The Virtual Design Team: A Computational Simulation Model of Project Organizations[1]

JOHN C. KUNZ[2], RAYMOND E. LEVITT, YAN JIN

Abstract

The long range goal of the "Virtual Design Team" (VDT) research program is to develop computational tools to analyze decision making and communication behavior and thereby to support true organizational (re)engineering. This article introduces the underlying theory, the implementation of the theory as a computational model, and results from industrial test cases. Organization theory traditionally describes organizations only at an aggregate-level, describing and predicting the behavior of entire organizations in terms of general qualitative predictions. We define and implement a "micro" theory of the structure and behavior of components of organizations, explicitly representing activities, groups of people called "actors," and organizational structure and policies for project teams. A VDT model can be "run" by a discrete event simulation. Emergent aggregate model output behaviors include the predicted time to complete a project, the total effort to do the project, and a measure of process quality. More detailed model behaviors include the time-varying backlog of individual actors and the "exceptions" associated with activities. The results are detailed and specific, so they can guide specific managerial interventions in a project team and can support sensitivity studies of the relative impact of different organizational changes. We conclude that such a theory is tractable and predictive for complex but relatively routine, project-oriented design tasks.

The application for which VDT offers unique new kinds of insights is where an organization is striving to shrink time to market dramatically for a product that is similar to ones it has previously developed. Reducing time to market dramatically almost always requires that previously sequential activities are executed more concurrently. In this situation, experienced managers can still correctly identify the required activities and estimate their durations and skill requirements; but they almost always underestimate the increased workload arising from exponentially higher coordination needs and the propagation of rework between the now highly concurrent activities. The VDT framework, which explicitly models information dependency and failure propagation between concurrent activities, has proven to be far more accurate, and to incorporate a wider range of parameters, than CPM/PERT process models for these fast-paced development projects.

---

[1] Published in the *Communications of the Association for Computing Machinery*, 41(11), pp 84-92, 1998.

[2] *Center for Integrated Facility Engineering, School of Engineering, Stanford University, Stanford, CA 94305*

(ORGANIZATION THEORY; ORGANIZATION DESIGN; INFORMATION PROCESSING; SIMULATION; ARTIFICIAL INTELLIGENCE; SYMBOLIC MODELING; COORDINATION THEORY; COLLABORATION TECHNOLOGY)

1. Introduction

Faced with increasingly competitive global markets and tight-fisted taxpayers, many private and public organizations now "reengineer" their organizations to improve their products or services and to reduce time between receipt of a new order and delivery of a requested product or service to a satisfied customer. When managers change existing work processes to reduce schedules dramatically, interdependent activities that were previously performed sequentially must be then performed concurrently. Organization theory predicts that coordination of concurrent interdependent activities is significantly more difficult and costly than coordination of the same activities performed sequentially. Yet traditional organization theory can neither predict the magnitude nor the specific actors and activities that require incremental coordination, even though coordination load and rework can grow exponentially as there is greater concurrency of complex, interdependent activities performed in parallel.

In contrast with today's empirical approach to developing organizations, engineers have long designed artifacts such as bridges and airplanes using computational models. The engineer models a design in the computer, analyzes it, changes it, and only after the design is well understood is it finalized and released for construction or manufacture. The vision of the Virtual Design Team (VDT) project is that managers should design organizations the same way engineers design bridges: by building and analyzing computational models of planned organizations and the processes that they support.

Our approach in the VDT project (Levitt, 94) is to extend organization theory so it considers individual organizational entities such as actors, activities, and both direct and coordination work. We represent this "micro" theory as a non-numeric (symbolic) model in the computer. We implemented the symbolic model using AI object-oriented symbolic representation tools and methods. To simulate the behavior of projects, we link the symbolic work process and organization model to a discrete event simulator that we developed.

1.1 VDT Analysis Objectives

In general, project clients, financiers and managers want answers to questions such as the following:

- Can a specific engineering team complete a project within a given (usually reduced) time schedule? If not, which specific disciplines or management groups can a manager augment, and what are the changes to estimated project costs, duration and quality of such additional staff?
- What are the predicted effects on project cost, duration and quality of particular detailed changes in the organization structure of a project team, e.g., decentralize certain decision approvals or formalize communication with more regularly scheduled meetings?

Normally, project managers rely on their experience and intuition to provide answers to these kinds of questions.

1.2 How Procurement Policy Affects Coordination Load—A Case Study

This brief case study illustrates the way a VDT model can provide theory and tools to predict the impacts of specific organizational changes on task performance, given schedule pressures, organization structure and project policies.

An aerospace company had done carefully regulated design and manufacture of a military system. The company now is adapting the design for commercial use. Simultaneously, it is changing its processes to become "agile," with specific objectives to outsource the manufacturing of significant subsystems and substantially decentralize engineering decision-making. The purposes of our modeling and analysis were, first, to predict the effects of different levels of decentralization on product delivery time, cost and quality; and second, to predict the effects of different levels of design engineering support on performance of new manufacturing subcontractors. The symbolic VDT model represents the structure and capabilities of organizational entities and also the activities in the engineering design process. For a simple, illustrative case example, Figure 1 shows that the VDT model links the organization chart and the activity diagram of modeled projects.

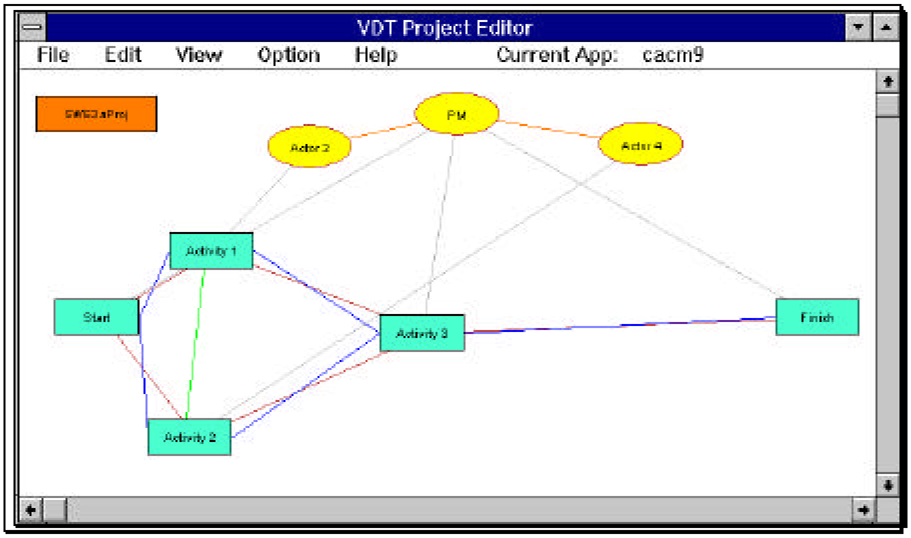

Figure 1: The VDT model links the organization chart (ellipses) and the activity diagram (rectangles) of a project. Relationships shown by lines include `Reports-to` among actors, `Responsible-for` among actors and activities, and `Successors`, `Reciprocal-Information-Dependence` and `Failure-dependence` among activities.

The model predicted that one design subteam would develop a high coordination "load," i.e., need for coordination, to support a new vendor. The predicted effect was that the total time and effort to complete the particular activity would both be significantly greater than estimated. In addition, since the at-risk activity was on the critical path of the project, the model predicted that the project would exceed its budgeted cost and duration. Several months after these predictions were discussed with managers, the project encountered the predicted cost and schedule overruns. Using what-if studies, the model also predicted that the activity and project duration impacts could have been managed with use of additional staff with appropriate skills. The what-if studies also predicted significant change in project performance given change in centralization of decision-making. Finally, the organization model predicted effects on project performance of additional issues such as degree of formalization of communications among organizational participants and structure of the engineering process.

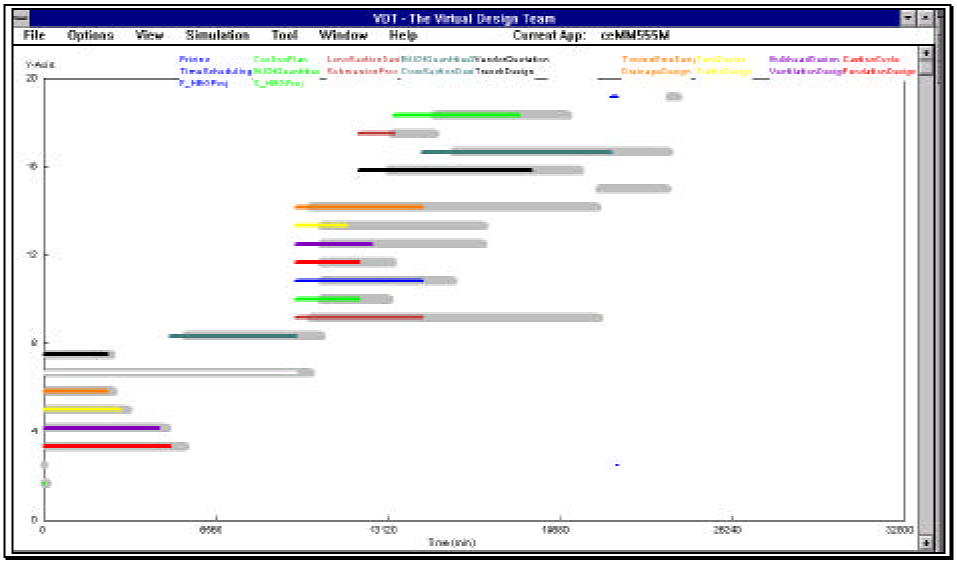

Figure 2: The VDT simulation produces a Gantt chart for an example project. Solid lines show the traditional Critical Path Method time projection. The broader gray lines show the more realistic VDT prediction considering both planned direct work and additional predicted rework and coordination among actors.

2. The VDT Micro Theory of Project Organizations

Organizational Engineering is the process of configuring an organization structure to accomplish a given high-level task while attempting to satisfy stated performance objectives. An organization includes human actors supported by information processing and communication tools.

The basic premise of the VDT model is that organizations are fundamentally information-processing structures. This view of organizations dates back to Weber's work in the early 1900s, and is elaborated in (March and Simon 1958), (Simon 1976), (Galbraith 1977). In this view, an organization is an information-processing and communication system, structured to achieve a specific set of tasks, and composed of limited teams (called "actors") that process information. Actors send and receive messages along specific lines of communication (e.g., formal lines of authority) via communication tools with limited capacity (e.g., memos, voice mail, meetings, etc.). Thus, for example, each modeled manager has specific and limited (boundedly rational) information processing abilities. Managers send and receive messages to and from other actors along pre-specified communication channels, choosing from a limited set of communication tools. In the organizational literature, coordination load is the complex set of requirements for coordination among the various actors in an organization. It is usually reduced to a single, ordinal measure of the level of interdependence among actors in the organization: *High*, *Medium* or *Low*. The VDT simulation system, in contrast, infers the coordination load, as discussed in Section 3.2.

Jay Galbraith's (1977) information processing view of organizations provides a foundation for modeling the information processing patterns of an organization and, by simulation, for determining overall information processing capacity of an organization. Galbraith views organizations as limited in their ability to process "exceptions"—requests for advice or direction when local knowledge or authority is insufficient to deal with the information processing requirements posed by an actor's activity. The organization's information processing capacity, in this view, is limited both by the bounded rationality (Simon 1976) of the actors or "nodes" in an organization and by the limited information carrying capacity of the information "channels" that connect actors.

Burton and Obel's (1984) simple but elegant model of organizations was more of a macro contingency theory model than VDT, but it provided important theoretical insights and continues to inspire us to simplify future versions of VDT. Masuch and Lapotin's (1989) AAISS system demonstrated the use of non-numerical computing paradigms derived from artificial intelligence to model organizational decision making in clerical tasks and to predict the impact of various aspects of structure on performance. Carley and her colleagues (1992) have extended the model of actors in AAISS to include learning and communication between actors. Computational organizational modeling also has a parallel in the work of several computer scientists (Gasser 1991 and Shoham 1993).

3. VDT Implementation

Figure 3 shows the inputs and outputs of the VDT simulation model. For a particular set of analyses, a set of organizational attributes are held fixed as control variables, and a small set of variables are varied as independent variables in the simulation.

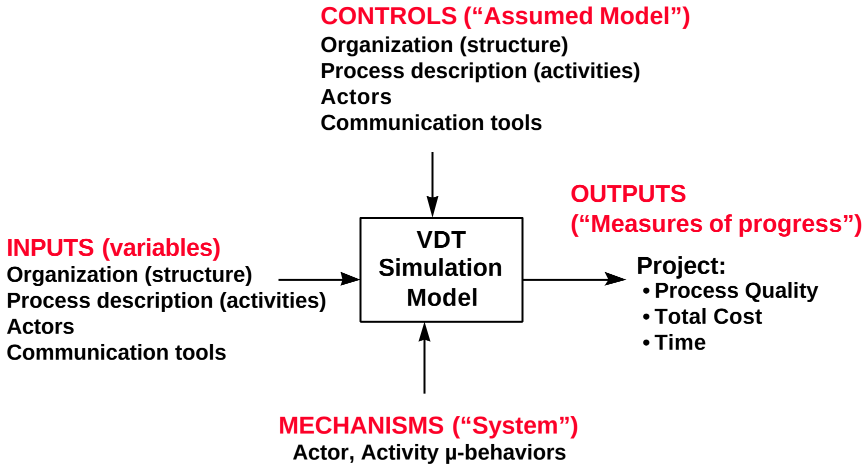

Figure 3. VDT Model Architecture. Given values for independent input variables that describe a project and a set of fixed assumptions, the VDT model simulates each activity being performed by responsible actors and computes overall project duration, cost and coordination quality. The microbehaviors consider both planned direct work and inferred requirement for coordination and rework.

The VDT model uses editable decision tables ("behavior matrices") to transform qualitative attribute values to quantitative values that are used in the discrete event simulation. For example, the simulation changes the Monte Carlo simulation probabilities for who makes a decision about whether to perform rework when a failure is detected, depending on the qualitative (e.g., "high," "medium," "low") degree of centralization of decision-making responsibility: With low centralization, the decision about whether to perform rework when an error is detected tends to be made by the actors themselves or their immediate subteam leaders; with high centralization, the decision is more likely to be referred to higher level subteam leaders or the project manager. A second decision table then determines the probabilities that managers at each level decided to rework, correct (quick fix) or ignore the error. The default decision tables make senior managers more conservative in carrying out rework, i.e., project managers order rework to be done more often than a subteam leader or subteam actor would. If the decision maker does not attend to the exception message and make a decision within a specified timeout period, the actor that initiated the original exception message delegates the decision to itself by default. Through this interplay between symbolic inference and numerical Monte Carlo simulation, VDT's emergent predictions replicate Galbraith's predicted tradeoff between speed of decision making and decision quality associated with greater centralization: As long as higher level managers have time to deal with exceptions, better quality decisions are the result; however, if they should become backlogged, high centralization can lead to excessive delays, and eventually to "delegation by default" as lower level managers or engineers take matters into their own hands and make their own, less conservative decisions.

To generate specific predictions about information processing capacity versus load at the level of individual actors or subunits, we operationalized and extended Galbraith's framework in our VDT micro theory of organizations. The remainder of this section summarizes how we implemented the VDT micro theory in the VDT model.

3.1. Activities

VDT shifts the object of analysis from an aggregated organization with one high level task to multiple individual actors and their assigned activities. The activity representation abstracts away the technical content of activities. Activities simply consume time and may (or may not) generate communications and exceptions. However, activities have attributes that the simulator considers to check the match among activities and actors, to generate coordination processes, and to derive overall task efficiency and effectiveness based on actor and activity performance. Based on the closeness of the match between the *complexity* of an activity and the *capability* of its responsible actor, the VDT model assigns an *actor processing speed* and a *verification failure probability,* i.e., the probability that each subtask comprising the activity "fails" in the verification that occurs as it is completed. Actors' responses to subtask failures depend on organization structure and policies; actors' responses to subtask failures affect the quality of the work process.

The VDT activity model represents (parentheses show type of attribute values):

- Duration (nominal time);
- Failure Dependence (list of activities)
- Requirement complexity (low, medium, high);
- Required skill (e.g., financial accounting, structural steel design);
- Solution complexity (low, medium, high);

- Subtask size (time to do one subtask within the activity, where activities are assumed to decompose into equal sized subtasks, and a subtask is the minimum amount of work that can be determined to have "failed.");
- Successors/Predecessors (list of activities)
- Uncertainty (low, medium, high);
- Work volume (time for an actor with "medium" level of "required skill" to perform the activity, assuming no rework).

Unlike conventional Critical Path Method (CPM) activity models, VDT's activity model explicitly represents the coordination among specialists assigned to interdependent activities. Thus, in addition to sequential dependency relationships among activities, VDT models activity coordination requirements in terms of *verification failure probability* arising from activity complexity, and *information exchange communication intensity* arising from activity uncertainty and interdependence. The former determines the probability that a subtask will fail when the simulator verifies the work of a responsible actor at the end of each subtask. Subtask failure and leads directly to communications about failures with the actor's supervisors; and a decision to perform rework on the failed subtask propagates rework to "failure dependent" activities. The communication intensity defines how frequently the actor responsible for an activity needs to communicate with the actors responsible for functionally interdependent activities to coordinate decision making. During the simulation, coordination activity, i.e., exception processing and communication among interdependent actors, and rework on failed and dependent subtasks, emerge as a result of direct work by actors with interdependent activities and assigned organizational roles.

3.2. Communications

Coordination requires information flow among actors in a project team. A *communication* represents a "packet" of information that is generated and sent by one actor, and received and processed by another. In the VDT model, communications may be: *work communications* or *coordination communications*. The latter are further subdivided into *information exchange communications, failure exception communications,* and *decision communications.*

The simulator dynamically creates "work item" communications that inform each responsible actor when an activity subtask is ready to be worked on by that actor. Using attention rules explained in Section 3.3, actors select subtasks or exceptions from their in-trays to process. Upon selecting an activity to process, the actor stochastically initiates information exchange communications to other actors based on the communication intensity and the reciprocal interdependence relationships of the activity on which it is currently working. An information exchange can be a request for coordination or a "for your information" message.

When a subtask completes—typically at the end of a work day—the simulator stochastically determines whether it has failed. The simulator generates failure exceptions when actors encounter failures in their subtask verification. As detailed in Section 3.4, generation of a failure exception initiates an exception-decision process. The simulator generates a decision communication after a manager makes a decision to rework or ignore the exception.

3.3. Actors and Information Processing

Because of its aggregated view of organizational information processing, the Galbraith framework says very little about how actors' attributes influence their information processing behavior. We model project teams as comprising a set of "actors" that can be either individual managers and engineers, or small, encapsulated subteams with undifferentiated members. Actors in a team are the entities that perform work and process information. By disaggregating organizations into actors and explicitly modeling their behavior, VDT generates emergent organizational behavior and performance resulting from the actions of, and the interactions among, individual actors.

VDT models actors in terms of their *capability, attention rules, action* and *organizational role*. Actor behavior has limited information processing capability and attention rules that select one communication at a time. These properties give actors behavior that is boundedly rational (Simon 1976).

The VDT actor model represents:

- Actor size (number of people, >= 1);
- Actor skills and skill level for each skill (high, medium, low);
- Responsibilities (Activities)
- Role in the organization (e.g., subteam, subteam leader, project manager);
- Task experience (high, medium, low);

The actors live in an overall organization that is dedicated to performing a particular project. The VDT project model represents:

- Centralization of decision-making responsibility (high, medium, low);
- Formalization of communications in memos, organized meetings (high, medium, low);
- How frequently the actor responsible for an activity needs to communicate with the actors responsible for functionally interdependent activities;
- Experience of the entire project team in working together before (high, medium, low);
- Probability that each subtask comprising the activity "fails" in the verification that occurs as it is completed and the failure causes internal rework (%) or rework in other activities (%);
- Likelihood that communication is non-activity related "noise." Processing noise consumes actors' time without contributing to activity performance.

VDT input includes the direct work requirement for an actor doing each activity. Coordination requirements for the responsible actors are inferred from these actor and activity attributes.

Actors in VDT have several kinds of behavior. Actors:

Allocate attention. Activity subtasks and communications accumulate in the in-tray of an actor to await processing. The actor's attention rules determine whether to interrupt an ongoing activity when a new communication enters the in-tray, and they select a new communication to process from the in-tray when a subtask or exception completes. VDT actor attention rules consider factors such as current activity priority, incoming communication priority, and the order in which communications enter the in-tray. Attention rules give actors boundedly rational attention allocation behavior. VDT's default actor attention rules select the highest priority item 50 percent of the time; they use LIFO and FIFO each 20 percent of the time; and they randomly select a communication from the in-tray ten percent of the time.

Process information. After an actor selects an activity or coordination item from the in-tray, VDT calculates the time required to process it based on the actor's processing speed (derived from the degree of the match between the attributes of an actor and the communication) and the work volume of the communication. During the time that an actor is processing a work subtask (typically about one day in duration), an incoming communication may arrive from another actor at each simulation event, as little as one minute apart). Whenever this occurs, the actor applies its attention rules and stochastically chooses whether to stop processing the current subtask to attend to the exception or communication.

Send communications to other actors. Actors use communications to coordinate with each other. VDT extends Galbraith's (1977) notion of communication channels by modeling them as relationships among actors, each supported by communication tools whose functional attributes affect the timing and quality of information transfer across that channel. Actors in VDT communicate with each other by sending informal communication items or by attending scheduled, formal meetings. To send a communication to another actor, an actor must select a communication tool. Actors use several criteria for choosing a tool, including actor preference, message priority, primary natural idiom in message, proximity of sender to recipient, and cost.

Generate and handle exceptions. VDT actors generate, communicate and process several kinds of exceptions (See next section.)

3.4. Exceptions and Decision-Making

Actor information processing and exception handling form the kernel of the VDT micro theory framework. Since we abstract much of the content of the design task, information processing related to direct design work merely consumes time of VDT actors. Processing exceptions, in contrast, requires VDT actors to route exceptions to authorized actors, who then make decisions about how to handle them.

Subtask failure is one kind of exception. A subtask is the smallest portion of an activity that can be evaluated—typically a day's work for an actor. Each subtask is verified when completed. This requirement is realistic for many kinds of engineering work, especially for design of highly regulated facilities such as power plants or offshore oil platforms.

When the verification process evaluates a subtask as having "failed," the simulator generates an exception for the responsible actor. The responsible actor must decide (stochastically) with whom to communicate to resolve the subtask failure exception, based on the level of centralization of the organization. The actor then sends the exception to the authorized decision maker for resolution. When and if the decision maker's attention rules select the failure exception for processing, the decision maker decides whether to ask the responsible actor to rework the failed subtask or proceed without doing rework. The actor's rework rules, which vary for actors with different roles, determine probabilitistically what the rework decision will be.

Requests for information represent a second kind of "exception." VDT models two different types of requests for coordination: *informal information exchange* and *formal, scheduled meetings*.

Depending on the level of uncertainty of a given activity, its responsible actor will initiate informal information exchange requests to obtain needed information more or less frequently with actors performing interdependent activities. The project manager schedules formal meetings. VDT generates meeting requests and sends communications to participating actors to request their attendance.

When an actor generates a failure or an information exception, it suspends work on the current subtask until the exception is resolved: like humans, it waits. While waiting, it can handle communications or other activities, but it suspends work on its chosen subtask until the exception is resolved by information or a decision from another actor or by default. Figure 2 shows activities that have significant coordination delays as actors await exception resolution.

3.5. Programming Implementation

The VDT system was implemented as an object-oriented, discrete event simulation. Using standard AI techniques, the model uses inheritance and the behavior methods using symbolic pattern matching. The VDT discrete event simulation of stochastic behavior uses Monte Carlo simulation. For example, actors stochastically choose items to attend to from their in-trays and decide whether or not to communicate with interdependent actors upon completion of each subtask. The level of the hierarchy to which a request for a rework decision is sent, and the outcome of the rework decision are also determined stochastically by Monte Carlo simulation.

Version 2 of VDT (Christiansen 1993) uses Kappa®, a C-based object oriented programming environment developed by IntelliCorp. The model was developed and the simulations were run on Sun Workstations and PC's. A single run of VDT for a large project (50 activities, 20 actors, one year project duration, one day typical subtask size) generates about a million simulation events and takes about 15 minutes on a Sun SparcStation IPX or a medium-class Pentium.

The commercial implementation of VDT by Vité™ is implemented in C++ to run on Windows and executes suites of simulations for typical projects at the rate of about one run per second.

4. Discussion

As discussed in Section 1.2, we used VDT to model an aerospace project prospectively. Unlike the optimistic predictions of project mangers and CPM models, the VDT simulation predicted the presence and major significance of bottlenecks in the organization and the process. We have also modeled more than 30 engineering design projects retrospectively. Our experimental results show qualitative consistency among the predictions of theory, experienced project managers, and simulations. We claim that, for the types of complex but relatively routine projects that we have modeled, VDT produces aggregate performance predictions that are qualitatively reasonable. Experienced project managers consistently find them both interesting and surprising.

After identifying activities and actors associated with predicted bottlenecks, a VDT user can propose decentralization of decision making, reassignment or change in the number or skills of workers on a subteam, better communication tools, or other changes in the structure of the team's organization. The user can model each proposed change in VDT and run simulations to predict changes in the VDT efficiency and effectiveness performance measures.

The VDT system is an early example of building symbolic models of social sciences theory. The theory is inherently qualitative, but symbolic models now allow computational representation and manipulation of qualitative conceptual entities, their attributes, relationships and behaviors. The computational implementation of theory is much more precise as a computational model than theory in classical text form. In addition, the computational symbolic model is executable and therefore inherently repeatable and testable. The symbolic model allows precise definition of important conceptual entities and the precise, testable specification of their functions, structure and behaviors.

Experience to date with VDT in the aerospace, construction and semiconductor domains has demonstrated that the VDT modeling approach and analysis tool can provide managers of fast-paced product development projects with valuable new kinds of insights about the tradeoffs between product performance objectives, development schedules (especially the degree of concurrency of activities), actor qualifications, and team organization and policies. As such it represents a step on the path toward our long range goal of developing analysis tools to enable true "organizational engineering': designing work processes and organizations like we currently design bridges, automobiles and airplanes—by testing and refining "virtual prototypes" of design alternatives in a computer, rather than by trial and error experimentation with full-scale prototypes.

References

Burton, R. M., and Obel, B., *Designing Efficient Organizations: Modeling and Experimentation*, North-Holland, New York, 1984.

Carley, K., Kjaer-Hansen J.,, Newell, A., and Prietula, M., "Plural-Soar: A Prolegomenon to Artificial Agents and Organizational Behavior, " In M. Masuch and M. Warglien (Eds.) *Artificial Intelligence in Organization and Management Theory* North-Holland, Amsterdam, 1992, pp.87-118.

Christainsen, R. T., *Modeling Efficiency and Effectiveness of Coordination in Engineering Design Teams.* Unpublished Ph.D. Thesis, Stanford University, September 1993.

Davis, S M., and Lawrence, P. R., *Matrix*, Addison-Wesley, Reading, Massachusetts; 1977.

Galbraith, J.R., *Organization Design*, Addison-Wesley, Reading, Massachusetts; 1977.

Gasser, L. and Huhns, M. N., (Eds.) *Distributed Artificial Intelligence II*, Pitman, London; 1989.

Levitt, R. E., Cohen, G. P., Kunz, J. C., Nass, C. I., Christiansen, T. R., and Jin, Y., 'The Virtual Design Team:' Simulating How Organization Structure and Information Processing Tools Affect Team Performance", in *Computational Organization Theory*, edited by K. M. Carley and M. J. Prietula, Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.

March, J. G., and Simon, H. A., *Organizations*, John Wiley, New York; 1958.

Masuch, M. and LaPotin, P., "Beyond Garbage Cans: An AI Model of Organizational Choice," *Administrative Science Quarterly*, 34, (1989), 38-67.

Shoham, Y., "Agent Oriented Programming," *Artificial Intelligence* 60 (1993), 51-92.

Simon, H. A., *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization*, Free Press, New York; 1976.

Thompson, J. D., *Organizations in Action: Social Science Bases in Administrative Theory*, McGraw-Hill, New York; 1967.