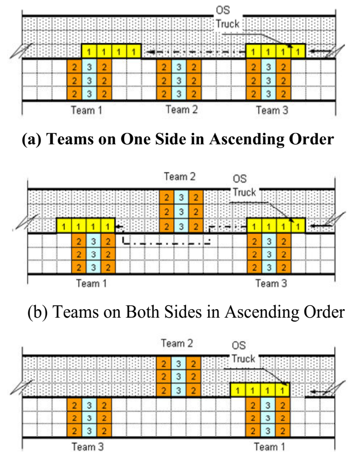
Table 1. Codes Used in the Bridge Model

Layer	Code	Description
Occupancy	0	Empty cell
	1	An OS truck before loading old section
	2	Crane
	3	Old section
	4	An OS truck after loading old section
	5	Empty space after removing the old section
	6	New installed panel
	8	An NP truck before unloading
	9	An NP truck after unloading
	10	For temporary use
	11	For temporary use
Control	0	Empty cell
	1	Truck moves north
	2	Truck moves south
	4	Truck moves west
	5	Static object
	6	Truck temporarily stops – waiting delay
ID	0	Empty cell
	1- 40	OS Truck
	41 - 80	NP Truck
	91 - 99	Work team



(a) Teams on One Side in Ascending Order

(b) Teams on Both Sides in Ascending Order

(c) Teams on Both Sides in Descending Order

Figure 5. Examples of site Layouts

(6) Define zones for Cell-DEVS models. A zone defines a region of the cellular space that will use a different local computing function or condition. In this case study, the cell space is divided into two zones as shown in Figure 4. The difference of the rules in the two zones is that a truck takes different moving directions when it meets an obstacle.

(7) Develop rules for each Cell-DEVS model. Rules are applied to each layer to control the movement of objects. The user can define his/her own rules to detect and give possible solution to the spatial conflicts.

(8) Develop DEVS models. External events are collected through input ports and the *external transition function* defines how to react to such inputs. At the moment the duration for the present state expires, desired results are spread through output ports by activating the *output function*, which will trigger the *internal transition function*, causing a state change (Wainer 2002). These functions are defined for each DEVS model.

Main-Processing. Because the worksite is explicitly represented by cells, the objects can be controlled by rules and each object can be identified by its ID. Consequently, the resource allocation and site layout can be considered. Figure 5 shows three patterns of site layouts. In pattern A, teams are on one side of the bridge and their ID numbers are in ascending order; in pattern B, teams are on both sides of the bridge in ascending order, while in pattern C, teams are on both sides of the bridge in descending order. When a truck is moving from west to east, it checks if it is passing a team and if this team is its corresponding team. If the truck finds its corresponding team, i.e., the truck's ID number matches that of the team, it will stop for loading old sections or unloading new panels; otherwise, it will move on. In this way, the team layout and the order of the team's ID numbers determine where the trucks should stop. Thus, the results of different worksite layouts can be compared. Resource combination is also determined in this phase. The acronym TSON is used to indicate resource combinations, e.g., TSON 5235 means a combination of 5 *Teams*, 2 *Saws*, 3 *OS Trucks* and 5 *NP Trucks*.

Post-Processing

(1) Delays resulting from spatial conflicts: As presented in Table 1, we use codes to indicate spatial conflicts on the *Control* layer. If all the trucks always move straight on an east-west axis, there will be no spatial conflicts or delays. However, trucks may change directions to turn around obstacles such as cranes or other trucks, which results in changing direction delays. In other cases, a truck has to stop temporarily and give the priority to another one, which results in waiting delays. Assuming that trucks always move on the bridge at a constant speed of 10 km/h, it takes about one second for a truck to go from one cell to the next. Thus, we can calculate the delays resulting from spatial conflicts. Occasionally, two or more trucks may get onto the bridge at the same time and one of them has to wait for a short while. We counted all of the three types of delays based on 45 combinations of TSON. As shown in Figure 6, we found that the changing direction delays are the major reason for the delay and Pattern A always has a significantly less delays than the other two patterns.

(2) Sensitivity Analysis: Sensitivity analysis is done by changing the number of each resource for each site layout pattern. The results for patterns and resources (Figure 6) show that the productivity is very sensitive when the number of NP trucks is less than 5, the number of teams less than 5, the number of OS trucks less than 3, and the number of saws less than 2. Thus, the TSON 5235 is the optimum combination from the productivity perspective, which consists with the results of MicroCYCLONE. The productivity results between MicroCYCLONE (Zhang et al. 2007) and Cell-DEVS are compared (Figure 7). Based on the 45 combinations of three site layout patterns, we found that: (1) The results from both modeling techniques are similar; (2) In most combinations, the productivity of Pattern A is a little higher than the other two patterns; (3) In most combinations, the productivity of MicroCYCLONE is higher by about 5% (1-2 panels) when the speed of trucks is constant (10 km/h). The reason for this is that the delays are very short (Figure 8), compared with the simulation time (9 hours). However, when a truck is changing direction, it will slow down and longer delays should be taken into account. To make the model more realistic, we slowed down the truck's turning speed and kept the other conditions to compare the effect of turning delays on the productivity using combination TSON 5235 and pattern B (Fig. 5(b)). Table 2 shows that the productivity dropped to 89% when the turning speed changed from 10 km/h to 1 km/h.

Table 2. Effect of Turning Speed on Productivity (TSON 5235 Pattern B)

Turning Speed (km/h)	Productivity (Panels/9h)	Rate
10	36	100%
2	34	94%
1	32	89%

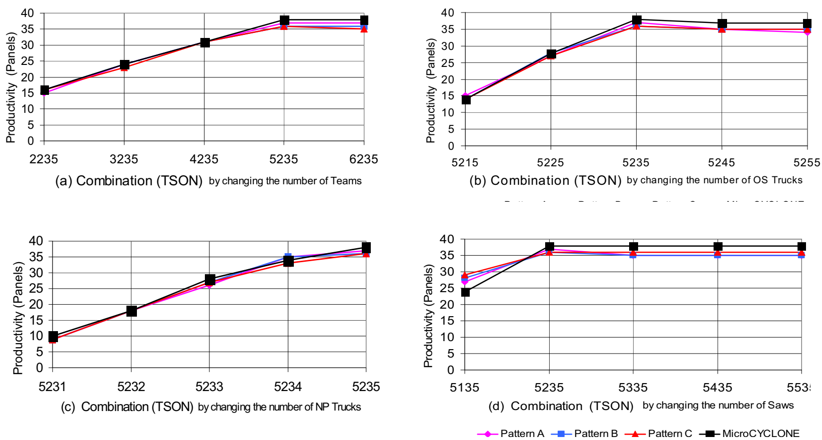


Figure 6. Sensitivity Analysis Using MircoCYCLONE and Cell-DEVS

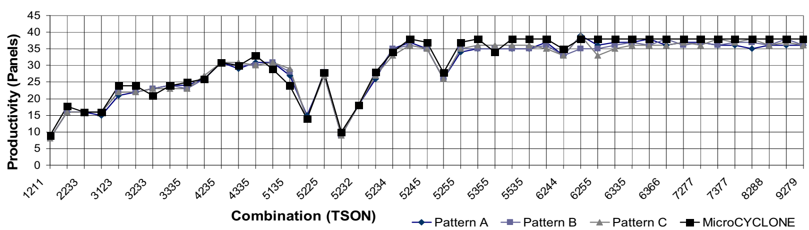


Figure 7. Comparisons of Productivity between MicroCYCLONE and Cell-DEVS

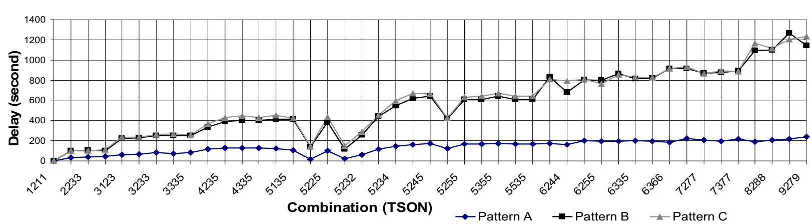


Figure 8. Comparisons of Delays in Different Patterns

Conclusions and Future Work

Cell-DEVS simulation is a general-purpose simulation tool for broad domains and can be used to simulate complex construction operations, especially when spatial constraints are crucial to the project. The proposed Cell-DEVS system integrates three phases, which facilitates arranging worksite layouts, visualizing recourse allocation, controlling the movement of objects and animating the simulation results. More information (occupancy, moving direction and ID, etc.) is integrated in the Cell-DEVS model, which makes it possible to identify and trace a specific object.

The programmability and the capability of defining rules make the Cell-DEVS system more flexible and capable of detecting and resolving spatial conflicts during the simulation. The simulation results of Cell-DEVS show that spatial conflicts can be decreased by selecting appropriate patterns and combinations. The difference between the results of MicroCYCLONE and Cell-DEVS illustrates that in some cases the impact of spatial constraints on productivity is significant and should not be neglected. Based on the present study, it has been found that Cell-DEVS simulation is an effective tool for space-related analysis in construction project.

Future work will include the following: (1) Investigating more case studies to further validate the Cell-DEVS approach in construction modeling; (2) Investigating new features of CD++, such as parallel processing and dynamic features for contingencies and dynamic releasing/allocating recourses; and (3) Exploring 3D visualization of simulation results (Khan et al. 2005).

References

Halpin, D.W. (1977). “CYCLONE: Method for Modeling of Job Site Processes.” *J. Const. Div.*, ASCE, 103(3), 489-499.

Kamat, V.R. (2003). *VITASCOPE – Extensible and Scalable 3D Visualization of Simulation Construction Operations.* Doctoral dissertation, Virginia Polytechnic Institute and State University, Virginia.

Kamat, V.R., and Martinez, J. C. (2006). “Interactive Collision Detection in Three-Dimensional Visualizations of Simulated Construction Operations”, *Engineering with Computers*, Springer, London, UK.

Khan, A., Venhola, W., Wainer, G. and Jemtrud. M.(2005). "Advanced DEVS model visualization". *Proc., IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation 2005*. Paris, France. 2005. VI.

Martinez, J.C. and P.G. Ioannou (1994). “General purpose simulation with Stroboscope.” *Proc., 1994 Winter Simulation Conf.*, 1159 - 1166.

Martinez, J.C. and P.G. Ioannou (1999). “General-purpose systems for effective construction simulation.” *J. Constr. Engrg. and Mgmt.,* ASCE, 125(4), 265-276.

Pang, H., Zhang, C., and Hammad, A. (2006). “Sensitivity Analysis of Construction Simulation Using Cell-DEVS and MicroCYCLONE.” *Proc., 2006 Winter Simulation Conf.*, Monterey, Calif., 2021-2028.

Tommelein, I. D., Levitt, R. E., and Hayes-Roth, B. (1992). ‘‘Site-layout modeling: How can artificial intelligence help?’’ *J. Constr. Engrg. and Mgmt.*, ASCE, 118(3), 594–611.

Wainer, G (2002), CD++: a toolkit to develop DEVS models, *Software-Practice and Experience*, Wiley. 32(3), 1261-1306.

Wolfram, S. (1986). *Theory and Application of Cellular Automata.* Vol. 1, Advances Series on Complex Systems, World Scientific, Singapore.

Zeilger, B.P. (1976). *Theory of Modeling and Simulation.* Wiley-Interscience Publication.

Zhang, C., Hammad, A., Zayed, T.M., Wainer, G., and Pang, H. (2007). “Cell-based Representation and Analysis of Spatial Resources in Construction Simulation.” *Automation in Construction*. No. 16, 436-448.

Towards Supporting Construction Equipment Operation Using Collaborative Agent-based Systems

Cheng Zhang,[1] and Amin Hammad[2]

[1] Dept. of Building, Civil & Environmental Engineering, Concordia University, Montreal, QC, PH (514) 848-2424, ext. 7074; FAX (514) 848-3171;
email: zha_che@encs.concordia.ca

[2] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, PH (514) 848-2424, ext. 5800; FAX (514) 848-3171;
email: hammad@ciise.concordia.ca

Abstract

A new approach based on collaborative agents is proposed to coordinate construction equipment operation in real-time. Cranes are taken as an example due to their frequent usage in construction for cooperative work. Literature related to agent-based applications is reviewed to investigate the feasibility and advantages of applying agents for equipment operation. Information about the equipment to be shared between agents is identified and the requirements for applying collaborative agent-based systems in construction are discussed. Lift task is decomposed into several steps comprising different actions for each crane, and negotiation is used to resolve the conflicts. A case study about a bridge rehabilitation project is used, where two cranes are working together to lift a panel on the bridge with height limits from the bridge structure. A simulation model is under development to create a virtual environment showing the work site, the bridge structure and two virtual cranes. In our preliminary test, agents can dynamically control the kinematic action of the two cranes respecting the functional constraints for safety and efficiency of operations.

Introduction

Operating equipment in a construction project should meet several requirements of capacity, safety and spatial constraints. Much research has been done about selecting equipment, simulating the work processes, training operators and optimizing the work paths to improve the efficiency and reduce conflicts in real construction operations. Training simulation for equipment operation has been used as an effective and cost-efficient training tool for the operators (Ritchie, 2004). Development in simulation software and visualization is making it possible to visualize simulation results (Kamat and Martinez 2001) and train equipment operators using virtual reality (Simlog, 2006).

Cranimation (2006) is a crane selection software, which calculates the outrigger forces for mobile cranes, the distribution of ground pressures for crawling cranes, and the minimum and maximum radius ranges. *LiftPlanner* (2006) is a 3D crane and rigging planning software, which produces drawings to plan and document critical lifts. The advantage of visualizing the work is that the user can simulate and check the functional constraints and interferences that may happen in reality between the 3D physical elements and virtual workspaces. However, these simulation tools focus on equipment working individually rather than coordinating the work of several cranes, such as the case of two cranes working together to lift a heavy or large object. The complexity of coordinating equipment requires more detailed planning and better real-time control of work. Ali et al. (2005) have proposed a path planning approach using a Genetic Algorithm (GA) for automating the path planning of two cooperative construction manipulators. However, they focus more on the path planning rather than considering the execution phase where the predefined path has to be changed to avoid the obstacles in a dynamic environment.

Artificial intelligence (AI) research aiming at the creation of unmanned construction systems, capable of performing complex tasks as well as human operators, has been carried out to control construction work on hazardous sites or for space and underwater constructions. These systems have been applied to perform emergency countermeasure and restoration work at disaster sites (Ban, 2002). It is mentioned that the efficiency of unmanned construction is roughly 60% to 70% of that of manned construction, but sharply decreases in cases where the machinery moves or high precision work is necessary (Ban, 2002). For example, collaborative equipment work is a common case in construction where communication and negotiation are essential to properly accomplish the work. Some research involving AI has been done to enhance communication between team workers and resolve problems in the construction industry. The concept of agents in AI refers to relatively independent and autonomous entities, which operate within communities in accordance with complex modes of cooperation, conflict and competition in order to survive and perpetuate themselves (Russell and Norvig, 2003). Using agents to plan and coordinate construction activities can simulate the manoeuvring of the equipment and enhance communication to reduce conflicts and improve efficiency. Agent systems have been used for construction claims negotiation (Ren, 2002) and dynamic rescheduling negotiation between subcontractors (Kim et al., 2003). However, little research has focused on real-time control for construction equipment operation using agents. Activities may need to be carried out in a multi-equipment environment to achieve a specific goal, such as two cranes working together to lift heavy or big objects. Multiple agents can be used to simulate such type of collaborative work. The distributed organization is able to adapt more easily to unforeseen modifications in the environment and, in particular, to possible malfunctions of certain agents (Ferber, 1999).

It is estimated that one crane upset occurs during every 10,000 hours of crane use. Approximately 3% of upsets result in death, 8% in lost time, and 20% in damage to property other than the crane. Nearly 80% of these upsets can be attributed to predictable human error when the operator inadvertently exceeds the crane's lifting capacity (Davis and Sutton, 2003). Therefore, in the present paper, cranes are taken as an example due to their frequent usage in construction for collaborative work. A new approach based on collaborative agents is proposed to coordinate construction equipment operation by providing real-time support. The objectives of this paper are: (1) To identify the requirements of applying collaborative agent-based systems; (2) To propose an approach for guiding operation in cooperative work considering engineering and spatial constraints; (3) To explore the feasibility of applying the proposed approach in real construction work using a case study; and (4) To visualize the work processes supported by the agent system using a simulation model.

Requirements for Applying Collaborative Agent-based Systems in Construction

Agents have separate but interdependent tasks to meet their final objective and to carry their work. When several agents are working together, it is necessary to define the relationships existing between their actions to improve the coordination of these actions. Coordination of actions is a matter of arranging the behaviours of the agents in time and space in such a way that the group action is improved either through better performance or through a reduction in conflicts (Ferber 1999). The following are the main requirements that should be considered when applying collaborative agent-based systems in construction focusing on cranes:

(1) Kinematic motion requirements: A loaded crane has a maximum of eight degrees of freedom (DoFs) (Reddy and Varghese, 2002), and path planning for manipulators having more than four DoFs is considered to be complex (Hwang and Ahuja, 1992). As mentioned by Reddy and Varghese (2002), there can be multiple solutions to configure the DoFs of the manipulator for a particular location of the end-effector (i.e., the hook); therefore, simplifying the representation and avoiding the complexity of inverse kinematics should be considered. Figure 1 (Davis and Sutton, 2003) shows an example of a hydraulic crane that has four DoFs for the movements of the boom and hook. The scope of the present work is limited to these four DoFs.

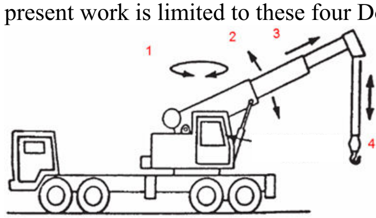


Figure 1. The DoFs of a crane

(2) Engineering requirements: In addition to following the kinematics relationships, the operation of a crane should respect the constraints imposed by the working ranges and load charts. The working range shows the minimum and maximum boom angle according to the length of the boom and the counterweight. Load charts give the lifting capacity based on the boom length, boom angel to the ground and the counterweight. Rules should be developed to represent these constraints which are stored in a database. One important rule is that the distance between two hooks should be equal to the length of the object, and crane load lines must be kept plumb at all

times for multiple crane lift (Shapiro et al., 2000). Other rules include avoiding hoisting and swinging or hoisting and luffing at the same time; and avoiding motion when a crane is travelling. Furthermore, the ground support during lift should meet certain requirements to avoid soil failure problems.

(3) Collaboration requirements: Among the research groups working on path planning in robotics, only a few are focusing on path planning of cooperative manipulators (Sivakumar et al., 2003). Coordination-by-planning technique is the most traditional approach in AI, which is based on breaking actions down into two phases. In the first phase (planning), a set of plans are produced including a set of actions to be carried out by agents to achieve a goal. The selection of a correct representation of actions becomes even more crucial than in the planning for single agents (Ferber, 1999). In the second phase (executing), one plan is selected and then executed. Due to the dynamic environment, re-planning may be needed.

(4) Environment perception requirements: Most of the previous research has been focusing on path planning with assumptions of the site containing static obstructions (Sivakumar et al., 2003). The present work tries to enhance the communication during the execution phase when agents negotiate about conflict resolution. Therefore, knowing the position of each part of the boom and detecting any obstacle on the moving path is essential to ensure that the work is done properly while meeting the kinematics and engineering requirements. Sensors can be used to detect the collisions in real time (Bosche et al., 2006).

(5) Task requirements: Task requirements include spatial and temporal requirements. Spatial requirements define original and destination locations, movement path, and the workspace required. Temporal requirements include the start time, end time, and the duration of a task.

(6) Realization requirements: The volume of data to be exchanged in order to coordinate the actions should be limited to avoid information overflow. The computational complexity should be confined within a certain range to find the trade-off between time and cost.

Computing Aspects

Planning phase

In the planning phase, plans have to be made by the two agents (representing two cranes). Goals are defined in general and are shared by the two agents; however, the tasks of individual agent are different. Each agent should generate a plan for an individual crane based on the environment and the location of the crane while meeting the collaboration requirements.

(1) Representation of goal: Origin (ob, P_o, Φ_o) represents the original position P_o and orientation Φ_o of the object ob. P_o (x_o, y_o, z_o) is given by the coordinate of the reference point of ob. Destination (ob, P_d, Φ_d) represents the destination position P_d and the orientation Φ_d of ob. Duration (t_1, t_2) represents the start time t_1 and the end time t_2 of a task;

(2) Representation of actions: Different movements of a crane can be decomposed into a series of actions. Taking a hydraulic crane as an example, the movement of the crane includes the following actions:
Base movement: BaseMove, BaseStop;
Boom movement: BoomRaise, BoomLower, BoomExtend, BoomRetract, BoomSwing;
Hook movement: HookHoist, HookLower, HookStop, HookGrip, HookRelease.

(3) States description:
State j: ObjectLocation (ob, P_j, Φ_j): object ob is at position P_j with orientation Φ_j;
CraneLocation ($crane_i$, P_{ij}, Φ_{ij}, θ_{ij}, α_{ij}, l_{ij}, P_{ij}^h): crane i is at location P_{ij}, with base orientation Φ_{ij}, boom swing angle θ_{ij}, boom angle to the ground α_{ij}, boom length l_{ij}, and hook position P_{ij}^h; HookGrip ($crane_i$, ob): the hook of $crane_i$ is gripping ob;
Distance ($hook_i$, $hook_{i+1}$, d_j): the distance between two hooks is d_j;

(4) Generating plans based on negotiation between agents
Generating a plan may be seen as a state space search. Most implementations of search algorithms should be assisted by appropriate domain heuristics to find a good/optimal path within a reasonable time (Reddy and Varghese, 2002). Based on the requirements identified in the previous section, the kinematic motion requirements and engineering constraints are integrated to generate reasonable plans for each crane. The whole plan can be divided into tasks which consist of sub-tasks or a set of crane actions. Three major tasks category can be defined as: pre-lift task, lifting task, and post-lift task. The pre-lift task includes the actions for positioning the cranes on site, and attaching the load to the hook; the lifting task, which can be divided into sub-tasks, is the main body of the work; and the post-lift task, which includes detaching the load and moving to another place. The lifting task combines several milestones on the moving path, which can be used as the target when re-planning is needed to reduce the search time. Each task is fulfilled by taking actions to change the states of the crane. Figure 2 shows an example of the movement of one crane. S_j represents different states after the actions.

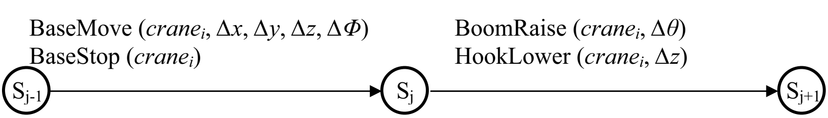


Figure 2. Actions and state changes

Figure 3 shows the schematic representation of the multi-agent system. Three major agents are involved to plan the path and execute the task. The site state agent is responsible for collecting states from the equipment agents and the environment model, including both static and dynamic information. The two equipment agents first generate actions individually based on their own knowledge.The study of Varghese et al. (1997) has shown that no industry-wide standard for heavy lift planning practices exist at present. The experts rely primarily on experience to develop the plans or optimization. Three major criteria should be taken into account: Lift path clearances, capacity during lift, and ground support during lift. Moreover, requirements also limit the possible movement of each crane, which reduces the actions that can be taken by agents. All the information

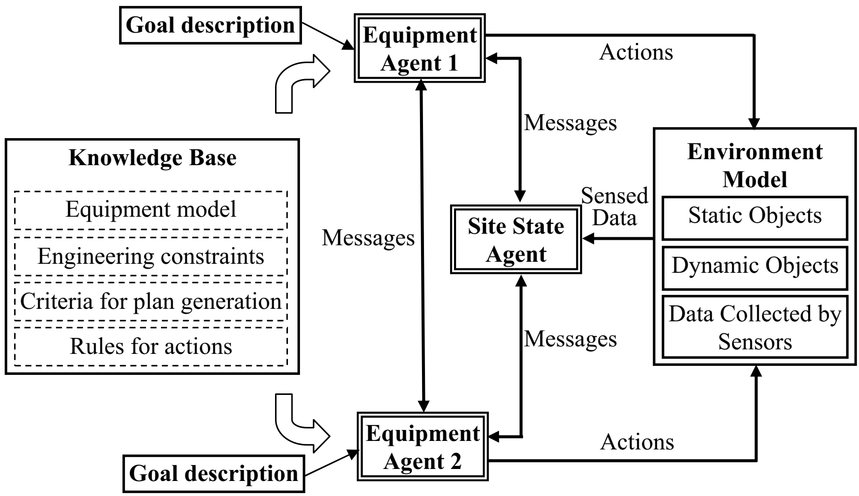


Figure 3. Negotiation between agents

needed is encoded in a knowledge base. In one scenario, one agent is given the priority to generate the actions and the partial plans to move the object lifting half of its weight. The other agent can follow by taking reacting actions or reject the actions due to its own constraints. Rules are developed to check the possibility of each action based on the requirements described above. The priority of an agent may change according to specific rules. Through negotiation, an effective plan can be generated based on possible combinations of movements of cranes from one step to another.

Executing phase

Obstacles in the environment may move and are not static during the executing phase. Due to the dynamic changes in the construction site, whenever an action is going to be undertaken, the agents can detect spatial conflicts that may happen on the path. Sensors are used to position the boom and hook location and detect any spatial obstacle nearby. If any change is needed in the plan, the agents will communicate and negotiate with each other to decide the next action while respecting engineering constraints and other requirements (Figure 3). The same rules used in the planning phase are applied again to check the possibility of further actions.

Case Study

The re-decking project of Jacques Cartier Bridge in Montreal is used to demonstrate the proposed collaborative agent-based system. The deck of this bridge was replaced in 2001-2002. The existing deck was removed by saw-cutting the deck into sections. Each section was removed by two telescopic cranes and a new panel was installed using the same cranes. Figure 4 shows two telescopic cranes positioned on both sides of the section to be replaced.