Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

EXPECTING THE UNEXPECTED: REPRESENTING, REASONING ABOUT, AND ASSESSING CONSTRUCTION PROJECT CONTINGENCIES

G. Ryan Anderson
Nilufer Onder

Department of Computer Science
1400 Townsend Dr., Michigan Technological University
Houghton. MI 49931, U.S.A.

Amlan Mukherjee

Civil and Environmental Engineering Department
1400 Townsend Dr., Michigan Technological University
Houghton. MI 49931, U.S.A.

ABSTRACT

Planning, scheduling and effective management of contingencies are crucial for the successful management of construction projects. In this paper we explore a mathematical representation of construction processes that can be used to infer alternative futures of a project as it unfolds. The representation has its foundations in temporal constraint networks. We present algorithms that can traverse the network in time, reason about the constraints driving a construction project, and present the combinatorial possibilities of futures that can emerge from one or more constraint violations during project implementation. The algorithms will aid construction managers to anticipate and react to crisis scenarios as they evolve in time. Our broader goal is to use the contingency information and the user responses to reveal the cognitive strategies used by humans to manage complex crisis scenarios.

1 INTRODUCTION

Planning, scheduling and effective management of contingencies are crucial for the successful management of construction projects. In this research effort the fundamental questions we have explored are: Can we build a system that will allow construction managers to anticipate and mitigate crisis scenarios dynamically during the execution of a construction project? Can such a system supplement the diverse body of discrete event simulations in construction? Finally, can it be used to complement human judgment and decision-making rather than automate the decision process - thus promising adaptive human-machine cognition in construction decision-making?

In this paper we explore a mathematical representation of construction processes that can be used to infer alternative futures of a project as it unfolds. The representation has its foundations in temporal constraint networks that can be used to represent and reason about constraints driving a construction project, and the combinatorial possibilities of futures that can emerge from one or more constraint violations during project implementation. We present algorithms that can traverse the network in time, as well as query probabilities of alternative futures and calculate associated risks. The algorithms will aid construction managers to anticipate and react to crisis scenarios as they evolve in time. In particular, with our representation and algorithms we can predict contingencies and calculate the likelihood and effect of contingencies during the implementation of a construction project. Our broader goal is to use the contingency information and the user responses to reveal the cognitive strategies used by humans to manage complex crisis scenarios. Our representation is sufficiently general to apply to existing Discrete Event Simulations (Hajjar and AbouRizk 2002, Martinez and Ioannou 1999) and the data collected can be used for post simulation visualizations (Kamat and Martinez 2001).

2 BACKGROUND

In this section, we briefly discuss two points of departure for this research: construction planning and simulation in civil engineering; and temporal reasoning and contingency planning in artificial intelligence.

2.1 Construction Simulations

The study of uncertainty in construction has predominantly involved estimating cost overruns and schedule delays in projects and estimating input parameters in simulating construction operations and processes. In both of these areas uncertainty is characterized by statistical distributions that describe the underlying variability. *Risk* is defined as a result of uncertainty, referring to a lack of predictability about structure, outcomes, or consequences during planning (Hertz and Thomas 1983) . *Risk analysis* thus involves estimating the probabilities needed as input data for the evaluation of decision alternatives.

Traditionally contingency is budgeted into construction cost estimates as a fixed percentage of the total cost

1-4244-1306-0/07/$25.00 ©2007 IEEE

(Mak and Picken 2000), based on previous experience with similar projects. Among other deterministic approaches, contingencies are calculated based on the risk associated with individual activities. Such approaches are limited in quantifying the degree of confidence associated with the contingencies identified. Touran (2003) explored probabilistic methods of assessing and allocating contingencies to construction projects. His premise was that the events causing delays and budget impacts during construction projects occur randomly according to a Poisson process. This assumption applies for independent events because Poisson processes require event arrivals to be independent and non-overlapping such as arriving change-orders. It can be applied during the early stages of project development and provides a probabilistic alternative to predicting contingencies more accurately than the deterministic approaches. In addition, the approach accounts only for events that are independent and does not take into account dependent events that occur due to cascading constraint violations during the project implementation.

Research in construction simulations has produced general purpose platforms such as Simphony (Hajjar and AbouRizk 2002) and STROBOSCOPE (Martinez and Ioannou 1999) that have been very useful in modeling construction processes and operations. They emphasize optimizing resource use and allocation. Simphony has been applied to simulating projects that are repetitive in nature, specifically tunneling construction projects (Hajjar and AbouRizk 1999). Such simulation systems have primarily focused on using statistical approaches to quantifying uncertainty associated with different model inputs and parameters to increase the accuracy of simulation output. A Bayesian method to update the input penetration rates of a tunnel boring machine was used in a Simphony simulation of a tunneling operation (Chung, Mohammed, and AbouRizk 2004). Similarly, STROBOSCOPE allows statistical distributions to be incorporated in the model to reflect variations in simulation input parameters (Lee and Arditi 2006).

The Simplified Discrete Event Simulation Approach (SDESA) enables modeling concurrent interruptions such as random equipment failures or prescheduled worker break times (Lu and Chan 2004). When these events are included in the simulation, activity end times can be predicted more accurately.

The approach in this paper is to explore uncertainties arising from the underlying structure of the construction management domain. Construction management domain structure representation captures relationships between processes and events, using semantics that provides the ability to automatically reason and infer about different project outcomes. One such research effort uses belief networks to model relationships between construction processes and events to calculate the risks arising from the combined interactions between the identified variables (McCabe, AbouRizk, and Goebel 1998, Nasir, McCabe, and Hartono 2003).

This work builds on the approach that uses temporal semantics (Mukherjee and Rojas 2003) to reason about construction activities and unexpected events resulting from constraint violations. This research extends the existing temporal semantics to represent activity schedule information, project constraints and associated variable information using an unified temporal constraint network. The network provides a platform to query different project futures resulting from violations of individual or combined constraints. Uncertainty associated with a particular future is its probability of occurrence conditional to what has already transpired. The system is dynamic because the network is constantly updated to reflect recent decisions taken by managers. In this paper we develop the ability to anticipate possible futures resulting from the defined constraints. We have limited the scope of this paper to predominantly focus on temporal constraints, though this approach can also be used to represent resource constraints.

2.2 Temporal Reasoning and Contingency Planning

Simple Temporal Networks (STNs) (Dechter, Meiri, and Pearl 1991) and their associated polynomial time algorithms are widely used in temporal reasoning. An STN includes nodes which represent events and edges between pairs of nodes which represent temporal constraints between events. The temporal constraints have a form such as $2 \leq x_2 - x_1 \leq 5$ meaning that the duration $[x_1, x_2]$ between time points x_1 and x_2 can be between 2 to 5 time units. Deadlines are represented by posing constraints between the starting time (T_0) and the desired time point. A *shortest path algorithm* is used to check if the temporal constraints are consistent. In construction domains, we need to explicitly represent the reasons for the variability of durations. We also need to reason about about the past, current, and future status of the world. Therefore we have developed and used an extension to STNs as explained in Section 3.2.

Contingency planners are systems that can compose plans with branches. The contingency branches are created automatically by reasoning about the setting in which the plans will be operated. In one setting, it is assumed that contingencies might arise, but execution time constraints will not allow the detection of these contingencies. Therefore, *conformant plans*, i.e., plans that contain steps to avoid or handle contingencies in advance of execution time are prepared (Bertoli, Cimatti, and Roveri 2001, Hyafil and Bacchus 2003, Hoffmann and Brafman 2005). The second setting includes *full-observability planners* which assume that every component of a state including contingencies will be detectable during execution (Boutilier, Dean, and Hanks 1999). Therefore, the plans contain steps that can handle every possible contingency upon its detection during execution. The third type includes *partial-observability planners* which constitute the middle ground: some contingencies are detectable, while others are not (Draper, Hanks, and Weld 1994, Majercik and Littman 1999, Bonet and Geffner 2000, Hansen and Feng 2000). Therefore, the plans contain steps suitable for the type of contingency.

It is important to note that contingency planners model only contingencies arising from unknowns at the initial state and the contingencies that result from executing plan steps. One notable exception is described in (Blythe 1994) where external events are modeled explicitly and Bayesian belief networks are used to compute the probability of plan failures. Another important note is that contingency planners do not reason about which contingencies should be planned for. The exceptions are (Onder and Pollack 1999) where the system can reason about which failure points are more important to consider, and (Meuleau and Smith 2003) where the number of contingency branches are limited for cognitive and computational simplicity.

The research in this paper builds on contingency planning by first shifting the focus from automatically constructing contingency plans to identifying possible contingencies and presenting them to construction managers for effective decision making. Second, our framework allows the recording of manager responses and this helps in understanding the cognitive models underlying human decision making under crisis situations.

3 REPRESENTATION

The representation developed in this paper builds on previous work (Mukherjee and Rojas 2003) on representing and reasoning about construction information. The construction management domain is conceptualized as a planning and constraint satisfaction problem. The project plan and schedule are developed in keeping with the constraints and the objective of minimizing the total project duration and cost. The constraints are resource constraints and /or temporal constraints. The former allocates resources (material, labor and equipment) to activities and the latter define relationships between the activities in time and space. A violation of one or more of these constraints can result in crisis scenarios. Events are defined as constraint violations, that may result from external happenings such as bad weather or from consequences of previous decisions that result in issues such as space conflicts between labor crews. We define each of the terms below.

3.1 Definitions and Assumptions

The input to the simulator is a project schedule that was prepared by a construction manager and consists of project activities and project constraints. Formally, an *as-planned schedule* is defined by a set of *activities*, $A_1, A_2, \ldots, A_k$, and a set of *constraints* $C_1, C_2, \ldots, C_n$. Each activity A_i has a beginning point $A_{i,b}$ and an ending point $A_{i,e}$. A constraint relates two time points and associates penalties with various degrees of delay. As an example consider a constraint of the following form:

$$\begin{array}{ll} 0 & : 1 \leq A_{1,e} - A_{1,b} \leq 5 \\ 1 & : 6 \leq A_{1,e} - A_{1,b} \leq 10 \\ \infty & : 11 \leq A_{1,e} - A_{1,b} \end{array}$$

The above constraints mean that no penalty is incurred if activity A_1 takes up to 5 days, a penalty of 1 is incurred if A_1 takes between 6 to 10 days, and the project will fail if A_1 takes longer than 10 days. When only a single time interval is associated with an activity, we write only the constraint and no associated penalties are written. In such a case, the penalty is zero if the constraint is satisfied, and the project fails it the constraint is violated. Each activity is assumed to take at least one unit of time. Thus, for each activity A_i, $1 \leq A_{i,e} - A_{i,b}$ is an implicit constraint. Theoretically, each of the k^2 time point pairs has an associated constraint. In practice, most of these constraints are null and only the activities that depend on each other have constraints.

The environment emulates activities, events and processes pertaining to construction projects. It is characterized by a set of entities, each of which describes a unique aspect of the environment. For example, weather and production rate are entities in the environment. An activity is an emulation of a real life construction operation and is represented by an interval which has the same length as its duration. Activity intervals are dynamic in nature, as activity durations may change during the construction process.

The values of the variables associated with an activity constitute a vector denoted by $\vec{V_i}$. In addition to activity specific variables regarded as "local" variables, we define a vector of "global" variables to represent the conditions that apply to every activity. As an example consider a simple environment with local variables pertaining to productivity ($v_{i,1}$) and labor status ($v_{i,2}$), and global variable weather ($v_{g,3}$). Assume that decking (A_1) and hoisting(A_2) are the activities. Then a snapshot of the environment is as follows:

$$\vec{V_1} = [100\% \text{ , working}] \tag{1}$$
$$\vec{V_2} = [\text{ } 0\% \text{ , on-strike}] \tag{2}$$
$$\vec{V_g} = [\text{ rain}] \tag{3}$$

An *event* may result from external happenings or as consequences of previous internal decisions that affects the environment. An event E is represented as a triple $< E_{prec}, E_{eff}, E_{prob} >$. E_{prec} is a set of variable values that enable the occurrence of the event. E_{eff} is the set of the effects the event has on the environment. The set of variables affected by an event can be different from the variables included in the preconditions. E_{prob} refers to the probability of event occurrence given that the enabling preconditions are satisfied. For example, a drop in labor productivity due to adverse weather can be represented as:

$$\begin{aligned} E_{prec} &= \{weather = snow, location = outdoor\}, && (4)\\ E_{eff} &= \{productivity = 50\%\}, && (5)\\ E_{prob} &= 0.75 && (6)\end{aligned}$$

An *event occurrence* at any point of time is a reassignment to the vector $\vec{V}_i$. Events can occur across one or more activities resulting in reassignment to the vectors corresponding to each of the activities.

3.2 The Traversal Algorithm

We represent the simulated world as a *Temporal Network with Activities and Events (TONÂE)* which is an extension of temporal constraint networks (Dechter, Meiri, and Pearl 1991). The inputs to the system are an as-planned schedule and a list of events that can occur. At the initialization phase, the as-planned schedule is converted into the TONÂE representation. In particular, the beginning and ending time points of each activity are represented as *activity nodes* (shown as circles in Figure 1). The temporal constraints are represented as labels on the edges connecting activity nodes. For example, in Figure 1, the link between $A_{1,b}$ and $A_{1,e}$ shows that activity 1 takes 2 units of time. The link between $A_{1,e}$ and and $A_{2,b}$ shows that activity 2 should start right after activity 1 ends. For clarity of exposition, we have shown only singleton constraints in the example. Our implementation can handle interval constraints and multiple constraints with associated penalties as described in the previous section.

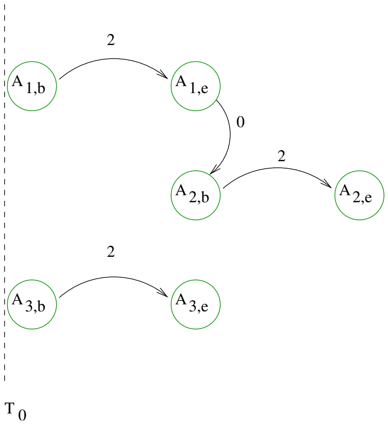


Figure 1: The as-planned project.

After the initial TONÂE is constructed, the simulation starts and changes the graph at every time tick. The main process is shown in Figure 2.

```
Require: An as-planned schedule APS, (activities and constraints);
    event information E-SET (set of events)
Ensure: A complete simulation of the construction environment.
WORK-G ← INITIALIZEGRAPH(APS).
while the project has not terminated do
    WORK-G ← ADVANCETIME(WORK-G,E-SET).
end while
return project statistics
```

Figure 2: The main traversal algorithm.

The heart of the process is the ADVANCETIME method which is shown in Figure 3. ADVANCETIME advances and keeps track of the state of the simulation using the concept of *present nodes*. At each time point, every activity that is ready to run is assigned a present node. A present node is shown as a triangle in Figure 4, and keeps all the information related to a running or finished activity. The values of all the variables specific to an activity are stored within this node. The link from an activity beginning node to a present node shows how much time has passed since the beginning of the activity. The link from a present node to an activity ending node shows duration of the remainder of the activity. For instance, in Figure 4, the links from $A_{1,b}$ and $A_{3,b}$ into their respective present nodes are both zero because both activities have started at the present time (T_0). In Figure 5 the present nodes are at time T_1. The link between $A_{1,b}$ and Y_1 is labeled 1 to reflect the fact that activity A_1 started 1 time unit ago.

```
Require: WORK-G, a TONÂE;
    event information E-SET (set of events)
Ensure: A simulation step of the construction environment.
for each running activity $a_i$ in WORK-G do
    E-SET$_i$ ← PIMGENERATE($a_i$, E-SET,“traverse”)
    for event $e$ in E-SET$_i$ do
        Create beginning and ending nodes for $e$ and link
        them to the present node of $a_i$, i.e., to $Y_i$.
    end for
end for
WORK-G ← COMPUTEEVENTEFFECTS(WORK-G).
WORK-G ← CALCREMAININGDURATION(WORK-G).

for each running activity $a_i$ in WORK-G do
    REMOVEENDINGEVENTS(WORK-G, $a_i$)
end for
WORK-G ← INCREMENTY(WORK-G).
return WORK-G
```

Figure 3: The ADVANCETIME algorithm.

The third and final type of node in a TONÂE is an event node which is shown as a square. Similar to activities, each event has a beginning point, an ending point, and a duration. An event such as snow is a global event and affects all the activities. An event such as a labor strike is local

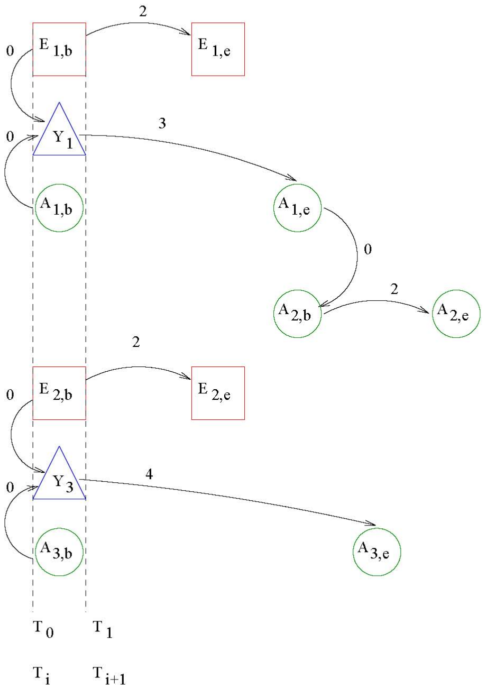


Figure 4: Project initialization at time T_0.

and affects a single activity. While simulating the passage of time, all the events that affect a particular activity are identified and inserted into the TONÂE. This is performed by calling the PIMGENERATE method shown in step 2 of Figure 3. The basic function of PIMGENERATE is to use the event descriptions to identify which external events might affect the running activities. For example, snow affects outdoor activities but not indoor activities. Recall that each event in E-SET has an associated probability of occurrence. The "traverse" parameter causes PIMGENERATE to sample from this probability to determine if an eligible event will actually occur. Each event that will occur is inserted into the TONÂE as shown in lines 3 through 5 of Figure 3. In this paper, our focus is to automatically infer and present possible crisis scenarios to the user. Therefore, we leave the details on how to compute event probabilities and on how to compute the combined effects of several events on the same variable outside the scope of this paper. We have implemented simple modules for PIMGENERATE, COMPUTEEVENTEFFECTS, and CALCREMAININGDURATION (the latter two are explained below.)

Once all the events are recorded in the TONÂE, their combined effects on the project activities must be computed. For example one event might cause 50% drop in productivity, while another causes 25%. Determining the exact rate of productivity requires domain knowledge an is handled by COMPUTEEVENTEFFECTS in line 7 of Figure 3. When the combined effects are resolved, the new values of the variables are recorded in the present node of each activity.

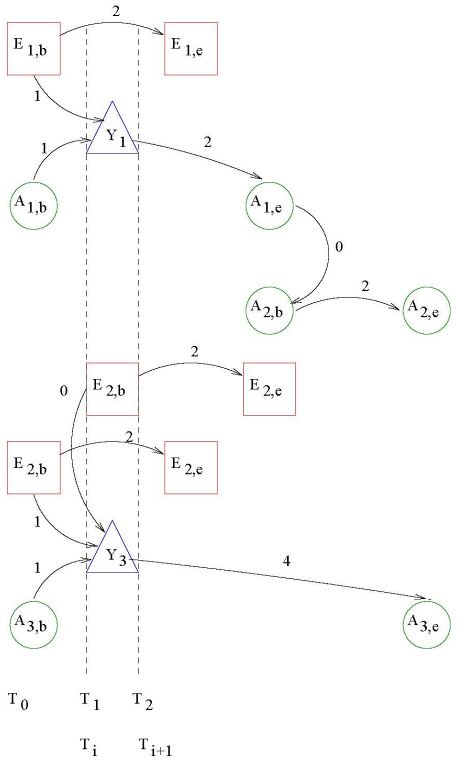


Figure 5: Project status at time T_1.

The next task is to determine the effect of the new variables on the activity durations. For example, a drop in productivity increases the duration of an activity proportionally. The exact proportions are again domain dependent and are handled by the CALCREMAININGDURATION method in line 8 of Figure 3. For example Figure 1 shows that the as-planned duration of activity A_3 is 2. After the insertion of event E_2 in Figure 4 the duration increases to 4, and after the insertion of event E_3 in Figure 5 the duration increases to 5 (5 comes from 1 + 4, the addition of the labels of the links from $A_{3,b}$ to Y_3 and from Y_3 to $A_{3,e}$).

The effects of extended durations are computed and the user is alerted if there are constraint violations. The next step is to advance the state of the simulation to the next time step. This is done in steps 9 through 12 of Figure 3. First, the effects of the events that are ending at the current time point are removed. Next, the present nodes and their associated links are updated to reflect the passage of time.

In the next section we describe the querying algorithm which computes the possible event combinations at a given instance.

3.3 The Querying Algorithm

The query algorithm (Figure 6) can be viewed as a depth-first exploration of the possible futures of a project. In lines 1 through 3, PIMGENERATE is called using the "query" parameter so that it returns all the possible events rather than a sampled set of events that will occur. In line 4, a Cartesian product of all the events are taken.

```
Require: WORK-G, a TONÂE;
    event information E-SET (set of events);
    $\theta$ probability of the status shown in WORK-G occurring
Ensure: A set of possible project outcomes.
 1: for each running activity $a_i$ in WORK-G do
 2:   E-SET$_i$ $\leftarrow$ PIMGENERATE($a_i$, E-SET,“query”)
 3: end for
 4: E-COMB $\leftarrow$ GENERATESUBSETS($\cup$ E-SET$_i$)
 5: for event-combination $ec$ in E-COMB do
 6:   for event $e$ in $ec$ do
 7:     Create beginning and ending nodes for $e$ and link
        them to the present node of $a_i$, i.e., to $Y_i$.
 8:   end for
 9:   WORK-G $\leftarrow$ COMPUTEEVENTEFFECTS(WORK-G).
10:   WORK-G $\leftarrow$ CALCREMAININGDURATION(WORK-G).
11:   for each running activity $a_i$ in WORK-G do
12:     REMOVEENDINGEVENTS(WORK-G, $a_i$)
13:   end for
14:   if the project ended then
15:     return project statistics
16:   else
17:     WORK-G $\leftarrow$ INCREMENTY(WORK-G).
18:     QUERY (WORK-G, E-SET, $\theta \times$ probability($ec$))
19:   end if
20: end for
```

Figure 6: The QUERY algorithm.

Each element of the resulting Cartesian product represents a possible future of the project. For example, assume that event E_1 with probability 0.2 can affect activity A_1, and events E_2 and E_3 can affect activity A_2 with probabilities 0.3 and 0.6, respectively, and the events are independent. Then there are 8 possible event combinations with the following probabilities: $\{$ $\{$<none, none, none>, $0.8 \times 0.7 \times 0.4\}$, $\{< E_1$, none, none>, $0.2 \times 0.7 \times 0.4\}$, $\{< E_1, E_2$, none>, $0.2 \times 0.3 \times 0.4\}$, $\ldots$, $\{< E_1, E_2, E_3 >$, $0.2 \times 0.3 \times 0.6\}$ $\}$. In the remainder of the query algorithm, steps similar to the traversal algorithm are taken for each event combination (the loop at line 5). After the state of the simulation is advanced, the algorithm is called recursively to consider the possible events in the next time step (line 18).

It is important to note that there are an exponential number of possible events at each time step. Our current work focuses on using Monte Carlo techniques that limit the number of possible futures shown to the user.

As expected, the above approach to querying produces an exponential number of possible project futures. To limit the number, we have implemented a combination of the traversal algorithm and Monte Carlo sampling. In this technique, a single element of the aforementioned Cartesian product is sampled based on its probability of occurence. This sampled future is recursively queried, which results in a future being sampled for the first sampled future. This recursion continues until the query determines that the project has either been completed or has failed. At this point, we have sampled a single, linear, threadlike future of the current time point in the simulation. This process is repeated N number of times, and the result of each iteration corresponds to a complete sample. The result is a collection of N most statistically probable futures which can then be analyzed for completion times and other information to be reported to the user.

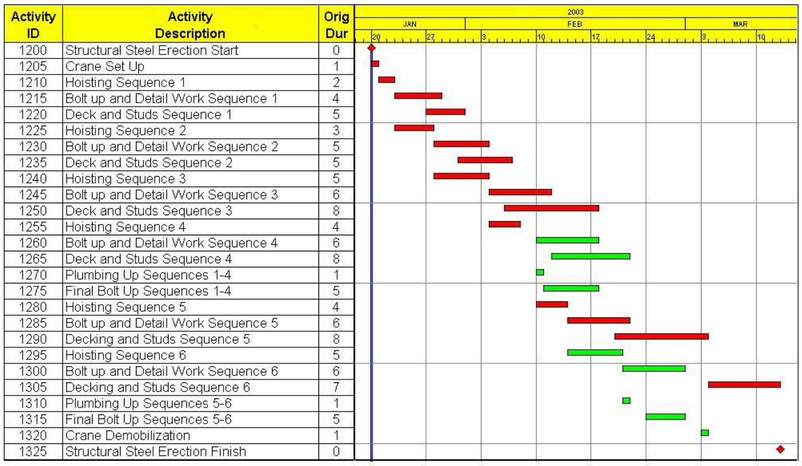
4 CASE STUDY: STRUCTURAL STEEL ERECTION

In this section we will analyze schedule information from a case study involving the construction of a steel frame building. The information for the case study compiled and documented (Daccarett and Mrozowski 2005) the construction process of the structural steel frame including detailing, fabrication and erection. In this paper we specifically focus on temporal relationships between activities. We briefly discuss how the traversal and querying algorithms can be applied to analyze problems such as the construction of a structural steel frame office building.

4.1 Project Description

The case study described involved the construction of a structural steel framed office building in 2003. The building has four stories, has 80,000 square feet of built area, weighs approximately 400 tons of structural steel or about 10 pounds per square foot. Fabrication and erection cost $9 per square foot. A total of 964 pre-fabricated structural steel members were used in the construction. The standard bay size in the building is 30 feet by 30 feet and there are 3 bays along the width and 7 bays along the length of the building. For this case study we used the schedule shown in Figure 7 and the construction plan used in constructing this project.

During the planning stages the steel frame was divided into 6 sequences as illustrated in Figure 9. Sequences represent the order in which a zone or section of the frame will be erected and were designed to improve efficiency of the erection process. The project was planned so that the construction operation involved in the sequences could be completed in parallel. Hence, while the erection crew was hoisting sequence 2 the decking crew could place the metal deck on the sequence 1 that had already been completed. This deck on sequence 1 in turn provided a work platform when the crew were hoisting sequence 3 above sequence 1. This reduced the fall distances and helped in satisfying work-site safety regulations because fall distances greater than 25 feet or two floors require separate fall protection on site (OSHA 1996).



Activity ID	Activity Description	Orig Dur
1200	Structural Steel Erection Start	0
1205	Crane Set Up	1
1210	Hoisting Sequence 1	2
1215	Bolt up and Detail Work Sequence 1	4
1220	Deck and Studs Sequence 1	5
1225	Hoisting Sequence 2	3
1230	Bolt up and Detail Work Sequence 2	5
1235	Deck and Studs Sequence 2	5
1240	Hoisting Sequence 3	5
1245	Bolt up and Detail Work Sequence 3	6
1250	Deck and Studs Sequence 3	8
1255	Hoisting Sequence 4	4
1260	Bolt up and Detail Work Sequence 4	6
1265	Deck and Studs Sequence 4	8
1270	Plumbing Up Sequences 1-4	1
1275	Final Bolt Up Sequences 1-4	5
1280	Hoisting Sequence 5	4
1285	Bolt up and Detail Work Sequence 5	6
1290	Decking and Studs Sequence 5	8
1295	Hoisting Sequence 6	5
1300	Bolt up and Detail Work Sequence 6	6
1305	Decking and Studs Sequence 6	7
1310	Plumbing Up Sequences 5-6	1
1315	Final Bolt Up Sequences 5-6	5
1320	Crane Demobilization	1
1325	Structural Steel Erection Finish	0

Figure 7: Case Study: Project schedule

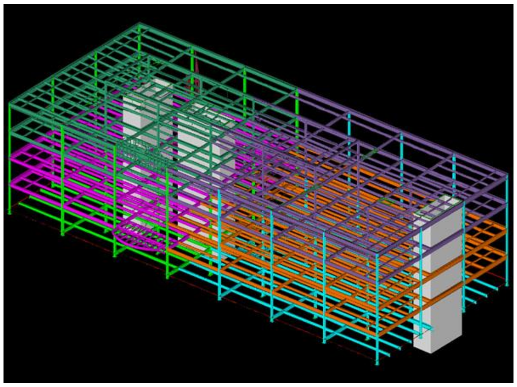


Figure 8: Case Study: Building steel frame

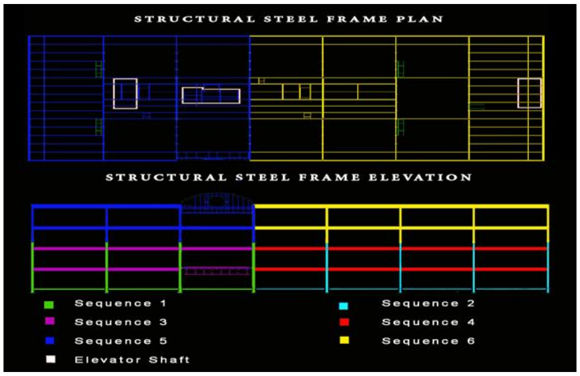


Figure 9: Case Study: Building steel frame

Figure 10 shows the completed hoisting activity for sequence 3. Sequences 3 and 4 consisted of 365 pieces of steel while sequences 1 and 2 consisted of 166 pieces of steel. This is a hoisting rate of about 36-40 pieces a day.

Seven people formed the erection crew used for this project. The crew consisted of one foreman, two iron workers hooking steel, two iron workers connecting steel, a crane operator, and an oiler. In order for the crews to be completely utilized it was important to minimize crew idle time. This meant that the hoisting crew would move from hoisting sequence to sequence with minimum time in between and would be followed in pursuit by the bolting and connection and decking crews.

In the next section we show how this narrative can be expressed using our system.

4.2 Constraint Information

In this section we will define the constraints that govern a very simple representation of this project using our proposed representation scheme. We will also explore how we can reason about events and possible futures given the identified constraints and our reasoning algorithms. Our primary information sources while studying this case study were the documented schedule information and the documented plans and project site photographs, some of which have been illustrated in Figures 8, 7, 9, and 10.

Based on the project narrative in the previous section we abstract the steel erection construction process to consist of construction of the six identified sequences. Erection of each sequence in turn consists of the following activities:

- Hoisting
- Bolting and Detail work
- Decking

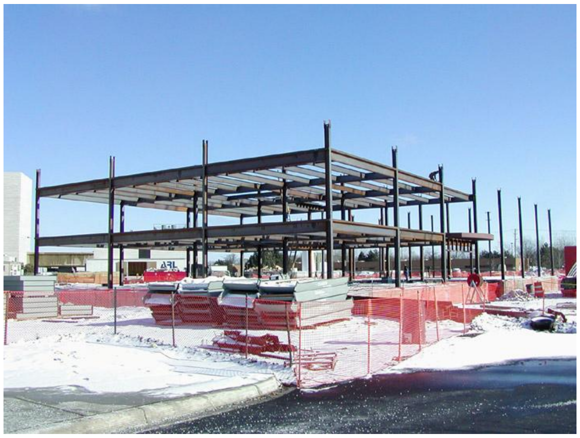


Figure 10: Case Study: After completion of sequence 3

Each of these activities need to be completed to erect a sequence. For the ith sequence each of these activities can be represented as $< H_{i,b}, H_{i,e} >, < B_{i,b}, B_{i,e} >, < D_{i,b}, D_{i,e} >$. Hence, according to our representation, as per the schedule information we can write the temporal constraint relationships for hoisting sequence 1 as:

$$\begin{aligned} 0 &: 1 \leq H_{1,e} - H_{1,b} \leq 2 \\ k^* &: 2 + k = H_{1,e} - H_{1,b} \end{aligned} \tag{7}$$

$$\begin{aligned} 0 &: 6 \leq B_{1,e} - B_{1,b} \leq 7 \\ k^* &: 7 + k = B_{1,e} - B_{1,b} \end{aligned} \tag{8}$$

$$\begin{aligned} 0 &: 5 \leq D_{1,e} - D_{1,b} \leq 6 \\ k^* &: 6 + k = D_{1,e} - D_{1,b} \end{aligned} \tag{9}$$

Constraints (7, 8, 9) express information about durations of each of the activities and the associated penalties when the activities get delayed. Constraint (7) expresses that the hoisting activity for sequence one should have a duration of no more than 2 days to not incur a penalty. For durations beyond 2 days or when there is a delay of k days, the activity incurs a penalty which is computed by a function of k denoted by k^*. This delay represents extra costs due to longer equipment usage and or longer labor hours.

Similarly we can write the relationships between the activities that comprise a single sequence as follows:

$$\begin{aligned} 0 &: 0 = B_{1,b} - H_{1,e} \\ k^* &: k = B_{1,b} - H_{1,e} \end{aligned} \tag{10}$$

$$\begin{aligned} 0 &: 0 \leq D_{1,b} - B_{1,e} \leq 0 \\ k^* &: k = D_{1,b} - B_{1,e} \end{aligned} \tag{11}$$

The temporal constraints (10, 11) describe the relationships between the activities in the construction of the first hoisting sequence. Constraint (10) requires that the bolting and the hoisting activities have a finish to start relationship with a lag of 0. Similarly constraint (11) ensures that the relationship between the bolting and decking activities is also finish to start with lag 0.

Next we define temporal relationships to describe the constraint between the construction of different sequences. The governing constraints that can be identified are:

- Constraint 1 (C_1): In order to maintain work-flow and efficiently utilize equipment lag time between hoisting activities for sequence i and $(i+1)$ should be = 0
- Constraint 2 (C_2): In order to maintain labor-flow and efficiently manage crew on site, the lag time between bolting and connection activities for sequence i and $(i+1)$ should be = 0
- Constraint 3 (C_3): In order to ensure safety and maintain appropriate fall distance during the hoisting operations, the decking operation for operation i should start before the hoisting operation for the sequence $(i+2)$

if the above equations are considered as hard constraints then, we can frame them as:

$$C_1 \rightarrow \quad \infty : H_{i+1,b} - H_{i,e} > 0 \tag{12}$$

$$C_2 \rightarrow \quad \infty : B_{i+1,b} - B_{i,e} > 0 \tag{13}$$

$$C_3 \rightarrow \quad \infty : H_{i+2,b} - D_{i,b} > 0 \tag{14}$$

Based on the violation of these constraints (7 - 14) events can be reasoned about. The simulation reasoning mechanism is required to identify violations of one or more of these constraints and then identify resulting scenarios. For example, if an activity is delayed, the traversal algorithm will propagate delays and calculate corresponding penalties. After that, the query algorithm will check with the different combinations of constraints C_1, C_2 and C_3 and the resulting futures.

5 CONCLUSION AND FUTURE WORK

We have presented a formal framework that can aid in construction decision making by reasoning about possible crisis situations. We implemented the basic traversal, delay propagation, and event generation functions. Our current work involves completing the query methods. This involves obtaining the possible futures of a project using the TONÂE representation and converting these results into an appropriate graph. For example, a linear schedule (Weber 2005, Ch. 14) can be constructed to show any differences between the expected and the simulated project where the simulation might exhibit buffer violations. This capability is an important feature of the TONÂE representation because users can utilize a representation that they are accustomed to, and view the results of the contingencies encountered.

Our future work involves taking into account the cognitive load of the construction managers, recording the decision making patterns of experienced and novice construction managers and incorporating the ability to learn and automatically suggest response plans to crisis scenarios.

ACKNOWLEDGMENTS

This work was supported by the NSF grant *SES 0624118* to Amlan Mukherjee. We thank the anonymous reviewers for their helpful comments, and MTU graduate students Collin Osenroth and Matt Watkins for their help.

REFERENCES

Bertoli, P., A. Cimatti, and M. Roveri. 2001. Heuristic search + symbolic model checking = efficient conformant planning. In *Proc. 17th Intl. Joint Conf. on Artificial Intelligence (IJCAI-01)*, 467–472.

Blythe, J. 1994. Planning with external events. In *Proc. 10th Conf. on Uncertainty in Artificial Intelligence (UAI-94)*, 94–101.

Bonet, B., and H. Geffner. 2000. Planning with incomplete information as heuristic search in belief space. In *Proc. 5th Intl. Conf. on Artificial Intelligence Planning & Scheduling (AIPS-00)*, 52–61.

Boutilier, C., T. Dean, and S. Hanks. 1999. Decision theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Chung, T. H., Y. Mohammed, and S. M. AbouRizk. 2004. Simulation input updating using bayesian techniques.

Daccarett, V., and T. Mrozowski. 2005. Aisc digital library: Construction management of steel construction. Available via <http://www.aisc.org/> [accessed 01/10/2007].

Dechter, R., I. Meiri, and J. Pearl. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.

Draper, D., S. Hanks, and D. Weld. 1994. Probabilistic planning with information gathering and contingent execution. In *Proc. 2nd Intl. Conf. on Artificial Intelligence Planning Systems (AIPS-94)*, 31–36.

Hajjar, D., and S. AbouRizk. 1999. Simphony: an environment for building special purpose construction simulation tools.

Hajjar, D., and S. AbouRizk. 2002. Unified modeling methodology for construction simulation. *Journal of Construction Engineering and Management* 128 (2): 174–185.

Hansen, E. A., and Z. Feng. 2000. Dynamic programming for POMDPs using a factored state representation. In *Proc. 5th Intl. Conf. on Artificial Intelligence Planning & Scheduling (AIPS-00)*, 130–139.

Hertz, D. B., and H. Thomas. 1983. *Risk analysis and its applications*. New York: Wiley.

Hoffmann, J., and R. I. Brafman. 2005. Contingent planning via heuristic forward search with implicit belief states. In *Proc. of 15th Intl. Conf. on Automated Planning & Scheduling (ICAPS-05)*, 71–80.

Hyafil, N., and F. Bacchus. 2003. Conformant probabilistic planning via CSPs. In *Proc. 13th Intl. Conf. on Automated Planning & Scheduling (ICAPS-03)*, 205–214.

Kamat, V., and J. C. Martinez. 2001. Visualizing simulated construction operations in 3d. *Journal of Computing in Civil Engineering, ASCE* 15 (4): 329–337.

Lee, D., and D. Arditi. 2006. Automated statistical analysis in stochastic project scheduling simulation. *Journal of Construction Engineering and Management* 132 (3): 268–277.

Lu, M., and W.-H. Chan. 2004. Modeling concurrent operational interruptions in construction activities with simplified discrete evetn simulation approach (SDESA.

Majercik, S. M., and M. L. Littman. 1999. Contingent planning under uncertainty via stochastic satisfiability. In *Proc. 16th National Conf. on Artificial Intelligence (AAAI-99)*, 549–556.

Mak, S., and D. Picken. 2000. Using risk analysis to determine construction project contingencies. *Journal of Construction Engineering and Management, ASCE* 126 (2): 130–136.

Martinez, J., and P. Ioannou. 1999. General-purpose system for effective construction simulation. *Journal of Construction Engineering and Management* 125 (4): 265–276.

McCabe, B., S. M. AbouRizk, and R. Goebel. 1998. Belief networks for construction performance diagnostics. *Journal of Computing in Civil Engineering, ASCE* 12 (2): 93–100.

Meuleau, N., and D. Smith. 2003. Optimal limited contingency planning. In *Proc. 19th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-03)*, 417–42.

Mukherjee, A., and E. Rojas. 2003. Reasoning about actions and events in situational simulations.

Nasir, D., B. McCabe, and L. Hartono. 2003. Evaluating risk in construction - schedule model (eric-s): Construction schedule risk model. *Journal of Construction Engineering and Management, ASCE* 129 (5): 518–527.

Onder, N., and M. E. Pollack. 1999. Conditional, probabilistic planning: A unifying algorithm and effective search control mechanisms. In *Proc. 16th National Conf. on Artificial Intelligence (AAAI-99)*, 577–584.

OSHA 1996. Construction safety and health outreach program: Steel erection. Available via <http://www.osha.gov/> [accessed 3/30/2007].

Touran, A. 2003. Probabilistic model for cost contingency. *Journal of Construction Engineering and Management, ASCE* 129 (3): 280–284.

Weber, S. C. 2005. *Scheduling construction projects: Principles and practices*. New Jersey: Pearson Prentice Hall.

AUTHOR BIOGRAPHIES

G. RYAN ANDERSON is an M.Sc. student at the Department of Computer Science at Michigan Technological University. His research interests include artificial intelligence, cognitive science, operating systems, temporal reasoning, planning, and artificial intelligence applications in construction management. His web page can be found via <http://www.cs.mtu.edu>.

NILUFER ONDER is an Associate Professor at the Department of Computer Science at Michigan Technological University. Her research interests are artificial intelligence, planning, planning under uncertainty, and decision making under uncertainty. Her web page can be found via <http://www.cs.mtu.edu>.

AMLAN MUKHERJEE is an Assistant Professor at the Department of Civil and Environmental Engineering at Michigan Technological University. His research interests are artificial intelligence technologies such as agent based modeling and temporal logic, construction engineering and management, interactive simulations, construction engineering education, and cognitive modeling of expert decision making. His web page can be found via <http://www.cee.mtu.edu>.