Computational & Mathematical Organization Theory 2:3 (1996): 171–196

The Virtual Design Team: A Computational Model of Project Organizations

YAN JIN
Department of Mechanical Engineering and the IMPACT Laboratory, University of Southern California, DRB101, Los Angeles, CA 980089-1111, jin@usc.edu

RAYMOND E. LEVITT
Department of Civil Engineering, Stanford University, Stanford, CA 94305-4020, levitt@ce.stanford.edu

Abstract

Large scale and multidisciplinary engineering projects (e.g., design of a hospital building) are often complex. They usually involve many interdependent activities and require intensive coordination among actors (i.e., designers) to deal with activity interdependencies. To make such projects more effective and efficient, one needs to understand how coordination requirements are generated and what coordination mechanisms should be applied for given project situations. Our research on the Virtual Design Team (VDT) attempts to develop a computational model of project organizations to analyze how activity interdependencies raise coordination needs and how organization design and communication tools change team coordination capacity and project performance. The VDT model is built based on contingency theory (Galbraith, 1977) and our observations about collaborative and multidisciplinary work in large, complex projects. VDT explicitly models actors, activities, communication tools and organizations. Based on our extended information-processing view of organizations, VDT simulates the actions of, and interactions among actors as processes of attention allocation, capacity allocation, and communication. VDT evaluates organization performance by measuring emergent project duration, direct cost, and coordination quality. The VDT model has been tested internally, and evaluated externally through case-studies. We found three way qualitative consistency among predictions of the simulation model, of organization theory, and of experienced project managers. In this paper, we present the VDT model in detail and discuss some general issues involved in computational organization modeling, including level of abstraction of tasks and actors' reasoning, and model validation.

Keywords: organization design, simulation, organization modeling, organizational analysis tools

1. Introduction

In the early 1990s, the Statfjord Sub-sea Satellites Project was undertaken to produce oil from deep ocean wells in the Norwegian sector of the North Sea. The goal of this project was to design, manufacture and place unmanned sub-sea oil production modules on the ocean floor. Since they would be expensive to access once placed, the Statfjord modules were designed to very high quality standards to ensure that they would operate reliably, maintenance-free, for extended periods. After this project started, its work plan was changed to reduce its development schedule from three years to two years. To fit this new schedule, the design phase of this project had to be reduced from 22 months to 15 months. As a result, many sequential activities in the original plan had to be carried out concurrently.

Several questions arose from the schedule change to which the Statfjord project manager needed answers:

- Could the original design team complete the design within 15 months, instead of 22 months? If not, which specific design disciplines or management groups should be augmented?
- What specific changes, if any, could the manager usefully make in the organization structure of the 25-person design team, e.g., decentralization of certain design approvals or decisions?
- If decision-making authority were decentralized to save time, what would be the impact on design cost and quality?
- Would project time be saved by investing in advanced communication technologies (e.g., CAD file sharing or video conferencing)?

The Statfjord project managers could only answer these questions intuitively, relying on their experience, because no extant technology and/or theory could provide explicit answers. While the Critical Path Method (CPM) models sequential interdependencies through explicit representation of precedence relationships between activities, it does not take into account reciprocal information requirements between concurrent activities, nor the impacts of actor interactions. At the same time, contingency theory (Galbraith, 1977) can provide only limited answers to these questions because of its aggregated view of organizations and its relatively general definitions of contingency factors.

Our research on the Virtual Design Team (VDT) attempts to develop a computational organization model, called VDT, to answer the questions. The VDT research was initiated in the late 1980s with a long term goal to develop new theory and tools that could extend the reach of both contingency theory (Galbraith, 1977) and network-based management tools like CPM, and to provide answers to these kinds of questions for project organizations engaged in complex, but relatively routine tasks. VDT explicitly represents organizations' tasks (e.g., the design tasks in the above example), their actors (i.e., the particular designers and managers in the above example), and organizational structures. For a given task and organizational setting, VDT can generate emergent organizational performance through simulation of micro-level actions of, and interactions among, the actors in the organization.

Our initial VDT model was developed based on two observations about collaborative, multidisciplinary work in large, complex projects. First, organizational tasks in project organizations can be divided into two parts: the primary *production work* that directly adds value to final products, and *coordination work* that facilitates the production work. For a given project, the amount of production work is usually determined based on the specifications of the product to be produced. Therefore, the variation of production work as a function of organization design is relatively low. The nature and amount of required coordination work, however, may vary considerably, depending on how the project team is organized: level of centralization and formalization, decision-making policy, task assignment, available communication tools, actors' team experience, etc. A model of how coordination work is generated and dealt with by team actors should thus be useful for researchers to understand organizational behavior of project teams and for project managers to analyze their organization's performance for better team design.

Second, although the extant contingency theory provides qualitative insights about the extent of coordination work given aggregated project parameters (Galbraith, 1977; Thompson, 1967), it does not say anything about which specific activities and actors are the bottleneck for coordination, and what specific steps can be taken to resolve coordination overload problems. We need an elaborated version of contingency theory with contingency factors set at more specific levels.

Advances in computer modeling technology, such as object-oriented programming and model-based reasoning techniques, have made it possible to address human coordination issues through a computational approach by explicitly representing tasks, actors' behavior, and coordination actions. Creating an effective conceptual model that can take maximum advantage of state-of-the-art computer technology is a research challenge with a high potential payoff.

In the course of developing the VDT model, we encountered a number of general organizational modeling questions including:

- What is the appropriate level of abstraction that can capture reality at a sufficient level of detail and, at the same time, avoid becoming too complex or too "realistic" to comprehend (Burton and Obel, 1995a)?
- To what extent should organizational tasks be explicitly represented so that actors' action, communication, and skill can be captured properly (Carley and Prietula, 1994)?
- How can we validate the computational organizational model? If the model is relatively abstract, can we find ways to link the representation of, and predictions from the abstract model to the real project data so that the model can be comparable with real projects?

The VDT model combines the CPM project modeling approach with organization theory to address the issue of abstraction. The validity of the model is tested by comparing simulation results with theoretical predictions and historical data from real engineering projects. In the following sections, we present our extended information-processing view of organizations and introduce the top-level concepts of the VDT model. Sections 3 and 4 describe how VDT models organizational tasks and organizational actors, respectively, to make coordination work explicit and measurable. Section 5 describes how organization structures are defined and used as a set of variables for organizational analysis. Section 6 presents an overview of VDT system architecture. Finally, in Section 7 we discuss the general organizational modeling issues mentioned above in the context of model validation, related work, and our future work.

2. An Extended Information-Processing View of Design Teams

Organizations, including project organizations, need information flows to function, and strive to create efficient information flows to be effective. An organization processes information to coordinate and control its activities. Since Weber's fundamental work in the early 1900s (Weber, 1924), many organization theorists have adopted an information-processing view of organizations (March and Simon, 1958; Galbraith, 1977). In this view, an organization is an information-processing and communication system, structured to achieve a specific set of tasks, and comprised of limited capacity, "boundedly rational"

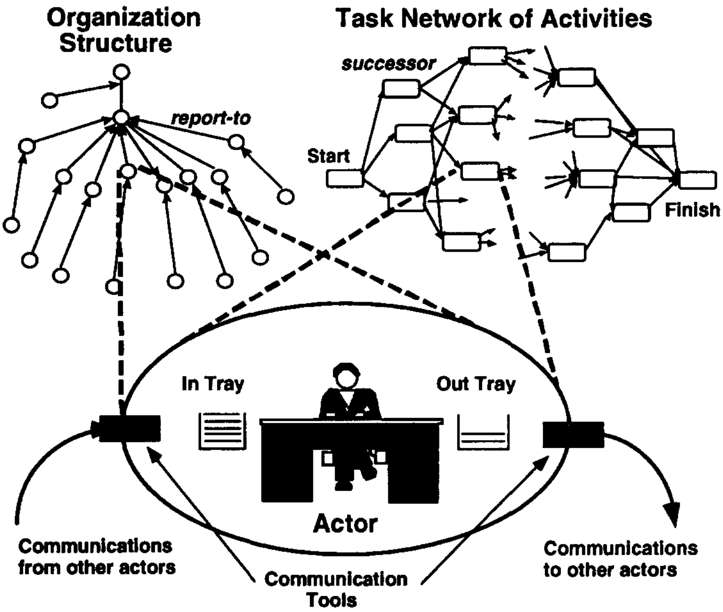

Figure 1. An overview of the VDT model.

information processors (individuals or sub-teams). These information processors 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).

This information-processing view of organizations provides a foundation for our VDT model. In VDT, the information-processing view has two implications. The first is we can model design teams as information-processing structures that are composed of *tasks* generating information to be processed, *actors* processing and communicating information, *communication tools* linking actors for communication, and an *organization structure* that constrains actors' information-processing and communication behavior. Figure 1 shows an overview of this information-processing structure in which tasks, actors, communication tools, and organization structure are the key conceptual components.

The second implication is that for a given design project team, both primary production work (i.e., design) and coordination work (i.e., communication and decision-making carried out to facilitate design) can be viewed as information-processing. We can, therefore, model the amount of information processing work in terms of *work volume*.[1] This uniform way to represent the contents of organizational tasks provides a strong means of abstraction. For a given project, let the total work volume of the project be TW, production work volume

[1]In VDT, we use *work volume* to represent the amount of information-processing work. Work volume is an attribute of a piece of work (e.g., an activity, a work item, a communication item) and is associated with required skill set. Work volume is expressed in units of time and represents the nominal time taken by one person with a medium level of the needed skill set to complete the work.

PW, and coordination work volume CW. We assume that

$$TW = PW + CW \tag{1}$$

Furthermore, PW can be divided into two parts: originally planned production work PW_o and production rework PW_r arising from the failure of original production work.

$$PW = PW_o + PW_r \tag{2}$$

From (1) and (2) we have:

$$TW = PW_o + PW_r + CW \tag{3}$$

For a given project task, PW_o is given, and $PW_r + CW$ may vary depending on the characteristics of the task and the effectiveness of the organization (i.e., project team) working on the task. The ratio

$$R_c = (PW_r + CW)/TW \tag{4}$$

provides a rough measurement of coordination load relative to originally planned production work load, and is a function of both task complexity and organization capacity. If a task is "perfectly simple"—i.e., there is almost no associated coordination requirement; or if the design team working on the task is composed of "perfect" designers and managers organized in a "perfect" way—i.e., with high skills relative to task complexity (Galbraith, 1977), the value of R_c can be close to 0. At the other extreme, the value of R_c can be close to 1, meaning that the project will never finish due to endless rework and coordination.

Between the two extremes, we believe, there exists a range in which the variation of R_c can be at least partially controlled by adjusting certain organization design variables. The question here is "Can we create a model that can estimate PW_r and CW at a sufficient level of detail so that we can use the model to analyze the performance of different organization designs to achieve the 'best' efficiency or minimum R_c?"

Our experience with VDT has shown that for routine design projects, the answer is yes. For routine design projects, the project tasks can be pre-specified as a precedence network and activity interdependencies are relatively well understood. Furthermore, actors involved in a routine design project are highly institutionalized such that their behavior is more professional than social, and thus relatively easy to model. In VDT, we have taken a Monte Carlo simulation approach to predict PW_r and CW. VDT simulation takes PW_o, other task variables (described in Section 3), and organization settings (described in Sections 4 and 5) as input, and produces emergent PW_r and CW through simulation. Before a simulation starts, each actor in VDT is assigned a position in the team organization and one or more project activities (production work) to work on, as shown in Figure 1. During simulation, an actor processes incoming information items through his/her in-tray and sends outgoing information items to others through his/her out-tray via selected communication tools. The incoming items include production work, information, and decisions received from others, whereas the outgoing items include requests for information, answers to requests, exception reports, and decisions. Besides production work, actors in VDT spend time on information exchange, exception-reporting, and decision-making. Furthermore, an actor may have to wait for a decision after reporting an exception, if his/her supervisor is too busy to work on the exception report. Based on this information-processing model, the coordination work volume CW in (2) and (3) can be divided into three parts: CW_{cm} for information exchange communication work volume, CW_{ct} for decision-making work volume, and CW_{wt} for waiting time. So we have

$$TW = PW_o + PW_r + CW_{cm} + CW_{ct} + CW_{wt}$$

The following sections describe our models of organizational tasks, actor actions, and organization structures, and show how the VDT simulation generates emergent PW_r, CW_{cm}, CW_{ct} and CW_{wt} based on given organizational tasks and project team designs.

3. Modeling Organizational Tasks

Project organizations are task-driven. They have specific tasks (e.g., to design a hospital building) that must be finished by a certain time (e.g., the end of 1996) and cannot cost more than a budgetary limit (e.g., $50 million). Usually, the top-level organizational task needs to be divided into smaller sub-tasks, called *activities* in this paper, so that they can be carried out by individual actors or small groups of actors. Activities represent primary production work (i.e., design work for a design team). As an activity is carried out by its responsible actor, coordination work may occur, depending on both the work content and the type of dependency between this activity and related activities. Although project managers seek to define activities that are independent from each other, the division of tasks almost invariably creates dependencies among the activities and thus generates a need for coordination.

There are two basic requirements for a VDT task model. First, the model must capture enough details of both work contents and activity dependencies so that both production work (PW) and coordination work (CW) can be generated. The challenge here is how to make the model simple, but effective, across many specific types of design projects. The second requirement is to be able to map the model attributes to accessible, real project data, so that the model is comparable with real project information and that the insights generated from the model are realistic. The research issue associated with this requirement is "Can we define a methodology to link real project information to the VDT task model?"

3.1 Activity Dependencies

In the organization literature, task dependencies have been considered as an important environmental measurement of uncertainty (Lawrence and Lorsch, 1967; Galbraith, 1977). Although this aggregated account of task dependency may be used to show how uncertain an overall organizational task is, it does not provide insights into specific dependency relationships between particular activities and their impact on organizational performance, nor into what coordination mechanism should be employed to manage a particular dependency.

In VDT, several kinds of dependencies among activities are explicitly represented and treated as the sources of coordination work. Following Thompson (1967), VDT models pooled, sequential, and reciprocal dependency relationships among activities.

Pooled dependency: Since we model project organizations, each activity is part of the overall project and is thus always in a pooled relation with other activities. Following Thompson, rules and standards, e.g. about how to deal with exceptions, are used to coordinate this kind of interdependency.

Sequential dependency: VDT adopts the *successor* relationship used in CPM (Critical Path Method) networks to represent sequential dependency between activities. An activity $Actv_B$ is a finish-to-start successor of $Actv_A$ if $Actv_B$ can start only after $Actv_A$ is completed. If $Actv_B$ can start some length of time after $Actv_A$ is started then $Actv_B$ is a start-to-start successor of $Actv_A$, etc.

Reciprocal dependency: VDT's task model captures two types of reciprocal dependencies. One is information related, and the other is work related. An *information related* reciprocal relation represents a mutual information requirement dependency between two activities. For example, the mechanical design and structural design activities of a building design project may be carried out in parallel. The structural designer needs spatial and weight information about mechanical equipment from the mechanical designer; and the mechanical designer may need to know the size and location of structural members to plan where mechanical equipment can be located. *Work related* reciprocal relation describes whether an exception (e.g., design change, error detected) generated within one activity will have any impact on the work of another. For the above example, if a design change is made in the mechanical design, then the structural designer may have to choose a different beam size; similarly, if the structural design is changed, then the mechanical design may have to be reconsidered because equipment sizes and/or locations may need to be changed. The VDT coordination load modeling methodology captures these reciprocal relationships through a series of manual analyses of the requirements and solutions of each activity (Christiansen, 1993).

3.2 Production and Coordination Processes

The activity dependency relationships described above explicitly represent the potential need for coordination work but do not define *when* and *how much* coordination work is needed. In VDT, the amount and the content of production work are defined explicitly as attributes of activities. Coordination work is implicit, and generated stochastically by VDT based on activity complexity, uncertainty, and task-actor skill match.

It has been pointed out that the level of abstraction of an organization model is determined by the modeling purpose (Burton and Obel, 1995a). Our purpose for modeling is to predict emergent coordination work volume (CW) and rework volume (PW_r) as dependent variables of both task situation and organization design. To achieve this goal, our process model is centered around describing *how much time is needed for a given project orga-*

nization to finish a specified task rather than explicitly treating design as a knowledge-based, problem-solving process. From the information-processing view of organizations described above, we assume that

- An activity representing production work has a preset *skill requirement* and amount of work described by work volume (see footnote 1).
- While processing production work of an activity, an actor will probabilistically need to communicate with relevant actors to get required information. The frequency of required communication depends on the *reciprocal dependency* with other activities and the activity's level of *uncertainty*. Communications may take place via informal information exchange between two actors, or in formally scheduled meetings among two or more actors.
- While processing production work of an activity, a small portion of the activity (typically one day's work), called a *work item*, may fail stochastically. The failure probability of a work item, called *verification failure probability* (VFP) depends on the *complexity* of the activity and the match between the activity's *skill requirement* and its responsible actor's skill level. This work failure will trigger a process of exception-report and decision-making. Failed work items need rework to maintain *production quality* described below.

An activity in VDT is defined by its *work volume, skill requirement, complexity* and *uncertainty*, and by its relationships to other activities. These attributes not only explicitly define the production work but also implicitly define the derived coordination work needed to facilitate the production work. Moreover, depending on how the project team is organized and how tasks are assigned to actors, the required volume and locus of coordination work (e.g., exceptions and decisions) will be different. Consequently the time needed to carry out the coordination work may vary.

While task complexity and uncertainty are treated in the organization literature as variables describing the task environment faced by an organization, complexity and uncertainty in VDT are associated with activities and affect the volume of both production work and coordination work. This change in focus from an abstract, overall task to specific activities allows us to analyze the lower level contingency factors (e.g., making two sequential activities parallel).

3.3 Process Efficiency and Quality

A VDT simulation produces several outputs, including the amount of production work PW, coordination work CW, and thus the combination of the two, total work TW. Since TW represents man-hours needed to finish the project, the smaller the TW, the more efficient the project. In VDT, we measure the project direct cost efficiency E_c and time efficiency E_t

$$E_c = PW_o/TW; \tag{5}$$

$$E_t = PD/SD \tag{6}$$

where PD is planned project duration and SD simulated project duration. For a given project, the bigger the values of E_c and E_t are, the more efficient the project is. From equations (1), (5) and (6), it is obvious that excessive coordination (i.e., CW_{cm} and CW_{ct}) and waiting (i.e., CW_{wt}) will decrease the project efficiency.

For an organization design A and its redesign B, the differences

$$\Delta E_c = E_{cB} - E_{cA} = PW_o * (TW_A - TW_B)/TW_A * TW_B \quad \text{and}$$
$$\Delta E_t = E_{tB} - E_{tA} = PD * (SD_A - SD_B)/SD_A * SD_B$$

represent the impact of the organization redesign on the project efficiency.

Besides efficiency, VDT also measures process quality. Since VDT does not model the engineering content of products, it cannot judge the quality of the final products. Instead, we measure process quality or effectiveness in terms of how well task failures and coordination requests are dealt with by actors.

When a task fails, the organization may or may not detect the failure. If the failure is detected, the organization can respond in ways ranging from completely reworking the failed activity and all related activities, to ignoring the failure and proceeding directly with related concurrent tasks and future tasks. We take the position that detection of task failure is not in itself an indicator of poor quality; rather it is the organization's response to the detected failures that determines the *verification quality* Q_v of its work processes. We view the proportion of detected failures that get reworked as a measure of the quality of an organization's work processes. Let PW_f denote the total detected failed production work volume. Then the verification quality can be expressed as

$$Q_v = PW_r/PW_f \tag{7}$$

Another aspect of process quality is the extent to which requests for coordination among interdependent actors are attended to. If actors are so busy that requests for coordination lie unattended in their "in-trays" then interdependent tasks will receive inadequate coordination. The proportion of attended requests for coordination will thus be viewed as a second measure of process quality—*coordination quality,* Q_c—that VDT can generate. Let $(CW_{cm-req} + CW_{ct-req})$ denote the total work volume of coordination requests generated from the simulation and $(CW_{cm-att} + CW_{ct-att})$ the work volume of coordination requested that are actually attended to by the receivers during the simulation. Then the coordination quality for a project simulation can be expressed as

$$Q_c = (CW_{cm-att} + CW_{ct-att})/(CW_{cm-req} + CW_{ct-req}) \tag{8}$$

The notion that the quality of an organization's work processes affects the quality of its ultimate product (e.g., a hospital building) has been demonstrated convincingly by several researchers in the facility engineering domain (Fergusson, 1993). During the 1970s and 1980s, US manufacturing and service organizations changed their focus from measuring the quality of completed products to reducing the variance, and hence enhancing the quality, of work processes. From an engineering viewpoint, we argue that VDT's approach to modeling process quality is a logical next step up the chain of quality control—i.e., we propose to

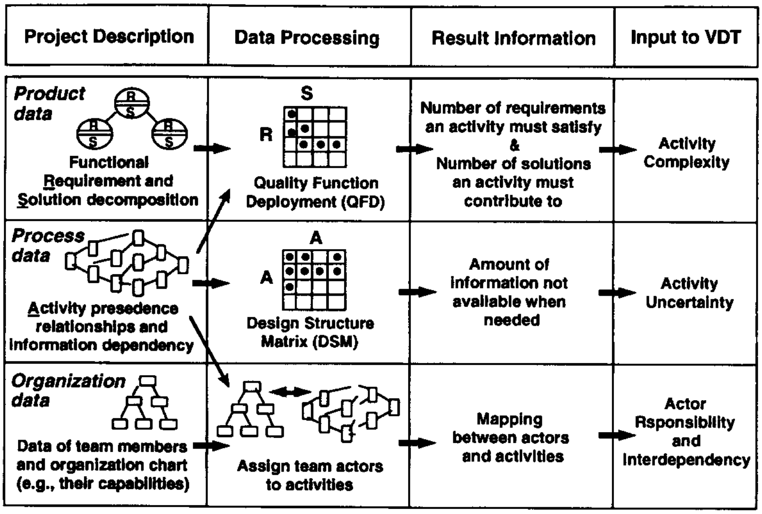

Figure 2. Processes of transforming real project data descriptions into VDT inputs.

measure the quality of the organizations that determine the quality of work processes that, in turn, determine the quality of its products.

3.4 Link to Real Projects

Mapping between an organization task model and accessible real project data is the second requirement described above. VDT's activities are described in terms of complexity, uncertainty and interdependency. Therefore, in order to simulate a real engineering project in VDT and relate the simulation results to real project performance, a link between these task properties and real project data is needed. As part of the VDT task model, Christiansen (1993) developed a methodology that maps real project information into VDT task model through a set of well defined engineering management analyses. This model uses an adaptation of the Quality Function Deployment (QFD) (Hauser and Clausing, 1988) and Design Structure Matrix (DSM) (Gebala and Eppinger, 1991) techniques to derive interactions between requirements and engineering solutions, dependence among design activities in an activity precedence network, sequence-induced activity uncertainty (an activity needing information from one that starts later has increased uncertainty) and relations between members of the project team. A detailed description wazzu of the process of modeling coordination load can be found in (Christiansen, 1993). Figure 2 shows an overview of this model.

4. Modeling Micro-Level Behavior of Actors

Because of its aggregated view of organizational information processing, the Galbraith (1977) framework says very little about how particular actors' attributes influence their information processing behavior. We model project teams as a set of actors that can be either individual managers and engineers, or small sub-teams 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 by simulating actors' actions and interactions.

In VDT, actors have two basic behaviors, *attention allocation* and *information processing*. During simulation, actors perform *production* and *coordination* actions as composites of these two fundamental behaviors.

4.1 Fundamental Actor Behaviors

Attention and time are scarce resources in individuals and organizations. Neither all alternatives nor all the consequences of any one of them can be known (March and Simon, 1958). In VDT, we operationalize this classic behavioral view of organizational problem solving at the actor level via two micro-level assumptions: *actor attention allocation assumption* and *actor capacity allocation assumption.*

4.1.1 Attention Allocation Attention allocation in VDT is related to how an actor chooses which task to work on when it faces alternatives. Based on observations of design team managers conducted by Cohen (1992), VDT models attention allocation based on the following assumption.

Actor Attention Allocation Assumption:

- An actor has an "in-tray" and an "out-tray" (see Figure 1); all incoming information-processing requests, including work items and coordination requests, are stored in the in-tray waiting for the actor's attention.
- Each item in an actor's in-tray has certain priority and specific time of arrival. The actor chooses one item at a time from the in-tray stochastically based on either priority, or time of arrival, or random selection.

VDT adopts a simple attention allocation rule proposed by Cohen (1992) based on his observations of several multi-disciplinary engineering projects. This rule suggests that among all the item selections actors make from their in-trays, 50% are based on priority of the items, 20% are based on the length of time in the in-tray (i.e., FIFO); 20% are based on the most recent item in the in-tray (i.e., LIFO), 10% of those selections are random. In VDT, we further assume that item priorities are measured on a scale from one to nine, with nine being the highest. Production work items have priority 5, a request for information from a reciprocally interdependent peer has priority 5, and a decision about how to handle an error (see subsection 4.2.2) has priority 8.

Although our attention allocation assumption is based on limited observations, it is consistent with the notion of *bounded rationality,* a key concept in understanding organizational behavior (March and Simon, 1958). Actors do not always have enough time and/or effective tools to make rational choices (i.e., based on priority) about what to work on. In VDT, the impact of this bounded rationality is that a project manager does not always pay attention to the most urgent exception reports, and designers may miss important requests from their peers. For example, despite the high priority of an exception report, a project manager may not have a chance to attend to the report within a reasonable length of time. As a result, the reporting actor has to make a decision about how to handle the exception in a "delegation-by-default" mode. Overloaded managers will cause more delegation-by-default decisions by their subordinates. Since our default cultural assumption is that lower level actors are less likely to understand the need for rework, this will lead to a reduction in the percentage of errors that receive rework, and hence to lowered process quality. VDT captures this intuitively correct emergent behavior dynamically through simulation, based on its attention allocation model.

4.1.2 Information Processing As described above, VDT models both production (i.e., design) and coordination processes in terms of information-processing. An activity has a certain amount of work volume to be processed. During activity processing, coordination work may be generated. In VDT, we model information-processing based on the following assumption.

Actor Capacity Allocation Assumption:

- An actor has a certain information-processing capacity determined by its skill type (e.g., civil, mechanical), skill level (e.g., high, medium or low), and allocable time (e.g., two days or one week).
- An information processing work item with certain work volume, whether production work or coordination work, can be processed and completed by an actor if the actor allocates sufficient capacity to the work item.

For a given actor A working on activity B, we assume that the actor has a certain information processing speed IPS_{AB} that is determined by actor A's skill set, activity B's complexity, and the match between A's skill set and B's skill requirement, then actor A's capacity for B can be expressed as

$$CP_{AB} = IPS_{AB} \times \Delta T \qquad (\Delta T = A\text{'s allocable time for } B) \tag{9}$$

If the original planned production work volume of activity B is PW_{OB}, then, ideally (assuming no coordination is needed and no exceptions will occur), actor A can complete activity B if actor A allocates enough capacity CP_{AB} such that

$$CP_{AB} > PW_{OB} \tag{10}$$

or allocates enough time ΔT such that

$$\Delta T > PW_{OB}/IPS_{AB} \tag{11}$$

This capacity allocation assumption has three implications: 1) Information processing not only needs attention but also takes time; 2) The information content of activities and work

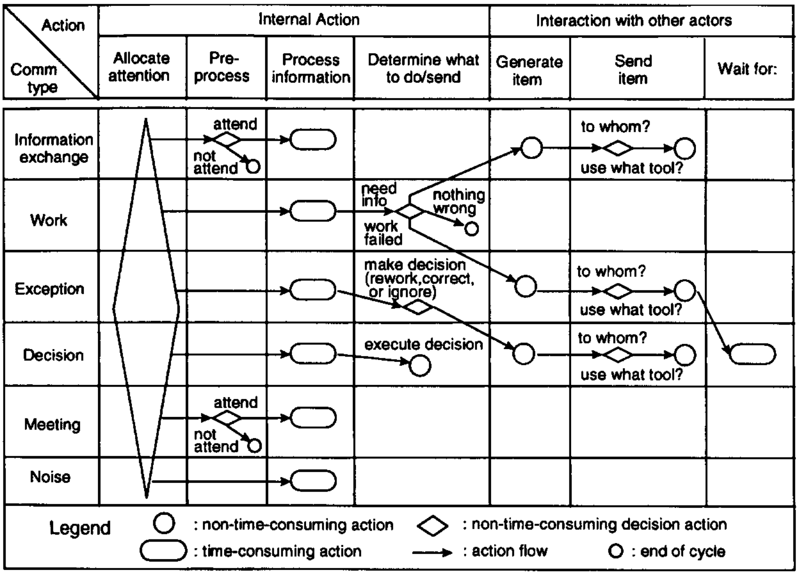

Figure 3. Actor action cycles.

items is captured by its *skill requirement*; and the information processing volume by its work volume; 3) An actor has only limited capacity to allocate.

4.2 Actors' Micro-Level Actions

While the fundamental actor behaviors described above represent task-independent actor characteristics, actor actions are carried out in a specific task context (e.g., production or coordination) to achieve certain task goals (e.g., to complete the project). Actor actions in VDT are centered around how communication items are generated, sent, received, and processed.

4.2.1 Information Flow and Communication Tools in VDT Like in real organizations, information flows constitute the dynamic *life* of the virtual organizations simulated in VDT. In the VDT model, flowing information items are called *communication items.* A communication item can be a *work item* representing a small piece of production work, or a coordination item, being either *information exchange, exception,* or *decision.* Communication items received in an actor's in-tray have the attributes of *sender* and *receiver, priority,* and *time of arrival.* During simulation, work items flow from activities to actors' in-trays; and a series of communication items are generated by actors and sent to other actors' in-trays using communication tools. Figure 3 provides an overview of VDT information flow.

In real organizations, information does not flow in a vacuum. Communication technologies and media are employed to carry information among actors. In VDT, we explicitly model "communication tools" such as face-to-face-conversation, telephones, voice mail systems, facsimiles, and e-mail. Following Nass and Mason (1990), VDT models communication tools based on a number of functional attributes: *synchronicity* (synchronous, partial, asynchronous); *cost* (low, medium, or high); *recordability* (whether or not a permanent record of the communication available routinely); *proximity to user* (close or distant); *capacity* (number of messages that can be transmitted concurrently); and *bandwidth* (low, medium or high) representing the capability of the tool for communicating information represented in each of the *natural idioms supported* (i.e., text, schematics, etc.).

For example, voice mail is partially synchronous, low cost, recordable, close proximity, high capacity for concurrent transmission, and high bandwidth for spoken voice, but low bandwidth for text, schematics or geometry. Telephone is similar except that it is synchronous, not routinely recordable, and has low capacity for concurrent transmission. In contrast, electronic mail is asynchronous, has high bandwidth for text and has high capacity for concurrent transmission. Thus, a manager who wants to send a textual communication to a large number of individuals simultaneously will choose a tool such as voice mail or electronic mail rather than the telephone. In contrast, the need for synchronous communication (arising from priority) will encourage the use of the telephone as opposed to the other two tools; and a communication to coordinate dimensions or layout of components will likely use facsimile or CAD file sharing, rather than telephone.

In the following subsection, we explain how different communication items are generated, attended to, and communicated to other actors through communication tools in different action cycles (see Figure 3).

4.2.2 Actor Action Cycles In the VDT model, actors process both production work and coordination work through various action cycles. An action cycle for an actor is defined as a process, starting from picking up an item from the actor's in-tray and finishing at the point when the actor is ready to turn its attention to its in-tray again to pick up the next item. Depending on what is picked up, an actor will go through one of the action cycles described below.

Processing Production Work As shown in Figure 3, action cycles start from attention allocation. When an actor picks up a work item, A, then a direct work process starts. The actor first allocates ΔT_A to complete the work item. After the work item is completed, the actor checks, stochastically, whether there is a need to communicate with reciprocally related actors; and whether there is an error in the completed work item. An error or exception may occur in a work item if the work item fails to pass a verification check. Like many aspects of VDT, work item verification is performed stochastically.

If there is a need to communicate, then the actor will spend a certain amount of time generating a *Request-For-Information* item, and then send it to the actors whose responsible activities are reciprocally dependent upon the current activity. If the work item is considered to have failed in verification, then the actor will generate an exception item, determine who should be the decision-maker for this exception based on the organization configuration, send the exception to the decision-maker, and then wait for a decision. Waiting is terminated when 1) a decision from the responsible supervisor (to ignore the error, do a quick fix, or completely rework the failed work item) arrives in the actor's in-tray, or 2) the waiting time reaches $\Delta T_{wait_{max}}$, the maximum waiting time set up by the organization for actors to assume delegation by default.

It is important to note that, while processing a work item, an actor may be interrupted by incoming communication items. In this case, the actor will re-allocate attention, with the interrupted work having a higher priority than it had before. If the interruption has a higher priority than the interrupted work item, or if the actor uses arrival time or random selection to choose the next work item, then the interrupted work item is discontinued and placed back into the in-tray, with a priority equal to its initial priority and a revised work volume equal to its remaining work volume.

Information Exchange If an actor's attention allocation process "selects" an information-exchange item, then the actor will decide whether to respond to the item or not. This decision is influenced by the organization's "matrix strength" (Davis and Lawrence, 1977) described below. Since actors in "weak" matrix organizations are not co-located, they tend to rely more on formal meetings to achieve coordination. Actors in weak matrix organizations, therefore, tend to prefer attending scheduled meetings over *ad hoc* information exchange. In contrast, co-located actors in "strong" matrix cultures learn to coordinate informally, and are thus more likely to decide not to attend formal coordination meetings.

If the actor decides to respond to the communication, then it will spend a certain amount of time processing the information. If the decision is not to respond, then the information-exchange item will be discarded and the number of non-attended communications will increase by 1.

Exception and Decision-Making When an actor picks up an exception, it will spend a certain amount of time making a decision. In VDT, we assume that a manager has three choices for the decision: instruct the actor who referred the exception to completely *rework* the failed item, to partially *correct* the item, or to *ignore* the item. Again, which choice to make is determined stochastically, depending on the culture of the organization. We assume that higher level managers have a more global understanding of the consequences of work item failure on the activities performed by other actors, and are thus more likely to require that rework be performed when failures are detected. While this assumption seems to hold for most of the facility engineering organizations we have studied, the opposite turned out to be the "rework culture" for a software organization we modeled in which programmers wanted to correct all known bugs, while managers wanted to ship software to meet deadlines, even with known, non-serious bugs. This kind of behavioral assumption for actors can be directly modified by a user of VDT.

Processing Decisions Processing a supervisor's decision about how to handle a verification failure is a relatively simple action cycle. After picking up a decision item, an actor spends a certain amount of time processing the decision. Then, if the decision is to *rework,* the failed work item will be put back into the actor's in-tray; if the decision is to *correct,* then the failed work item will be put back into the in-tray with half of its original work volume; if the decision is to *ignore,* the failed work item will be discarded, and the actor then can proceed with a new round of attention allocation and information processing.

Attending Meetings VDT models formal communication among actors in regular meetings. Regular meetings are set up before simulation. During simulation, actors who are supposed to attend a series of meetings receive a meeting notice before each meeting. Notices of formal meetings normally have higher priority than ordinary work items. After picking up a meeting notice, an actor will decide whether or not to attend the meeting depending on the level of formalization of the organization. For highly formalized organizations, actors are more likely to attend formal meetings, but less likely to respond to requests for information exchange; and conversely. If an actor decides to attend a meeting, then it will spend the required time in the meeting. The time the actor spent will be converted to work volume for performance calculation. If the actor's decision is not to attend the meeting, the actor will ignore the meeting notice. If an actor misses a scheduled meeting, then the verification failure probability (VFP) for future work goes up, since the actor may have missed important coordination information.

Processing Noise The current version of VDT can model only one organization or project team at a time. Thus it cannot explicitly capture interactions among different projects being performed simultaneously by an organization. To consider the influence of other projects or of outside organizations unconnected with the subject project (e.g., life insurance vendors), VDT models *noise.* Noise in VDT is defined as any communication item that is not related to the current project. For example, for a building design project, designers involved in the project may receive information from their functional departments that does not have any relation to the current project. As part of the environment, noise can impact organizational performance by consuming the attention and time of actors.

5. Organization Structure

One of the fundamental questions to be answered by organizational modeling is how changes in organization structure affect an organization's performance. In VDT, we chose to address this question by modeling, through simulation, how organization structure variables control or influence actors' micro level actions, and consequently the organization's emergent performance, for a given task. An organization structure in VDT represents a pattern of decision-making and communication among actors (Baligh and Damon, 1980; Baligh and Burton, 1981; Malone, 1987). It affects organizational performance by enforcing constraints on actors' decision-making actions through a control structure and centralization policy, and affects communication actions through a communication structure and formalization policy.

5.1 Control Structure and Centralization

A control structure is defined by *Supervise/Report-To* relationships among actors. It is often represented as an organization chart. VDT represents control structures as either flat hierarchies or multiple level hierarchical structures. *Supervise/Report-To* links determine with whom actors should communicate up the chain of supervisors when a work item fails; and the *level of centralization* determines at what level of the hierarchy a decision about the failure should be made.

For example, in a highly centralized structure, most decisions are made at the top of the control structure by project managers. Thus, when an engineer detects an exception, the actor reports the exception to the sub-team leader, and the sub-team leader passes the exception to the project manager for a decision. As a first order effect, this leads to higher quality rework decisions, given our cultural assumption about facility project organizations described previously. However, if high level managers become backlogged (e.g., because the organization was designed that way), they may not be able to attend to the exception reports fast enough. In these cases, the exception reporting actors will likely assume delegation by default and make "ignore" decisions. As a result, the process quality may be reduced. In contrast, in a decentralized organization, decisions for many exceptions are made by the sub-team leaders or even by the engineers themselves. Therefore, a decentralized organization has fewer communications sent to and processed by high-level managers and can generally save coordination time. Decentralization, however, may decrease process quality if lower level actors make less conservative rework decisions based on their limited perspective on the overall project. Again, VDT replicates this commonly observed organizational phenomenon through its attention allocation and information-processing models. In particular, it can predict when centralized decision making may lead to lower quality because of delays in handling exceptions caused by an overloaded project manager.

5.2 Communication Structure, Formalization and Matrix Strength

In addition to the control structure, VDT represents communication structure by *coordinate-with* relationships among actors. The communication structure of an organization defines who can talk to whom. In the current VDT model, we assume that communication requirements for information exchange between designers are purely task dependent. That is, *coordinate-with* relationships among actors are derived directly from the reciprocal relationships among their responsible activities. For example, if activity *A* is reciprocal-with activity *B*, then their responsible actors, *Actor-A* and *Actor-B*, are linked to each other by a coordinates-with relation.

While a communication structure defines who can talk to whom, the level of formalization of the organization defines how frequently they will send communications to each other, instead of communicating through formally scheduled meetings. A more formalized organization relies on scheduled formal meetings for coordination and reduces the frequency of informal inter-actor communications; and conversely.

Whereas the level of formalization affects the frequency of requests for informal coordination, an attribute of organization culture—*matrix strength*—affects the likelihood that a request for a formal meeting or an informal information exchange will be attended to. As described in Section 4.2.2, actors in "weak" matrix organizations tend to prefer attending scheduled meetings over *ad hoc* information exchange and those in strong matrix cultures are more likely to decide not to attend formal coordination meetings. We view matrix

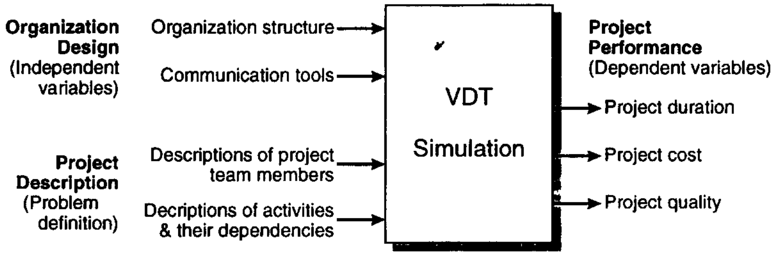

Figure 4. Input and output of VDT simulations.

strength as an attribute of organization culture since it reflects actors' preferences for formal versus informal information exchange.

6. The VDT Simulation System

Figure 4 shows input and output variables for a VDT simulation. We differentiate three kinds of variables. *Problem definition variables* define a given project, e.g., activity dependency network, number of actors. They are usually held constant throughout a suite of simulation runs. Project description can be derived using the load model, schematically represented in Figure 2, from the information of products, processes and the team to be studied. *Independent variables* define organization design, e.g., control structure, level of centralization, and use of communication tools. For a given project description , one may change organization designs to see their effect on the performance of the project. *Dependent variables* are outputs of each simulation run and change as a function of the independent variable settings. The VDT simulation output includes cost, duration and process quality.

The VDT model has been implemented as an object-oriented, discrete event driven simulation system. Activities, actors, communication tools, work items, exceptions, decisions, and information exchange items are all implemented as objects (i.e., data structures that store both the state and the behavior of the concepts they represent). As shown in Figure 5, the VDT system has a *Graphic Organization Editor* for graphically entering and changing task and organization data which is converted into an *Organization and Process Description Language* (OPDL) based ASCII. The core of VDT is a *Simulation Engine* for simulating actors' micro-level actions. A *Graphic Organization Monitor* is used to display actors' micro-level variables during simulation, e.g., number of items in an actor's in-tray; number of exceptions generated, reworked, and ignored; change in actors' verification failure probability. VDT has a set of *Behavior Matrices* represented in OPDL that describe the underlying assumptions about actors' behavior and the organization's culture.

The VDT system was developed based on IntelliCorp's Kappa™, an object-oriented programming environment. VDT runs on both Sun Workstations under Unix and PCs under Windows. A single run of VDT on a Pentium (100MHz) PC for a large project (50 activities, 20 actors, one year project duration, one day work item size) generates upwards of a million simulation events and takes about 15 minutes.

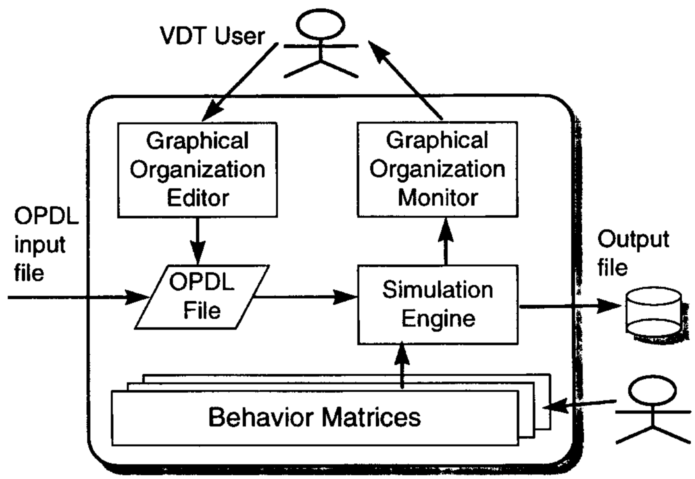

Figure 5. The VDT system architecture.

7. Discussion

7.1 Dealing with Complexities

How to deal with complexity is a crucial issue for organization modeling. It can be said that organization models differ when they choose different ways to deal with task and actor complexities.

Task Complexity The complexity of task domains may vary from simple toy-problems in a hypothetical organization to highly complex engineering design problems such as design of a refinery that involves thousands of components. Which task domain should be considered depends on the purpose of organization modeling. While simple hypothetical tasks provide restricted contexts to study general features of organizations, complex and real tasks provide richer environment descriptions for studying organizational behavior at a more detailed level. Since VDT is developed to provide detailed insights for project managers, we use complex and real tasks.

Complex and real tasks are not easy to model. One way to deal with task complexity is abstraction. Less abstract (or close to real) task descriptions often set up specific requirements and constraints of the tasks. The actors working on the tasks must infer "how" (i.e., detailed actions) to accomplish the tasks through knowledge-based reasoning. More abstract task descriptions, such as those in the VDT model, describe tasks in terms of time and resource requirements and the details of "how" are held uniform and constant. In VDT, "how" to complete a task simply means to spend a certain amount of time on it.

Actor Complexity Human actors are complex. It is difficult to construct a model that sufficiently, coherently and mechanistically describes their behavior. From our VDT modeling experience, we found that there are two ways to reduce the complexity. The first is to choose an appropriate behavior level, as shown in Figure 6. The lower (or bottom)

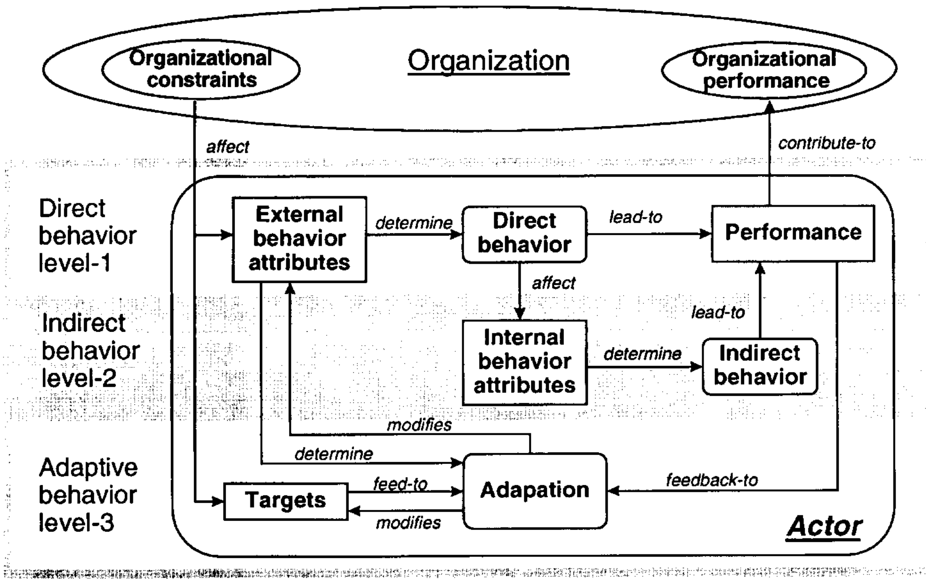

Figure 6. Actor behavior levels for organization simulation.

level models are more complex than the higher level ones. Our purpose for VDT simulation is to analyze how coordination among actors working on relatively routine tasks may impact organizational performance. Thus a Level-2 model was appropriate. If the purpose is to simulate interplay between technologies and learning in organizations, then the adaptive model should be used.

The second way to reduce actor complexity is to abstract actor behavior content corresponding to the task abstraction described above. Behavior content abstraction may vary from treating actors as simple information processing nodes that allocate their time to different work items, to treating actors as intelligent agents (Jin and Levitt, 1993) that reason about task requirements and constraints and infer the actions that must be taken to complete the task. Again, different purposes of simulation may require different levels of abstraction. If allocation of attention and time is important, the "information processing node" model will be simple and useful. If the issue concerns how knowledge distribution impacts organization behavior, then an "intelligent agent" model will be more useful, even though it is complex. Our experience with VDT is that organizations with institutionalized actors working on routine tasks can be modeled effectively with the "information processing" model. It is simple, yet produces relatively accurate predictions.

7.2 Model Validation

Model validation is an important part of computational organization modeling. In developing the VDT model, we had several specific questions related to validation: Does the simulation result make sense (i.e., does it have "face validity")? Does the model capture the underlying features or characteristics of project organizations (i.e., does it have construct validity)? Can we generate theories based on the model (i.e., does it have concept validity)?

Following previous work on the validity issue in social science by Campbell and Stanley (1963) and Cook and Campbell (1976), we have addressed the above questions through extensive testing of both internal (or content) validity and external (or construct) validity.

In VDT, internal validity is related to whether relevant and only relevant concepts (or representation constructs) are included in the model, and whether the concepts are correctly implemented. Burton and Obel (1995b) point out that the purpose of modeling should guide conceptualization so that simplicity and realism can be balanced. Our purpose for modeling is to explicate the performance impacts of lower level (i.e., more specific) contingency factors: e.g., the impact of introducing a specific communication tool (e.g., voice mail), adding reciprocal information interdependency between certain activities, or changing the skill level of an actor who is the bottleneck in a project.

In light of the information-processing view of organizations, we conceptualized fundamental task processes and actors' micro-level behaviors based on our experience with, and observations of, engineering project organizations. We mapped organizational variables (e.g., level of centralization) into actor behavioral constraints (e.g., selecting the decision-maker for exceptions based on the level of centralization) in accordance with organization contingency theory (Galbraith, 1977; Thompson, 1967). Internal validation was carried out through a systematic testing process using intentionally designed small projects. Since our model represents an elaborated version of organization contingency theory, its aggregate simulation results should be consistent with contingency theory predictions. Therefore, we compare aggregate simulation results with theoretical predictions to evaluate the model's internal validity.

VDT's external validity is related to how well the model's predictions agree with observable real project information. Our external validation was based on case-studies. We have conducted more than 20 case-studies of different kinds of projects, from a three-year petroleum refinery design project (Cohen, 1992), to a 12-week software development project (Chachere *et al.,* 1994), to validate VDT's predictions externally. The case-studies conducted so far have been retrospective. We collected information about an already completed project through a set of structured interview processes (Christiansen, 1993), and created a VDT model of the project. After confirming that the simulation results for the original project setting matched well with the actual data for the project (e.g., real project duration and cost), we then introduced variations such as adding specific communication tools (e.g., voice mail) for actors to communicate with each other, or changing the decision policy to more (or less) centralized. We showed the simulation results of these variations to the real project managers and/or domain experts of the project for evaluation. In general, we found good qualitative agreement between real project data, VDT predictions and predictions of the underlying theory. Validation results for these examples can be found in (Levitt *et al.,* 1994) and (Christiansen, 1993).

7.3 Related Work

Our research on VDT has been inspired by a number of previous computational organization models. Cyert and March's (1963) pioneering simulation of department store and can manufacturing organizations provided early examples of the theoretical insights that could be gained from simulating organizational decision making in fine-grained detail.

The "Garbage Can" simulation model (Cohen *et al.,* 1972) of organization anarchies is quite relevant to VDT. First, our capacity allocation assumption is similar to the Garbage Can model's energy allocation assumption. Second, the way Garbage Can uses structures to restrict access between problems and solutions is similar to the VDT organization structure that constrains actors' access to activities and their exceptions. The difference between the two frameworks is that VDT models project organizations with clear goals and well-understood technologies rather than " organizational anarchies".

Burton and Obel's (1984) simple but elegant model of M-form and U-form 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 through ongoing sensitivity testing of its various behavioral parameters.

Carley's (1992) model of "organizational learning and personnel turnover" addresses the issue of organizational design and explicitly represents boundedly rational actors. Carley's model is different from the VDT model in that 1) Carley adopts the Level-3 model in Figure 6 (i.e., the actors in her model can learn), whereas the VDT actors are at Level-2 in Figure 6; and 2) Carley's model deals with relatively simple and stylized classification and choice tasks, whereas the VDT model covers more complex and real tasks.

Masuch and Lapotin's (1989) AAISS system showed the use of non-numerical computing paradigms to model organizational decision making in clerical tasks. They demonstrated the ability to model subtle effects such as the degree of actor commitment, i.e., an actor's willingness to perform a task rather than delegate it. Carley and her colleagues (1992) developed the Plural-Soar model in which actors can learn and communicate with each other. Like these models, VDT uses non-numeric representation of attributes and reasoning together with numerical summation of duration. Unlike these models that represent simple organization problems (i.e., clerical tasks, and warehouse tasks) at a relatively detailed level (i.e., reasoning about tasks), VDT models complex organizational tasks (e.g., refinery design) at a relatively abstract level (i.e., stochastic choices on tasks).

Our experience with VDT has shown that, for our purposes, developing a detailed task model is important for organization modeling (Carley and Prietula, 1994). The level of detail at which to model both tasks and actors' reasoning depends on complexity of the task, modeling purposes, and available modeling technologies (Jin and Levitt, 1993). For the engineering project organizations modeled by VDT, creating a task model to the level of detail of those in Plural-Soar (Carley *et al.,* 1992) and I-AGENTS (Jin and Levitt, 1993) is almost impossible due to the complexity of the tasks. On the other hand, a simple task model like that in "Garbage Can" is too abstract to make coordination work explicit and activity-dependent. We see the VDT task model as lying between these two "extremes", and find it to be adequate for modeling engineering project organizations.

7.4 Future Work

To create a computational organization model, one must decide which approach to take (i.e., mathematical, heuristic-based, or model-based), at which level to model actor behaviors (i.e., direct, indirect or adaptive), and how to deal with the task and actor complexities. In general, the decisions depend on the purpose of the simulation. For example, if the objective is to model highly institutionalized engineering project teams working on highly routine tasks, then a model-based, relatively abstract organization model may function well. To study the impact of new information technologies and learning behavior of industrial corporations, adaptive organization models will be required.

Our current research is moving in two complementary directions. One is more theoretical: using the current VDT model, we are attempting to understand the "information flow dynamics" of work processes "flowing through" organizations, by looking more deeply into the roles of interdependence among activities and the mechanisms of coordination. We are currently designing a set of simulation experiments in which VDT will be used for "intellective" simulation, i.e., simulation with quasi-realistic models of organizations in which values such as centralization, actor skill levels, etc., are ranged across their full spectrum of possible values (Burton and Obel, 1995). By conducting many simulation runs of quasi-realistic organizations, while varying one or two attributes of activities, actors or organization structure, we hope to identify nondimensional parameters of information flow associated with work processes in organizations that can be used to classify different information flow regimes (akin to the way that the nondimensional Reynolds Number is used in fluid mechanics to identify laminar vs. turbulent fluid flow regimes). In a similar vein, a system like VDT could be used to test some of the original ideas about the effectiveness of alternative coordination approaches for handling different types of interdependency first proposed by Thompson (1967) and extended by Malone and Crowston (1991).

The second direction our research is moving in is a natural extension of the current VDT framework. There is a need to add the capability for explicitly modeling multiple organizations (i.e., project teams) working on interacting projects so that models like VDT can be applied to model enterprises operating in a changing technological environment. This extension of scope will allow researchers and managers to address issues related to management of matrix organizations and design of virtual corporations (Davidow,1992) that operate over the Internet and cross the boundaries of time zones and countries.

8. Acknowledgment

There is, of course, a real team behind the Virtual Design Team research program. We acknowledge the support and friendship of past and present members of the VDT research team at Stanford with whom we have shared many passionate discussions and at least as many glasses of wine. In particular, the work described in this paper was advanced by the Ph.D dissertation work of Geoff Cohen, who developed the original KEE/Lisp-based version of VDT—need we say more—and Tore Christiansen, who extended the framework to represent task failures so that VDT could model process quality. Our colleague, John Kunz, has inspired and patiently taught Geoff, Tore and several current students to implement their ideas as " executable symbolic models." Rich Burton has been a frequent and always constructive critic of the VDT research program; his guidance has moved us towards clearer formulation of key theoretical issues as we proceed. Gaye Oralkan and Lars Christensen provided valuable comments as "alpha reviewers" of our manuscript. We are grateful to

Kathleen Carley and two reviewers of the paper who provided excellent suggestions for polishing the paper.

We dedicate this paper to the loving memory of Larry Rosenberg, for his inspired creation and stewardship of NSF's "Information Technology and Organizations" program, and for encouraging us to persist in trying to exploit computational modeling to develop a new micro-contingency theory of organizations.

This research was supported by The Center for Integrated Facility Engineering (CIFE) at Stanford University and National Science Foundation under grant #9122541

9. References

Baligh and Burton (1981), "Describing and Designing Organizational Structures and Processes," *International Journal of Policy Analysis Information Systems,* 5, 251–266.

Baligh and Damon (1980), "Foundation for a Systemic Process of Organization Structure Design," *Journal of Information Optimization Science,* 1, 133–165.

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

Burton, R. M., and B. Obel (1995a), "The Validity of Computational Models in Organization Science: From Model Realism to Purpose of the Model," *Journal of Computational and Mathematical Organization Theory,* 1(1), 57–71.

Burton, R. M., and B. Obel (1995b), *Strategic Organization Diagnosis and Design—Developing Theory for Application.* Norwell, MA: Kluwer Academic Publishers.

Carley, K. (1992), "Organizational Learning and Personnel Turnover," *Organization Science,* 3(1), 20–46.

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

Carley, K., and M. Prietula (1994), "ACTS Theory: Extending the Model of Bounded Rationality," in K. M. Carley and M. J. Prietula, (Eds.) *Computational Organization Theory,* Hillsdale, NJ: Lawrence Erlbaum Associates.

Chachare, J., W. Nasrallah, and G. Okhuysen (1994), "Coordination and Communication in a Software Engineering Company." Unpublished term paper for CE251, Stanford University, Palo Alto, CA.

Christainsen, R. T. (1993), "Modeling Efficiency and Effectiveness of Coordination in Engineering Design Teams." Unpublished Ph.D. thesis, Stanford University, Palo Alto, CA.

Cohen, M. D., J. G. March, and J. P. Olsen (1972), "A Garbage Can Model of Organizational Choice," *Administrative Science Quarterly,* 17(1), 1–25.

Cohen, G. P. (1992), "The Virtual Design Team: An Information Processing Model of the Design Team Management," Unpublished Ph.D. thesis, Stanford University, Palo Alto, CA.

Cyert, R. M., and J. G. March (1965), *A Behavioral Theory of the Firm.* Englewood Cliffs, NJ: Prentice-Hall.

Davidow, W. H. (1992), *The Virtual Corporation.* New York: HarperBusiness.

Davis, S. M., and P. R. Lawrence (1977), *Matrix.* Reading, MA: Addison-Wesley.

Fergusson, K. J. (1993), "Impact of Integration on Industrial Facility Quality." Unpublished Ph.D. thesis, Stanford University, Palo Alto, CA.

Galbraith, J. R.(1977), *Organization Design,* Reading, MA: Addison-Wesley.

Gebala, D., and S. D. Eppinger (1991), "Methods for analyzing design procedures," *Third Intnl. ASME Conference on Design Theory and Methodology,* Miami, Florida.

Hauser, J., and D. Clausing (1988), "The house of quality," *Harvard Business Review,* May 1988.

Jin, Y., and R. E. Levitt (1993), "i-AGENTS: Modeling Organizational Problem Solving in Multiagent Teams," *International Journal of Intelligent Systems in Accounting, Finance and Management,* 2(4), 247–270.

Lawrence, P. F., and J. W. Lorsch (1967), *Organizations and Environment.* Boston, MA: Harvard Business Press.

Levitt, R. E., G. P. Cohen, J. C. Kunz, C. I. Nass, T. Christiansen, and Y. Jin (1994), "The 'Virtual Design Team': Simulating How Organization Structure and Information Processing Tools Affect Team Performance," in K. M.

Carley and M. J. Prietula, (Eds.) *Computational Organization Theory,* Hillsdale, NJ: Lawrence Erlbaum, Associates.
Malone, T. (1987), "Modeling Coordination in Organizations and Markets," *Management Science,* 33, 1317–1332.
Malone, T., and K. Crowston (1991), "Towards an Interdisciplinary Theory of Coordination," MIT Sloan School Working Paper #3294-91-MSA.
March, J. G., and H. A. Simon (1958), *Organizations,* New York: John Wiley & Sons.
March, J. G. (1988), *Decisions and Organizations,* Oxford, UK: Basil Blackwell.
Masuch, M. and P. LaPotin (1989), "Beyond Garbage Cans: An AI Model of Organizational Choice," *Administrative Science Quarterly,* 34, 38–67.
Mintzberg, H. (1979), *The Structuring of Organizations,* Prentice-Hall, Englewood Cliffs, NJ: Prentice-Hall.
Nass, C., and L. Mason (1990), "On the Study of Technology and Task: A Variable-Based Approach," in J. Faulk and C. Steinfield (Eds.), *Organizations and Communication Technology* Beverly Hills, CA: Sage Publications.
Scott, W. R. (1992), *Organizations: Rational, Natural, and Open Systems,* 3rd ed., NJ: Prentice-Hall.
Simon, H. A. (1969), *The Science of the Artificial.* Cambridge, MA: MIT Press.
Simon, H. A. (1976), *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization.* NY: Free Press.
Thompson, J. D. (1967), *Organizations in Action: Social Science Bases in Administrative Theory,* New York: McGraw-Hill.
Weber, M. (1924) *The Theory of Social and Economic Organization* (1947 trans. A. H. Henderson and T. Parsons, Glencoe, IL), Free Press.

Yan Jin is Assistant Professor of Mechanical Engineering in the University of Southern California and Associate Director of the USC IMPACT (Improving Productive with Advanced Collaboration Technology) Laboratory. He received his Ph.D. degree in Naval and Information Engineering from the University of Tokyo in 1988. Since then, Dr. Jin has done research on knowledge-based planning, distributed problem solving, organization modeling as well as their applications to computer integrated manufacturing, collaborative engineering and project management. Before joining USC in 1996, Dr. Jin worked as a senior research scientist at Stanford University since 1991. Dr. Jin has been a principal developer of the Virtual Design Team model. His current research interests include collaborative engineering, computational organization modeling, and agent-based systems.

Raymond E. Levitt is Professor of Civil Engineering in Stanford's Construction Engineering and Management Program and Associate Director of Stanford's Center for Integrated Facility Engineering (CIFE). Dr. Levitt earned MS and Ph.D. degrees in Construction Management at Stanford and a BSCE at the University of Witwatersrand. He was on MIT's Civil Engineering faculty from 1975–1980, and moved to Stanford in 1980. Dr. Levitt was awarded ASCE's Huber Civil Engineering Research Prize in 1982, ENR's Marksman Award in 1985, and the Commitment to Life Award of the National Safe Workplace Institute in 1988. He is a founder and Director of Design Power, Inc., in Cupertino, California, a software and consulting company providing knowledge-based engineering automation solutions. Dr. Levitt is the Principal Investigator of the Virtual Design Team research project.