Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

REASONING ABOUT ACTIONS AND EVENTS IN SITUATIONAL SIMULATIONS

Amlan Mukherjee

Department of Civil Engineering
University of Washington
Seattle, WA 98195, U.S.A.

Eddy M. Rojas

Department of Construction Management
University of Washington
Seattle, WA 98195, U.S.A.

ABSTRACT

In this paper we have applied an interval representation of time to represent and reason about activities, events, actions and situations relevant to the construction domain. The first part of the paper formally defines the situational simulation environment and develops a set of temporal axioms which can be used to 1) Express precedence constraints between time intervals and 2) Capture the causal relationships between actions and events. The second part of the paper looks at an agent reasoning mechanism used to perceive and predict actions and foresee future consequences of present actions within the simulation environment. Agent reasoning is based on awareness derived from a knowledge base of facts which captures the causal nature of events in the construction management domain.

1 INTRODUCTION

1.1 Situational Simulations in Construction Engineering and Management

Barab et al. (2001) argue that the core of cognitive science and the resultant pedagogical models are based on the Cartesian philosophy of mind-matter dualism. This has separated the learner and the learning context resulting in a disembodiment of concepts from their contexts. Situational Simulations are temporally dynamic clinical exercises with the objective of exposing participants to rapidly unfolding events and to the pressures of quick decision making. When implemented through virtual environments situational simulations can provide participatory and contextually rich educational environments. In a field like construction engineering and management, where context specific knowledge and awareness is imperative, situated learning promises to challenge students' capabilities and improve their understanding of concepts and their inter-relationships.

In the field of construction engineering and management, this understanding has led to efforts at creating various gaming and simulation environments such as Superbid (AbouRizk 1993), STRATEGY (McCabe et al., 2000), ICMLS (Sawhney et al., 2001) and VIRCON (Jaafari et al., 2001). Some of these efforts have been inspired by earlier research efforts in the area such as CONSTRUCTO (Halpin, 1970).

1.2 Planning and Constraint Satisfaction in Construction Management

Management in construction may be regarded as a problem in planning. Project management tools like Microsoft Project are used to schedule construction projects. Work Breakdown Structures (WBS) allow planners to organize the project into lists of inter-dependent activities. The Critical Path Method (CPM) further uses activity network diagrams to plan and control the progress of the construction process.

A closer look at the activity network diagram reveals that it is a directed graph, in which vertexes represent activities and edges represent constraints. The direction of the graph provides the precedence order of the activities. The edges can be treated as precedence constraints. Two activities connected by an edge are constrained by a precedence relationship. Precedence constraints reflect finish-to-start, start-to-finish, finish-to-finish and start-to-start relationships between activities. Constraints are also defined by the availability of resources. Activities are dependent on the availability of specific quantities of labor, equipment and material. In the absence of the necessary resources, activities cannot be successfully completed. Resource constraints and precedence constraints are activity and project specific, and are defined during the planning phase of the construction project.

There is a third element that deserves attention. During the construction process precedence and resource constraints are violated by various events which occur in the construction environment. For instance, a day of bad weather could create delays in all outdoor activities. Such an event would have a cascading effect on the whole activity network and delay future activities bound to it by precedence constraints. Similarly, resource constraints may be violated in the event of a material delivery being delayed. This introduces the problem of dealing with the effects of various events occurring in the construction environment during the implementation of the project.

The above understanding allows us to conclude that the construction management domain can be divided into two different phases. The first phase can be modeled as a planning and constraint satisfaction problem, during which the most optimal plan, in terms of time and cost, is developed keeping in mind well defined project specific precedence and resource constraints. The solution to this problem usually results in the 'As-Planned' schedule. The second phase is the implementation phase during which the construction manager has to deal with events and respond to constraint violations by taking corrective measures.

1.3 Reasoning in Situational Simulations

Situational simulations concentrate on simulating the construction environment and the generation of events during the implementation phase. Constraint violations resulting from event occurrences, are situational scenarios. Participant skills are tested by how well they can take corrective measures to satisfy violated constraints.

As discussed earlier, precedence and resource constraints are specific to projects which are being simulated. Events on the other hand are causal in nature and relate to the construction environment. Projects built in similar environments would tend to face the same kind of events. For instance during the implementation phase most construction projects need to deal with events like space shortage, labor strikes and untimely material delivery, which can impact or delay the 'As-Planned' schedule of work.

Representation of information, pertinent to the construction engineering and management domain, based on an accurate abstraction of real life construction processes is the first step in creating situational simulation environments. Such environments should also have the capacity to reason about inter-relationships and abstract rules pertinent to construction processes, besides being responsive to participant interaction and design. This will allow the simulations to create consistent situations and evolve with the progress of time. It may be useful to develop a representation for precedence and resource constraints, and the causal relationships between events. The semantics of the representation can be used by an agent to reason about constraint violations in the present and possible futures of the environment.

1.4 The Agent

Event generation in the simulation is distributed between two entities, the Agent and the Event Generator. The agent is the reasoning arm, while the Event Generator is the executive arm. The functions of the agent are two fold: The agent reasons and advises the event generator on the possible events that can be generated in a given environment. It also reasons about possible events that will be triggered in the future or present because of participant interactions in the immediate past. The scope and implementation of the event generator is beyond the scope of this paper, so actions taken by the event generator on the agent's advice will be referred to as 'Agent actions' since it is important to understand the impact of these actions on the environment. A later section in the paper deals in depth with agent reasoning.

2 REPRESENTATION

This section uses the language of First Order Logic to formally represent information about situational simulations for the construction domain. First order logic can be used to represent domains which are composed of objects which have individual identities and properties that distinguishes them from other objects (Russell & Norvig, 2002). The objects are also related to each other by relations and functions. Knowledge representation and reasoning is widely studied using first order logic.

To start with, the situational simulation environment is formally defined and characterized. The second part of this section looks at notions of an interval represention of time. Based on definitions of the environment and axioms of time, actions and events are defined. The section concludes with a justification for using an interval representation of time in representing actions and events in the situational simulation.

2.1 Definitions

2.1.1 Time-Points and Time Intervals

Time-points are always represented by positive integers, and signify specific points in the continuum of time. In the simulation it is defined as the smallest discrete interval of time within the scope of the simulation and is referred to as the *discrete granularity*($\wp$).

A series of consecutive time points make up a time interval. Every time interval is associated with an ordered pair of integers. The integers define the start and end time-points of the time interval. Hence, a time interval i which stretches from the third day to the fifth day of the simulation, will map onto the ordered pair *{3,5}* where the discrete granularity of the simulation is a day. When a time interval represents an activity, the time-points represent the early start and early finish points for the activity.

$$\forall i \cdot \exists J, K \cdot J < K. \tag{1}$$

The interval duration is given by $K - J + 1$

$$i : \{J, K\};\ i.start = J,\ i.end = K$$

The convention followed in this paper is that all time points are represented in the upper case, while all time intervals are represented in the lower case.

2.1.2 Environment

The environment sets the scene for the situational simulation. It is the participant's perception of the simulated construction project. It is interactive, temporally dynamic and virtual in nature. The environment emulates activities, events and processes pertaining to construction projects. It is characterized by a set of entities, each of which describes an unique aspect of the environment. For example, weather and production rate are entities in the environment.

2.1.3 Activity

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. Activities take time from start to completion. Activity intervals are dynamic in nature, as activity durations may change during the construction process.

A two-dimensional time-activity plane, Figure 1, is helpful in visualizing how activities span time. It is similar to the Gantt Chart representation of the activity schedule. The time-activity plane has simulation time represented on the x-axis and the activity intervals on the y-axis. The y-axis is discrete and each unit represents an activity. On the time axis, each unit represents the discrete granularity of the simulation which is the smallest unit of time in the simulation and is represented by a time point.

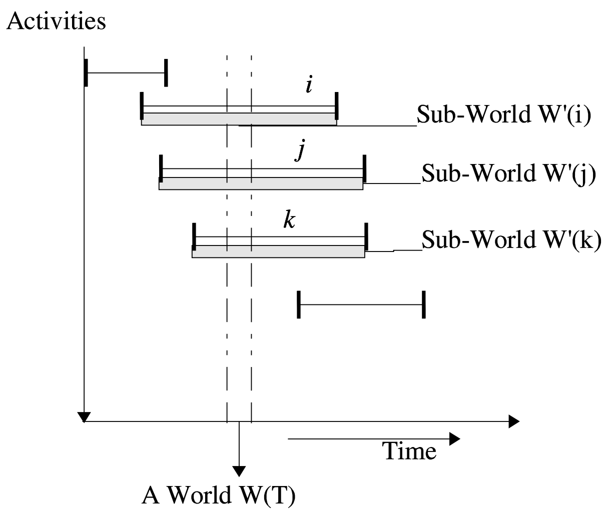

Figure 1: Worlds and Sub-Worlds in the Activity-Time Plane

2.1.4 Variables

A Variable is a symbolic representation of an entity. They are discrete in nature and can only take up values from the discrete finite domain $[s_1, s_2, s_3]$ which is the set of all possible attributes of the entity described by the variable. Every entity has an unique set of attributes. Hence, every variable maps on to an unique domain. A set of variables completely characterizes the environment. Precedence and resource constraints are relationships between the variables and determine the values taken up by the variables. The environment (E) being a composition of the enities is expressed as a set of the variables defining its entities.

$$E = [v_1, v_2, v_3, \cdots v_n];$$

$$where\ Domain\ of\ \ v_i = [s_1, s_2, s_3]$$

In the above equations the symbol v_i is a variable which describes an unique entity v_i in the environment. There is a closure on the set of variables in the environment. Hence as the simulation progresses, the set of variables may take up different attributes from the domain, but new variables may not be added to the set.

A combination of n such variables completely defines the simulation environment, as expressed in the second equation. As the environment changes, variables take up different values from their domains to reflect the change. This information is expressed through boolean predicates, which state the truth regarding time intervals over which variables hold particular values. The truth that the entity represented by the variable v_i has the attribute value s_1 over the time interval t is represented by the predicate:

$$v_i(s_1, t)$$

For example, the entity Weather is represented by the variable $weather$ which can take values from the domain $[sunny, rainy, snowy]$. The predicate $weather(sunny, t)$ signifies that the weather in the environment will hold sunny over the time interval t.

Reasoning in the environment uses conditions which are conjuctive clauses of the predicates. Hence, the condition representing snowy weather and null productivity over the interval t is represented by the sentence:

$$weather(snowy, t) \wedge prod(null, t)$$

Variables are homogeneous over the time intervals in which they hold. Unless otherwise altered, it may be assumed in the above example that the weather will hold snowy for the entire interval t.

2.1.5 Variable Classification

Just as entities in the environment are represented by variables, similarly activities are represented by variables too. Often these variables are specific to the activities that they describe. This calls for a classification of variables in the environment. All variables which describe entities specific to particular activities are *Activity Specific Variables*; variables which describe entities that are relevant to the whole time-activity plane are *Global Variables*. For example, weather is a global variable since its effects can be felt across activities. However, the availability of a particular material or special equipment may be specific to a particular activity. The unavailability of earth-moving equipment will not effect a concrete pouring operation, even though it may delay a concurrent earth moving operation.

It is possible for more than one concurrent activity to have instances of the same variable with differing values at the same point of time. For example, the labor efficiency for an activity might be 100%, while that of a concurrent activity may be 80%. Thus, the activity specific variable representing labor efficiency can take up two different values in two different contexts at the same time. We define *contexts* as dynamic time intervals identical to the activity intervals. For all intents and purposes, the contexts are set by activities. Activity specific variables are always in context to the activity which they define and at any point of time there can be multiple instances of the same activity specific variable, each in context to a different activity. However, within the same context no variable can have multiple instances, as that would mean an entity having more than one attribute at the same time.

This understanding helps us formally define *Activity Specific Variables* as variables which can have multiple instances across contexts at the same time, and *Global Variables* as variables which can only have a single instance across all contexts at every point of time.

2.1.6 World

A world is a snap shot of the environment at a specific time point T, as shown in Figure 1. The time point is the granularity of the simulation and is usually represented as a day. Progress from day to day in the real world translates to progress from time point to time point in the simulation. The simulation thus moves from one world to the next. Symbolically the world at the time point T is denoted by $W(T)$ which is given by the set of all variables which defines the environment at that time point.

$$W(T) = E|_T$$

2.1.7 Sub-World

The set of all variables in the environment which belong to the same context is defined as a sub-world. The sub-world is therefore a subset of the world where all the variables are specific to a particular activity which defines the context, as shown in Figure 1. For the context defined by the activity interval i the sub-world at the time point T is the set of m variables which describe entities in the activity and is denoted by:

$$W'(i) = \{v_{i1}, v_{i2}, \cdots v_{im}\} \in E|_T$$

At any time point T if the ongoing activity intervals defining the concurrent contexts are i, j, k then we can say that the set of environment variables is given by:

$$W(T) = [W'(i) \cap W'(j) \cap W'(k)] + W'(\mathcal{G})$$

and $\mathcal{G}$ is an uniquely defined pseudo context for global variables.

$$W'(\mathcal{G}) = Set\_of\_Global\_Variables$$

2.2 Axioms Representing Time

This section investigates the interval representation of time as proposed by Allen and Ferguson (1994). A review of their axioms has not been reproduced here due to limitations of space. On the basis of the axiomatic relations $Meets$, $Before$ and $After$ defined on intervals by Allen and Ferguson, precedence relations between intervals have been axiomatized to aid representation in the situational simulation domain. A diagrammatic representation of the axiomatized time intervals can be seen in Figure 2. The precedence relations developed include consequence, coincidence, precedence and concurrence. A brief review of the relations $Meets()$, $Before()$ and $After()$ has been provided below.

Meets: Two time intervals *i* and *j* are said to meet if and only if *i* precedes *j*, and yet there is no time between *i* and *j* and *i* and *j* do not overlap. In terms of discrete time points this can be axiomatized as:

$$\forall i, j \cdot \exists I, J, K, L \cdot (i = \{I, J\}) \wedge (j = \{K, L\}) \wedge (J = K - 1) \Rightarrow Meets(i, j) \equiv i : j. \quad (2)$$

It can be proved from this axiom that a time interval cannot meet itself, excepting when *{I,J}* coincides with *{K,L}*, in which case the intervals *i* and *j* collapse and become time points. This rules out possibilities of circular models of time.

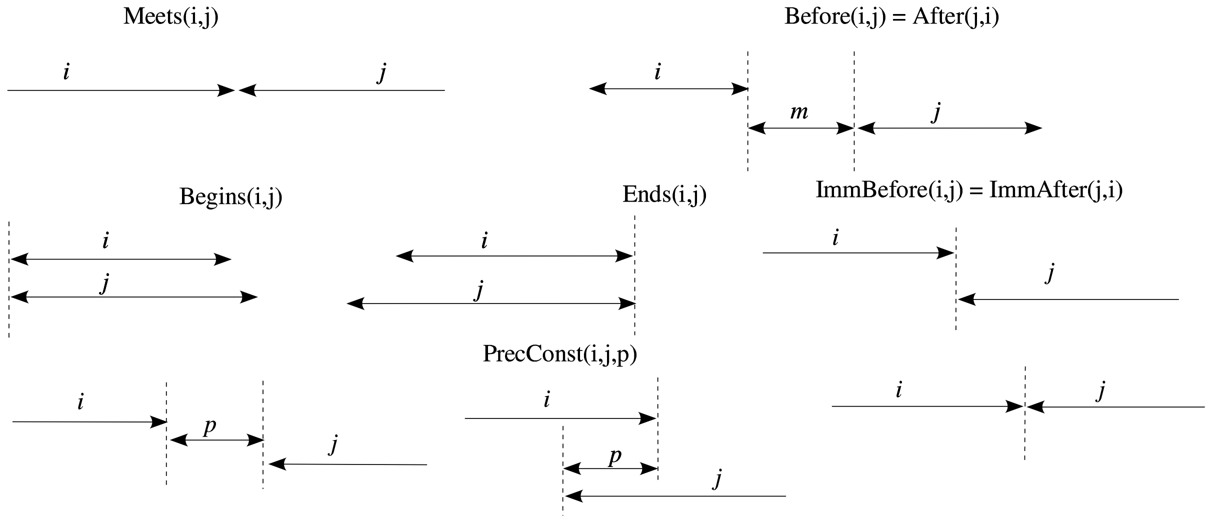

Figure 2: Axioms Defined on Intervals

Before: A truth value for *Before(i,j)*, implies that interval *i* starts *before* interval *j* and can be expressed as:

$$Before(i, j) \equiv \exists m \cdot Meets(i, m) \wedge Meets(m, j) \equiv i \prec j. \quad (3)$$

The inverse of *Before* is *After*, which is defined as:

$$After(i, j) \equiv \exists m \cdot Meets(j, m) \wedge Meets(m, i) \equiv i \succ j. \quad (4)$$

Precedence constraints between time intervals representing activities, actions and events as described in this paper will be based on the definitions of the following axioms.

Consequence Axiom 1 *An interval i is said to be immediately before another interval j if i precedes j and meets it. Inversely, the interval j is said to be immediately after the interval i*

$$\forall i, j \cdot ImmBefore(i, j) \Rightarrow (i \prec j) \wedge (i : j) \equiv ImmAfter(j, i) \Rightarrow (j \succ i) \wedge (j : i). \quad (5)$$

Coincidence Axiom 2 *Intervals i and j are said to have a coincident point of beginning if there exists a time interval t which comes immediately before both i and j. Similarly, if there is a time interval which comes immediately after two time intervals i and j then they are said to have a coincident point of ending.*

$$\forall i, j \cdot Begins(i, j) \Rightarrow \exists t \cdot ImmBefore(t, i) \wedge ImmBefore(t, j). \quad (6)$$

$$\forall i, j \cdot Ends(i, j) \Rightarrow \exists t \cdot ImmAfter(t, i) \wedge ImmAfter(t, j). \quad (7)$$

Precedence Axiom 3 *Any two time intervals i and j will always have a finish to start precedence relationship defined by the time interval p.*

Combining (5), (6) and (7) we get:

$$\forall i, j, p \cdot PrecConst(i, j, p) \Rightarrow (ImmBefore(i, p) \wedge ImmAfter(j, p)) \vee (Begins(j, p) \wedge Ends(i, p)). \quad (8)$$

$$\forall i, j \cdot PrecConst(i, j, 0) \Rightarrow i \prec: j$$

Concurrence Axiom 4 *Given that $\wp_{now}$ is the discrete granularity which represents the current time in the simulation, all time intervals which span across it are said to be concurrent.*

$$\forall t ConCurrent(t) \Rightarrow \wp_{now} \subset t \quad (9)$$

2.3 Actions, Events and Situations in Situational Simulations

Temporal logic has been used in this section to expressively represent actions, events and situations. Actions are triggers which create events and situations. Some outcomes of actions are bad weather, material delivery, reallocation of resources, labor strikes etc. In the simulation environment actions occur instantaneously in time, at the the starting time point of the interval of the event they trigger.

Events reflect the effects of real life episodes on resource and precedence constraints within the construction domain. All events span over time intervals. Each event is accociated with three sets of variables: the *Pre-Condition* set, the *Event Condition* set and the *Consequence* set and is triggered by an unique action. Member variables of the event condition and pre-condition sets are identical. However, the variables in the two sets must have different attribute values. The change in attribute values is triggered by actions. The event is reflected by the event condition set of variables. Future effects of the event, are captured in the consequence set, which is a set of assertions about values of variables in the future. The compound predicates $Pre\_Cond(t)$, $Event\_Cond(t)$, $Consequence(t)$ are conjunctive clauses of simple predicates which assert attributes of the member variables over the time interval t during which the conditions specified by the pre-condition, event condition and consequence sets hold, respectively. They are also homogeneous over the time intervals in which they hold.

For example, in the event of a labor strike that lasts for the time interval t, productivity (represented by the variable $prod$) for all activities is reduced to 0 due to a 0% availability of labor (represented by the variable $labor$). In this case the event condition set is $\{labor(null, t), prod(null, t)\}$ across all activity contexts. The pre-condition set is $\{labor(100\%, t'), prod(100\%, t')\}$ where the predicate $Meets(t', t)$ is true. This event represents a violation in a resource constraint. The action to create the labor strike event can only be taken if the pre-condition set is fulfilled in the immediately previous time point. This is expressed as:

$$\forall t \cdot Act\_Labor\_Strike(t.start) \Rightarrow \\ \exists t' \cdot labor(100\%, t') \wedge prod(100\%, t') \\ \wedge Event(Labor\_Strike, t) \wedge ImmBefore(t', t)$$

The fact that the pre-condition set is a necessary condition for an action to occur and that every action triggers an event is used to axiomatize the definition of an action as:

$$\forall t \cdot Action(t.start) \Rightarrow \exists e, t' \cdot Pre\_Cond(t') \\ \wedge Event(e, t) \wedge ImmBefore(t', t) \qquad (10)$$

The converse of the above axiom does not hold, however, if the pre-condition set holds, then it may be concluded that the action may probably be taken. This is axiomatized as:

Axiom 5 *A specific action can be predicted to probably occur at* $T + 1$ *if for all contexts concurrent at both* T *and* $T + 1$ *there exists at least one context* i *such that* $W'(i)$ *at* T *has a subset of variables which satisfy the pre-condition set of the action.*

The pre-condition set could have also included other non-null values of productivity and labor, for the logical precondition of a labor strike to be fulfilled. However, in order to avoid disjunctive reasoning in the present level of research, the precondition set is being limited. This limitation will be dealt with in future.

Consequences of an event are assertions about the future that are direct outcomes of the event. The consequence set is a set of variables that assert attributes of entities in a future time interval, which is directly affected by the occurence of the event. For example, in the case of the event $Labor\_Strike$, the productivity of all activities may take a while to recover, and continue to be at 50% for the time interval t'' immediately after the $Labor\_Strike$ event is over. The time intervals t and t'' are tied by the precedence constraint predicate. Hence, the labor strike event may be defined in first order predicate logic as:

$$\forall t \cdot Event(Labor\_Strike, t) \Rightarrow \\ \exists t' \cdot labor(null, t) \wedge prod(null, t) \\ \wedge prod(0.5, t'') \wedge PrecConst(t, t'', 0)$$

This allows us to generalize the definition of the event as:

$$\forall e, t \cdot Event(e, t) \Rightarrow \exists t', p \cdot Event\_Cond(t) \\ \wedge Consequence(t') \wedge PrecConst(t, t', p) \qquad (11)$$

Information about actions and events stored in the knowledge base is based on event and action definitions discussed here.

If the pre-condition and event condition sets of variables for an event belong to the same context, then the event is specific to a particular context and is effectively an *activity or context dependent event*. However, if the pre-condition and event condition sets of variables are Global variables only, then the event is a *global or independent event*. For example, because weather is a global variable, an event related to it will be a global event. By definition variables in the event condition and pre-condition sets have to be either global or context specific. They cannot be mixed. The consequence set may, in both cases, still have variables from across activity contexts.

Situations are events which result in immediate constraint violations and demand immediate user intervention to carry on with the simulation. All events may not create immediate constraint violations, and hence may not create situations.

2.4 Participant Interactions and Agent Actions

Participants interact with the environment by changing values of the variables that represent specific entities. By changing the contexts or resource variables, participants can reallocate resources between activities. Participants can interact only with variables within their jurisdiction. Global variables are beyond their control (e.g., the participant cannot change the weather). Access is limited to context specific variables which describe the resource requirements of the activities.

The agent has greater access to variables than the participant does. It can access all global variables and context specific variables. However, in taking actions that affect the environment, the agent is not allowed to change eternal truths about the environment (e.g., the agent may not change the attribute of an excavation activity from outdoor to indoor).

All agent actions are essentially operators which transform a set of pre-conditions to a set of event conditions. Because participant interactions are limited to resource reallocation and replacement within the environment, they cannot directly create events in the environment. However, reallocation of resources might result in resource constraint violations which, when perceived by the agent, will indirectly create events. Hence participant interactions can only create the $Pre\_Cond()$ set, but only agent actions can transform a $Pre\_Cond()$ set to a $Post\_Cond()$ set.

2.5 Interval vs. Time Point Representation

The axioms and definitions described so far have been based on a representation which uses time intervals rather than time points. Time intervals can be represented as a series of time points and the simulation itself traverses from time point to time points. This section uses the constructs of finite state machines to justify the use of intervals over time points.

The situational simulation can be looked upon as a Finite State Machine (FSM) which is defined as a model of computation consisting of a set of states, a start state, an input alphabet, and a transition function that maps input symbols and current states to successive states. It can be described by the tuple:

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{I}, \mathcal{R}, \mathcal{L} \rangle$$

where $\mathcal{S}$ is a finite set of states, $\mathcal{I} \subseteq \mathcal{S}$ is the set of initial states, and $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ is the transition relation, specifying the possible transitions from state to state. $\mathcal{L}$ is a function that labels states with the atomic propositions from a given language. Such a tuple is called a Kripke structure (Kripke, 1962).

States in the situational simulation environment are expressed using sets of variables. The two sets of variables defined so far are the world set $W(T)$, which characterizes the environment at each time point, and the sub-world set $W'(i)$ which characterizes the environment, in terms of intervals, over activity contexts. Closure on the set of variables and the attribute domains, limits the number of possible worlds and sub-worlds to a finite number. The set of all worlds is denoted by $\mathcal{W}$. If there are n variables, each of which can take up at most, m attributes, then the cardinality of the set $\mathcal{W}$ is at most n^m. In reality the number is lower, because there will be many worlds which are mathematically accountable but absurd in reality.

We can also, define the set of all possible sub-worlds specific to a particular activity context i. This is denoted by $\mathcal{W}_i$. This set does not completely define the activity environment, because it leaves out the global variables. However, if we augment it with the set of global variables, then we get a set of states that completely defines the context. This is denoted by $\mathcal{W}_i^+$.

$$W^+(i) = W'(i) + W(\mathcal{G}); \;\; W'(i) \in \mathcal{W}_i$$

$$\mathcal{W}_i^+ = \{W^+(i)\};$$

$W^+(i)$ is the augmented sub-world. $\mathcal{W}$ and $\mathcal{W}_i^+$ are equivalent to $\mathcal{S}$ in the Kripke structure. $W(T)$ $(W(T) \subseteq \mathcal{W})$ corresponds to an initial state defined on a time point. Similarly, $W^+(i)$ is a an initial state defined on the interval i.

An action in the environment creates transitions in state. Differences between attributes of variables, in the pre-condition and event condition sets of an event triggered by an action, indicate a transition in state. The critical question is, do the actions create changes in states defined in $\mathcal{W}$ or in $\mathcal{W}_i^+$?

Actions creating dependent events operate on variables specific to the context specific sub-world. Actions creating independent events operate on the set $W(\mathcal{G})$. Since the set of variables in the sub-worlds have been augmented with the global variables, independent events are essentially multiple instances of the same action across all contexts. Hence, actions creating context dependent and independent events can create state transitions in $\mathcal{W}_i^+$. This representation allows multiple state changes to occur in the environment, each in a different context, at any point of time. In other words, an interval representation allows simultaneous context dependent events across different activity contexts without breaking the semantic structure.

It is very difficult to define unique state change actions as operators on $\mathcal{W}$, without breaking the semantic structure. Only actions triggering independent events can be expressed as state transitions. Simultaneous context specific dependent events cannot be expressed by this representation within the defined action-event semantics. Hence, state transitions defined by actions are best suited for the set of states $\mathcal{W}_i^+$ that uses an interval representation of time.

The Kripke Structure that can be defined for the context i in the simulation environment is:

$$\mathcal{M} = \langle \mathcal{W}_i^+, \mathcal{I}, \mathcal{R}, \mathcal{L} \rangle$$

where,

$$\mathcal{I} = W^{+}(i);\ \ \mathcal{R} \subseteq \mathcal{W}_i^{+} \times \mathcal{W}_i^{+}$$

$$\mathcal{L} \in Set\_of\_all\_Events$$

In effect we have a FSM for each context allowing simultaneous activities and events; actions serving the purpose of state transition operators and events providing a language to express changes in state. This is the rational behind using an interval representation of time.

3 AGENT REASONING

3.1 Overnight Inference

The agent comes into play between every consecutive discrete time point in the simulation, in other words, between any two consecutive 'days' during the simulation of the project. Hence, the name 'overnight inference'. The agent infers after the time point T and before the time point $(T+1)$. It has complete access to information in the environment encoded in terms of sets of variables $W(T)$ and $W(T-1)$.

Since sub-worlds cannot change state during the inference process, the agent's inference environment is static. It is also discrete, since there are a finite number of possible states. The environment is also non-episodic, because the agent needs information from $W(T)$ and $W(T-1)$. Finally, from the agent's point of view, the environment is non-deterministic as it cannot predict all user interactions or event generator decisions in the immediate future.

3.1.1 The Reasoning Mechanism

It may be noted here that for every event, the set of event conditions may be referred to as the post-condition set for the action triggering the event. The pre-condition set of the action and event are identical. The following assumptions of closure can also be made:

- Event Closure: An occurrence of an event implies that an action occurred. This is expressed as:

$$\forall e, t \cdot Event(e, t) \Rightarrow \exists Action(t.start) \qquad (12)$$

- Attribute Closure: This reflects a closure on the attributes and variables and implies that any change in attributes of variables implies that an event has occured. This is expressed as:

$$\forall t, t' \cdot Pre\_Cond(t') \wedge Post\_Cond(t) \wedge Meets(t', t) \Rightarrow \exists e \cdot Event(e, t) \qquad (13)$$

On the basis of the definition of an action (10), definition of an event (11), variable unification (Variables unify when they both describe the same entity and take up the same attribute value, two sets of variables unify when there is a one-to-one unification between the members of the sets) and the assumptions of closure (12 & 13), we can state the following theorem:

Theorem 1 *A specific action can be inferred to have occurred in between time points $(T-1)$ and T if for all contexts concurrent at both $(T-1)$ and T there exists at least one context i such that $W'(i)$ at $(T-1)$ has a subset of variables which unify with the pre-condition set of the action and $W'(i)$ at T has a subset of variables which unify with the post-condition set of the same action.*

Given a context i for which there exists sets S and S' which unify with the pre-condition and post-conditions sets of the same Action at $(T-1)$ and T respectively, it is required to prove that there was an $Action(T)$.

Proof: We know that $S \subset W'(i)$ at $(T-1)$ and $S' \subset W'(i)$ at T. By definition, the member variables of S and S' are identical but have different attribute values. Let us assume that S and S' are homogeneous over time intervals m and n. A change in the values of variables occurs at $(T-1)$ and at T. Therefore, $m.end = (T-1)$ and $n.start = T$. Hence by (2) we can say $Meets(m, n)$. Now by (13)

$$S(m) \wedge S'(n) \wedge Meets(m, n) \Rightarrow Event(e, n)$$

by (12)

$$Event(e, n) \Rightarrow Action(n.start)$$

Hence, there was an $Action(T)$.

Corollary 1 *For every Action taken there is a set of assertions which predict the future of the simulation environment.*

This is proved because of the fact that every action implies an event, which, in turn, implies a consequence set (10 & 11). Prediction by the agent for possible future actions follows Axiom 2.3.1.

3.2 Implementation and Correctness

A theorem prover was implemented for the stated theorem using the Forward Chaining algorithm (Russell & Norvig, 2002). Given the perceptions of the world in terms of $W(T)$ and $W(T-1)$, the agent classifies the variables into sub-worlds specific to contexts that are concurrent in both (T) and $(T-1)$. It then isolates all contexts in which variables have registered changes in attribute values. Then for each of these contexts it unifies the variables with action and event definitions in the knowledge base (KB). All inferences are added as facts to the Assertion set, thus allowing reasoning in future worlds to be based on perceptions and outcomes of the past. Figure 3 illustrates the algorithm.

```
Agent (Assertions, KB)
{ Perceive W(T), W(T-1) from Assertions
    {For each context i concurrent in W(T) and W(T-1)
        {if W'(i) at T != W'(i) at (T-1)
            {For each event e in the KB
                {Check for unification of Pre_Cond
                  and Post_Cond set of e with W'(i);
                 Infer actions taken in T;
                 Infer Consequences of actions in T;
                 Predict possible actions in (T+1);
                 Add inferences to set of Assertions;
            }  }
        }
    }
}
```

Figure 3: Agent Algorithm

All agent inferencing and reasoning is done on the basis of assertions about actions and events in a knowledge base of facts. The inference mechanism is sound and complete within definitions of actions and events defined in the knowledge base. Hence, if an action is defined in the knowledge base, then it will always be predicted everytime its pre-condition set is fulfilled. Also, an event defined in the knowledge base will always be inferred if it has occured. However, if there is a combination of variable attributes which the participant can change but which are not documented in the knowledge base, then they will simply go unnoticed. Hence, an efficient implementation of this agent lies in developing an accurate knowledge base of facts, and creating appropriate closures on participant interaction.

4 FUTURE RESEARCH

Agent reasoning so far is limited to dealing with simple conjunctive clauses only. Disjunctions are difficult to deal with because they often make the problems computationally intractible.

The success of the reasoning mechanism suggested in this paper depends upon how accurately the action and event definitions capture the causal nautre of events in the real world. Research is being conducted to identify appropriate pre-condition, event condition and consequence sets so that the environment can appropriately simulate the reality of events.

Assertions about the future made from the consequence set of events are not eternal truths, but ones that can be changed because of future user interactions. For example, an event might lead to a resource unavailability in a future interval, which the agent can predict. However, before the interval starts, the participant might reallocate resources and restore availability, and falsify a previously made assertion. This kind of non-monotonic behaviour is rather unsettling for the agent. This limitation can be dealt with by introducing default logic constructs, which is part of an ongoing effort.

REFERENCES

Abourizk, S. & Halpin, D. 1990. Probabilistic Simulation Studies for Repetitive Construction Processes. *Journal of Construction Engineering and Management*, ASCE, 116(4): 575-594

Allen, James F., & Ferguson George, 1994. Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation Special issue on Actions and Processes.*

Barab, S.A., Hay, K.E., Barnett, M. & Squire, K. 2001. Constructing Virtual Worlds: Tracing the Historical Development of Learner Practices. *Cognition and Instruction*, 19(1): 47-94.

Halpin, D. & Woodhead, R. 1970. *A Computerized Construction Management Game.* Department of Civil Engineering, University of Illinois at Urbana-Champaigne.

Jaafari, A., Manivong, K. & Chaaya, M. 2001. VIRCON: An Interactive System for Teaching Construction Engineering and Management. *Jour. Construction Engineering and Management*, ASCE, 127(1): 66-75.

S. A. Kripke. 1962. Semantical considerations on modal logic. In Proceedings: *A Colloquium on Modal and Many-Valued Logics*, Helsinki.

McCabe, B., Ching, K.S. & Savio,R. 2000. STRATEGY: A Construction Simulation Environment. *Proceeding of the Construction Congress VI.*

Russell, Stuart J. & Norvig, Peter 2002. *Artificial Intelligence: A Modern Approach, 2nd Edn.* Prentice Hall.

Sawhney, A., Mund, A. & Koczenasz, J. 2001. Internet-based Interactive Construction Management Learning System. *Jour. Construction Education*, 6(3): 124-138.

AUTHOR BIOGRAPHIES

AMLAN MUKHERJEE is a doctoral candidate in the Department of Civil Engineering, at the University of Washington. In 2001, he received an MS in Civil Engineering from the State University of New York at Buffalo. His research interests lie in applying AI technologies (like multi-agent situational simulations) to construction engineering and management and in engineering education. <amlan@u.washington.edu>

EDDY M. ROJAS is an Assistant Professor in the Department of Construction Management at the University of Washington. He received his Ph.D. in Civil Engineering in 1997, an MA in Economics in 1997 and an MS in Civil Engineering in 1995 from the University of Colorado at Boulder. His undergraduate civil engineering degree is from the University of Costa Rica in 1991. His research interests include modeling, simulation and visualization of construction engineering and management processes; engineering education and construction economics. <er@u.washington.edu>.