IMPLEMENTING A GENERAL PURPOSE FRAMEWORK USING MULTI-AGENTS FOR CONSTRUCTION MANAGEMENT EDUCATION

Amlan Mukherjee

Dept. of Civil Engineering
University of Washington
Seattle, WA 98195, USA

Eddy M. Rojas

Dept. of Construction Management
University of Washington
Seattle, WA 98195, USA

William D. Winn

College of Education
University of Washington
Seattle, WA 98195, USA

ABSTRACT

The need for contextually rich educational experiences in construction education has led researchers to explore alternatives using gaming and simulation environments. The work done so far has concentrated on special and general purpose simulations of construction operations with limited interactivity. We claim that it is necessary to have an extensible general purpose framework, for developing a wide variety of situational simulations of construction management processes, that can be used to create interactive environments for training managers. We have introduced such a framework and developed it using multiple autonomous agents. The Virtual Coach is an implementation of the developed framework. It was tested on a class of 19 construction management senior students, and proved to be an useful educational tool.

1 Introduction

In traditional construction education the learner and the learning context are detached. Concepts are presented as fixed, well-structured, independent entities and classroom activities are disconnected from authentic context resulting in fragmentation and specialization of courses and educational experiences. McCabe, Ching and Savio (2000) argue that current civil engineering coursework teaches only the theories of construction management (CM) and that students may encounter difficulties in applying theoretical principles when exposed to real world situations upon employment. Sawhney, Mund and Koczenasz (2001) state that civil and construction engineering curricula does not allow the inclusion of issues of importance to construction, or the significance of hands-on experience and interaction with practitioners. Case studies, class projects that involve interaction with the industry and even internships are useful for bridging such a gap but are limited as they do not provide students the opportunity to explore implications of management decisions as the risks of financial losses are high.

This understanding has led researchers to explore alternatives in construction education using gaming and simulation environments such as Superbid (AbouRizk 1993), STRATEGY (McCabe, Ching, and Savio 2000), ICMLS (Sawhney, Mund, and Koczenasz 2001) and VIRCON (Jaafari, Manivong, and Chaaya 2001). Some of these efforts have been inspired by earlier research undertakings in the area, such as CONSTRUCTO (Halpin and Woodhead 1970) and AROUSAL (Ndekugri and Lansley 1992). Simphony (Hajjar and AbouRizk 1999) and STROBOSCOPE (Martinez and Ioannou 1999) have also developed simulations that deal with construction operations like tunneling and earthmoving. Simulation languages like STROBOSCOPE and CYCLONE have also provided a general and special purpose framework for simulating construction operations and CM processes, with absent or limited interactivity.

These efforts have provided a stepping-stone toward creating participatory, contextually rich educational environments. However, they have been limited to providing special purpose simulations of specific construction processes or providing general purpose frameworks for developing simulations for construction operations with limited or absent interactivity. A literature survey in greater detail has been presented in Mukherjee and Rojas (2003a) to support this claim.

We claim that general purpose interactive situational simulations of CM processes can be used to develop effective learning environments for construction managers. Such environments allow them to understand the interdependencies between various constraints which govern the CM environment and develop better decision making skills. The first step toward creating such general purpose situational simulations, that can be programmed to simulate a variety of CM processes, is to develop a framework to support such programmability and scalability. In this paper we have introduced a multi-agent framework for developing such general purpose simulations. We have also developed an implementation of a situational simulation using the framework, for a hypothetical construction project, and tested its usefulness on a group of 19 CM seniors.

2 Situational Simulations as Educational Environments

A *situational simulation* is a temporally dynamic, interactive simulation. In their simplest form *simulations* of construction processes use a set of initial conditions and parameters, and a well-defined model to project outcomes regarding the simulated operation. For example, given information regarding the availability of trucks and loaders, their unit costs and the amount of earth to be moved a process simulation would be able to project the total time and cost for an excavation operation. Situational simulations also have a well-defined model and a set of initial conditions, but as the simulation proceeds the system generates events which reflect various real life situational scenarios that might arise independently or as a consequence of user interaction, and expect the user to react to such events. How the simulation evolves is completely dependent on the model used, the way the events are generated and user interaction.

A situational simulation is a part machine (computer software/hardware) and part human environment. The machine is responsible for simulating the CM environment using construction domain specific knowledge while being sensitive to how human participants react to it. For example, given the knowledge that labor when overworked will tend to produce lower quality work the machine would infer a "re-work" event when the human participant tries to crash activities by making labor work over time too often. It can also create a "bad weather" event that disturbs progress on outdoor activities. The human participant is expected to try and finish the simulated project within time and budget constraints as they would in real life. Thus their responsibility is to constantly take challenging decisions regarding resource allocation and time cost trade-offs. As the simulation proceeds, there are a large number of ways to complete the simulated project. The project completion depends on the reactions of the human participant and the reactions of the machine.

The interactivity and the emphasis on learning based on contextually rich information allows situational simulations to be classified as learning environments that are based on theories in situated cognition (Winn 2002). Such environments expose participants to clinical exercises that help them explore future consequences of present decisions and the sensitivity of their contexts to such decisions, and over time develop better decision-making skills. The Virtual Gorilla Project at the Atlanta Zoo (Allison et al. 1997) and the Virtual Puget Sound (Windschitl and Winn 2000) and the Surgical Simulator (Oppenheimer and Weghorst 1999) efforts at the Human Interface Technology Laboratory, at the University of Washington are successful instances of such learning environments.

Extensive use of situational simulations have also been seen in the politico-military arena (Allen 1987, Goldhammer, and Speier 1959) and in natural disaster relief management (Ritchie 1985). A general purpose framework for situational simulations dealing with CM processes could be useful for developing a very wide variety of training environments for the construction engineering and management domain.

3 Proposed General Purpose Multi-Agent Framework (GPMAF)

A general purpose framework (GPF) provides a protocol that allows us to develop many different special purpose simulations. The common protocol will allow a community of developers to share, extend and build on simulations and foster collaboration in CM education. The participants of such a community can belong to academia and industry, with the common goal of training construction managers. In this section we have discussed the conceptual foundations of the proposed GPF, the components that make it up and the framework itself.

3.1 Conceptual Foundations of the GPF

In developing the foundations of the GPF, we studied the CM domain and tried to classify the pre-construction and construction phase processes into specific classes of problems. This abstraction is the first step toward creating the GPF.

During the pre-construction phase the problem at hand is that of creating a resource loaded activity schedule, also referred to as the "As-Planned" schedule. This can be classified as a constraint satisfaction problem (CSP). Such problems can be solved using a search based constraint solver. A number of research efforts support this claim. Succur and Grobler (1996) suggested a CSP formulation for construction project planning. They developed a structure that can represent precedence constraints (which they refer to as temporal constraints) and implicit resource constraints; and they provided a solution to the CSP using forward-checking constraint propagation algorithms like pruning and conflict

resolution. Hammond et al. (2000) suggested the use of a partitioned dependency structure matrix (DSM) to represent information about a schedule, which on closer analysis proves to be a CSP in which each matrix is a state representation of precedence and resource dependencies in a schedule. WorkPlan (Choo et al. 1999) also used resource and precedence constraint satisfaction in the WorkPlan implementation. It is safe to claim that given the appropriate constraints, the "As-Planned" schedule can be generated using search based constraint solvers that return sequences of state transformations between an initial state representation of a schedule and a goal state representation (a resource loaded "As-Planned" schedule) while assigning resources to all activities, in keeping with resource and precedence constraints.

During the construction phase managers aim to complete the project within constraints of budget and time as encoded in the "As-Planned" schedule. However, in reality, circumstances seldom permit the "As-Built" schedule to be identical to the "As-Planned" schedule. Projects get derailed from the "As-Planned" implementation because of violations in resource and precedence constraints caused by unexpected events like labor strikes, undelivered material and bad weather. Construction managers face the challenge of completing the project while constantly making critical decisions that satisfy the constraints encoded in the "As-Planned" schedule by reallocating resources, rescheduling activities and making time-cost trade offs. Hence, the manager's job during the construction phase is akin to a planning problem.

Planning problems make use of domain structure to generate relevant plans. Unlike search based problem solvers which are dependent on a specific set of successor functions to affect the environment, planners have a greater degree of autonomy and can create plans which are sensitive to context specific information. Specifically, during the construction phase managers are responsible for maintaining constraint satisfaction by taking corrective measures and dynamically updating the plan based on context specific knowledge of the present and anticipated futures of the environment. A discrete state representation that is incapable of representing multiple dynamic relationships overlapping in time, is not enough to describe such complex scenarios. Instead, a formal language is necessary to describe autonomous reasoning in such environments.

The first step toward developing the GPF was to agree on a "language" that could be used to represent and reason about activities, actions and events in the CM domain. The semantics of such a language would have to be based in a concept that applies generally across all scenarios in the CM domain. Such a language was developed using the semantics of interval temporal logic (Mukherjee and Rojas 2003b). The general concept is that an environment can be defined using a set of variables each of which can take up values from specifically defined continuous or discrete ranges. Each such variable is also attached to an interval of time which specifies the interval over which the value of the variable is valid. For example, the Weather is represented by the variable $weather$ which can take values from the domain $[sunny, rainy, snowy]$ and the predicate $weather(sunny, t)$ signifies that the weather in the environment will hold sunny over the time interval t. Combinations of changes in the validity intervals of one or more such variables representing the environment signify actions in the environment. Post conditions of such actions signify events. The pre-conditions of such actions need to be fulfilled for the events to be triggered by the actions. The pre-conditions and post-conditions for any action-event combination can be used to encode constraint information. The action-event combination thus represents constraint violations and the effect of such on the simulated CM environment.

From the above analysis we can conclude that the CM domain can be abstracted to a planning problem during the implementation phase and a constraint satisfaction problem during the pre-construction phase. It involves satisfaction of resource and precedence constraints, and reasoning processes, which govern actions and events in the construction environment. The foundations of the GPF lie at the very heart of this general understanding. Given a language to represent and reason about CM constraints such an understanding can provide the basis to simulate a diverse set of scenarios in the CM domain. The next question to be answered is: how can the developed conceptual understanding be used to develop a GPF for creating situational simulations?

3.2 The GPMAF

The GPF is a protocol that can be used by developers to put together different simulations using the conceptual foundation of constraint satisfaction, planning and the semantics of interval temporal logic to represent and reason about the CM domain. It is akin to an API for a programming language that can be used to program situational simulations for the CM processes. Hence, the GPF components will belong to one of the following fixed classes: agents, entities, operators and bases. Members of these classes can combine according to a well defined grammar to form operations, which are the basic building blocks of any situational simulation programmed using the framework. In this section we have defined each of the four mentioned classes and discuss the grammar that governs the framework.

An agent is anything that can perceive its environment through sensors and can act upon that environment through effectors (Russell and Norvig 2002). In the context of this paper all discussions about agents will refer to *software agents*. Software agents are programs which can

autonomously create changes in their environment based on its understanding of the condition of the environment. The *environment* is the formal definition of the semantics underlying the software simulation (Mukherjee and Rojas 2003b). Agents reasoning logically and acting autonomously (free of human control) toward a goal, can be attributed a notion of intelligence. They are aware of the repercussions of their actions on the environment and dynamically integrate their experiences into existing reasoning mechanisms. In the suggested multi-agent environment, each agent handles a specific reasoning aspect of the environment.

Agents are responsible for simulating the environment by generating current events that are an outcome of past participant interactions or, by randomly generating seed events. Secondly, the agent can predict future consequences of present circumstances; as warning flags for the participant and also as a guideline for effectively planning the future of the environment. Finally, the agent can depict the sensitivities of the environment to user decisions. This allows it to portray differences between the "As-Built" and the "As-Planned" trends. In order to accomplish the first two duties, the agent needs to be perceptive to changes in the environment effected by the participant as well as be able to effect changes in it. It must also have awareness regarding the context specific causal reasoning about actions and events, which governs the environment.

Each agent has a finite set of operators associated with it. Operators are reasoning mechanisms attributed to each agent. Agents use operators to reason autonomously and make changes to the environment. Changes to the environment are made by changing values of variables and/or variable collections which are referred to as entities. The nature of variables and their classification has been discussed in an earlier publication (Mukherjee and Rojas 2003b). Variables can be classified into discrete and continuous variables depending on the nature of the values they take up. Each variable can also be classified as activity specific (defines an aspect of a specific activity) or global (defines an aspect of the environment applying to all activities). Combinations of variables can also be classified into the following sets of disjoint entities.

- As-Planned Data, As-Built Data
- Activity Dependent Events, Global Events

Agents function by implementing operators to change the values of entities

Pro and Information Attitudes (Woodridge and Jennings 1995) (inferred and factual information) are inherited by agents from knowledge bases (KB), databases (DB) and feedback from user interaction (UI). This allows the agent to reason autonomously. Knowledge bases contain event definitions and data bases contain "As-Planned" cost and schedule information about the project being simulated. The framework consists of utility functions which are not operators but can allow any of the agents to access the bases or to do routine repetitive tasks like calculating remaining durations of activities or updating the floats on the schedule.

The basic unit of the GPF is an operation. In an operation an agent inputs an information entity and outputs it to another information entity using a specific operator. Situational simulations built using the GPF can be expressed as a combination of operations, in series and/or parallel. This sets the grammar for creating simulations using the Agent-Operator-Base-Entity components of the framework, as illustrated in the Figure (1).

In the parlance of the Java programming language the GPF can be expressed as:

```
public interface Operation1{
      void O11(Environment E);
      void O12(Environment E);
      . . .}
public class Agent1 implements Operation1{...}
public abstract class Variable{
      . . .
      //Status of a variable: global or local
      public boolean global_local;
      . . . }
public class DiscreteV extends Variable{...}
public class ContinV extends Variable{...}
public class Environment{
      . . .
      //List of Discrete variables
      public DiscreteV discrete_list;
      //List of Continuous variables
      public ContinV contin_list;
      . . . }
public class Simulation() implements Runnable
{
      public static Agent1 A1;
      public static Agent2 A2;
      public static Environment E;
      . . .
      run(){
          . . .
          //A typical operation
          A1.O11(E);
          . . .      }
      public static void main(String Args[]){
        run();
      }
      public static void utility1(){ ... }
}
```

An implementation of such a framework would have definitions of multiple *Agents* each implementing a particular *Operation* interface. The current implementation of the framework called the Virtual Coach, has three agents: Logical Agent (LA), Mathematical Agent (MA), and the Visualization Agent (VA). It also has the following events defined: bad weather, poor quality work, labor strike, no material delivery and cost hike. Each of these events represents a resource constraint violation. The implementation

also includes utility functions which read from the data base, the knowledge base, runs the scheduler and calculates remaining duration. In the next section we will discuss the Virtual Coach implementation in greater detail.

There are three interfaces to the developed framework. One is the programmer's interface, the second one is the developer's interface and the third one is the user interface to the developed situational simulation. We will discuss the third interface in the next section. The first two interfaces that allow the framework to be extensible. An implementation of such a framework will require the developer to input the following to simulate a specific project of their choice:

- "As-Planned" cost and schedule information
- Definitions of variables characterizing the simulation (they can add to the defaults)
- Definitions of anticipated events using pre and post conditions for the associated constraint violations
- Realistic probabilities of defined events based on historical data to enable the simulation to generate reasonable scenarios.

The programmers have access to the source code and are free to add more operations to each of the existing agents and/or to add more agents to the framework with dedicated operators. Thus the developer can either use the functionalities provided by the current implementation of the multi-agent framework to simulate projects of their choice or they can add more functionality to extend the current framework.

4 The Virtual Coach Implementation

The Virtual Coach is a particular implementation of the discussed GPMAF. It is a situational simulation that is run by three agents: the Logical Agent (LA), the Mathematical Agent (MA) and Visualization Agent (VA).

MA operators are Unite and Compute, while LA operators are Inference and Event Generation. Entities are defined as the different classes of information in the simulation environment. Every agent operation takes an information entity as an input and transforms it to another information entity (Fig. 1). Atomic entities can be combined to create super entities when the super entity is a logical parent of the atomic entities.

Systemic reasoning in the Virtual Coach is based on a mathematical model defined by Rojas and Mukherjee (2003). It deals with reasoning about how events affect the net equilibrium of the system. If the project is executed "As-Planned", then the system equilibrium is not affected. However, every time there is an event, which results in a crisis, the equilibrium is disturbed. This allows the simulation to constantly give the participant feedback regarding progress as compared to the "As-Planned" implementation.

These graphs can be seen in the lower left corner of the "As-Planned" vs "As-Built" screen.

The logical agent can create events and also infer events, which follow as a result of user interactions with the simulated environment. It can create events in the situational simulation by violating developer defined constraints. It can also predict future constraint violations based on its ability to infer from facts in the knowledge base. A default knowledge base can be used or developers can create their own knowledge bases. A detailed discussion of how the agent functions can be found in, Mukherjee and Rojas (2003b). In Virtual Coach, events could be generated as a result of the following constraint violations:

- No work can be done unless necessary material and labor are available
- Outdoor activities cannot be productive during snowy weather
- Overworking a labor crew reduces productivity and increases chances of rework
- Labor hired on an emergency basis costs more and is less productive
- Schedule constraints

In the Virtual Coach information visualization and user interactivity are handled by the visualization agent. The function of the VA is to make sure that the information being displayed to the user is consistent in with the information in the simulation. The VA is also responsible for encoding participant reactions and passing them onto the other system agents in a format that can be easily processed.

The Virtual Coach implementation currently runs a situational simulation for a twelve activity hypothetical project with realistic constraint violations and event information. Figures (2) and (3) provide screen shots of a preliminary deployment of the system. Figure (2), is the resource allocation screen, which informs the participant of the total available resources in the environment and the total resource requirements specific to each ongoing activity in the simulation. Each activity panel also has a graph showing the "As-Planned" rate of work completion versus the "As-Built" rate of work completion. The participant is allowed to assign more or less than the planned requirements depending on availability to accelerate or decelerate the project.

In the absence of the necessary resources, the participant is also allowed to hire more labor and purchase more material at a premium price. This allows the participant to accelerate the project, at a higher cost, and is often an option to keep the project on schedule. While the direct costs go up, the participant does gain in terms of indirect costs by saving time.

Finally, Figure (3) illustrates the report about progress at the end of a week. The participant can view the current state of the schedule compared to the "As-Planned" schedules.
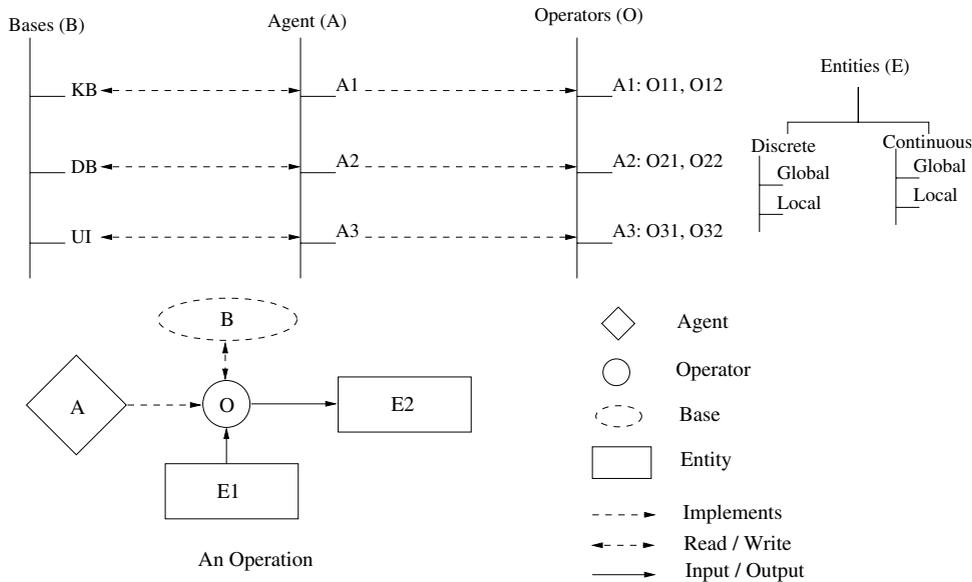
Figure 1: The Agent-Operator-Entity-Base Framework

He/she can also keep track of direct costs, indirect costs and space requirements by following the graphics at the lower left hand corner of the viewer. The lower right hand corner of the viewer allows the participant to monitor the values of discrete and continuous environment variables and keep track of the possibilities of events that may occur in the near future. They can also keep track of recent events that have just occurred. This is important in allowing them to make future resource allocations. The final goal of the participant is to steer the project through generated scenarios and complete within budget and time constraints.

5   Implications in Construction Education

The Virtual Coach situational simulation was administered to a sample of 19 senior level construction management students, as part of a Project Management class. The students were made to take a pre-test and post-test before and after they ran the simulation respectively. The pre-test and post-test were identical, requiring the students to rank (on a scale of 1-10), in their opinion, the importance of a list of factors in developing a plan for a 12 week period of a construction scenario. They were provided with a list of constraints governing the scenario. The constraints included schedule considerations, budget limitations and the possibilities of events like bad weather, material delivery delays and labor shortage.

Four of the priority ratings assigned by the students, before and after using the simulation, were summed and compared using a paired-sample t-test. The ratings selected for analysis were those that related to the schedule and resource constraints and the need to anticipate delay on a project (giving priority to critical activities in case of delay, attention to space restrictions on site, anticipating future material delivery delays, accelerating activities to create buffer for anticipated delay.) The difference between the ratings was significantly different:

```
Pre-Test:  Mean = 21.26 Std. deviation = 4.92
Post-Test: Mean = 25.31 Std. deviation = 4.70
T-statistic: t(18) = 3.32 p-value < .01.
```

Based on qualitative feedback (post simulation survey) from the students (84% of them thought that the Virtual Coach was a useful educational tool), the statistical significance of the post- and pre-test results and the high differential values of the confidence interval we can conclude that an intervention using situational simulations could be useful in construction education. It is also encouraging that the students improved on issues related to the specific constraints that were programmed into the simulation. This reflects the constraint satisfaction philosophy that rules the underlying framework. It also indicates that curriculum developers could program simulations with constraints that are important for students to attend to.

This study is preliminary and the results are only indicative. They however, do indicate the importance of exploring the use of situational simulations in construction management education.

REFERENCES

AbouRizk, S. (1993). "Stochastic simulations of construction bidding and project management," Microcomputers in Civil Engineering, 8, 343-353.
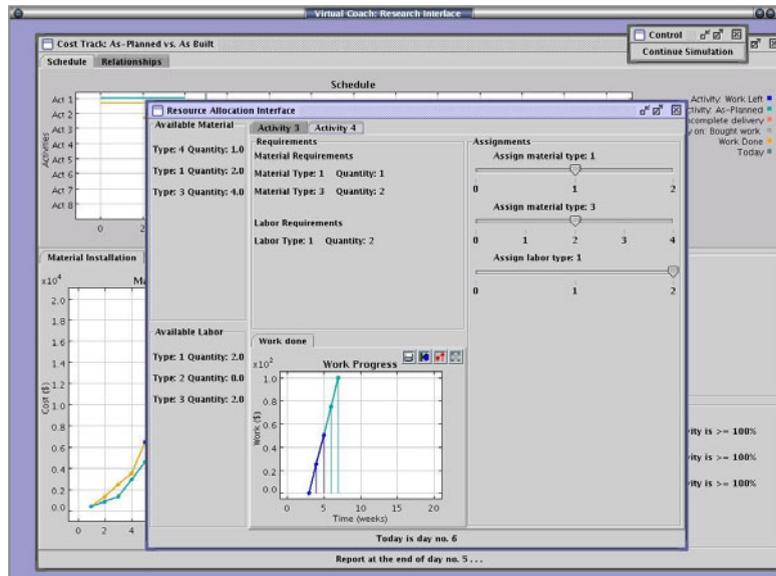
Figure 2: Resource Allocation Interface

Allen, T. (1987). "War games", McGraw-Hill, New York.

Allison, D., Wills, B., Hodges, L. F., and Wineman, J. (1997). "Gorillas in the bits". Paper presented at the VRAIS annual conference, Albuquerque, NM.

Choo, H.J., Tommelein, I.D., Ballard, G., and Zabelle, T.R. (1999). "WorkPlan: Constraint-based Database for Work Package Scheduling." Journal of Construction Engineering and Management, 125 (3), 151-160.

Goldhammer, H. and Speier, H. (1959). "Some observations on political gaming," World Politics, 12 (1), 71-83.

Hajjar, D. and AbouRizk, S. (1999) "Simphony: an Environment for building Special Purpose Construction Simulation Tools" In the Proceedings of the 1999 Winter Simulation Conference. P.A.Farrington, H.B.Newbhard, D.T.Sturrock and G.W.Evans, eds.998-1006.ACM Press, New York, USA.

Halpin, D. and Woodhead, R. (1970). "A Computerized Construction Management Game." Department of Civil Engineering, University of Illinois, Urbana Champagne.

Hammond, J., Choo, Hyung Jeong, Austin, Simon, Tommelein, Iris D., and Ballard, Glenn. (2000)"Integrated Design Planning, Scheduling, and Control with Desplan" Proceedings of the Eighth Annual Conference of the International Group for Lean Construction, IGLC-6, Brighton, UK, July 17-19.

Jaafari, A., Manivong, K., Chaaya, M. (2001). "VIRCON: interactive system for teaching construction management," Journal of Construction Engineering and Management, 127 (1), 66-75.

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

McCabe, B., Ching, K.S., and Savio, R. (2000). "STRATEGY: a construction simulation environment," Proceedings of Construction Congress VI, ed. Kenneth D. Walsh, 115-120.

Mukherjee, A. and Rojas, E. (2003a) "Applying Situational Simulations for General Purpose Situational Simulations in Construction Management." Presented at Construction Research Council, Ph.D. Research Symposium, November 14, Nashville, TN.

Mukherjee, A. and Rojas, E.(2003b) "Reasoning about Actions and Events in Situational Simulations" In Proceedings of Winter Simulation Conference, December 7th-10th, New Orleans, LA.

Ndekugri, I. and Lansley, P. (1992). "Role of simulation in construction management." Building Research and Information. 20 (2), 109-115.

Oppenheimer, P. and Weghorst, S. (1999). "Immersive Surgical Robotic Interfaces" In Proceedings of Medicine Meets Virtual Reality, 242-248.

Ritchie, G. (1985). "Atlantis: The Basis for Management Simulation Development." Simulation/Games for Learning. 15 (1), 28-43.

Rojas, E., and Mukherjee, A. (2003) "Modeling the Construction Management Process to Support Situational Simulations." Journal of Computing in Civil Engineering, 17(4), 273-280.

Russell, Stuart J. and Norvig, Peter 2002. Artificial intelligence: a modern approach, 2nd. Ed. Prentice Hall.
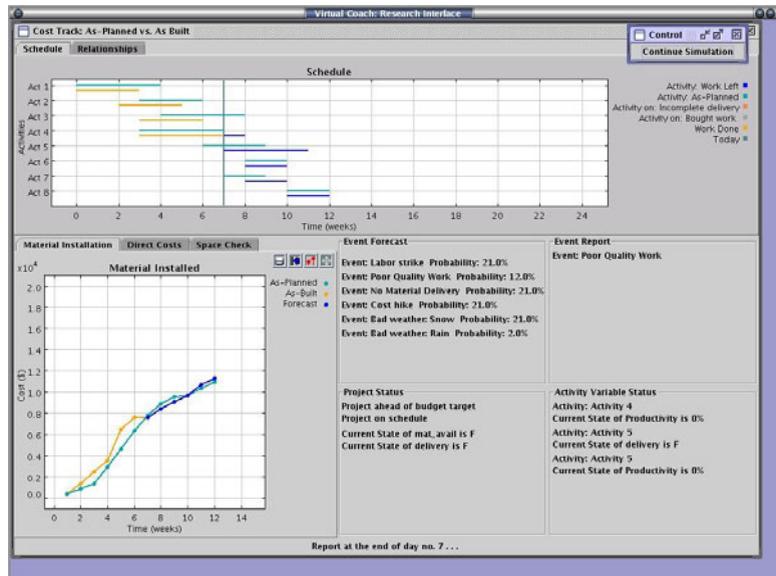
Figure 3: End-of-day Report

Sawhney, A., Mund, A., and Koczenasz, J. (2001). "Internet-based interactive construction management learning system," Journal of Construction Eneducation, 6 (3), 124-138.

Succur, M. and Grobler, F.(1996) "Construction Planning through Multi-Agent Constraint Satisfaction" Proceedings of the Third Congress of Computing in Civil Engineering, Anaheim CA.

Windschitl, M., and Winn, W. D. (2000). "A virtual environment designed to help students understand science." Proceedings of the International Conference of the Learning Sciences,In B. Fishman and S. O'Connor-Divelbiss, eds., 290-296. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Winn, William D.(2002) "Learning in Artificial Environments: Embodiment, Embeddedness and Dynamic Adaptation" Tech., Inst., Cognition and Learning 1

Woodridge, M. and Jennings, N.R. (1995) "Intelligent Agents: Theory and Practice" Submitted to Knowledge Engineering Review.

AUTHOR BIOGRAPHIES

AMLAN MUKHERJEE is a doctoral candidate in the Department of Civil Engineering, at the University of Washington. His research interests lie in applying interactive simulations to CM and engineering education. <amlan@hitl.washington.edu>

EDDY M. ROJAS is an Associate 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 and an undergraduate civil engineering degree from the University of Costa Rica in 1991.His research interests include modeling, simulation and visualization of construction management processes; engineering education and construction economics. <amlan@hitl.washington.edu>

WILLIAM D. WINN is Director of HITL's Learning Center and Professor in the College of Education at the University of Washington. He obtained his Bachelor's degree in Modern Languages from Oxford University, England, and his MA in Comparative Literature and PhD in Instructional Technology from Indiana University. He has written extensively on the ways in which cognitive and constructivist theories of learning can help instructional designers make effective decisions in selecting teaching strategies. He has also developed a theoretical framework that is concerned with how information should be presented in virtual worlds. <billwinn@u.washington.edu>