Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1995), *9*, 145–158. Printed in the USA.

The Virtual Design Team: Modeling organizational behavior of concurrent design teams

YAN JIN,[1] RAYMOND E. LEVITT,[1] TORE R. CHRISTIANSEN,[2] AND JOHN C. KUNZ[1]

[1] Department of Civil Engineering, Stanford University, Stanford, CA 94305-4020, U.S.A.
[2] Det Norske Veritas Research AS, Vertasveien 1, N-1322 Hovik, Norway.

(RECEIVED May 5, 1994; ACCEPTED November 11, 1994)

Abstract

Concurrent engineering is a systematic approach to the integrated, concurrent design of products and the related processes of manufacturing and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from concept through disposal, including quality, cost, schedule, and user requirements. To achieve successful concurrent-engineering design, one needs an integrated framework, a well-organized design team, and adequate design tools. The research on concurrent engineering to date has focused on developing communication infrastructure, design tools, and product data representations. Little attention has been paid to developing tools to address the organizational issues involved in concurrent engineering. The authors' research on the Virtual Design Team (VDT) attempts to develop a computerized analysis tool to support the systematic design of organization structures for concurrent engineering projects. VDT is a computer simulation system. It takes descriptions of design tasks, actors (i.e., designers and managers), and organization structure as input, and produces predicted historical records of the actors' design and coordination behavior, project duration, cost, and design process quality as output. VDT has been applied to model more than ten realistic engineering projects, and the results are qualitatively consistent with the predictions from theory and project managers. The VDT framework for modeling concurrent-engineering teams is described, and examples of VDT applications are presented to demonstrate the effectiveness of the Virtual Design Team approach to modeling the organizational behavior of concurrent design teams.

Keywords: Concurrent Design; Coordination; Organization Design; Simulation

1. INTRODUCTION

With the growth of multinational corporations, competition in the world marketplace is relentless. Those who can get the highest quality, price-competitive product to market in the least time are going to be winners. To respond to this challenge by merely cutting prices and work forces to maintain profit has been proven to be a misguided strategy; rather, the success results from understanding customer needs, developing a product to meet those needs, bringing that product to market quickly and at a fair value. Since 1982, research has been conducted to explore a new product development paradigm called *concurrent engineering* to increase the competitiveness of manufacturers. Recent practice has shown the effectiveness of the new engineering paradigm (Carter & Baker, 1992).

Concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from concept through disposal, including quality, cost, schedule, and user requirements (Carter & Baker, 1992). To make concurrent-engineering design successful one needs an integrated framework, a well-organized design team, and adequate design tools.

Research on concurrent engineering to date has focused on developing communication infrastructure, design tools, and product data representation. Little attention has been paid to developing tools to address the organizational issues involved in concurrent engineering. It has been recognized recently that barriers to concurrent engineering are cultural, organizational, and technological. A success-

Reprint requests to: Dr. Yan Jin, Department of Civil Engineering, Stanford University, Terman Engineering Center Rm 297, Stanford, CA 94305-4020, U.S.A. Phone: (415)723-2918; fax: (415)725-8662; E-mail: jin@cive.stanford.edu.

ful implementation of concurrent engineering requires that these issues be identified and solved up front.

Our research on the Virtual Design Team (VDT) attempts to develop a computerized analysis tool to support the systematic design of organizations for concurrent-engineering projects. VDT explicitly incorporates information processing and communication models from organization theory that allow qualitative predictions of organizational performance. The input to VDT are the descriptions of design tasks, actors (i.e., designers and managers), organization structure, and the communication tools (e.g., facsimile, voice mail, electronic mail, meetings) available to each actor. The output of VDT is a prediction of the total processing time required to complete all subtasks (a surrogate for total labor cost of design), the duration to complete the entire design project along the longest or "critical" path through activities, and verification and coordination quality. VDT's behavior has been validated extensively for internal consistency. Its behavior also compares well with theoretical predictions about, and the observed behavior of, concurrent design teams in several facility engineering domains (Cohen, 1992; Christiansen, 1993).

In this paper we describe the organizational issues of concurrent design in Section 2 and the VDT approach in Section 3. Section 4 describes the VDT framework in detail, and Section 5 presents examples of VDT application. Finally, we compare our research with related work in Section 6, and describe our future work plan in Section 7.

2. ORGANIZATIONAL ISSUES IN CONCURRENT DESIGN

Successful implementation of concurrent engineering depends on project requirements, team organization, task arrangements, and technology (or tools) available to the team. To explicitly address the functional interactions among different design activities, the activities need to be carried out in parallel. As a result, coordination among the actors who are responsible for the activities becomes the critical part of the whole engineering design process. Better team organization facilitates communication and information sharing between team members and leads to efficient coordination. From an organization design point of view, the following questions must be addressed to achieve successful concurrent engineering:

Control structure and policy: What kind of control structure should be implemented, more hierarchical or flatter? Who should report to whom? Given a control structure, what decisions should be made at which level of the hierarchy?

Communication structure and policy: Who can talk to whom? Who should talk to whom about what? Should the team have formal meetings frequently? Who should attend what meetings? Should team members meet or talk to each other informally whenever they need?

Technology or tools: What tools should be used for communication? Is it necessary to introduce new communication tools such as voice mail, E-mail, and video conference? Is it necessary to introduce new CAD tools? How should actors choose their tools?

Task arrangement: How should tasks be arranged—more concurrently or more sequentially? What will be the consequence of introduction of more concurrency? Who should be responsible for which task? How are tasks interrelated with each other? How do these relations affect relations between the responsible actors?

Effectiveness and efficiency: How do we measure project performance (i.e., effectiveness and efficiency) as a whole? What are the organizational and individual factors that contribute to effectiveness, and what are those that contribute to efficiency? How do we trade efficiency for effectiveness, and vice versa?

Although some of these questions are straightforward if a specific task situation is given, the answers to many of the questions are not so obvious. Organization theory can provide aggregated and qualitative answers, but not detailed prescriptions. Concurrent engineering management experts—of whom there are few—address the question based on their experience. We propose a computational organization design approach that allows systematic analysis of team design and provides relatively detailed information about how changes in organization design may impact on team behavior and performance.

3. THE VIRTUAL DESIGN TEAM APPROACH

To address the organizational issues described above, we need a methodology for organization design. Design of artifacts to meet human needs—whether they are physical artifacts such as buildings, or social artifacts such as business organizations—is a ubiquitous human activity and can be broken down into the following generic steps: *requirement definition, synthesis, analysis, evaluation*, and *acceptance* or *recycling* based on the evaluation of performance (Levitt et al., 1991). *Analysis* plays an important role in this process since it is the basis of *evaluation* and iterating *synthesis* for optimal design.

Although extant organization theory (Thompson, 1967; Galbraith, 1977; Mintzberg, 1979) allows aggregated analysis and predictions about the organizational performance of engineering teams under given circumstances, its aggregated view of organizational behavior prevents it from providing specific prescriptions for organization design in a concurrent-engineering context. There is a need for a computational framework in which organizational issues of concurrent engineering can be explicitly analyzed and suitable organizational structures can be identified.

Engineering design teams are composed of different and specialized participants working on complex design tasks with different values, interests, and capabilities. Building a computational model of organizations such as concurrent design teams is difficult because of the complexity of human organizations and the requirement for detailed predictions of team behavior and performance. From a computational modeling point of view, there are three basic issues that must be addressed: (1) what is the unit of analysis for studying organizational behavior – the team or the actors? (2) How do we model engineering design activities – detailed or abstract? (3) How do we validate the model?

To explicitly model coordination among design participants, we treat an actor as the unit of analysis and generate emergent organizational behavior of the design team by simulating actions of and interactions among the actors. We assume that we understand the behavior of "typical" individuals better than that of organizations or teams, especially for highly institutionalized engineering work (Meyer & Rowan, 1993). The simulation may help us to understand the impact of organization design on organization behavior and performance.

To avoid the extreme complexity of real organizations, we adopt an information processing perspective to abstract the real design actions and interactions as described below. Since time and attention of actors are the resources of organizations (March, 1988), our approach attempts to explicitly model how actors allocate their attention and spend their time.

To assess the validity of the model, we choose to compare the predictions produced by the model with those derived from organization theory and those collected from real design projects. The theoretical predictions can be used to justify the qualitative validity of the model, and real project data can be used to assess the accuracy of the simulation prediction.

4. THE VIRTUAL DESIGN TEAM

The Virtual Design Team (VDT) is a computational discrete event simulation model of concurrent design teams. VDT predicts the emergent behavior of a design team for a given organization design through simulating actions of and interactions among the team participants. The goal of the VDT research project is to develop computerized analysis tools to support the systematic design of organization structures for concurrent-engineering projects.

4.1. An information processing perspective on concurrent design

The basic premise of the VDT model is that organizations are fundamentally information processing structures – a view of organizations that dates back to Max Weber's work in the early 1900s, and that is elaborated in the work of March, Simon, and Galbraith (March & 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 information processors termed "actors" – individuals or undifferentiated specialist subteams. Actors send and receive messages along specific lines of communication through communication tools; they pay attention to a selected message in their "in-tray" and spend a certain amount of time to process the message. To capture these characteristics and constraints, VDT employs explicit descriptions of tasks, communications, actors, tools, and structures. Thus, for example, each modeled manager has specific and limited (boundedly rational) information processing abilities; and managers send and receive messages to and from other actors along prespecified communication channels, choosing from a limited set of communication tools. The view of organizations that we have implemented is presented in Figure 1.

Viewing concurrent design teams from an information processing perspective has two important implications. First, we abstract much of the content of the engineering design process. An actor performing a design task spends a certain amount of time on the task. The amount of time depends on the size, complexity, and uncertainty of the task, and the skill level of the actor. Work is verified periodically and can be found to fail. If the work done within a given period of time is found to contain an error, rework on the failed work will frequently solve the problem. Second, not all design tasks can be viewed as the information processing tasks described above. We found that *routine* design projects have a predefined activity network (Moder et al., 1983) and match fairly well with our information processing model. Therefore, we limit our domain to routine design projects. In fact, many design projects in engineering domains are routine according to this criteria, rather than innovative.

4.2. A process model of concurrent design

In VDT, a design task is represented as a dependency network of activities. These activities consist of the design, review, and approval of a series of components or subsystems of the artifact to be designed. An activity is VDT's unit of analysis for modeling task-related issues including information processing requirements, activity interdependency, complexity, uncertainty, and, hence, coordination requirements. The responsibility of an actor for an activity is determined based on the organization structure discussed below.

4.2.1. Design activities

Each design activity requires a specialist to spend a certain amount of time to accomplish it. In order to model as little detail as possible about activities but still retain

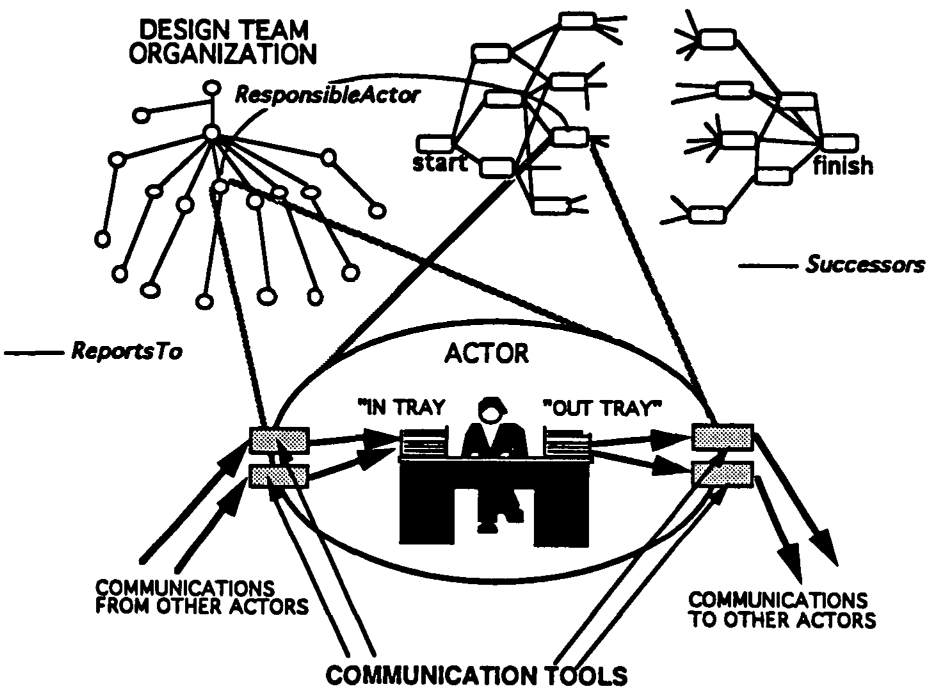

Fig. 1. Overview of the Virtual Design Team. VDT models the design task, actors, organization structure, communication tools, and project policy. The design task is broken down into a precedence network of activities. Actors are information processors with skills and attention allocation rules for selecting items from an "in-tray." The organization structure is defined by supervision and communication relationships among the actors. Each activity is assigned to a single actor.

the accuracy of performance prediction, we model information processing requirements in terms of work volume and work type. *Work volume* is the time needed for an actor with average skills to accomplish the activity. *Work type* is the specialized skill or "craft" an actor must possess to carry out the activity effectively.

Activities in VDT are composed of indivisible components called *tasks*. A task is the minimum amount of work that an actor can choose to process at a certain time, typically one day. After a task is processed by an actor, the result of processing can be either "successful" or "failed," meaning the work was effective or not, respectively. The probability with which a task may be determined to have "failed" is called *Verification Failure Probability* (VFP). VFP is associated with a combination of an activity and it's responsible actor and is determined based on the activity's complexity, the actor's skill level, and the match of the activity's skill requirement and the actor's skill set.

Contingency theorists have characterized tasks in term of *complexity* and *uncertainty* (Galbraith 1977; Thompson, 1967; Nadler & Tushman, 1988). In the organization literature, complexity and uncertainty are treated as variables describing the task environment faced by an organization as a whole. In VDT, we operationalize the concepts of complexity and uncertainty at the activity level (rather than at the overall project level). Complexity has been viewed as the number of different items or elements that must be dealt with simultaneously (Scott, 1992). In VDT, higher activity complexity results directly in higher verification failure probability, and indirectly in more coordination to deal with rework following task failures and possibly in poorer process quality and efficiency. Uncertainty has been defined as the difference between the amount of information required to perform the task and the amount of information already possessed by the organization (Galbraith, 1977). In VDT, higher uncertainty of an activity results in more frequent information exchange communication among the responsible actors. The complexity and uncertainty of an activity have values of either *high, medium*, or *low*.

4.2.2. *Activity interactions and coordination*

Interactions among actors are the main force that affects the organizational performance of concurrent design teams. Dependency relationships between activities require the responsible actors to interact and coordinate with each other. The more concurrent the design activities are, the more coordination will be required among the actors. Radical concurrency may actually cause the design to take longer due to the overwhelming requirement for coordination among the actors. A key goal of the VDT simulation is to let the simulation identify coordination requirements, generate coordination tasks, and simulate the impact of coordination tasks on the team performance.

Following Thompson (1967), VDT models pooled, sequential, and functional relationships among activities. Since we are concerned with single design projects, we assume that all activities have *pooled* interdependence with each other. Therefore, the performance of each activity contributes to the overall organizational performance. Activities are *sequentially* interdependent when the accomplishment of certain activities is a prerequisite for another activity to start. Activities are *functionally* interdependent – reciprocally interdependent in Thompson's framework—with each other if information produced from one activity must be communicated to another activity and this information may result in rework in the other activity.

Dependency among activities determines the requirement for coordination among responsible actors. To capture the intensity or magnitude of coordination, VDT uses information exchange *communication intensity* (CMI) and *verification failure probability* (VFP) to describe activities. The values of the variables are derived from each activity's complexity and uncertainty described above.

4.2.3. Modeling real design projects

VDT's activities or tasks are described in terms of complexity, uncertainty, and interdependence. Therefore, in order to simulate a real engineering project in VDT, one must derive these task properties from the real project data. VDT uses a set of engineering management techniques to specify those task properties (Christiansen, 1993).

As shown in Figure 2, Functional Decomposition (FD) provides a means for creating a hierarchy of customer requirements and their associated technical solutions in an intended product design (Willems, 1988). We use the FD approach to develop a complete set of product requirements and solutions. We then use QFD analysis to identify the interdependence among activities based on the assumed technical interactions among their requirements and solutions (Hauser & Clausing, 1988). We also use QFD to specify the complexity of each activity by analyzing the number of interacting requirements and solutions associated with that activity. The Design Structure Matrix (DSM) technique (Gebala & Eppinger, 1991) analyzes

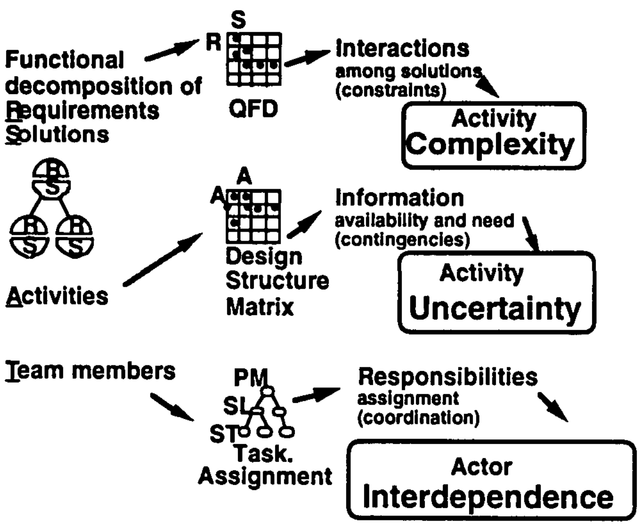

Fig. 2. A model of coordination load for design teams. This model uses Quality Function Deployment (QFD) (Hauser & Clausing, 1988) and Design Structure Matrix (Gebala & Eppinger, 1991) to derive interactions between requirements and engineering solutions, dependence between design activities, and relations between members of the project team.

the information flow among interdependent project activities and assesses the relative uncertainty associated with the requirements of each activity based on the sequencing of the activities. We use these techniques together to specify the complexity and uncertainty levels of each activity.

4.3. Communication and tools

Coordination in a concurrent design team requires information flow among actors. To explicitly capture this information flow, VDT defines a *communication* as an elementary packet of information that is generated and sent by one actor, and received and processed by another. Each communication has its own type and *amount of information* contained (i.e., work volume).

4.3.1. Communication types

In our work volume and work type activity models, a communication in VDT contains a certain amount of work volume, indicating how much time it will take to process the communication. The semantics of the communication are captured by the communication type. At present, VDT has five communication types, namely, work communications, information exchange, exceptions, decisions, and noise.

Work communication: Activities generate work communications and send them to their responsible actors. A work communication is a design task as described in Section 4.2.1. It contains information specifying work volume and associated activity. A work communication can be viewed as a request of design.

Information exchange: An information exchange is initiated by an actor based on the communication intensity and the reciprocal relationships of the activity for which the actor is responsible. An information exchange can be a request for coordination or just a message "for your information." Upon receiving an information exchange, an actor may choose to attend to or to ignore the communication, depending on the actor's backlog and on the culture of the organization as discussed below.

Failure exception: When an actor identifies a task failure, it generates a failure exception communication and sends the communication, together with the failed task, to a decision maker for a decision on how to deal with the failure.

Decision: When a decision maker receives a failure exception, he/she will spend some time to process the exception and make a decision stochastically whether or not the failed task should be reworked. When a decision is made, the decision maker than creates a decision communication and sends it to the actor who initiated the exception.

Noise: Finally, VDT recognizes that some communications received by individuals are irrelevant to accomplishing the task; nevertheless, sorting through and processing these communications, called *noise*, consumes time of the design team participants.

Not all communications are of equal importance for the completion of a given task. Each communication is assigned a priority (on an integer scale from 1 to 9) by VDT based on the relative status of the sender and receiver and the type of communication. A communication also has a lifetime after it arrives in an actor's in-tray, depending on the type of communication tool through which the communication was transmitted. For example, a communication transmitted by a telephone dies after one minute if it is not attended to. An E-mail communication will have a longer lifetime. When a communication exceeds its lifetime, it is removed from the actor's in-tray.

4.3.2. Communication tools

Each communication is transmitted through a tool selected by an actor. The VDT framework represents each tool in terms of values on a set of variables that are theorized to affect both the choice of tool and the results of that choice. The adoption and behavior of tools are then defined in terms of the relationships among the tool variables and the characteristics of the task, the actors, and the organizational structure. In the present version of the VDT, tools are characterized by their *synchronicity* (synchronous, partial, asynchronous), *cost* (low, medium, or high), *recordability* (whether or not a permanent record of the communication is available routinely), *proximity to user* (close or distant), *capacity* (volume 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 text, but low bandwidth for geometry; the telephone is similar except that it is synchronous, not recordable, and has low capacity for concurrent transmission; and electronic mail is asynchronous and has high 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 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.

4.4. Actors and information processing

Actors in VDT represent either individual managers and designers, or small, undifferentiated subteams in a concurrent design environment. Actors in a design team are the entities that perform the design work. By disaggregating organizations into actors and explicitly representing actor behavior, VDT can generate the emergent organizational behavior of concurrent design teams from the actions of, and interactions among, individual actors.

4.4.1. Actor description

VDT models actors in terms of the actors' capability, attention, action, and organizational role. An actor's *capability* is described by its "discipline" (i.e., "craft," such as civil engineer or project manager), skill level (high, medium, or low), task experience for a given class of task (high, medium, or low), and team experience with other team members (high, medium, or low). Based on these variables and a given design task, VDT calculates the actors' *information processing speed*, which determines the time required for the actor to solve the design task. For example, a civil engineer may work slowly when assigned a task with mechanical engineering skill requirements, even though the actor's civil engineering skill level is very high. We further assume that actors with a higher level of task experience and team work experience can work faster since they spend less time determining the task requirement and coordinating with other team members, respectively.

An actor's *attention* determines which task the actor will choose when there are alternatives. An "in-tray" metaphor was proposed by Cohen (1992) to address this issue. Using the in-tray metaphor illustrated in Figure 1, the attention problem becomes which item in an actor's in-tray should be picked up by the actor. Based on the limited observations of design team managers conducted by Cohen (1992), VDT models actors' attention in terms of the priorities of the items in the actors' in-tray and the actors' *attention allocation rule* that specifies probabilities by which actors choose items based on the items' priorities, their times of arrival (i.e., FIFO – first-in first-out, or LIFO – last-in first-out), and random selection. As described above, items in each actor's in-tray can be assigned priority based on the relative status of the sending and receiving actors, communication type, and communication tools. The priority of a communication can also change over time, e.g., a communication's priority might rise as the deadline approaches and then decay to zero.

Another important attribute that governs actors' behavior is their organizational role. In a design team, an actor may play a project manager role, a subteam leader role, or a designer (or subteam) role. Actors playing different roles have different decision-making authority and different decision-making behavior. We assume that project managers tend to demand more rework on failed tasks, whereas subteam leaders and designers or design subteams are more likely to ignore the failure of tasks and proceed without doing rework.

4.4.2. *Actor behavior*

Actors' *actions* in VDT include attention allocation, information processing, communication, and decision making for exceptions.

Allocate attention. Tasks including design tasks and communications arrive in actors' in-trays and wait for processing. Actors allocate their attention to incoming tasks and communications based on their attention rules. The simple attention allocation rule proposed by Cohen (1992), based on his observations, was that actors use priority about 50% of the time to choose the next item from their in-tray to work on, length of time in the in-basket or FIFO is used 20% of the time, the most recent item in the in-tray or LIFO is used 20% of the time, and the actor chooses items randomly 10% of the time.

Process information. After selecting a task or communication item from the in-tray, an actor calculates the time required to process it based on the actor's information processing speed and the work volume of the task. During the time when an actor is processing a task or communication, an incoming communication from other actors may arrive at any time. Whenever this happens, the actor applies the attention allocation rule stochastically to determine whether to stop processing the current task to attend to the new task or communication.

Communicate with others. Coordination among actors is accomplished through communications. Communications take actors' time from doing ordinary design work and generate coordination work load. Actors in VDT communicate with each other by sending communication items to each other or by attending meetings. Meeting schedules are set up at the project level based on project coordination requirements. Actors generate communication items for information exchange or failure exceptions stochastically, based on the activities' complexities, uncertainties, and the actors' capabilities. Decision communications are generated by actors in response to exceptions.

After receiving a communication or a meeting request, an actor must decide whether to attend to it or not. We assume that the probability of choosing to attend to an informal communication or a formal meeting depends on an aspect of organizational culture – the "strength of the matrix."

By *matrix strength* we mean the extent to which actors are located in discipline-based functional departments and influenced by functional managers ("weak matrix"), versus co-located with other discipline specialists in dedicated project teams and influenced more by the project manager than by their respective functional managers ("strong matrix") (Davis & Lawrence, 1977). VDT assumes that the co-located actors in the strong matrix will be more likely to attend to informal communications, while the actors in a weak matrix will be culturally biased toward communicating in formal, scheduled coordination meetings.

Generate exceptions and make decisions. An actor generates an exception when verification indicates a task failure, stochastically. To resolve the exception, the actor chooses a decision maker based on whether the organizational control is more centralized or decentralized. In a more centralized organization, most decisions are made by high-level managers, whereas, in less centralized organizations, decisions are often made by subteam leaders or designers themselves. After receiving an exception, an actor must make a decision about whether the failed task should be partially or fully reworked, or whether the responsible actor should proceed without rework. The probabilities for each choice are based on the actor's position in the organization. If the actor who sent the task failure exception for decision does not receive a decision after waiting for a given period of time (e.g., because the exception did not attract the attention of the decision maker), the actor assumes "delegation by default" and decides locally whether to rework or to proceed by ignoring the failure.

4.5. Organization structure

One of the fundamental questions in organizational modeling is to determine what changes when an organization's structure changes, and how this affects the organization's performance (Mintzberg, 1979, 1983). Since organization performance in VDT emerges from the simulated actions of, and interactions among, actors, we chose to address this question by identifying variables that control the actors' behavior. Thus, in VDT, organization structure affects organizational performance by enforcing behavioral constraints on individual actors.

Organization structure is defined by a set of attributes of and relationships among actors. VDT differentiates between formal control structure and information communication structure.

A *formal control structure* is a hierarchy of *reporting-to* (or *supervise*) relationships between the actors and has a certain level of *centralization*. Reporting-to links guide actors to determine with whom they should communicate when a task fails; and the level of centralization determines at what level of the hierarchy a specific decision should be made. For example, in a highly centralized organization structure, decisions are made by project managers. Thus, when an engineer actor detects an exception, the actor reports the exception to the subteam leader. The subteam leader then passes the exception to the project manager for a decision. In a decentralized organization, however, the decisions for exceptions are often made by the subteam leaders or even by the engineers themselves. Therefore, in decentralized organizations, fewer communications are sent to and processed by high-level managers. This reduces both the need for communication and the need for information processing.

An *informal communication structure* is defined by *coordinate-with* relationships among the actors and has a certain level of *formalization*. If activity A is *reciprocal-with* activity B, then their responsible actors must be linked via a *coordinate-with* relationship. Coordinate-with links specify who can talk to whom, and the level of formalization determines the frequency of the communication. For example, a highly formalized organization relies on scheduled formal meetings for coordination and reduces the frequency of informal interactor information exchanges. The organization matrix strength described above also affects the strength of the communication structure. Since actors in weak matrix organizations are often not co-located, they tend to use informal interactor communications less often, relying on formal meetings instead. We also call this matrix strength *organization culture* since it reflects actors' informal social relationships.

4.6. Team effectiveness and efficiency

As indicated above, VDT views the time and attention of actors as resources of organizations (March, 1988) and measures organization efficiency and effectiveness by looking at how these resources are consumed. The measurement of efficiency of a design team working on a design project can be defined by project duration and total cost (e.g., total design work-hours) spent to accomplish the project. The measurement of effectiveness of a project team is somewhat more difficult. Since we are not modeling the content of products being designed, it is impossible to judge the effectiveness from the quality of the product. Instead, we choose to measure the quality of the design process as an indicator of project effectiveness. VDT models the quality of the design process in terms of how 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 future tasks. We take the position that the detection of task failure is not in itself an indicator of poor quality; rather, it is the organization's response to detected failures that determines the quality of its work processes. In general, we argue that managers at higher levels have a more global understanding of the consequences of task failure on interdependent tasks performed by other actors, and are thus more likely to require that rework be performed when failures are detected. We view the proportion of detected failures that are reworked as a measure of the quality of an organization's work processes. With this refinement, the information flow model can model the trade-off implicit in decentralization: It can lead to faster decision making, but only with a penalty in process quality.

Another, more subtle, 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 be inadequately coordinated. Actors' work loads and attention rules, and the communication tools available to actors for coordinating with each another, will affect the rate of response to requests for coordination in a design team. The proportion of attended requests for coordination thus will be viewed as another measure of process quality that VDT can generate.

In summary, VDT uses the time and cost required to accomplish a design project as the team efficiency measurement. The team effectiveness is assessed based on two process quality measurements: *verification quality* – the proportion of reworked failed tasks, and *coordination quality* – the proportion of attended communications.

4.7. The VDT simulation environment

VDT operationalizes Galbraith's (1977) information processing model of organizations by explicitly incorporating specific tasks and actors with attention allocation capabilities, and by addressing coordination issues at the microlevel in terms of explicit interaction among team participants. The VDT simulation environment can be characterized by a number of objects representing tasks, actors, and organizations, as shown in Figure 1, and the organizational processes that facilitate coordination among team participants. The model is formal in that it includes the basic concepts of, and predicts behavior based on, a set of widely accepted theories. VDT is implemented on a Sun Microsystem IPX Sparcstation using Kappa, an object-oriented programming environment from IntelliCorp,and the SIMLIB, a discrete-event simulation system we developed on top of Kappa.

4.7.1. Simulation design

Figure 3 illustrates how VDT produces a set of team performance measures (dependent variables) based on given organization structure, communication tools (independent variables), and a description of team actors and project activities (state description). As part of the organization structure, centralization policy determines the probability of how "high up in the hierarchy" decisions are made on how to deal with exceptions; and formal-

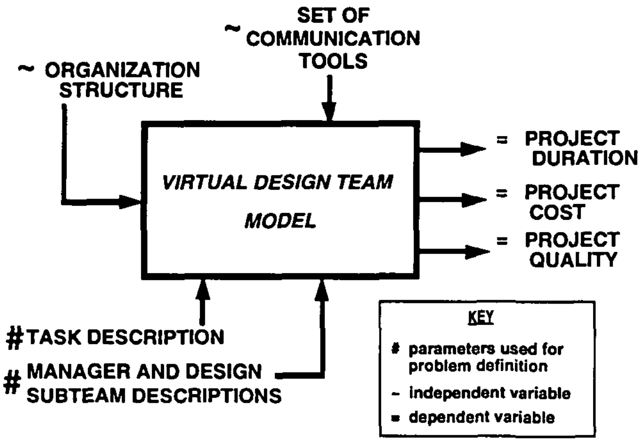

Fig. 3. The function of the Virtual Design Team. VDT simulates changes in different aspects of project team performance, given changes in organization structure, communication tool availability, and project policy.

ization policy determines the degree to which project communication is made up of formal meetings versus informal information exchanges. Links among actors such as *Report-to* and *Coordinate-with* can be set as input to VDT, and changing the links results in changes in team performance. Organizations with different "matrix strengths" will give different priority to formal versus informal communication and lead to different team performance. The state description variables are set up to model the particular project under study and are kept constant for changes in the independent variables. Different projects will thus have different state descriptions. No systematic study of the relationship between state description variables and dependent variables is carried out in the present research, although any of the state variables in the current study could be treated as independent variables in a different set of experiments. For example, VDT's user can vary the task description (e.g., to study the effect of a shorter schedule with more concurrency) or the actor descriptions (e.g., to study the effect of employing more highly skilled actors in key positions) while holding structure and/or communication tools constant.

The output from VDT includes project duration, total cost, and project quality measurements such as *verification quality* (ignored exceptions/total exceptions), *coordination quality* (nonattended communication/total communication), *schedule quality* [(actual duration − scheduled duration)/scheduled duration], and *budget quality* (rework volume/work volume). Besides the performance results, VDT also records the dynamic behavioral data of actors such as the number of items in an actor's in-tray at each time, time spent waiting for decisions, etc., and progress data of activities, such as work completed, amount of rework, etc.

4.7.2. The OPDL language and graphical interface

In order to make it easy for students and project managers to create input files for simulation in VDT, we developed a high-level language called OPDL, an Organization and Project Description Language. OPDL is a computer language for describing and simulating organizational behavior and performance of teams working on engineering projects. Using OPDL, a user can program project activities, project policy, actors and organizations, load the program into VDT, and then simulate the project's performance. OPDL is not only an interface to VDT but has been designed as a more general language for formal descriptions of organizations and projects. Figure 4 shows part of an OPDL program.

VDT views organizational performance as the results of actors' microlevel processes. To understand how the microlevel processes contribute to organizational performance, VDT has a graphic interface to show how many items there are in the in-tray of certain actors, how many meetings and communications have been attended to so far, and how verification failure probability changes as a result of the actors' decisions on whether to do rework and/or to attend to communications, etc. Through the graphic interface, one can clearly understand who (which actor) is overloaded and who is spending excessive amounts of time waiting for approval from supervisors.

5. APPLICATION OF THE VIRTUAL DESIGN TEAM – EXAMPLES

VDT has been applied to model more than ten realistic industrial concurrent design projects ranging from refinery design, subsea module design, and electrical substation extension design to construction management. We found three-way qualitative consistency among predictions of the simulation model, of organization theory, and of experienced project managers. In the following, we describe two examples and demonstrate how VDT can be used to analyze the organizational performance of concurrent design teams.

5.1. Impact of tools and organization structure on project duration

We applied VDT to model a routine petroleum refinery design project having a total design and construction cost of approximately $130 million, a planned duration of 20 months, and, at its peak, approximately 120 managers, engineers, designers, and support staff located in two offices (Cohen, 1992). Our focus in this case was to see how different organizational structure and communication tools may impact on organizational performance.

All actor and task descriptions were derived from this project and held constant. The predefined actor attention

```
(Activity Achitectural_design
    :WorkVolume               6000    % An integer [1000]
    :TaskNumber               100     % An integer [10]
    :Uncertainty              High    % High/[Medium]/Low
    :RequirementComplexity    Medium  % High/[Medium]/Low
    :SolutionComplexity       High    % High/[Medium]/Low
    :CraftRequirement         Architecture
                                      % [Civil]/Mechanical/
                                      %   Electrical/Management/
                                      %   Architecture
)

(Actor Architect-John
    :Role                     SubTeam % [SubTeam]/SubTeamLeader/
                                      %   ProjectManager
    :NumberOfParticipants     1       % An integer [1]
    :Skill                    Medium  % High/[Medium]/Low
    :TaskExperience           Medium  % High/[Medium]/Low
    :ResponsibleFor           Actv_1  % An activity [Null]
    :Craft                    (Architecture High)
                                      % High in architecture,
                              (Mechanical Low)
                                      % Low in Mechanical [Null]
    :SupervisedBy             PM4     % An actor [Null]
)
```

Fig. 4. Part of an OPDL program. This is part of a program describing a building design project. Comments headed by "%" explain possible values separated by "/", and default values denoted within [ ].

rules and tool selection rules were initially derived from a series of interviews with actors on this project and then compared to observed managerial behavior in a second petrochemical design project. Since we were concerned with organizational structure and communication tools, we selected two independent variables—level of centralization of decision making and presence or absence of voice mail. As explained above, the theory predicts that decentralizing decision making and adding voice mail should each decrease the project's duration.

Our candidate organization had a decentralized structure and provided voice mail to its designers. To model different levels of centralization, we changed the level in the hierarchy to which design approval exceptions were routed from subteam managers (decentralized) to the design manager (centralized). To give actors voice mail capabilities, we reset the synchronicity attribute of the existing telephone tool from synchronous to partially synchronous, the recordability attribute to recordable, and the capacity attribute to high capacity. Values of all other variables in the model were set to an average value such as "medium."

By independently varying the level of centralization, and the availability of voice mail, we examined the impact of these two variables on the duration of individual tasks and on overall project duration. The results of the simulations are shown in Figure 5. The numbers in each cell show the mean and standard deviation (for three simulation runs) of project duration, in working days. Standard deviation is the number shown in parentheses. The ">" symbols indicate prediction from Galbraith's theory (1977). The simulation results indicate that:

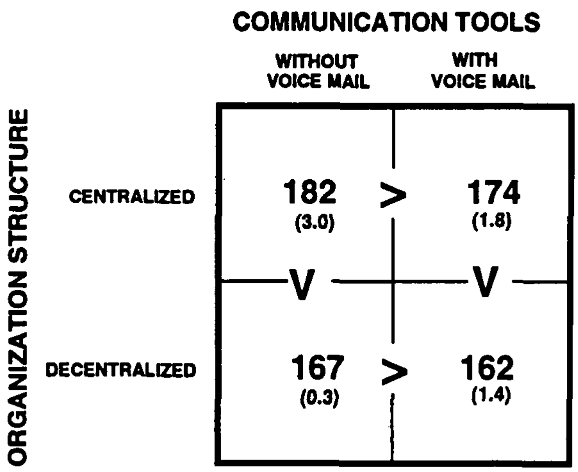

Fig. 5. Impact of communication tools and organization structure on project duration. VDT contingent predictions of change in project duration compare qualitatively with predictions based on Galbraith's theory. Numbers in each cell show the mean and standard deviation (for three simulation runs) of project duration, in working days. Standard deviation is the number shown in parentheses. The ">" symbols indicate predictions of Galbraith's theory.

- Centralized decision making leads to longer task duration than does decentralized decision making.
- Voice mail improves performance for both centralized and decentralized organization structures.
- The interaction between voice mail and centralization is not significant.

5.2. Trade-off between efficiency and effectiveness

In another test example, we applied VDT for the Statfjord subsea oil pumping module satellites project (Christiansen, 1993) to investigate the trade-off between effectiveness and efficiency with respect to the design team's level of centralization. The engineering design part of this project, which we modeled, was budgeted at $1.2 million and 22 months, and involved 13 to 20 engineers (including the project manager). The project team participants were all co-located and worked exclusively on this project, so the project team was strongly project-oriented and had a *strong matrix structure* (Davis & Lawrence, 1977). After the project was under way, the original project plan was reengineered to reduce the schedule from 3 years to 2 years. As a result, the engineering design part of the project had to reduce its schedule from 22 months to 15 months. The resulting schedule was "radically concurrent" and required extensive coordination in order to ensure consistency between engineering activities normally performed in sequence.

Figure 6 shows three-way predictions of how the level of centralization of decision making in the Statfjord organization impacts on project duration. The expected behavior from contingency theory is based on the assumption that managers have a more global view of different parts of the project, and thus will tend to choose rework, rather than locally suboptimize performance by "quick-fix" corrections or ignoring failures. Also, decisions from higher level managers will often be delayed by other items in the manager's in-tray (Galbraith, 1977). In VDT, higher centralization gives higher probability that decisions will be made by managers and most decisions will be to rework rather than ignore the failed tasks. As a result, higher centralization leads to longer project duration. The prediction from simulation is qualitatively consistent with both theory and the project manager.

Figure 7 shows VDT simulation results for the effect of centralization on verification quality (ratio of uncorrected nonconformance to total number of nonconformance) together with the prediction from the project manager and the qualitative prediction from contingency theory. In this case, the theoretical assumption that managers have a more global overview leads to a prediction of fewer uncorrected nonconformance for higher centralization and more uncorrected nonconformance (by subteams) for lower centralization. Again, the prediction from simulation is consistent with the predictions from the project manager and from Galbraith's theory (1977).

From Figures 6 and 7 it is clear that there is a trade-off between project efficiency (duration) and effectiveness (verification quality) for organization structure design (level of centralization). One advantage of the VDT sim-

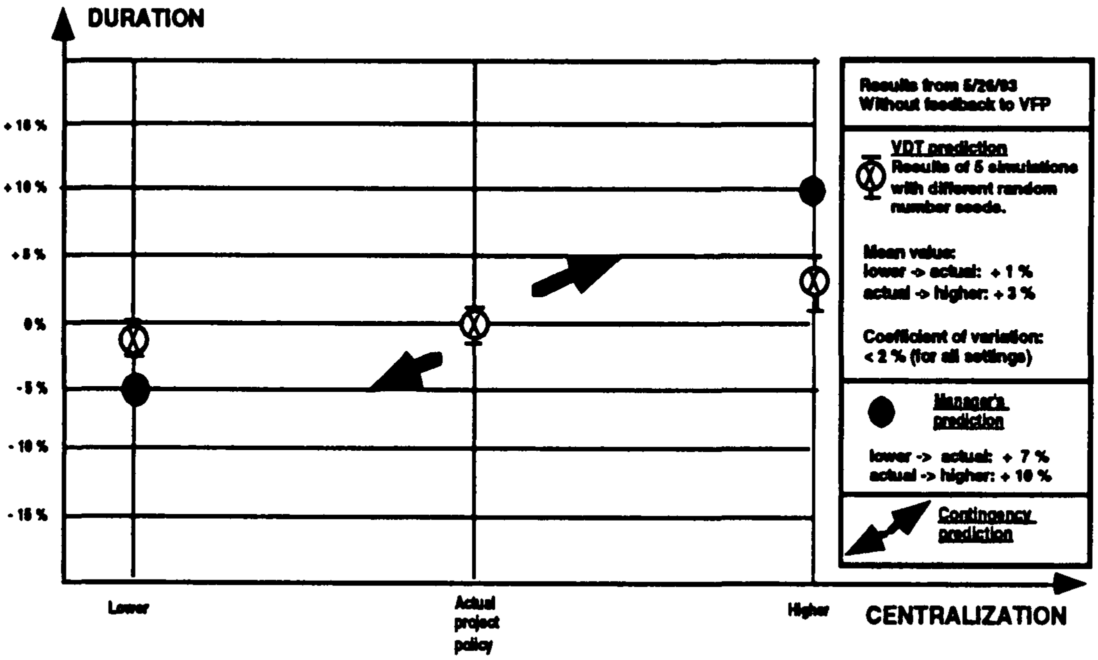

Fig. 6. Statfjord Satellites Project – effect of centralization on duration. This figure shows the results of computer simulations for the effect of centralization on project duration, together with the prediction from the project manager and the qualitative prediction from contingency theory.

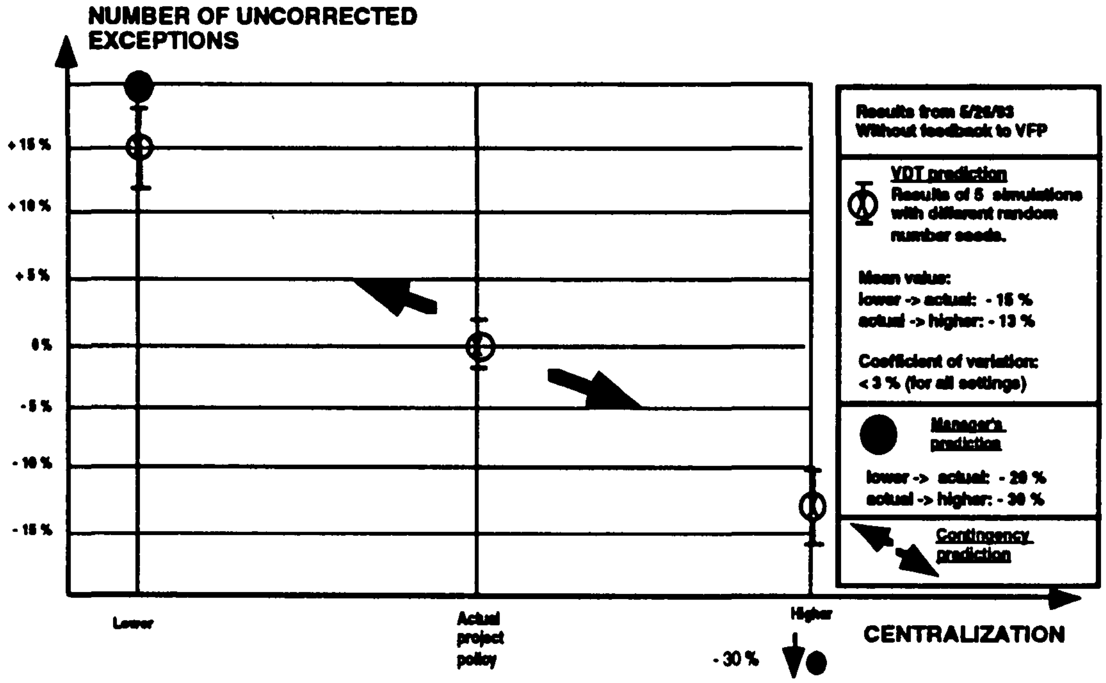

Fig. 7. Statfjord Satellite Project—effect of centralization on effectiveness. This figure shows the results of computer simulations for the effect of centralization on verification quality (ratio of uncorrected nonconformance to total number of nonconformance), together with the prediction from the project manager and the qualitative prediction from contingency theory.

ulation approach is that a project manager can "play" with VDT through simulations to determine which organization design is the "best" given project efficiency and effectiveness requirements.

In the above examples, we demonstrated the validity of VDT by comparing simulation predictions with predictions from theory and the experienced project managers. Although we do not claim quantitative validity of the system at present, we expect that the detailed quantitative results from VDT simulations will provide qualitative but useful detailed insights into organization design for concurrent-engineering projects.

6. RELATED WORK

VDT provides a computational framework to investigate organizational issues involved in concurrent engineering. The basic approach taken by VDT is computational organization modeling. There have been many studies concerned with organization modeling and concurrent engineering management. For example, early computer simulation models of organization include Cyert and March's (1965) pioneering work on the behavioral theory of business firms, and Cohen and his colleague's Garbage Can model (Cohen et al., 1972) of organizations like universities. These models provided examples of theoretical insights that could be gained from simulating organizational behavior.

Recent work on organization modeling applied artificial intelligence and object-oriented programming techniques. Masuch and LaPotin's DoubleAISS is one of the first computational models that applies AI and object-oriented tools to study organizational decision making, based on the interplay of actors, skills, actions, issues, and structure (Masuch & LaPotin, 1989). Carley and her colleagues extended Soar and built the Plural-Soar system to examine the three-way relationship between individuals' skills, job requirements, and schemes for coordinating individuals within the organization (Carley et al., 1992). Like these systems, VDT uses an AI-based nonnumeric representation of attributes and reasoning together with numerical computation of variables like duration. Unlike these systems, VDT uses a more abstract description of tasks and actors to model realistically complex concurrent-engineering design projects and emphasizes the impact of organization structure on project efficiency and effectiveness.

On the other hand, researchers in the concurrent-engineering field have recognized that the barriers to concurrent engineering are cultural, organizational, and technological in nature. Karandikar et al. (1993) developed a model, a measurement tool, and a methodology—the Readiness Assessment for Concurrent Engineering (RACE)—to assist concurrent-engineering implementors in identifying the barriers and prioritizing implementation actions. Adachi and his colleagues (1994) developed a framework for supporting the organization and structuring of product development teams in a concurrent-engineering environment. Like these studies, VDT provides a framework to analyze relationships between organization structure and team performance given the concurrent-engineering context. Unlike these studies, which are based on empirical data analysis, VDT adopts a computer model-based approach. The concurrent-engineering environment is represented as a computer model, and the analysis and design of the organization structure are carried out through model-based simulations. Once the model is validated, the model-based approach provides more flexibility for organization analysis and design.

7. SUMMARY AND FUTURE WORK

We presented a computational approach to organization analysis and design for concurrent-engineering projects. In summary, 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. We are conducting experiments to validate the quantitative predications from VDT simulations.

Besides these aggregate predictions, which can be compared to predictions derived from managers' or contingency theory, VDT generates a wealth of data during each simulation run about the workloads and activities of individual actors. We have developed graphic visualization tools to facilitate inspection of these detailed performance data. For example, if centralization results in a longer project duration, it is possible, using VDT, to determine where bottlenecks are occurring. Thus, by inspecting actors' work history logs, we can determine which actors have spent excessive amounts of time waiting for decisions from supervisors. We can then inspect the depth of the in-trays of these actors' supervisors over the course of the project to determine which supervisors were overloaded with exceptions.

Once the key bottlenecks have been found, a user of VDT can propose decentralization of decision making, reassignment of subordinates to reduce the supervisor's span of control, better communication tools, or other changes in the structure of the design team's organization. Each proposed change can then be modeled in VDT and simulations conducted to see whether it produces a better overall result in terms of VDT's efficiency and effectiveness performance measures.

We plan to extend VDT in three respects. First, we will continue to validate and calibrate VDT. We are offering a course at Stanford, "Organization Design for Projects and Firms." Students in this course will help to calibrate VDT by using the simulation system to model a real organization as their term project.

Second, VDT has been developed in the facility engineering domain. We plan to use VDT to model project teams in other domains such as aircraft design and software engineering. We believe that applications of VDT to other engineering domains will result in new requirements and lead to a more general model of design teams.

Third, the current VDT models actors in terms of nominal variables, such as skill and task experience, and the rules for allocating attentions and selecting communication tools. Our ongoing work tries to introduce cognitive aspects (e.g., actors' aspirations, interests, and knowledge) and learning capability into the model of actors (Jin & Levitt, 1993). By doing so, we expect to be able to model the adaptation behavior of organizations emerging from the simulation, and consequently to relate actors' cognitive aspects and dynamic behavior with organization design. We also plan to make VDT capable of explicitly simulating multiple projects so that we can study inter-organization issues using VDT.

ACKNOWLEDGMENT

This research was supported, in part, by the National Science Foundation under grant #IRI-9122541 and by seed research grants from the Center for Integrated Facility Engineering at Stanford University. The opinions expressed in this paper are solely those of the authors.

REFERENCES

Adachi, T., Shih, L.C., & Enkawa, T. (1994). Strategy for supporting organization and structuring of development teams in concurrent engineering. *Int. J. Human Factors Manuf. 4(2)*, 101–120.

Carley, K., Kjaer-Hansen, J., Newell, A., & Prietula, M. (1992). Plural-Soar: A prolegomenon to artificial agents and organizational behavior. In *Artificial Intelligence in Organization and Management Theory* (Masuch, M. and Warglien, M., Eds.), pp. 87–118. North-Holland, Amsterdam.

Carter, D.E., & Baker, B.S. (1992). *Concurrent Engineering—The Product Development Environment for the 1990s*. Addison-Wesley, Reading, MA.

Christiansen, R.T. (1993). Modeling efficiency and effectiveness of coordination in engineering design teams. Ph.D. Thesis. Stanford University, Stanford, CA.

Cohen, M.D., March, J.G., & Olsen, J.P. (1972). A Garbage Can Model of organizational choice. *Administrative Sci. Quart. 17(1)*, 1–25.

Cohen, G.P. (1992). The Virtual Design Team: An information processing model of the design team management. Ph.D. Thesis. Stanford University, Stanford, CA.

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

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

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

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

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

Hauser, J., & Clausing, D. (1988). The house of quality. *Harvard Business Rev. May–June.*
Jin, Y., & Levitt, R.E. (1993). i-AGENTS: Modeling organizational problem solving in multiagent teams. *Int. J. Intel. Sys. Accoun. Fin. Mgmt. 2(4)*, 247–270.
Karandikar, H.M., Fotta, M.E., Lawson, M., & Wood, R.T. (1993). Assessing organizational readiness for implementing concurrent engineering practices and collaborative technologies. *Proc. Second Workshop on Enabling Technologies Infrastructure for Collaborative Enterprise*, pp. 83–93. IEEE Comput. Soc. Press, New York.
Levitt, R.E., Jin, Y., & Dym, C.L. (1991). Knowledge-based support for management of concurrent, multidisciplinary design. *J. Artif. Intell. Eng. Des. Anal. Manuf. (5)2*, 77–95.
March, J.G., & Simon, H.A. (1958). *Organizations*. John Wiley, New York.
March, J.G. (1988). *Decisions and Organizations*. Basil Blackwell, Oxford, UK.
Masuch, M., & LaPotin, P. (1989). Beyond Garbage Cans: An AI model of organizational choice. *Admin. Sci. Quart. 34*, 38–67.
Meyer, J.W., & Rowan, B. (1993). Institutionalized organizations: Formal structure as myth and ceremony. *Amer. J. Sociology*, 343–63.
Mintzberg, H. (1979). *The Structuring of Organizations*. Prentice-Hall, Englewood Cliffs, NJ.
Mintzberg, H. (1983). *Structuring in Fives: Designing Effective Organizations*. Prentice-Hall, Englewood Cliffs, NJ.
Moder, J.J., Phillips, C.R., & Davis, E.W. (1983). *Project Management with CPM, PERT and Precedence Diagramming*. Van Nostrand Reinhold, New York.
Nadler, D., & Tushman, M. (1988). *Strategic Organization Design—Concepts, Tools and Processes*. Scott, Foresman, Glenview, IL.
Scott, W.R. (1992). *Organizations: Rational, Natural, and Open Systems*, 3rd ed. Prentice-Hall, Englewood Cliffs, NJ.
Simon, H.A. (1976). *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization*. Free Press, New York.
Thompson, J.D. (1967). *Organizations in Action: Social Science Bases in Administrative Theory*. McGraw-Hill, New York.
Willems, P. (1988). A functional network for product modeling. Technical Report PLI-88-16, IBBC-TNO, Rijkswijk, Netherlands.

Yan Jin is a Research Associate in the Department of Civil Engineering, Stanford University. He earned the Ph.D. degree in naval engineering from the University of Tokyo in 1988. Since then Dr. Jin has been doing research on knowledge-based planning systems, distributed problem solving and multiagent systems, organization modeling, and their applications to marine traffic control, computer integrated manufacturing, collaborative design, and concurrent-engineering project management. Dr. Jin's current research interests include agent-oriented collaborative design, computational organization modeling, multiagent systems, and coordination science.

Raymond Elliot Levitt is a Professor of Civil Engineering in Stanford University's Construction Engineering and Management Program and an Associate Director of Stanford's Center for Integrated Facility Engineering (CIFE). Dr. Levitt earned the M.S. and Ph.D. degrees in construction management from Stanford and the B.S.C.E. degree from the University of Witwatersrand. He was on MIT's Civil Engineering faculty from 1975 to 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 the Director of Design Power, Inc., Cupertino, CA, a software and consulting company providing knowledge-based engineering automation solutions. The Virtual Design Team research in which Dr. Levitt is currently engaged integrates his organization theory and artificial intelligence interests.

Tore R. Christiansen is working as a principal research engineer in the department for information systems of DNV Research, the strategic research company of the Det Norske Veritas ship classification society. He studied aeronautical engineering at the University of London and the Massachusetts Institute of Technology, and has experience as a developer and user of computational mechanics and computer-aided engineering. He obtained the Ph.D. degree from Stanford University, Department of Civil Engineering, in 1993, working as part of the VDT project team. His thesis, as well as his current interest, lie in the area of using computer tools to understand and improve the efficiency and effectiveness of project organizations.

John C. Kunz is a Senior Research Associate at the Center for Integrated Facility Engineering, Department of Civil Engineering, Stanford University, where he teaches and does research on nonnumeric symbolic modeling in engineering. He was the IntelliCorp Chief Knowledge Systems Engineer and Director of Manufacturing Applications. He has led AI systems applications development in diverse areas including experiment design, power plant control, petroleum exploration, project management, configuration, and factory scheduling. He is a member of the Editorial Board of several journals, including *AI/EDAM*. The Virtual Design Team research has extended his interest in modeling from products and processes to include social systems.