Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

MODELING ORGANIZATIONAL BEHAVIORS OF CONSTRUCTION ENTERPISES: AN AGENT BASED MODELING APPRAOCH

Jing Du

Department of Construction Science
The University of Texas at San Antonio
San Antonio, TX 78249, USA

Mohamed El-Gafy

School of Planning, Design and Construction
Michigan State University
East Lansing, MI 48823, USA

ABSTRACT

In an effort to explore the complexity of construction organizations, this paper introduces a comprehensive Agent Based Model called Virtual Organizational Imitation for Construction Enterprises (VOICE). Building its ground on the findings of organizational behavior, VOICE models three critical aspects of construction organizations including Work, Actors and Organization. Then different levels of organization processes are reproduced to simulate the transition from micro-level processes to the collective performance. As an attempt of developing a comprehensive, all-inclusive simulation model for construction organizations, this work sets a stepping-stone for future studies.

1 INTRODUCTION

Construction industry is in general suffering from significant performance problems (Du and El-Gafy 2012). One possible reason is the increasing 'complexity' of today's projects. Traditionally, the process of construction projects is regarded as an ordered, linear system, which can be planned, controlled and managed top down (Bertelsen 2003). However, recent development of the industry gives rise to a thinking that the process of construction projects may be not as ordered and predictable in its nature as it may look. Closer examinations revealed that construction projects are indeed complex, nonlinear and dynamic phenomena, or complex systems (Bertelsen 2003). It has been found that such complexity not only does come from the growing uncertainties in both technology and processes, but also from the increasing influence of intangible organizational factors (Chan et al. 2004). There does exist an urgent need to explore construction organizations from the perspective of organizational behavior.

This paper introduces the use of Agent Based Modeling (ABM) for exploring the organizational behaviors of construction organizations as complex systems. Unlike other simulation technologies, such as System Dynamics, ABM captures the emergent properties in a bottom-up manner, which has been proven to be a more effective tool for investigating complex systems (Macal and North 2007). We identified the critical building components of organizational behaviors based on previous studies (Robbins 2005), and then tailored them to meet the unique features of construction organizations. A simulation model that integrates all components was developed called Virtual Organizational Imitation for Construction Enterprises (VOICE). The rest of this paper introduces technical details of VOICE.

2 LITERATURE REVIEW

2.1 Organizational Behaviors

Following the findings of organizational science (Robbins 2005), individuals, teams and groups act interdependently in organizations to achieve an ultimate goal.

978-1-4799-7486-3/14/$31.00 ©2014 IEEE

On the one hand, psychosocial conditions of individuals, as the fundamentals of organizational processes, significantly affect the technical quality of any formal or informal group activities. On the other hand, the team runs as a group of distinct individuals and exerts noticeable impacts on behaviors (Robbins 2005). People-organization relationships play an irreplaceable role in forming organizational behavior and defining the outcomes. Building on Robbins and Langton's (1998) model of organizational behavior (Figure 1), work performance is human output dependent on individual behaviors, group behaviors and the organizational system, and affected by environmental change and stress.

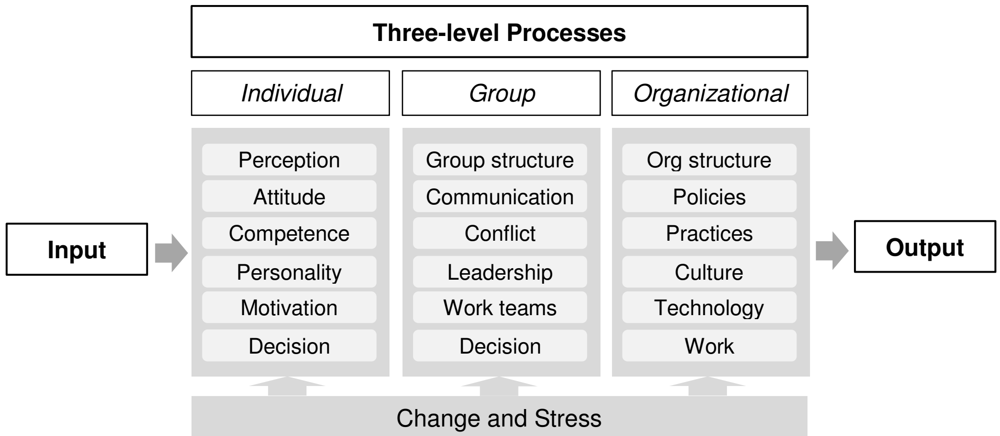

Figure 1. Model of Organizational Behaviors; adopted from (Robbins and Langton 1998)

Robbins and Langton's work (1998) constitutes the point of departure of this research as to develop a modeling architecture for simulating construction organizations. It suggests a system approach to be adopted in the investigation that considers interactions of a variety of process levels including individual level, group level and organizational level. An overall analysis on the construction literature has also highlighted the fact that construction organizational processes can be naturally grouped into three categories: individual factors, organizational factors, and work-related factors. These three categories of factors define organizational behavior in construction settings, which pictures a transitions from micro-level processes (such as individual actions) to collective performance (such as team performance).

2.2 Agent Based Modeling

Agent Based Modeling (ABM) is a suitable tool for investigating the transition from micro-level processes to collective performance in a range of problems (North and Macal 2007). It builds a common environment for heterogeneous and autonomous agents to share, and allows the agents to simultaneously interact with each other for self-interest (Ligmann-Zielinska and Jankowski 2007). Unlike top-down modeling approaches (e.g., System Dynamics, Discrete Event Simulation etc.), in ABM the collective behavior of the simulated system is not predefined, but emerges from individual agents who act based on what they perceive to be their own interests; thus ABM is capable of reproducing the emergent properties of the studied systems (Macal and North 2007).

ABM has been utilized by a small but growing community of scholars to tackle a range of difficult problems in construction engineering and management (Taylor 2010). Recently, there is an increasing use of ABM to study the complex behaviors of project organizations and network including organizational performance, work related behaviors, project team cooperation, and cross-cultural issues (Du and El-Gafy 2012; Horii et al. 2005; Jin and Levitt 1996; Taylor and Levitt 2007). Representative work is Virtual Design Team, or VDT (Jin and Levitt 1996). As a multi-agent based simulation platform, VDT models a project organization as a channel of communication, and human behaviors can be investigated in the context of information processing (Jin and Levitt 1996). As a well-documented and well-validated organizational simulation tool, it is widely applied to study project-based organizations to estimate project durations, costs, and quality. Following VDT, the authors developed an ABM simulation model, VOICE, as a comprehensive simulation platform to investigate a system of diverse work-related behaviors in construction organizations (Du and El-Gafy 2012). VOICE recognizes the importance of the systematic interdependency among different types of human behaviors, and treats them as the foundation of the organizational behavior.

3 MODEL ARCHITECTURE

Figure 2 illustrates the architecture of VOICE. It tailors Robbins' model of organizational behaviors (refer to Figure 1) to suit construction organizations, with 3 main components being modeled: (1) Work: construction organizations are project based organizations (PBOs), and thus projects and corresponding tasks are modeled as the sole input as that in Robbins' model; (2) Actors: project tasks are performed by the individuals in a construction organization, whose personalities, value and attitudinal factors affect the perceptions toward the tasks, leading to diverse micro-level actions (or behaviors) directly related to the work performance; and (3) Organization: a variety of organizational structures that arranges lines of authority, work and communications, and allocates rights and duties. In addition, key performance indicators of project team performance are modeled as the main output. The architecture illustrated in Figure 2 reflects the bottom-up process of organizational behavior (input-individual level process - group process - organizational process - output) as suggested by Robbins (2005). VOICE conceptualizes and integrates all components into a comprehensive and integral model.

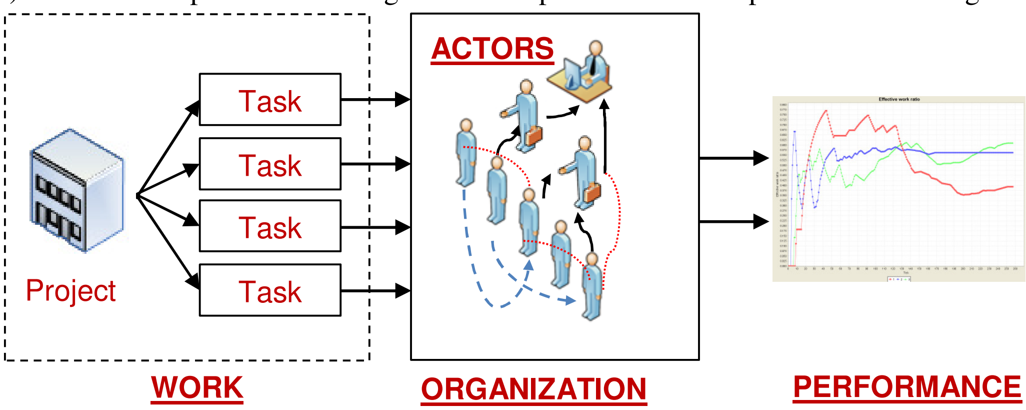

Figure 2. Model architecture of VOICE

3.1 Modeling Work

VOICE models projects and corresponding tasks as the sole input. A project is a multi-task effort that must be completed by a group of actors with each of them having different characteristics (e.g., size, complexity, mistake percentage, etc.). In VOICE, a project generates a list of tasks according to the predefined task arrays (a set of arrays indicating the sequences and attributes of a list of tasks). A task is a basic executable work effort that is assigned to corresponding actors sequentially for processing. Tables 1 and 2 illustrate the attributes of project and task in VOICE.

Table 1. Attributes of projects

Name	*Value*	*Remark*
ID number	[1, *]	A unique and universal ID number for each project
Size	(0, *]	Size of the project, measured as the sum of its tasks' work amount
Priority	[1, *]	An indicator of the project's priority; the higher the more urgent
Complexity	[1, *]	An indicator of the work difficulty of a project's tasks; higher number means more time for processing its tasks
Time limit	(0, *)	The time requirement of a given project, measured in the ticks
Cost limit	(0, *)	The budget of a given project
Mistake	[0,1]	The final mistake percentage of a project, it's a function of the mistake percentage of all its tasks

Table 2. Attributes of tasks

Name	*Value*	*Remark*
Task number	[1, *]	A universal number (but not unique) for each task
Project ID label	[1, *]	An indicator overridden from project ID number, showing which project this task belongs to; a particular task cannot be identified unless both task number and project label are given: Task.id=task.number & task.projectLabel
Work amount	(0, *]	The work amount of task, measured as the work hours when processed by an actor with competence=1. It relates to the size of the project: Project.size=$\sum$task$_i$.amount
Priority	[1, *]	An indicator of task's priority, which relates to the priority of the project; higher priority means the work can jump to the top of work list. The actual priority relates to the project priority: task.actualPriority=project.priority*task.priority
Difficulty	[1, *]	An indicator of the work difficulty of a task. The actual difficulty level of a task relates to the project complexity:
Authority level	1,2,3	Indicating which position level has the right to process or approve a give task; for example, task "deciding profit rate" is fairly high authority work, with authority level of 3
Approval	0, 1	0 means this task has not been approved ;1 means it has been approved
Dependence	0, 1	1 means following tasks are dependent on this task while 0 means not
Information status	0, 1	0 means this task can be processed without more information, while 1 means more information is needed
Concurrent indicator	1..,n	An absolute number indicating how many following tasks will be generated after the completion of this task, the default value is 1
Mistake	[0,1]	The final mistake percentage of a task, depending on the actor's processing quality
Starting address	-	The address of the 1st actor for this task; the task will be assigned to this actor, and then passed through all relevant actors.

VOICE reproduces a concurrent working environment, where multiple projects and tasks can enter into the model simultaneously. This is why labeling each generated task with project ID is important. VOICE also captures dependencies among tasks, hence the generation of every task is strictly dependent on an attribute "*task.dependence*". *task.dependence*=1 means the successive tasks of a given task are dependent upon the finish of it, and if it is still undergoing, the successive tasks will not be generated. Once generated, tasks are assigned to the designated actors according to the actual work flow, and such designation information is embedded in the attribute of "*starting address*". In an effort to embed task attributes in the simulation, matrices are used. For example, the following 3 by 20 matrix is used to record work amount of 20 tasks of 3 projects respectively.

```
double[][] AmountArray=new double[3][20]
AmountArray=[[5,5,10,15,15,10,20,20,25,10,5,5,10,15,20,15,10,10,5,5],[5,9,10,10
,15,12,20,10,25,10,10,10,10,15,10,15,10,10,5,10],[15,15,10,15,15,10,20,20,30,20
,5,5,10,15,20,15,10,10,5,8]]
```

In the simulation, when a task is generated, the corresponding "work amount" value will be read from the matrix above. In the same way, all the task attributes are assigned to each generated task according to relevant matrices (Figure 3). It should be noted that unlike existing organizational simulation models such as VDT (Levitt 2007), tasks in VOICE are modeled as instancialized objects. They are generated by the project, passed onto and through actors, and removed from the model only when they are finished. The work flow is reflected as the sequential execution of attributes matrices. We found it to be more efficient in the simulation compared to building a hybrid model of ABM and Discrete Event Simulation (DES).

3.2 Modeling Actors

Project teams are modeled as a group of actors who are able to execute the generated project tasks. VOICE follows a set of assumptions on the basis of behavioral decision theory (Cyert and March 2005): (1) there are multiple roles in a construction organization (e.g., president, manager and staff), and each role is represented by a type of agent; (2) each agent is imbued with bounded rationality; this moves toward more specific and particular work aims instead of following a common aim for the whole organization.; (3) each agent occupies a particular position inside the organization that defines what task(s) the agent does, and with whom the agent interacts. As a result, the authority position and work arrangement determine the agent's behaviors; and (4) each agent possesses specific knowledge, skills, and capabilities.

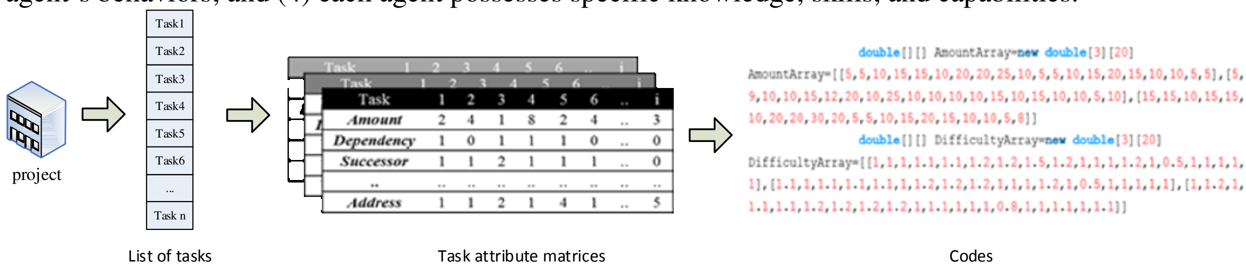

Figure 3. Convert project to task attribute matrices

3.2.1 Actor Attributes

Actor attributes are modeled as Table 3.

Table 3. Attributes of actors

Name	*Value*	*Remark*
Competence	(0, *]	An indicator of actors' work efficiency, measured with the processed work amount in unit time.
Exception handling time	(0, *]	An absolute quantity indicating the time needed to deal with exceptions (emerged tasks, e.g., staff's high work related stress, work quality issues).
Capacity	(0, *)	The most acceptable total work amount for an actor
Work quality	(0, *]	A relative quantity indicating the processing quality, measured in 1- percentage of processing mistakes.
Authority position	1,2,3	An indicator of the actors' position levels: president=3, manager=2, staff=1. This attribute is related to the task authority level: if task.authority>actor.authority, then the actor will ask for approval from higher authority position.
Assigning preference	1,2,3	When a task is assigned to the staff (only applicable for manager): 1 means assigning tasks to the staff who can finish the task in the shortest time; 2 means assigning tasks to the staff that can finish the task with the best quality; and 3 means assigning tasks to the staff with the smallest work pressure.
Quality preference	1,2,3	When a task delivered/submitted to the manager has quality issues: 1 means returning the task to who processed this task; 2 means reassigning the task to other subordinates that can finish the task with the best quality (only applicable for manager); and 3 means correcting the mistakes by self.
Quality threshold	(0, 1)	An absolute quantity indicating the quality threshold of an actor; if the delivered/submitted task's mistake percentage is bigger than the threshold, then the actor will take action based on quality preference.
Exception indicator	(0, *]	An indicator monitored by the president showing the total management exceptions happening in the organization.
Exception threshold	[1, *]	An absolute quantity indicating the exception threshold of the president; if the exception indicator is bigger than the threshold, then the president will set up a meeting to address these exceptions.
Salary rate	(0,*]	An absolute quantity showing the salary rate for the actor

3.2.2 Individual Behaviors

In VOICE, behaviors are represented as the most fundamental and generic activities of an actor that constitute his/her daily work related actions. Individual relational behaviors are different from a list of tasks: relational behaviors are abstract and conceptual activities that an actor might take to finish one or more tasks. For example, to evaluate subcontractors' quotes (a task with a specific target), an estimator may need to “communicate” and “coordinate” (work-related relational behaviors that are needed for fulfilling the task). The complex actions of each actor can be regarded as the result of the modification and combination of the behaviors in this list.

Table 4. Fundamental behaviors modeled in VOICE

Category	***Definition***	***Behavioral Modules***	***Description***
Routine work	The most generic and daily activities associated with the direct finishing of tasks	Prioritizing	Comparing the priorities of concurrent projects as well as relative priorities of tasks to make a work plan where most emergent task jumps to the top of work list
		Processing	Directly working on specific task. Processing tasks is demonstrated as the reduction of remaining work amount. Work time and quality depend on competence and work quality and task difficulty represent the reduction of tasks' work amount
		Submission	Submitting the finished tasks to supervisor/co-workers
Coordination	Activities to maintain the consistency of work flow	Assigning	The manager/coordinator/president assigns tasks to different subordinates based on ‘assigning preference’
		Requesting approval	Certain tasks need to be approved by higher management lines before processing, e.g., “determining profit margin”. This activity is considered to be a management exception which affects work effectiveness and efficiency.
		Approving job	Management approves the tasks per the request of subordinates. This activity is considered as a management exception which affects work effectiveness and efficiency.
		Conflict management	The manager/coordinator works on solving the exceptions attributed to conflicts. For example, the proposal team coordinator works with the engineering team coordinator to clarify information for proposal development.
Communication	Exchanging project task-related Information	Information exchange	Certain tasks need extra information from other actors before processing. Hence the communication is shown as changing of tasks' information status in VOICE. This activity is also considered as a management exception.
		Meeting	An event that clears work-exceptions by enhancing information exchange and solving problems. Meetings aim to eliminate management exceptions. They can enhance the communication, information sharing and other work related problems. The duration is determined by the number of management exceptions.
Trust-related	Believing other team members will perform their duties without supervision	Monitoring quality	Comparing the quality of delivered/submitted work with own quality threshold, and if quality is not satisfied, work will be reworked, returned, reassigned or reported based on “quality preference”. Unsatisfactory work is also a management exception.
Reciprocal	Extra activities generated by different reciprocal task interdependence	Stress-coping	If the total work amount of assigned tasks is over the capacity of actor, he/she may suspend the work and report work related overload to upper level management. The management, based on different preference, may return the work or reassign the work. This action is considered as a management exception.
		Correction/rework	If a job is returned by other departments/actors, the actor may redo/reassign/return/report the job according to different preferences. This action is considered as a management exception.

The above behavioral modules constitute the fundamental actions of and between team members. Figure 4 depicts the behavioral modeling in VOICE: It starts with work-related events that refer to those directly relevant to the completion of project related tasks, such as a proposal meeting between the estimators and the engineers, or a RFI for the suppliers. Once an event is perceived by an individual, a decision-making process starts. The decision-making process is a complex process involving several stages, affected by individual characteristics (e.g., preferences and roles in the team), rather than as an instantaneous stimulus-based response (Gwynne and Kuligowski 2009). The outcome of a decision-making process is the adoption of one or more behaviors from the behavioral modules as illustrated in Table 4, which in return, become the actions an actor would take. Ultimately, the action outcomes will affect the events, which represent the start of a new loop of the above process. Therefore, a typical behavioral modeling in VOICE is comprised of three components (Figure 4): Trigger events (E), decision-making process (D) and corresponding actions represented as the behavioral modules (A). The simulation of behaviors starts with a particular triggering event, such as a new task. Then it is divided into two sub-processes: (1) one models the decision-making process (D-process), which is represented as "what-if" evaluations made on every contingency. For example, "if current task needs approval?" and (2) the other models the action process (A-process), which refers to the actual work conducted by the actors and modeled as behavioral modules. For example, if the answer to "if current task needs approval?" is "Yes" then the action should be locating the upper management and sending the request for approval. In this way, although different actors may possess different behavioral flowcharts, the final behaviors and decision-making processes of a particular actor are simply a combination of these fundamental components.

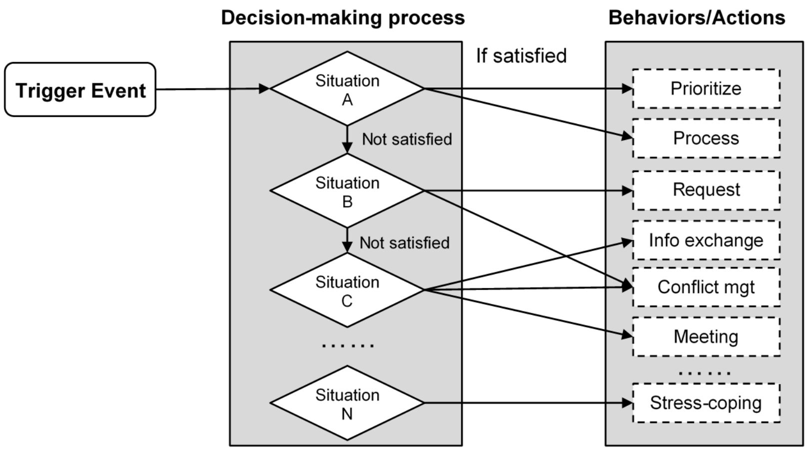

Figure 4. Mapping situations to behaviors

3.3 Modeling Organization

VOICE represents a construction organization with three types of structures: (1) administrative structure; (2) work process; and (3) information flow as the channel of information. Each component symbolizes a particular channel between actors when a related set of behaviors occurs.

3.3.1 Administrative structure

Administrative structure is the formal organizational structure that determines the arbitrary delegation pattern within an organization and reflects report directions. It can be demonstrated by a formal organizational chart (Daft 2009). In VOICE, this administrative structure is modeled as an association class connecting president, managers and staff. VOICE instancializes the gird, and then the administrative connection as a network projection. This network will be referred to when administration related activities are undergoing (such as a manager assigning a task to the subordinates), as shown in the following lines of code:

```
Projection grid = (Grid)FindProjection("VOICE/Grid") //instancialize grid
Network network = (Network)FindProjection("VOICE/Admin")//instancialize network
…
Iterator lowers = new NetworkPredecessor(network, this).query().iterator()
```

All administration related activities taking place among actors occur through the administration structure, such as "requesting and approving job" or "assigning job." VOICE represents the administrative structure by solid black lines, and the arrow represents the report direction. The current VOICE limits the administrative structure to a three-level pattern. Additional levels will be presented in future work.

3.3.2 Work process

Work process in the construction domain refers to the "necessary procedure in construction companies' execution of their business" (Cheng and Tsai 2003) that typically includes the planning and control of projects through conceptual planning, design, bidding, construction, and commissioning. VOICE conceptualizes the work process as the pattern of task sequence and work arrangement, i.e., the channel that allows a sequence of managerial activities to flow through actors.

In VOICE, work process is modeled as an association class connecting actors and tasks. Direct task processing is executed via the work process structure. For example, an estimator (staff) is required to conduct a WBS analysis on a project (this task is passed from project to actor via the work process structure), and he/she passes the result to another estimator for further work (the task is passed from one actor to another via the work process structure). The work process is represented by developing a Task-Actor Relation Table (TART), which reflects the task assignment for every actor. In a TART table, all the tasks are arranged according to the sequence of the task queue, which is shown as the column heads; all the actors are arranged in the row head. Then, the cells are marked if an actor is responsible for a task (e.g., an estimator helper is responsible for quoting subcontractors). The work process pattern is shown with zigzag lines inside the TART. Finally, the real world work process is translated to abstract task delivery pattern in VOICE.

3.3.3 Information flow

Information flow is the formal/informal communication connection that determines the coordination, message and knowledge movement. Because information flow is the most intangible structure compared to the other two structures, representing information flow requires intensive and reactive interview efforts except for document studies. In VOICE, information flow is modeled as a set of channels where work related information is transferred from one actor to another, by modeling it as a class of associations reflecting the coordination, message and knowledge movement among actors, while at the same time reducing uncertainty (Nonaka and Konno 1999). The movement of information is coded in the "ask-and-answer" fashion. For example, an estimator asks an accountant for a salary quote for site workers in order to finish an estimation task; this is modeled as passing the task with a null information index (task.infoStatus=0) to the accountant; and the accountant's answer can be modeled as changing the information index to applicable (task.infoStatus=1) and returning the task to the estimator. As a result, the reciprocal information exchange activities can be demonstrated with two monodirectional flows.

3.4 Performance Outcomes

Specific measures of the performance outcomes are presented as follows:

3.4.1 Time

The duration of a project (t_m) follows Equations (1) and (2):

$$\text{project.size}(t_{n+1}) = \text{Project.size}(t_n) - \sum \text{task}(t_n).\text{amount} \quad (1)$$

$$\text{if project.size}(t_m)=0, \text{ then } t_m \text{ is the time for a project} \quad (2)$$

Moreover, VOICE provides indicators for time used for different activities, as shown in Equations (3), (4) and (5):

$$\text{Total time} = \text{process time} + \text{communication time} + \text{idle time} \quad (3)$$

$$\text{Work time} = \text{process time} + \text{communication time} \quad (4)$$

$$\text{Effective time} = \text{process time} \quad (5)$$

Idle time is the span of time when the actors "have nothing to do" (normally this is due to waiting), and process time stands for the time directly used for processing the tasks, following Equation (6):

$$\text{Process time}=\text{task.actualDifficulty}*\text{task.amount}/\text{actor.competence} = (\text{project.complexity}*\text{task.difficulty})*\text{task.amount}/\text{actor.competence} \quad (6)$$

3.4.2 Quality

Quality of a project team is associated with "detected failed production work volume" (Jin and Levitt 1996). This research uses a mistake percentage to represent the failed production work amount of each task, which ranges from 0 to 1 (100%). Provided that processing tasks are the sole function of a project team in this research, the ultimate quality of the entire teamwork is thus described as one minus weighted sum score of task mistakes, as shown in Equation (7):

$$\text{project.quality}=1-\sum(\text{task}_i.\text{amount}*\text{task}_i.\text{mist})/\text{project.size} \quad (7)$$

where task_i.mist stands for the percentage of mistakes in a particular task and follows Equation (8):

$$\text{task.mist(after processing)}= \text{task.mist (before processing)} + (1-\text{staff.quality}) \quad (8)$$

3.4.3 Effectiveness

Following Jin and Levitt's (1996) work, the effectiveness of a project team is defined as the percentage of productive work time versus total work time. This research therefore formulates effectiveness as a ratio of effective work time (direct processing of tasks) to total work time. It is given by the Equation (9):

$$\text{project.effectivess}= \sum(\text{effective time}/\text{Total time}) \quad (9)$$

3.4.4 Efficiency

Levitt (2007) defines two types of efficiency for any project team. Among them, time efficiency is given by a ratio of estimated work duration to simulation duration. If setting simulation duration to a standard unit of time such as one day, the estimated work duration demonstrates the actual work (measured in man hours) that can be finished by the team within one day. It reflects the average work amount completed per time unit. This research follows a similar definition and measures efficiency as Equation (10):

$$\text{project.efficiency}= \sum\text{task}_i.\text{amount}/\text{Total time} \quad (10)$$

3.4.5 Work related pressure

Work related pressure is a subjective judgment made by team members regarding the relation between work environment and individual ability (Cox et al. 2010). Although it is hard to quantify, perceived work related pressure is directly related to the workload (Hall 2004). From a purely technical perspective, this research measures work related pressure as total work amount (measured in time) of all tasks at hand, following Equations (11) and (12):

$$\text{staff.workPressure}=\sum(\text{task.remainingAmount}\times\text{task.diff}/\text{staff.comp}) \quad (11)$$

$$\text{task.remaingingAmount}=\text{task.amount}-\text{finished amount} \quad (12)$$

4 DEVELOPMENT OF VOICE

A set of ABM development platforms were compared in terms of their flexibility, richness of libraries, development user interface, analytical functions, and support from developers. REcursive Porous Agent Simulation Toolkit Simphony, or Repast S (Collier 2003) was determined to be a suitable development platform for this research. Repast S is a pure Java-based implementation, which has been widely accepted recently in academia. It builds on Object-Oriented Programming (OOP) where each object can be naturally regarded as an agent in VOICE. Repast S is an open source platform so the functions provided can be easily tailored to the requirements of this research. In addition, developers of Repast S have built an interactive supporting forum where development problems can be solved immediately. A snapshot of Repast is provided below (Figure 5).

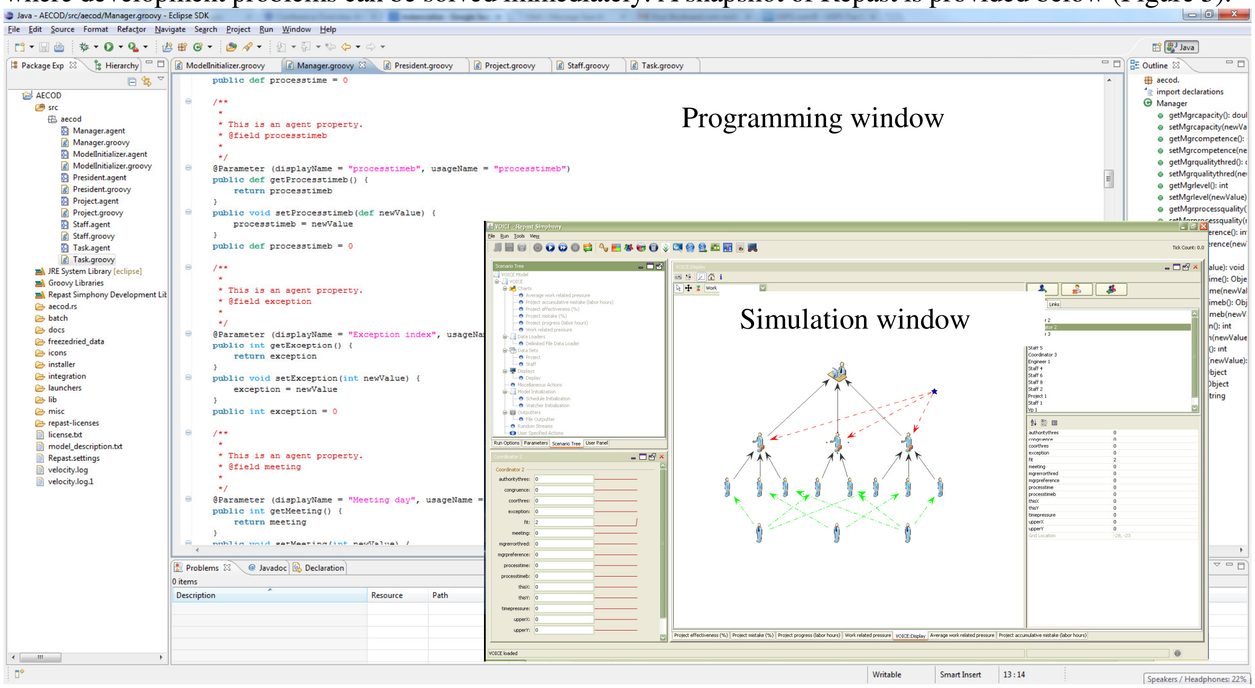

Figure 5. The runtime interface

5 IMPLEMETATION EXAMPLE

VOICE was used to model an Engineering Procurement and Construction (EPC) company to investigate the interdepartmental cooperation between the proposal development team and the engineering team. Data was collected in a survey and a series of interviews to the team members. Figure 6 illustrates the developed simulation model. As shown in this Figure, the simulation model captures the three-level organizational structure with 3 types of agents including vice presidents (of two teams), coordinators (of two teams), proposal staff and engineers (comparable to staff). The project tasks are mainly proposal development related, and information is needed from the engineers for proposal staff to develop the proposal.

A range of simulation experiments has been conducted to test the influences of a set of behavioral and organizational factors on the team performance in proposal development, which includes goal congruence (how well the work goals of two teams are aligned), task dependence (the average dependence level of all tasks), micro-management of the coordinators (shown as the number of communication iterations between the engineers and the proposal staff before the coordinates intervene; thus less iterations means leaning to micro-management) among others.

Figure 7 illustrates results of two Monte Carlo simulation experiments using VOICE. The left plot indicates that higher level of task dependence can increase the effectiveness of proposal development, although such influence can hardly be described using a linear function. Meanwhile, micro-management (increased number of iterations) does little to improve the effectiveness. The right plot examines the influence of goal congruence and micro-management on the quality (measured as the percentage of mistakes). As shown under different levels of goal congruence, micro-management may have differing impacts on the quality of work. If two teams are sharing aligned goals, micro-management is actually detrimental to quality.

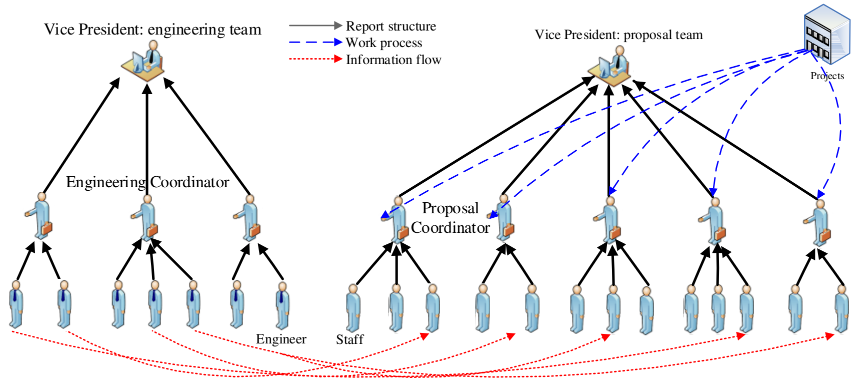

Figure 6. Snapshot of VOICE simulation

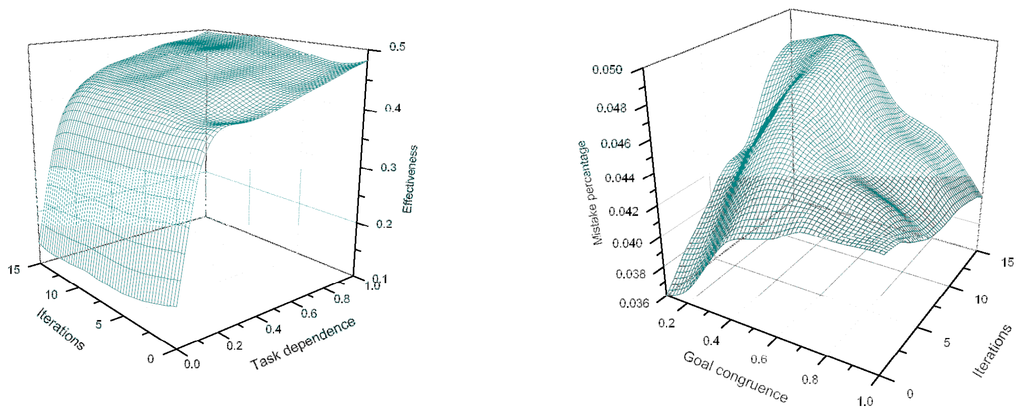

Figure 7. Example simulation results

6 DISCUSSION AND CONCLUSION

Although the complexity of present construction organizations has been well recognized, it lacks a comprehensive and all-inclusive simulation model that can help capture the complex organizational behaviors of construction organizations, especially the transition from micro-level behavioral processes to collective performance. Building on Robbins and Langton's (1998) model of organizational behavior, the performance of construction organizations is a systematic result of the three-level processes interacted with each other, including individual process, group process and organizational process. Such interaction and integration of three-level processes generate unexpected performance outcome which can be captured by ABM, as illustrated in the implementation example. The proposed model VOICE proves the applicability of modeling organizational behaviors of construction organizations using ABM. The main contribution of this work is to provide a simulation platform for organizational studies in the area of construction engineering and management. Scholars in this area may download the program code of VOICE at https://sites.google.com/site/dujresearch/.

Admittedly, this work is in its infancy. Modeling organizational behavior from simplified perspectives, VOICE may not be able to create realistic simulations for different organizational behaviors. The future work will be focusing on expanding the factors and processes modeled by VOICE to capture a wider range of organizational behaviors. More implementation case studies will also be conducted with VOICE to show comparisons of how an organization is performing in the real world versus VOICE. More real data from different companies will be collected in order to define behaviors, work process, and interactions. This will result in more realistic results.

REFERENCES

Bertelsen, S. (2003) "Construction as a Complex System." *Proceedings of IGLC 2013*, PA, USA, 11-23.
Chan, A., Scott, D., and Chan, A. (2004). "Factors Affecting the Success of a Construction Project." *Journal of Construction Engineering and Management*, 130(1), 153-155.
Cheng, M., and Tsai, M. (2003). "Reengineering of Construction Management Process." *Journal of Construction Engineering and Management*, 129(1), 105-114.
Collier, N. (2003). "Repast: An Extensible Framework for Agent Simulation." *The University of Chicago's Social Science Research*, 36(2003), 1-18.
Cox, T., Griffiths, A., and Rial, E. (2010). "Work Related Stress." *Occupational Health Psychology*, 31-56.
Cyert, R., and March, J. (2005). *A Behavioral Theory of the Firm*, Blackwell. Hoboken, NJ.
Daft, R. (2009). *Organization Theory and Design*, South-Western Pub. Mason, OH.
Du, J., and El-Gafy, M. (2012). "Virtual Organizational Imitation for Construction Enterprises: Agent-Based Simulation Framework for Exploring Human and Organizational Implications in Construction Management." *Journal of Computing in Civil Engineering*, 26(3), 282-297.
Gwynne, S., and Kuligowski, E. (2009). "Simulating a Building as a People Movement System." *Journal of Fire Sciences*, 27(4), 343-368.
Hall, D. S. (2004). "Work-Related Stress of Registered Nurses in a Hospital Setting." *Journal for Nurses in Staff Development*, 20(1), 6.
Horii, T., Jin, Y., and Levitt, R. (2005). "Modeling and Analyzing Cultural Influences on Project Team performance." *Computational & Mathematical Organization Theory*, 10(4), 305-321.
Jin, Y., and Levitt, R. (1996). "The Virtual Design Team: A Computational Model of Project Organizations." *Computational & Mathematical Organization Theory*, 2(3), 171-195.
Levitt, R. E. (2007). "The Virtual Design Team (VDT): A Computational Model of Project Teams." <http://cee.stanford.edu/programs/construction/what/pdfs/VDT_Overview_0711.pdf>.
Ligmann-Zielinska, A., and Jankowski, P. (2007). "Agent-Based Models as Laboratories for Spatially Explicit Planning Policies." *Environment and Planning B: Planning and Design*, 34(2), 316-335.
Macal, C., and North, M. "Agent-Based Modeling and Simulation: Desktop ABMS." IEEE Press Piscataway, NJ, USA, 95-106.
Nonaka, I., and Konno, N. (1999). "The Concept of 'Ba': Building a Foundation for Knowledge Creation." *The knowledge management yearbook 1999-2000*, 40-54.
North, M., and Macal, C. (2007). *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation*, Oxford University Press, USA.
Robbins, S. (2005). "Organizational Behavior (11th)." Prentice Hill.
Robbins, S., and Langton, N. (1998). *Organizational Behavior: Concepts, Controversies, and Applications*, Prentice Hall Englewood Cliffs, NJ.
Taylor, J. (2010). "Invitation to Submit Scholarly Articles Using Agent-Based Simulation to Tackle Challenging Civil Engineering Problems." *Journal of Computing in Civil Engineering*, 24, 465.
Taylor, J., and Levitt, R. (2007). "Innovation Alignment and Project Network Dynamics: An Integrative Model for Change." *Project Management Journal*, 38(3), 22-35.

AUTHOR BIOGRAPHIES

JING DU is an Assistant Professor in Department of Construction Science at the University of Texas at San Antonio. He received a Ph.D. in Construction Management from Michigan State University. His email address is jing.du@utsa.edu.

MOHAMED EL-GAFY is an Associate Professor in the MSU School of Planning, Design and Construction. He received a Ph.D. in Civil Engineering from Florida State University. His email address is elgafy@msu.edu.