



CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

**A Formal Identification and Re-sequencing
Process for Rapid Generation of
Sequencing Alternatives in CPM Schedules**

By

**Bonsang Koo, Martin Fischer,
& John Kunz**

**CIFE Technical Report #168
JANUARY 2007**

STANFORD UNIVERSITY

COPYRIGHT © 2007 BY
Center for Integrated Facility Engineering

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*

A Formal Identification and Re-sequencing Process for Developing Sequencing Alternatives in CPM Schedules

Bonsang Koo¹; Martin Fischer²; and John Kunz³

Abstract

Construction planners face many scheduling challenges during the course of a project. Limitations in time and resources frequently require planners to re-sequence existing activity sequences to expedite milestone or bottleneck activities. Re-sequencing activities within a CPM network, however, is time-consuming and error-prone because planners find it difficult to know which activities may be delayed and what the implications are to other activities once an activity is delayed. This paper presents a formal identification and re-sequencing process that supports the rapid development of sequencing alternatives in construction schedules. The identification process identifies activities in a CPM schedule that when delayed will expedite a project-critical activity. The re-sequencing process ensures that any sequencing conflicts that occur while re-sequencing activities are correctly resolved. The process builds on a constraint ontology that provides domain-specific representation of sequencing rationale for construction activities. The process also utilizes a classification mechanism that provides the heuristics to infer the role and “status” (i.e., whether an activity may or may not be delayed) between activities. The entire process ensures that all possible re-sequencing solutions are explored and developed consecutively. Validations performed using a prototype system, “Constraint-loaded CPM”

¹ Research Fellow, Construction and Economy Research Institute of Korea, 11th floor, Construction Bldg, 71-2 Nonhyun Dong, Kangnam Gu, Seoul, 135-701 Korea, bkoo@cerik.re.kr

² Professor, Department of Civil and Environmental Engineering and (by Courtesy) Computer Science, Director, Center for Integrated Facility Engineering, Stanford University, Stanford, CA 94305, fischer@stanford.edu

³ Executive Director, Center for Integrated Facility Engineering, Stanford University, Stanford, CA 94305, kunz@stanford.edu

(CLCPM), demonstrate improvement in planners' ability to understand the schedule logic and quickly devise alternative sequencing scenarios for execution.

Keywords: Constraint Modeling, Critical Path Method, Computer Aided Scheduling, Construction Management

1. Introduction

The ability to adapt to changing project demands is a critical skill for today's project planners. Changing demands from owners, unexpected site conditions, and procurement delays require planners to frequently modify existing activity sequences [1], [2]. Planners today rely on CPM-based scheduling tools to evaluate different sequencing alternatives for their feasibility and whether they will meet project deadlines.

When re-sequencing construction activities, planners need to determine the physical or technical impact or "role" an activity has on following activities. They also need to determine which activities may or may not be delayed. Distinguishing the role and "status" (i.e., whether an activity may be delayed) of activities in turn requires planners to understand the rationale and flexibility of constraints between activities.

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 (i.e., cannot be relaxed).

Distinguishing the role and status of activities in turn allows planner to determine which activities can or cannot be delayed, and helps them in prioritizing activities while developing sequencing alternatives. The current CPM framework does not allow planners to explicitly describe the rationale for constraints. Correspondingly, planners can only determine the role and status of activities in their minds, which is time-consuming and frequently prone to error. This in turn makes developing sequencing alternatives in CPM schedules difficult to implement and practically prohibitive.

Existing approaches for reducing schedule durations include time-cost trade-off (e.g.,

[3]- [6]) and resource-constrained scheduling (e.g., [7]-[10]). These methods assume that existing sequences in CPM schedules are fixed and that the primary method for acceleration is to add or reallocate resources. These approaches are thus not suited for generating re-sequencing alternatives, in particular where milestone tasks or major bottleneck activities need to be expedited.

“Replanning” techniques in AI planning systems [11]-[17] repair plans by relaxing existing sequences when plans fail during execution. These techniques define domain-specific representations (e.g., physical constraints, state constraints, etc.) for the rationale of activity sequences to complement the precedence relationships of CPM or PERT networks. In addition, they developed tailored heuristics that leverage the representation to enable automation support for re-sequencing of activities.

In this research project, such AI replanning methods were employed by formalizing the necessary representation and reasoning mechanisms tailored to the construction-scheduling domain in such a way that allows the computational support of re-sequencing activities in CPM schedules. Specifically, the authors formalized a constraint ontology, classification mechanism and an identification and re-sequencing process.

The constraint ontology explicitly distinguishes the rationale for activity sequences using specific constraints (e.g., *supported by*, *damaged by*), while assigning each constraint a role (“enabling,” “impeding”) and flexibility (“flexible,” “inflexible”) [18].

The classification mechanism leverages the ontology to automatically infer the role and status of activities with respect to the particular activity that requires re-sequencing. Specifically, the mechanism classifies the impact or role of activities as either “enabling” or “impeding,” and the “status” of activities as either “driving” (cannot be delayed) or “non-driving” (can be delayed) [18].

This paper focuses on the identification and re-sequencing process, which leverages the ontology and classification mechanism in a step-by-step process to expedite the development of sequencing alternatives while ensuring that alternatives developed by planners are logically sound. Specifically, it identifies all available options for expediting a particular activity. During re-sequencing of an activity, it identifies the methods (i.e., by relaxing a constraint or using available float) available for delaying a specific activity, and also uses the role and status of activities to resolve resource or workspace conflicts.

The process ensures that planners are aware of the available options so that sequencing alternatives are not missed or ignored. In addition, the process allows several alternatives to be developed consecutively, instead of finding a single optimal solution.

A prototype, “Constraint-Loaded CPM” (CLCPM), implements the described method and was used to perform retrospective cases and a charrette test. These validations demonstrate the power and generality of the formalizations in providing a faster and reliable way to develop sequencing alternatives.

The following section introduces a motivating case that exemplifies the manual steps planners need to take today in developing a sequencing alternative.

2. Motivating Case

Part of the construction schedule for one of Intel Corporation's fabrication plant (“FAB 22” project) is introduced to illustrate the steps required for developing sequencing alternatives. The project included two main buildings: the main fabrication building (“FAB”) and the Central Utility Building (“CUB”). The CUB houses boilers and chillers that support the main FAB building. Figure 1a shows the initial schedule for constructing the foundation, structural frame and process pipes of the CUB. One of the problems faced by the project manager was the need

to expedite the activity Install Process Pipes B. The process pipes connect the main FAB to the CUB, and thus it was critical to get the process pipes in position as soon as possible.

To install the process pipes earlier than scheduled originally (day 12 in Figure 1a), the project manager reversed the sequence between the activities Apply Fireproofing B and Install Process Pipes B, i.e., the planner elected to delay the activity Apply Fireproofing B activity so that the activity Install Process Pipes B could be performed earlier (day 11 in Figure 1b). This change required the fireproofing trade to wrap the process pipes to provide protection from the fireproofing material (shown as activity Wrap Pipes in Figure 1b). This alternative did not result from a thorough investigation of possible sequencing alternatives available to the project manager.

The example shows that construction planners modify activity sequences to coordinate multiple trades that share limited resources and workspaces [19]. However, planning decisions are often made without the evaluation of possible sequencing alternatives. This is in part due to the difficulty in generating sequencing alternatives using existing CPM-based scheduling tools, since the CPM framework does not provide a formalized approach for identifying and re-sequencing activities.

When developing sequencing alternatives, planners need to understand the rationale for activity sequences and also infer the role and status of activities. For example, the initial rationale for sequencing pipe installation work after fireproofing the 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. Figure 2a shows the rationale for these activity sequences denoted as *damaged by* and *supported by* constraints, respectively. Planners need to understand the rationale for constraints to determine the role activities have on following activities. For example, the *supported by* constraint between the activities Erect Frame B and Install Process Pipes B implies that the activity Erect Frame B is “enabling” since it provides physical support for the process pipes (Figure 2a). Similarly, the *damaged by* constraint between the activities Apply Fireproofing B and Install Process Pipes B implies that the activity Apply Fireproofing B is “impeding” the installation of process pipes (Figure 2a). 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 (Install Process Pipes B). The authors call this activity, an activity that is the focus of managerial attention, the “target” activity. The authors also generalize the role of sequencing constraints as “enabling” or “impeding.” For example, the *supported by* constraint is an enabling type, and the *damaged by* constraint is an impeding type of constraint.

In addition to the role of constraints, planners also need to know whether a constraint may or may not be relaxed (i.e., flexibility). Planners need to understand the flexibility of constraints to determine whether an activity is “driving” or “non-driving” with respect to the target activity.

The authors define a driving activity as an activity that cannot be delayed without delaying the target activity. For example, the activity Apply Fireproofing B is a critical activity (i.e., zero float). However, as shown in Figure 2b, the *damaged by* constraint is flexible, i.e., Apply Fireproofing B can be delayed. The activity is “non-driving.” Similarly, the activity Erect Frame B is also a critical activity. Although the *supported by* constraint between the activities Erect Frame B and Install Process Pipes B is inflexible, the activity Erect Frame B can still be delayed by relaxing the *damaged by* constraint between the activity Apply Fireproofing B and the activity Install Process Pipes B. Hence, the activity Erect Frame B can also be delayed and is also a “non-driving” activity (Figure 3b). The authors use the term “status” to describe whether an activity is “driving” or “non-driving.”

Applying the classification of the role and status of activities to the critical activities in the sample case enables the project manager to determine opportunities for expediting the target activity that may previously have been overlooked or ignored.

For example, Figure 3a shows the activities critical with respect to the target activity Install Process Pipes B. The activity Apply Fireproofing B is a critical activity that is impeding and non-driving, and as discussed, this was the actual activity delayed to expedite the target activity Install Process Pipes B. The activity Erect Frame A is also a critical activity linked to the target activity by the series of activities: Erect Frame B, Apply Fireproofing B, and Install

Process Pipes B. The authors call such paths an activity's "network chain." The role and status of this activity is determined by the role and flexibility of the constraints in this network chain. Similarly to the activity Apply Fireproofing B, this activity is also an impeding and non-driving activity. Thus, delaying the activity Erect Frame A can also expedite the activity Install Process Pipes B, and provides the project manager the option of developing a different sequencing alternative.

To summarize, the example shows that inferring the role and status of the activity based on the role and flexibility of constraints allows a clearer distinction of which activities can or cannot be delayed to expedite a particular target activity. By contrast, the current CPM framework only distinguishes the temporal aspect of constraints (i.e., Finish to Start precedence relationships) and the time criticality of activities, which can be misleading as it informs planners that activity sequences cannot be changed.

By delaying any of the impeding and non-driving activities, the project manager can expedite the target activity. As shown in Figure 3a, the authors call such activities, "candidate" activities. Assuming the project manager identifies and selects the activity Erect Frame A as the candidate activity to delay (Figure 3a, step 1; and Figure 3b, step 2), he then needs to relax one of the flexible constraints in the activity's network chain (Figure 3b, steps 2 and 3). There are two flexible constraints: the *damaged by* constraint and the *resource* constraint. Assuming he

chooses the *resource* constraint to relax, he subsequently expedites the target activity and its predecessors (Figure 4a, 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 (i.e., Frame Installation Crew) (Figure 4b, 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 (Figure 4b, step 7). 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 4b, step 8) and determines the activity Erect Frame A as the activity to delay (i.e., activity of lower priority).

With respect to the target activity, the activity Erect Frame A has positive float, or “target” float (i.e., the activity’s total float calculated with respect to the target activity, as to the last or “end” activity) (Figure 5a, step 9). The project manager tries to resolve the resource conflict by delaying the activity (Figure 5a, step 10). However, the conflict is still not resolved (Figure 5b, step 11). The activity Erect Frame A is still the lower priority activity. The activity now has zero target float, and further delaying the activity will in turn delay the target activity (Figure 5b, steps 12). The activity can only be delayed by relaxing either the *resource* constraint or the *damaged by constraint* (Figure 5b, step 13). Again, assuming the project manager decides to

relax the *resource* constraint, the relaxation of the constraint allows the activity to be delayed further, and correspondingly resolves the resource conflict (Figure 6a, steps 14 and 15). Finally, the project manager specifies a resource constraint to ensure that the logic remains correct (Figure 6b, step 16).

The example demonstrates that developing even a single sequencing alternative requires several steps to be performed manually. Planners need to continuously update the role and status of activities, use the role and status of activities to resolve resource or workspace conflicts, and determine the most suitable way to resolve the conflicts either by relaxing constraints or using available float.

In summary, the sample case illustrates the need for domain-specific representation and heuristics to support the correct and rapid re-sequencing of activities in construction schedules. Specifically, planners need to distinguish specific constraints with respect to their role and flexibility. Planners also need a way to quickly update the role and status of activities. Finally, planners need a formal approach that incorporates these concepts into a formal process and that guides planners throughout the steps involved in developing sequencing alternatives.

The following section introduces AI replanning methods that provided the design criteria and approach for developing these formalizations.

3. Points of Departure

Replanning techniques AI planning systems repair plans when a plan fails during execution while considering time and resources. For example, Nonlin [11] and SIPE [12] reason about the allocation of limited resources to various plan steps. These were some of the first planning systems to attempt automated replanning. OPIS [13] was developed specifically for scheduling and rescheduling and defined an opportunistic heuristic for rescheduling. GERRY [14] uses an iterative repair method to repair violated constraints. Optimum AIV [15] developed heuristics that assist planners in developing planning alternatives. Systematic Plan Adapter [16] and CABINS [17] use a case based reasoning approach for rescheduling.

These techniques have tailored representations (e.g., physical constraints in OPIS versus state constraints in GERRY) and heuristics (e.g., reactive scheduling in OPIS versus iterative repair in GERRY) to meet different domain-specific replanning requirements. On the other hand, they also incorporate similar methodologies. For example, 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 resource allocations. These systems also rely on CPM or PERT networks to represent the schedule to manage the number of activities in large projects. In addition, these planning systems are not designed as rigidly coded algorithms, but are designed in a way to promote planners in using their judgment by assisting planners to focus on the critical information and input only the relevant information.

Such AI replanning methods were employed for re-sequencing of activities in the construction-scheduling domain by developing a constraint ontology [18], classification mechanism [18], and an identification and re-sequencing process.

The constraint ontology provides the necessary representation for enabling the correct re-sequencing of construction activities. Specifically, the ontology defines four abstract types of

constraints: an impeding and enabling type of constraint for a constraint's role, and flexible and inflexible for a constraint's flexibility. A flexible constraint can be further assigned a degree of flexibility (DOF) using a high, medium and low scale. The ontology allows specific types of constraints to be defined, while associating each constraint with a default value for their role and flexibility. Table 1 shows the project-independent constraints compiled from existing literature [19]-[23] and actual construction schedules with which planners can describe their rationale using everyday terminology.

The classification mechanism provides the heuristics to automate the inference of the role and status of activities. Specifically, the classification mechanism leverages the constraint ontology to automatically infer the role (enabling, impeding) and status (driving, non-driving) of activities given a constraint-loaded CPM schedule. The classification mechanism consists of a network chain search algorithm and a set of inference rules. The network chain search algorithm identifies unique network chains or "paths" between an activity and the target activity. The inference rules generalize the relationship between activities and constraints within the context of network chains.

The identification and re-sequencing process encapsulates the ontology and classification mechanism and formalizes the steps involved in developing sequencing alternatives. The process also adopts the approach used by AI replanning systems. For example, process uses sequencing rationale in the existing plan from which to develop modified plans. The process also depends on a CPM network to manage construction activities of a large project, and uses a technique with which planners are familiar. Finally, the process has been designed to enable planners to make decisions during the steps, and hence rely on the planners' judgment and expertise to develop sequencing alternatives.

The following section describes the design requirements for the formal identification and re-sequencing process in detail.

4. Design Requirements for a Formal Identification and Re-sequencing Process

To formalize the set of steps for developing sequencing alternatives, the authors performed several paper-based Gedanken experiments. Gedanken experiments [20] are a method used in computer science to identify generic solutions by giving a system inputs and looking at the outputs and formulating an automation approach based on the outputs.

The sample case provides a good example of such an experiment. Table 2 summarizes the steps the project manager needs to take. In the identification process, the project manager needs to identify and select candidate activities by identifying impeding and non-driving activities on the critical path. In the re-sequencing process, the project manager has to perform several steps repetitively. He needs to continually expedite or delay activities until no further workspace or resource conflicts exist. When a resource or workspace conflict is identified, the project manager needs to first update the role and status of activities. Then he needs to decide which activity to delay. The project manager's rationale for determining the activity to delay is based on the role and status of the activity pair in conflict. Subsequently, he needs 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 chains.

Thus, a challenge was to automate these repetitive steps where possible, while ensuring that the steps result in correct sequencing alternatives. As shown in Table 2, many of the steps can be automated by utilizing the ontology and classification mechanism. It also entailed developing a priority rule that utilizes the role and status of activities to resolve workspace or resource conflicts. Another challenge was to automate the method of resolving the conflict; i.e., by using available target float or flexible constraints.

The sample 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.

Thus another challenge was to formalize the process so that it ensures that all candidate activities are accounted for, and that planners can continue to improve on their solution until all possibilities are exhausted. Finally, the process needs to be general enough so that it is applicable regardless of the different types of constraints used to describe sequencing rationale in a given CPM schedule.

The following section describes the formalized steps in detail.

5. Formal Identification and Re-sequencing Process

Figure 7 shows a flow chart for the formal identification and re-sequencing process. In the identification process (steps 1 to 4), users select a target activity and the system outputs a list

of candidate activities (i.e., activities on the critical path that are impeding and non-driving). Users can select one of these candidate activities to delay (Figure 7, steps 3 and 4).

In the re-sequencing process (Figure 7, steps 5 to 14), the system first identifies flexible constraints between the candidate activity and the target activity and outputs these constraints as a list for planners to choose from (step 5). If more than one flexible constraint exists, users can decide which constraint to relax by comparing the constraints' degree of flexibility (DOF) (step 6). The system relaxes the chosen constraint and subsequently expedites the target activity (step 8). Consequently, a workspace or resource conflict can occur (step 9). The system assumes that two overlapping activities requiring the same workspace (e.g., zone) or resource results in a workspace or resource conflict.

To determine which activity to delay, the system uses pre-defined priority rules that formalize planners' rationale for prioritizing activities based on the role and status of activities (step 13). As shown in Figure 8, 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 a 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 (Figure 8, 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) (Figure 8, step 13.3). If both activities are non-driving, then the priority cannot be decided based on the activities' status (Figure 8, step 13.4). As shown in the sample case, planners will try to re-sequence activities so that enabling activities are prioritized over impeding activities. Hence, as shown in Figure 8, enabling activities have priority for resource or workspace over impeding activities (step 13.5). If both activities have the same role, users have to decide which activity has lower

priority (Figure 8, step 13.6).

Once the system has identified which activity to delay, the system determines how that activity needs to be delayed. As shown in the 1st loop of Figure 7, the system first tries to resolve a conflict by using the target float. If the conflict is not resolved and no target float is available, the system uses the 2nd loop to identify constraints that users can select to relax. Once the workspace or resource conflict is resolved, the system instantiates a resource or workspace 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 7, the 3rd loop enables the system to identify a new set of candidate activities based on the modified CPM network. The system identifies 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 candidate activities exist and the process is terminated.

In summary, the identification process identifies and updates the candidate activities and thus ensures that all possible solutions are explored. The re-sequencing process ensures that the selected candidate activity is delayed correctly by resolving resource or workspace conflicts and informing planners of how the conflicts need to be resolved. The entire process enables sequencing alternatives to be developed thoroughly and consecutively, and allow planners to interact throughout the development process.

6. CLCPM System Architecture

Although the mechanisms formalized are platform independent, the authors implemented CLCPM to work as an “add-on” to Microsoft Project and coded in Microsoft

Visual Basic. Figure 9 shows an IDEF₀ diagram that describes how CLCPM assists users in developing sequencing alternatives.

As shown in the first module, CLCPM takes an existing construction CPM schedule as input. CLCPM provides users with a list of pre-defined constraints that include the constraints formalized in Table 1 (Figure 10). Users describe the rationale for precedence relationships by selecting one of these constraints or can create new constraints (Figure 10, (a) and (b)). Users can also customize the flexibility of a constraint to reflect the circumstances of a specific project (Figure 10, (c)). The output of the first module is a CPM schedule where the rationale for every precedence relationship is explicitly described, i.e., a constraint-loaded schedule.

In the second module (Figure 9), users select the target activity. CLCPM identifies activities that are time-critical with respect to the target activity, that is, activities that have zero target float. Of these activities, CLCPM uses the role and status of activities to identify candidate activities (i.e., activities that are impeding and non-driving). Hence, the output of the second module is a list of candidate activities.

The third and fourth module is equivalent with the steps described in the identification and re-sequencing process. CLCPM updates the role and status of activities each time users make a modification to the schedule and resolves resource or workspace conflicts using the pre-defined priority rules. The final output is a correctly developed sequencing alternative. CLCPM allows planners to repeat this process until all candidate activities are re-sequenced and ensures that all options for expediting a target activity are exhausted.

The overall architecture of CLCPM reflects the initial design intent, which was to involve planners in using their judgment, prevent planners from making unnecessary mistakes, and to expedite the process as much as possible. CLCPM only automates steps that are logically inevitable and does not require user input. Involving planners also prevents the development of

sequencing alternatives that may not be realistic due to impractical assumptions.

Figures 11 and 12 show how CLCPM is used to develop sequencing alternatives for the sample case, the Intel CUB schedule. The constraints used to describe the rationale were confirmed by the project manager of the project. Figures 11a and 11c show the role and status of the activities classified with respect to the target activity, Install Process Pipes B. Figure 11a shows that the only activities enabling the activity are the activities Formwork B, Build Slab B, Preassemble Frame B and Erect Frame B. Figure 11b shows the user interface where CLCPM provides a list of the role of activities. Figure 11c shows that activities on the critical path are non-driving activities. Hence these activities can be delayed to expedite the target activity.

Figure 12a shows the candidate activities identified by CLCPM, while Figure 12b shows a corresponding list of candidate activities. Figure 12c shows the final sequencing alternative which has the target activity expedited by 4 days. Figure 12c also shows that no further sequencing alternatives exist, as all activities on the critical path are driving, and by definition no further activities can be delayed. This alternative was confirmed by the project manager to be logically sound and executable.

7. Validation

Similar tests to that of the Intel CUB schedule were performed on two other retrospective cases: (1) the McWhinney Office Building project and (2) the Bay Street Retail Store project. As shown in Table 3, different phases of the projects were selected to ensure that CLCPM is general enough to develop sequencing alternatives for different types of constraints that exist in construction schedules. For each project schedule, a single target activity considered to be bottleneck or milestone activities was selected for the tests.

Table 3 shows that for the three project schedules, CLCPM successfully identified 13

candidate activities (CA). These candidate activities were confirmed to be correct by an experienced project scheduler. Table 3 shows that the project scheduler also agreed with the final sequencing alternatives developed, which all successfully expedited the target activity (TA). The results show that CLCPM correctly helps identify and develop sequencing alternatives for a wide range of different types of activities, which provides evidence of its generality, since the three schedules used different types of constraints (13 in total, [18]) to describe their rationale for activity sequences.

The authors also performed a “charrette” test [21] that involved using eight students at the Center for Integrated Facility Engineering (CIFE) at Stanford University to compare their ability to develop sequencing alternatives. Half the students used CLCPM, while the other half used Microsoft Project (MSP) to develop sequencing alternatives using the Intel CUB and the Bay Street Retail Store project. The testers were asked to identify the candidate activities for a single target activity in the two project schedules, and develop sequencing alternatives to expedite the target activities by delaying one of the candidate activities. The authors evaluated their performance by assigning points to mistakes (e.g., unresolved workspace and resource conflicts, incorrect constraints relaxed, etc.) and measuring the time taken to develop a sequencing alternative. Students using CLCPM scored on average 92 points (out of a total 100) compared to 52 points for students using MSP. Furthermore, students using MSP took on average twice as long to develop sequencing alternatives that were of poorer quality. The authors interpret the test as providing evidence for the power of the semi-automated identification and the re-sequencing process, as the results demonstrate that CLCPM significantly reduces the errors and time required to identify and develop sequencing alternatives.

8. Conclusion

This paper has presented a formal identification and re-sequencing process to support the correct and rapid development of sequencing alternatives in construction schedules. The formal process presents a novel approach to re-sequencing activities by adopting AI replanning methods previously unexplored in the construction-scheduling domain. The process employs the core concepts identified to be essential for re-sequencing activities: the role and flexibility of constraints, and the role and status of activities. It successfully utilizes these concepts and their implementations to formalize the steps involved in developing sequencing alternatives. The constraint ontology is used to determine which constraints can be relaxed. The classification mechanism is used to determine how activities should be delayed, how activities in resource or workspace conflict needs to be prioritized, and how these conflicts should be resolved. By automating the multiple steps, the process provides much needed computational support that makes evaluation and development of multiple sequencing scenarios practically executable in construction schedules.

Although the validations performed using CLCPM provide evidence of the power and generality of the formalizations, the cases used were limited in that all three schedules had less than 30 activities. Additional tests are planned to test the scalability of schedules, starting with schedules with 100 or more activities. CLCPM only handles Finish-to-Start relationships and focuses on construction-oriented constraints. Further research is required to extend CLCPM to handle the remaining types of precedence relationships as well as design and procurement constraints.

One extension to this research of interest is to utilize 4DCAD models to provide better visualization for the role and status of activities identified by CLCPM. Such integration efforts should provide a powerful communication medium for multiple stakeholders of complex

projects to experiment and explore sequencing scenarios prior to actual execution.

9. Acknowledgements

This research was supported in part by the National Science Foundation under grant 0301730. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors also thank the support of CIFE at Stanford University. The authors also thank Art Stout of Intel Corporation and Rich Creveling of Neenan Construction for providing access to their job sites and project schedules. A special thanks goes to Dean Reed of DPR Construction for his valued input and participation in the retrospective validation studies.

10. References

[1] Barrie, Donald S., and Boyd, C. Paulson, Jr. (1978). "Professional Construction Management." McGraw-Hill Book Co., New York, first edition.

[2] Milosevic, D. (2003). "Project Management ToolBox: Tools and Techniques for the Practicing Project Manager." John Wiley & Sons, Inc., Hoboken, NJ.

[3] Kelley, J. E. Jr. and Walker, M. R. (1959). "Critical Path Planning and Scheduling." Proceedings of the Eastern Joint Computer Conference, Boston, Massachusetts, 160-173.

[4] 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.

[5] 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.

[6] Zheng, D. and Ng, S. (2004). "Stochastic Time-Cost Optimization Model Incorporating Fuzzy Sets Theory and Nonreplaceable Front." *J. Constr. Eng. Mgmt.*, 131(2), 176-186.

[7] 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.

[8] Crandall, K. C. (1985). "Resource Allocation with Project Manager Control." *Construction Research Applied to Practice*, C. W. Ibbs, ed., ASCE, 1-17.

[9] 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.

[10] 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.

[11] Tate, A. and Whiter, A. M. (1984). “Planning with Multiple Resource Constraints an Application to a Naval Planning Problem.” *Proc. of the 1st Conference on AI Applications*, Denver, CO, 410-416.

[12] Wilkens, D. E. (1990). “Can AI Planners Solve Practical Problems?” *Computational Intelligence*, 6(4), 232-246.

[13] Smith, S. F. (1989). “The OPIS Framework for Modeling Manufacturing Systems.” Technical Report, TR-CMU-RI-89-30, Carnegie Mellon University, The Robotics Institute.

[14] Zweben, M., Davis, E., M. Deale. (1993). “Iterative Repair for Scheduling and Rescheduling.” *IEEE Transactions on Systems, Man and Cybernetics*, 23(6), 1588-1596.

[15] 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." Knowledge Based Scheduling, Fox, M. and Zweben, M. (editors), Morgan Kaufmann, San Mateo, CA, 451-469.

[16] 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.

[17] Sycara, K. and Miyashita, K. (1993). "Schedule Repair through Case-Based Reasoning." Technical Report, The Robotics Institute, Carnegie Mellon University.

[18] Koo, B., Fischer, M. and Kunz, J. (2006). "Formalization of Construction Sequencing Rationale and Classification Mechanism to Support Rapid Generation of Sequencing Alternatives." Technical Report Nr. 167, CIFE, Stanford University, Stanford, CA.

[19] Echeverry, D., Ibbs, W., and Kim, S. (1991). "Sequence Knowledge for Construction Scheduling." Journal of Construction Engineering and Management, ASCE, 117(1), 118-130

[20] Kähkönen, K. (1993). "Modelling Activity Dependencies for Building Construction Project Scheduling," Technical Research Centre of Finland, VTT Publications, ESPOO.

[21] Aalami, F., Kunz, J., and Fischer, M. (1998). "Model-Based Sequencing Mechanisms Used to Automate Activity Sequencing." Working Paper Nr. 50, CIFE, Stanford University, Stanford, CA.

[22] 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.

[23] Chua, D. K. H., Shen L. J., and Bok, S. H. (2003). "Constraint-Based Planning with Integrated Production Scheduler over Internet." *Journal of Construction Engineering and Management*, ASCE, 129(3), 293-301.

[24] 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.

[25] Thomsen, J, Levitt, R., Kunz, J., Nass, C., Fridsma, D. (1999). "A Trajectory for Validating Computational Emulation Models of Organizations." *Journal of Computational & Mathematical Organization Theory*, 5(4), pp. 385-401.

[26] Clayton, M., Kunz, J., and Fischer, M. (1998). "The Charrette Test Method." Technical Report Nr. 120, CIFE, Stanford University, Stanford, CA.

Figure Captions

Figure 1. CPM bar chart for initial and modified sequence of Central Utility Building.

Figure 2. Role and status of activities classified based on the role and flexibility of constraints.

Figure 3. Manual steps 1 to 3 for expediting the activity Install Process Pipes B.

Figure 4. Manual steps 4 to 8 for expediting the activity Install Process Pipes B.

Figure 5. Manual steps 9 to 13 for expediting the activity Install Process Pipes B

Figure 6. Manual steps 14 to 16 for expediting the activity Install Process Pipes B.

Figure 7. Automated (system) and manual (user) steps for the identification and re-sequencing process

Figure 8. Priority rules for resolving workspace and resource conflicts using the role and status

of activities.

Figure 9. IDEF₀ of CLCPM prototype.

Figure 10. CLCPM user interface for selecting (a), creating (b) and customizing (c) specific constraints.

Figure 11. CLCPM used to automatically infer the role and status of activities for the Intel CUB Schedule.

Figure 12. CLCPM used to identify the Candidate Activities (CA) and develop a sequencing alternative that expedites the Target Activity (TA) by 4 days.

Figures

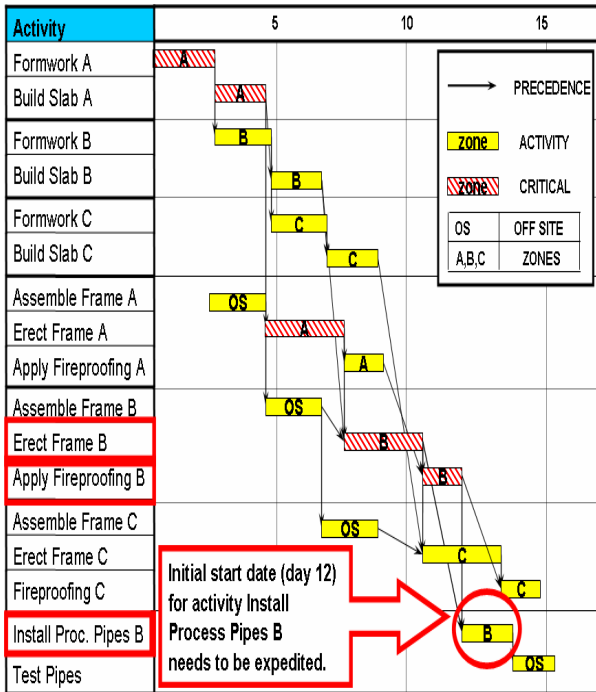


Figure 1a. Initial sequence for CUB.

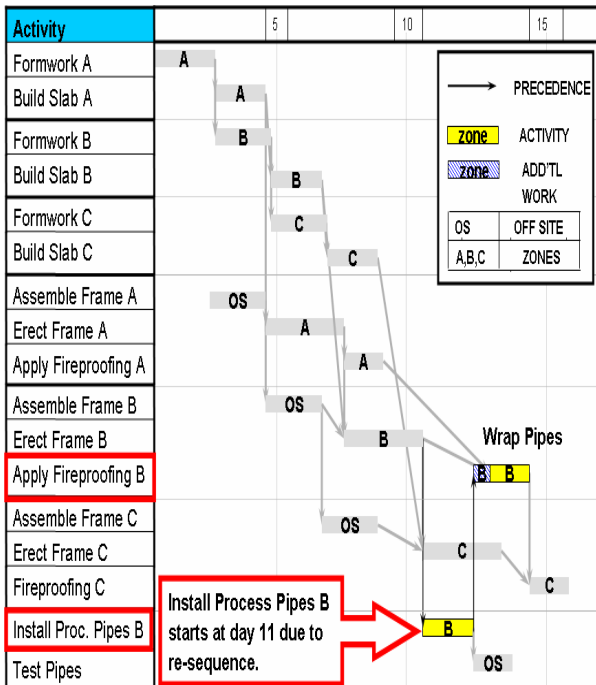


Figure 1b. Modified sequence for CUB.

Figure 1.

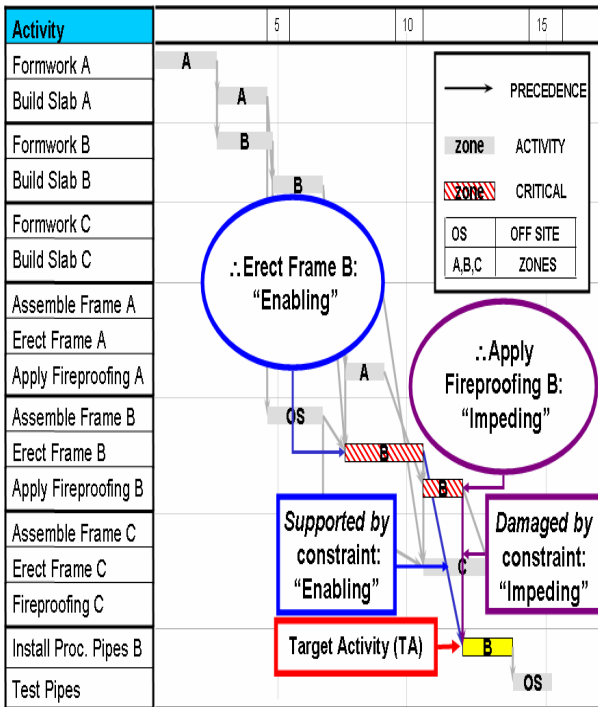


Figure 2a. Role of activities based on role of constraints.

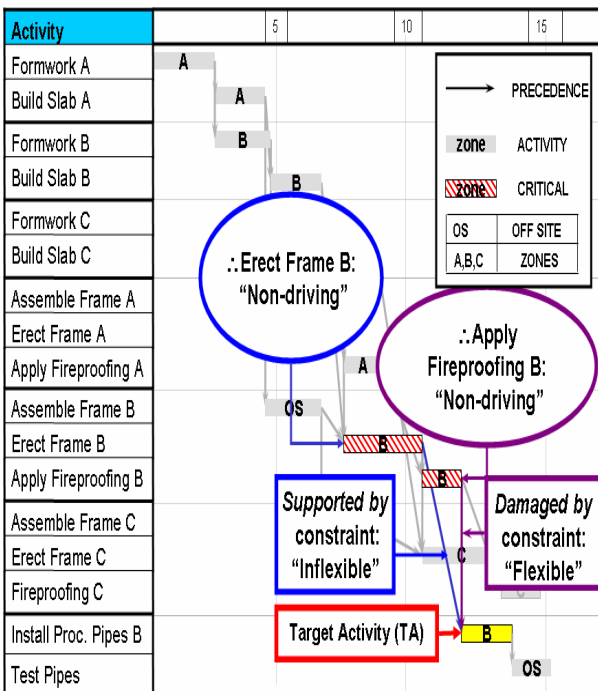


Figure 2b. Status of activities based on flexibility of constraints.

Figure 2.

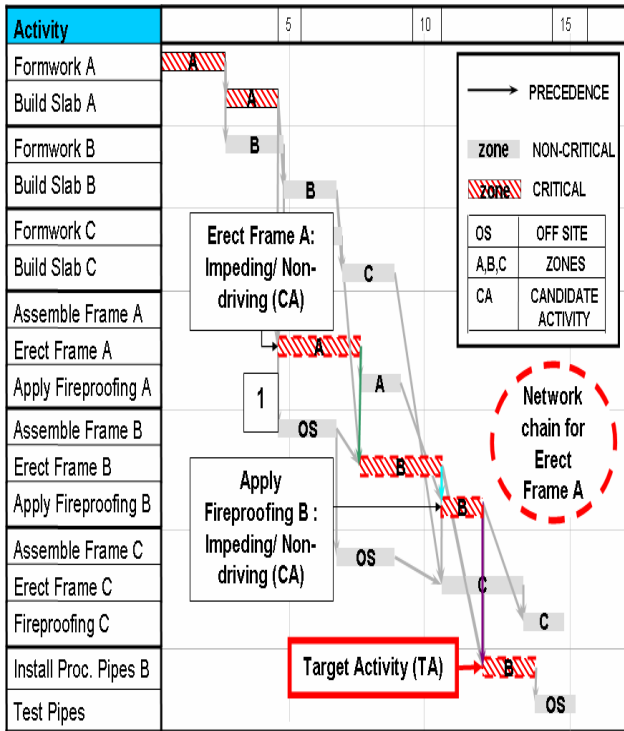


Figure 3a. Step 1.

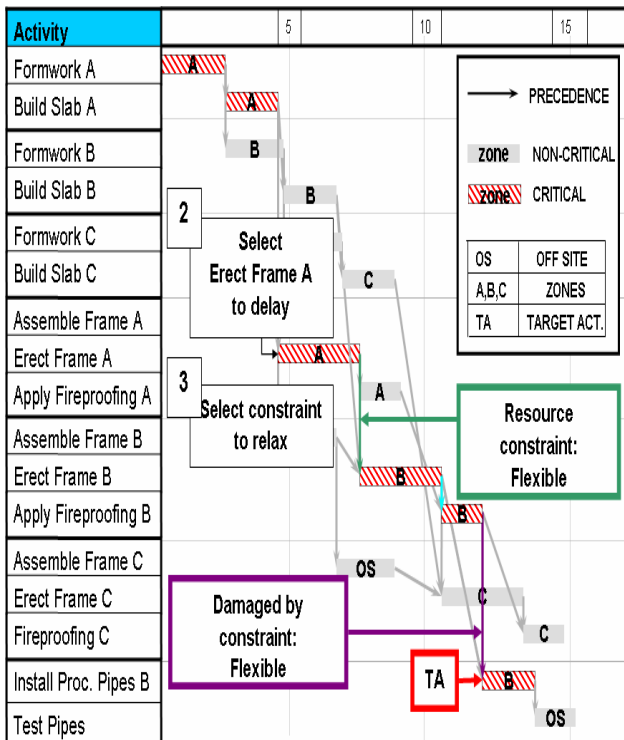


Figure 3b. Step 2-3.

Figure 3.

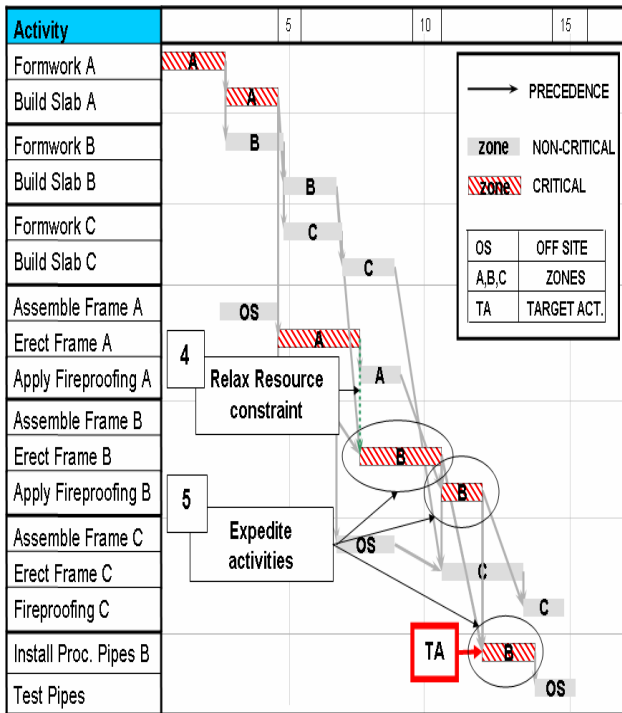


Figure 4a. Steps 4-5.

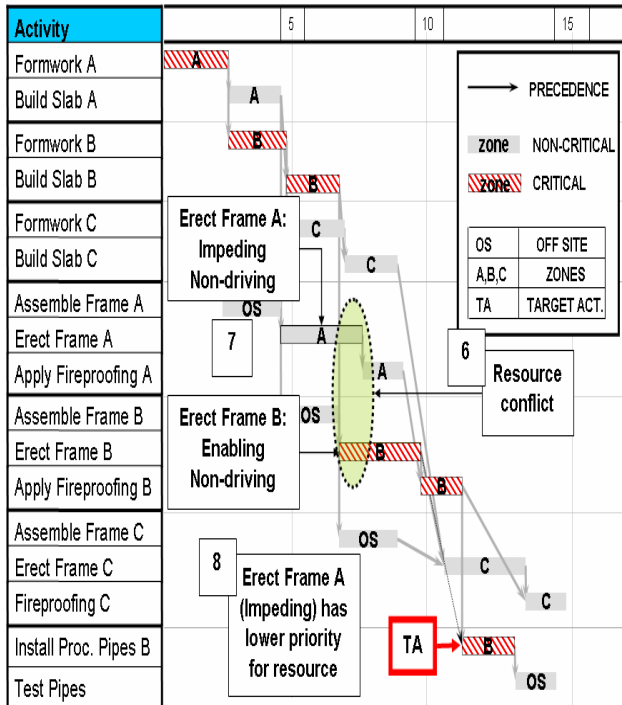


Figure 4b. Steps 6-8.

Figure 4.

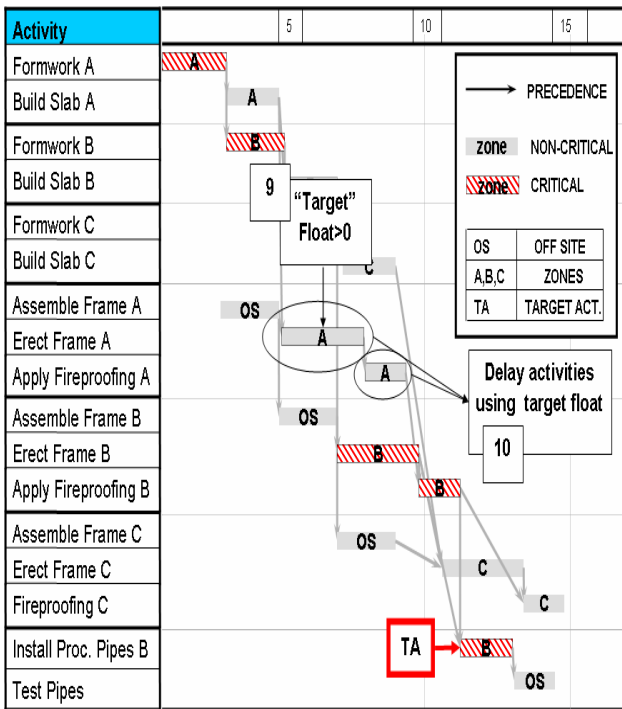


Figure 5a. Steps 9-10.

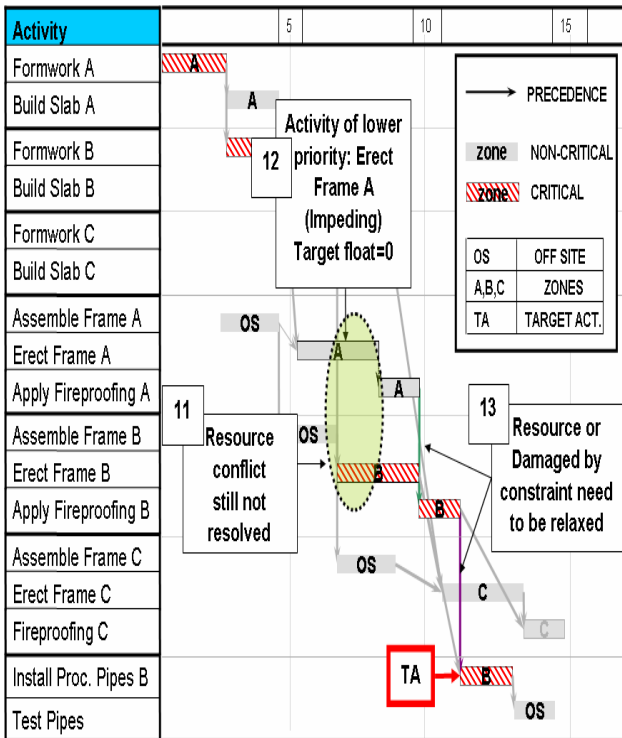


Figure 5b. Steps 11-13.

Figure 5.

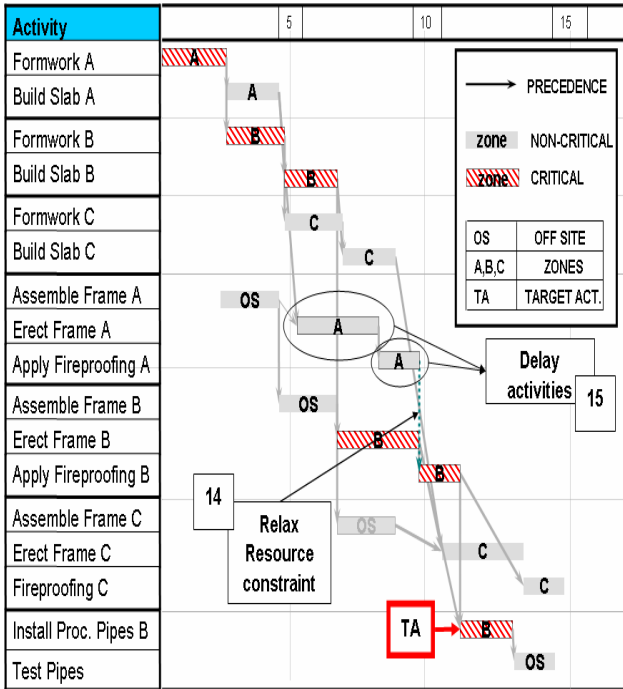


Figure 6a. Step 14-15.

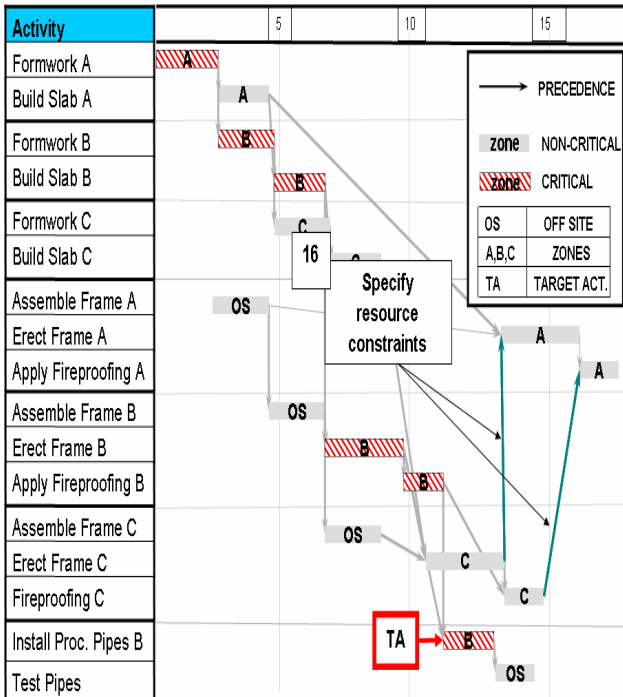


Figure 6b. Step 16.

Figure 6.

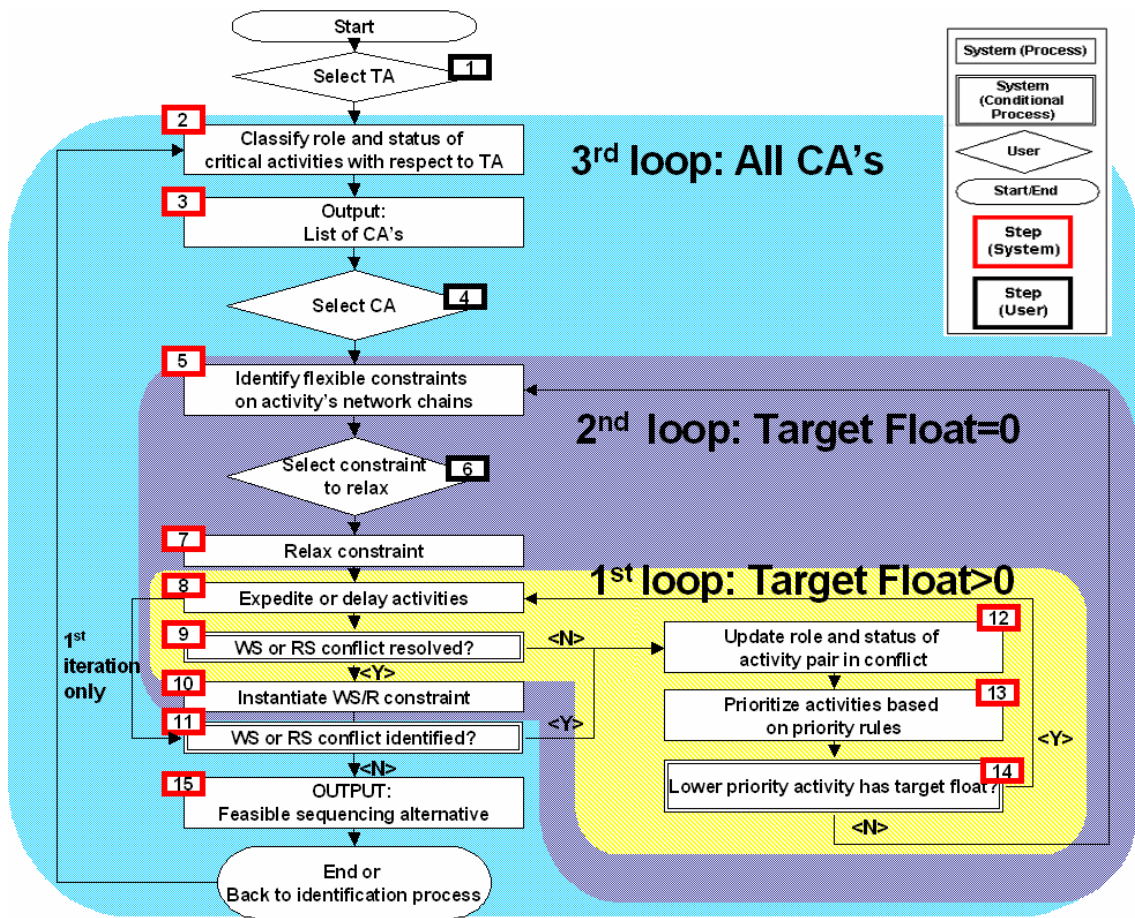


Figure 7.

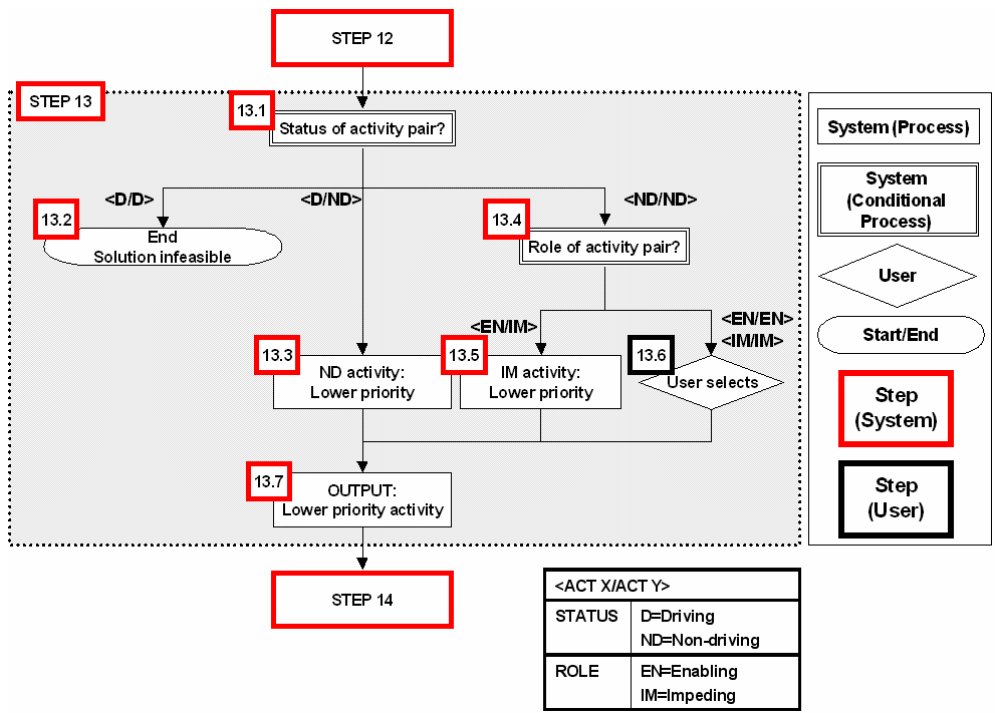


Figure 8.

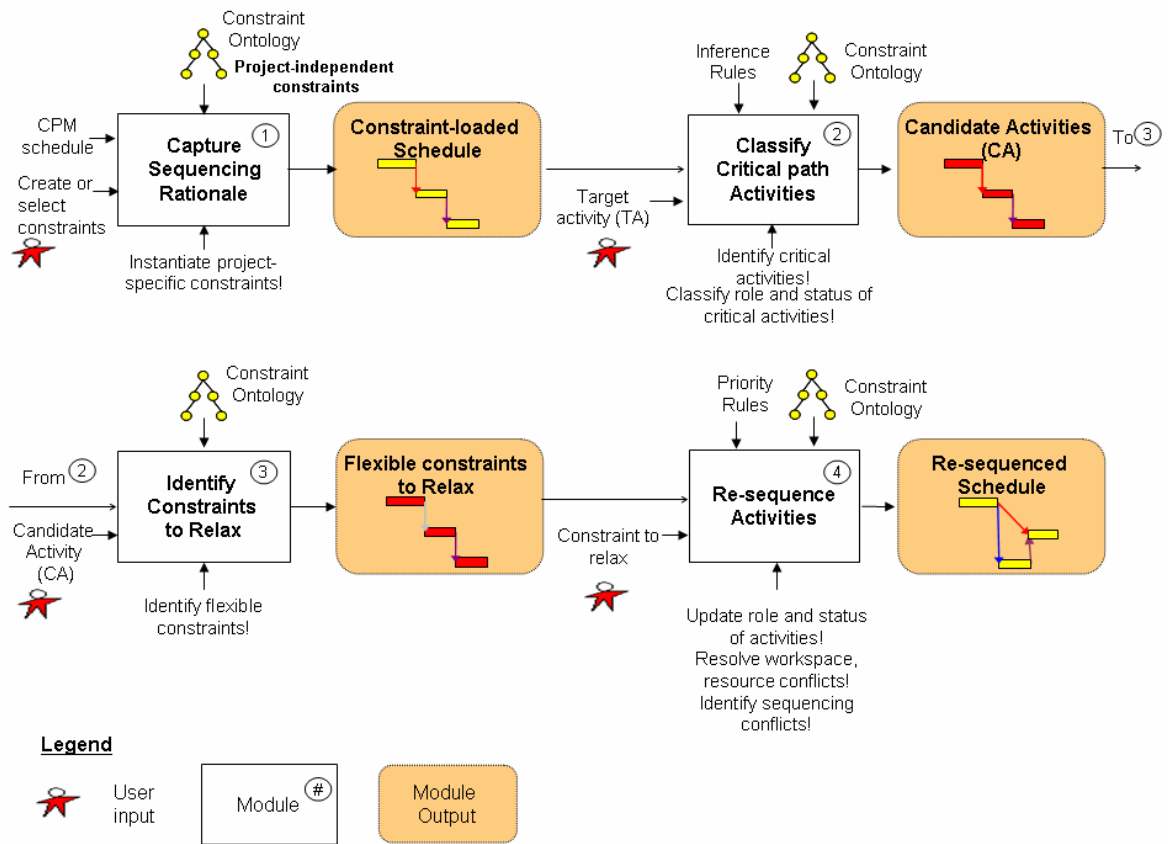


Figure 9.

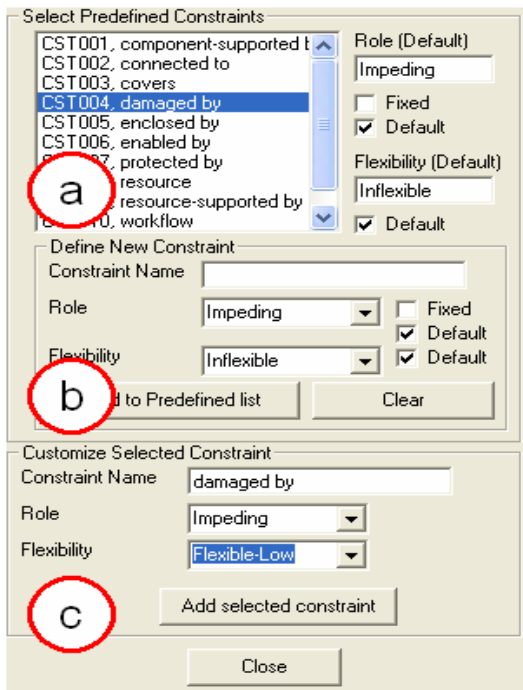


Figure 10.

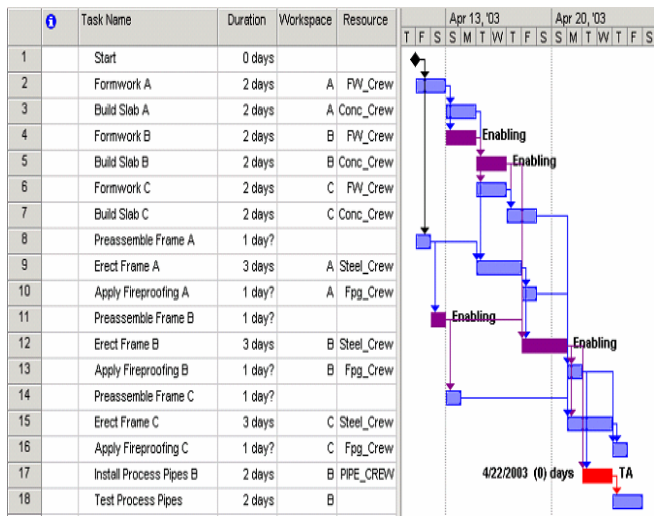


Figure 11a. Using CLCPM to identify the role of activities (enabling) for the Intel CUB schedule.

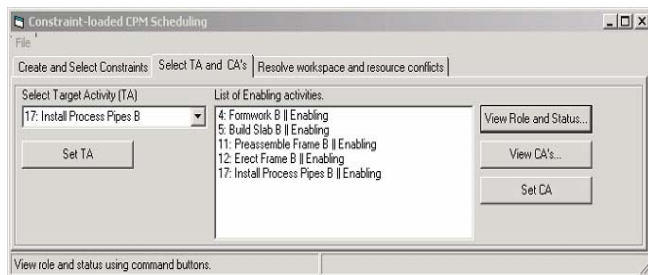


Figure 11b. CLCPM user interface shows list of enabling activities for the Intel CUB schedule.

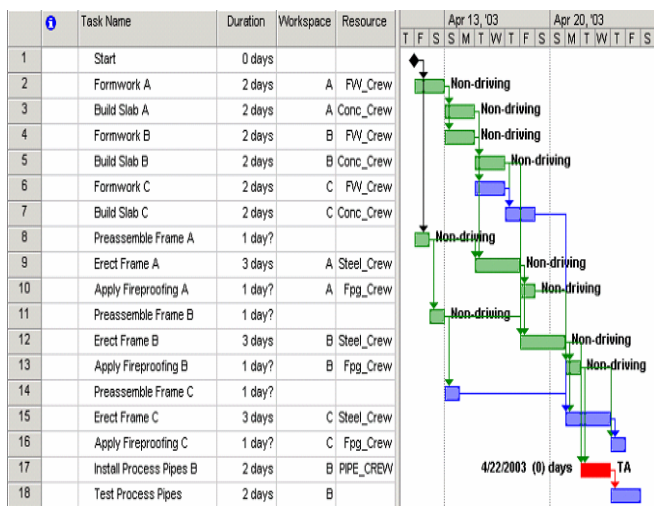


Figure 11c. Using CLCPM to identify the status of activities (non-driving) for the Intel CUB schedule.

Figure 11.

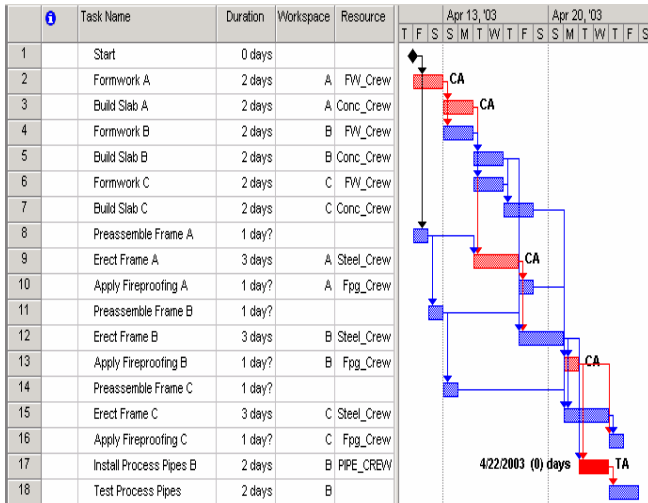


Figure 12a. Using CLCPM to identify the Candidate Activities (CA) for the Intel CUB Schedule.

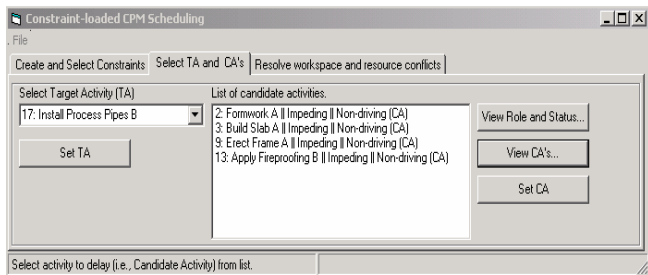


Figure 12b. CLCPM user interface shows list of Candidate Activities for the Intel CUB schedule.

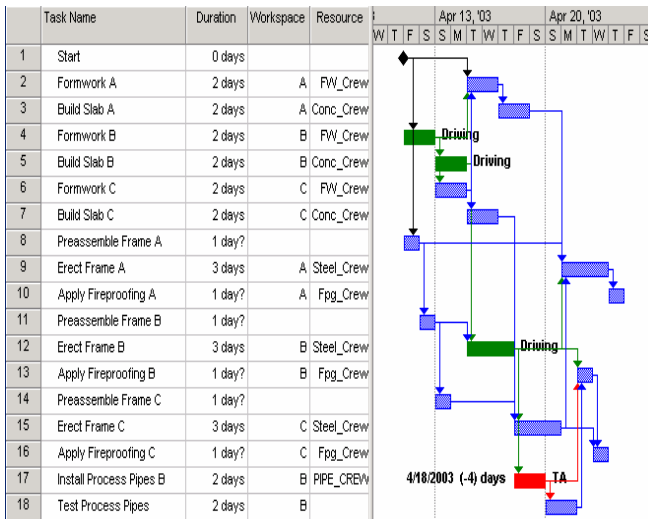


Figure 12c. Using CLCPM, the target activity Install Process Pipes B is expedited 4 days.

Figure 12.

Table Captions

Table 1. Project-independent constraints defined with default values for their role and flexibility.

Table 2. Summary of the manual steps and required formalizations for expediting the activity

Install Process Pipes B.

Table 3. Validation results using CLCPM for three retrospective cases.

Tables

Factor	Constraint	Role (default value)	Flexibility (default value)
Physical Component Relationships	Supported by	Enabling	Inflexible
	Connected to	Impeding	Inflexible
	Covered by	Enabling	Inflexible
	Enclosed by	Impeding	Flexible
	Closer to	Enabling	Flexible
	Protected by	Enabling	Flexible
Trade Interaction	Workspace	Impeding	Flexible
	Resource	Impeding	Flexible
	Damaged by	Impeding	Flexible
	Serviced by	Enabling	Inflexible
	Obstructed by	Impeding	Inflexible
Code Regulations	Safety	Impeding	Inflexible
	Inspection	Impeding	Inflexible
	Testing	Impeding	Inflexible

Table 1.

Processes	Required Steps	Steps in Sample Case (Figures 3 - 6)	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
	Expedite or delay activities	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
	Specify constraint	Step 16	Ontology

Table 2.

Projects	Intel CUB	McWhinney	Bay Street
Phase	Concrete and structural frame	Exterior closure	MEP and interior finishes
Target Activity	Install Process Pipes B	HVAC balance	Bookstore Turnover
Number of activities	18	27	27
Number of precedence relationships	25	37	36
Number of CA's (confirmed)	4 (4)	3 (3)	6 (6)
TA initial date	4/22	7/21	8/5
TA final expedited date	4/18(-4 days)	7/15(-6 days)	7/29(-7 days)
Number of sequencing alternatives (confirmed)	3 (3)	2 (2)	6 (6)

Table 3.