1-17.

Currie, K. W. and Tate, A. (1991). “O-Plan: the Open Planning Architecture.” Artificial Intelligence, 52(1), 49-86.

Darwiche, A., Levitt, R., and Hayes-Roth, B. (1988). “OARPLAN: Generating Project Plans by Reasoning about Objects, Actions and Resources”, AI EDAM, 2(3), 169-181.

Davis, E. W. (1973). “Project Scheduling under Resource Constraints-Historic Review and Categorization of Procedures.” AIIE Trans., 5(4), 297-312.

Davis, E. W. (1974). “CPM use in Top 400 Construction Companies.” Journal of the Construction Division, ASCE, 100(CO1), 39-49.

Davis, R., Shrobe, H., and Szolovits, P. (1993). “What is Knowledge Representation?” AI Magazine, 14(1), 17-33.

Dar, S. and Ramakrishnan, R. (1994). “A Performance Study of Transitive Closure Algorithm.” In Proc. of SIGMOD, 454-465.

Dym, C. L. and Levitt, R. E. (1991). “Knowledge-Based Systems in Engineering.” McGraw-Hill, Inc.

Dzeng, R. and Tommelein, I. (1997). “Boiler Erection Scheduling Using Product Models and Case-Based Reasoning.” Journal of Construction Engineering and Management, ASCE, 123(3), 338-347.

Echeverry, D., Ibbs, W., and Kim, S. (1991). "Sequence Knowledge for Construction Scheduling." Journal of Construction Engineering and Management, ASCE, 117(1), 118-130.

Feng, C. and Liu, L. (1997). "Using Genetic Algorithms to solve Construction Time-Cost Trade-Off Problems." Journal of Construction Engineering and Management, ASCE, 11(3), 184-189.

Fikes, R. E., and Nilsson, N. J. (1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving." Artificial Intelligence 2(3-4), 189-208.

Fikes, R. E., Hart, P. E., and Nilsson, N. J. (1972). "Learning and Executing Generalized Robot Plans." Artificial Intelligence, 5(4), 521-288.

Fikes, R. E. (1996). "Ontologies: What Are They, and Where's The Research? " KR 1996, 652-653.

Floyd, R. W. (1962). "Algorithm 97 (SHORTEST PATH)." Communications of the ACM, 5(6), 345.

Fox, M. S. and Smith, S. F. (1984). "ISIS- a Knowledge-based System for Factory Scheduling. Expert Systems, 1, 25-49.

Fox, M. S. and Gruninger, M. (1994). "Ontologies for Enterprise Integration." CoopIS 1994: 82-89.

Fondahl, J. W. (1961). "A Non-computer Approach to the Critical Path Method for the Construction Industry." Technical Report Nr. 9, The Construction Institute, Stanford University, CA.

Fondahl, J. W. (1991). "Development of the Construction Engineer: Past Progress and Future Problems." Journal of Construction Engineering and Management, ASCE, 117(3), 380-392.

Genesereth, M. R. and Fikes, R. E. (1992). "Knowledge Interchange Format, Version 3.0 Reference Manual." Technical Report Logic-92-1, Computer Science Department, Stanford University.

Gruber, T. R. (1991). “The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Base.” Morgan Kaufmann, San Mateo, CA.

Gruber, T. R. (1993). “A Translation Approach to Portable Ontologies.” Knowledge Acquisition, 5(2), 199-220.

Halpin, D. and Riggs, L. (1992). “Planning and Analysis of Construction Operations.” John Wiley and Sons, Inc., New York.

Hanks, S. and Weld, D. S. (1992). “The Systematic Plan Adaptor: A Formal Foundation for Case-Based Planning. Technical Report, 92-09-04, Dept, of Computer Science and Engineering, University of Washington, Seattle, WA.

Haymaker J., Kunz J., Suter, B., and Fischer, M. (2003). "Perspectors: Composable, Domain-Independent Reasoning Modules that Automatically Construct a Geometric Engineering View from Other Geometric Engineering Views." Working Paper Nr 82, CIFE, Stanford University, Stanford, CA.

Hegazy, T. (1999). “Optimization of Resource Allocation and Leveling Using Genetic Algorithms.” Journal of Construction Engineering and Management, ASCE, 125(3), 167-175.

Hendrickson, C., Martinelli, D., and Rehak, D. (1987). “Hierarchical Rule-Based Activity Duration Estimation.” Journal of Construction Engineering and Management, ASCE, 113(2), 288-301.

Hendrickson, C., Au, T. (1989). “Project Management for Construction.” John Wiley & Sons, Inc., New York, NY.

Hillier F., S., and Leiberman, G., J. (1995). “Introduction to Operations Research”, Sixth Edition, McGraw Hill Inc., New York, NY.

Howell, G. (1999). “What is Lean Construction-1999.” Proc. 7th Ann. Conf. Intl. Group for Lean Construction, 1-10, Berkeley, CA, 26-28.

Jiang, B. (1990). “A Suitable Algorithm for Computing Partial Transitive Closures in Databases.” Proc. IEEE 6th Int’l Conf. Data Engineering, Los Angeles, CA, 154-168.

Jakobsson, H. (1991). “Mixed-Approach Algorithms for Transitive Closure.” Proc. 10th Symp. Principles of Database Systems, Denver, Colorado, 199-205.

Kähkönen, K. (1993). “Modelling Activity Dependencies for Building Construction Project Scheduling,” Technical Research Centre of Finland, VTT Publications, ESPOO.

Kelley, J. E. Jr. and Walker, M. R. (1959). “Critical Path Planning and Scheduling.” Proceedings of the Eastern Join Computer Conference, Boston, Massachusetts, 160-173.

Kerrigan, S. and Law, K. H. (2003) “Logic-Based Regulation Compliance-Assistance.” 9th Int’l Conf. on Artificial Intelligence and Law (ICAIL 2003), Edinburgh, Scotland, UK.

Koo, B. and Fischer, M. A. (2000). “Feasibility Study of 4D CAD in Commercial Construction.” Journal of Construction Engineering and Management, ASCE, 126(4), 251-260.

Kim, K. and Paulson, B. C. Jr. (2003). “Agent-Based Compensatory Negotiation Methodology to Facilitate Distributed Coordination of Project Schedule Changes.” Journal of Computing in Civil Engineering, ASCE, 17(1), 10-18.

Kunz, J. C., Fischer, M. A., Levitt, R. E., and Teicholz, P. A. (1994). “Toward Desktop Engineering.” Proceedings of a Workshop in Honor of Steve Fenves, Carnegie-Mellon University.

Kunz, J. C., Jin, Y., Levitt, R. E., Lin, S-D., and Teicholz, P. A. (1995). “The Intelligent Real-Time Maintenance Management (IRTMM) System: Support for Integrated Value-based Maintenance Planning.” Technical Report Nr. 100, CIFE, Stanford University, Stanford, CA.

Kunz, J. C., and Fischer, M. A. (2002). “Research Process.” CIFE Research Questions and Methods, CIFE, Stanford University, Stanford, CA.

Lee, Y. and Gatton, T. M. (1994). “Construction Scheduling based on Resource Constraints.” AACE Trans., 3,1-6.

Lenat, D. B., and Guha, R. V. (1990). “Building Large Knowledge Based Systems: Representation and Inference in the CYC Project.” Addison-Wesley, Reading, MA.

Levitt, R. E. and Kunz, J. C. (1985). “Using Knowledge of Construction and Project Management for Automated Schedule Updating.” Project Management Journal, 16(5), 57-81.

Leu, S. S. and Yang, C. (1999). “GA-based Multi-Criteria Optimal Model for Construction Scheduling.” Journal of Construction Engineering and Management, ASCE, 125(6), 420-427.

Leu, S. S., Chen, A., and Yang, C. (1999). “Fuzzy Optimal Model for Resource-Constrained Construction Scheduling.” Journal of Construction Engineering and Management, ASCE, 13(3), 207-216.

Li, H., Cao, J., and Love, P. (1999). “Using Machine Learning and GA to Solve Time-cost trade-off Problems.” Journal of Construction Engineering and Management, ASCE, 125(5), 347-353.

Microsoft Corporation. (1998). “Microsoft Project 98.” Microsoft Corporation, Redmond. WA.

Meyer, W. L. and Shaffer, L. R. (1963). “Extensions to the Critical Path Method Through the Applications of Integer Programming.” Construction Research Series, No. 2. Urbana: Department of Civil Engineering, University of Illinois. IL.

Morad, A. A., and Beliveau, Y. J. (1994). “Geometric-Based Reasoning System for Project Planning.” 8(1), 52-71.

Mosehli, O. (1993). “Schedule Compression Using Direct Stiffness Method.” Can, J. Civ. Engrg., CSCE, 20(1), 65-72.

Mosehli, O., Lorterapong, P. (1993) “Least Impact Algorithm for Resource Allocation”, Can. J. Civ. Engrg., CSCE, 20 (2), 180-188.

Navinchandra D., Siriam D., and Logcher R. (1998) “GHOST: A Project Network Generator.” Journal of Construction Engineering and Management, ASCE, 2(3), 239-254, July.

Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W., and Musen. M. A. (2001). “Creating Semantic Web Contents with Protege-2000.” IEEE Intelligent Systems 16(2), 60-71.

Patterson, J. H. and Huber, D. (1974). “A Horizon-Varying, Zero-One Approach to Project Scheduling.” Management Science, 20(6), 990-998.

Paulson, B. C., Jr. (1971). “Man-Computer Concepts for Project Management,” Technical Report No. 148, Stanford: The Construction Institute, Department of Civil Engineering, Stanford University, Stanford, CA.

Paulson, B. C., Jr. (1973). “Project Planning and Scheduling: Unified Approach,” Journal of the Construction Division, ASCE, 99(CO1), 45-58.

Paulson, B. C. Jr. (1976). “Concepts of Project Planning and Control.” Journal of the Construction Division, ASCE, 102(CO1), 67-80.

Popescu C., and Borcherding J. D. (1975). “Developments in CPM, PERT, and Network Analysis.” Journal of the Construction Division, ASCE, 101(CO4), 769-784.

Prager, W. (1963). “ A Structured Method of Computing Project Cost Polygons.” Management Science, 9(3), 394-404.

Riley, D., and Sanvido, V. (1995). “Patterns of Construction Space Use in Multistory Buildings”, Journal of Construction Engineering and Management, ASCE, 121(4), 464-473.

Russell, A. D., and Wong W. C. M. (1993). “New Generation of Planning Structures.” Journal of Construction Engineering and Management, ASCE, Vol. 119, No. 2, 196-214.

Russell, S. and Norvig, P. (1995). “Artificial Intelligence: A Modern Approach.” Prentice Hall, Upper Saddle River, NJ.

Schlidt, H. (2002). "Java 2: The Complete Reference", Osborne McGraw-Hill, 5th edition, New York, NY.

Seibert, L., Seppanen, P., Kunz, J., and Paulson, B. (1996). "Value-added Assessment of Construction Plans." Technical Report Nr. 110, CIFE, Stanford University, Stanford, CA.

Siemens, N. (1971). "A Simple CPM Time-Cost Trade-Off Algorithm." Management Science. 17(6), B354-B363.

Smith, S. F. (1989). "The OPIS Framework for Modeling Manufacturing Systems." Technical Report, TR-CMU-RI-89-30, Carnegie Mellon University, The Robotics Institute.

Stefik, M. (1981). "Planning with Constraints (MOLGEN: Part 1)." Artificial Intelligence, 16(2), 110-137.

Sycara, K. and Miyashita, K. (1993). "Schedule Repair through Case-Based Reasoning." Technical Report, The Robotics Institute, Carnegie Mellon University.

Tamassia, R. and Ioannis, G. (2002). "Graph Algorithms and Applications." World Scientific Publishing Co., 1st edition, Singapore.

Tamimi, S. and Diekmann, J. (1988). "Soft Logic in Network Analysis." Journal of Computing in Civil Engineering, ASCE, 2(3), 289-300.

Tate, A. and Whiter, A. M. (1984). "Planning with Multiple Resource Constraints an Application to a Naval Planning Problem." In Proc. Of the 1st Conference on AI Applications, Denver, CO, 410-416.

Tommelein, I. D., and Ballard, G. (1997). "Coordinating Specialists." Tech. Report Nr. 97-8, Constr. Engrg. and Mgmt. Program, Civil and Envir. Engrg. Dept., Univ. of California, Berkeley, CA.

Tommelein, I. D. (1998). "Pull-driven Scheduling Techniques for Pipe-Spool Installation: Simulation of a Lean Construction Technique." Journal of Construction Engineering and Management, ASCE, 124(4), 279-288.

Tommelein, I., D., Riley, D., and Howell, G., A. (1999). "Parade Game: Impact of Work Flow Variability on Trade Performance." Journal of Construction Engineering and Management, ASCE, 125(5), 304-301.

Thabet, W. and Beliveau, Y. (1997). "Space-Constrained Resource-Constrained Scheduling System." Journal of Construction Engineering and Management, ASCE, 11(1), 48-59.

Warshall, S. (1962). "A Theorem on Boolean Matrices." Journal of the ACM*,* 9(1), 11-12.

Waugh, L. (1990). "A Construction Planner." Technical Report Nr. 32, CIFE, Stanford University, Stanford, CA.

Wiest, J. D., and Levy, F. K. (1969). "A Management Guide to PERT/CPM." Prentice Hall, Englewood Cliffs, NJ.

Wilkens, D. E. (1984). "Domain-Independent Planning: Representation and Plan Generation." Artificial Intelligence, 22, 269-301.

Wilkens, D. E. (1990). "Can AI Planners Solve Practical Problems?" Computational Intelligence, 6(4), 232-246.

Yang, Q. (1997). "Intelligent Planning: A Decomposition and Abstraction Based Approach." Springer-Verlag Berlin Heidelberg, Germany.

Zouein, Pierrette and Tommelein, Iris. (1993). "Space Schedule Construction." Proceedings of the 5th International Conference on Computing in Civil engineering, New York, NY, 1770-1777.

Zozaya-Gorostiza, C., C. Hendrickson and D. R. Rehak. (1989). "Knowledge-Based Construction Project Planning." Excellence in the Constructed Project, ASCE, 217-222.

Zweben, M., Davis, E., M. Deale. (1993). "Iterative Repair for Scheduling and Rescheduling." IEEE Transactions on Systems, Man and Cybernetics, 23(6), 1588-1596.

Zweben, M. and Fox, M. S. (1994). "Intelligent Scheduling." Morgan Kaufmann Publishers Inc., San Francisco, CA.