SIMULATION 64:3, 42-57

ISSN 0037-5497/94 $3.00 + .10
Printed in the United States of America

The Virtual Design Team: A Computer Simulation Framework for Studying Organizational Aspects of Concurrent Design

Yan Jin, Raymond E. Levitt and John C. Kunz
Department of Civil Engineering
Stanford University
Terman Engineering Center Room 297
Stanford, CA 94305-4020
Phone: (415)723-2918
Fax: (415)725-8662
Email: jin@cive.stanford.edu

Tore R. Christiansen
Det Norske Veritas Research AS,
Vertasveien 1, N-1322
Hovik, Norway

Successful concurrent design requires a well organized design team. Without analysis tools for organization design, managers must rely on their experience or trial-and-error, rather on systematic generation and evaluation of alternatives, to design their organizations. The goal of the Virtual Design Team (VDT) research project is to develop computerized analysis tools to support the systematic design of organization structures for complex, project-oriented tasks. The Virtual Design Team is a computational discrete event simulation model that incorporates qualitative reasoning concepts derived from Artificial Intelligence research. VDT explicitly incorporates information processing and communication models from organization theory and allows qualitative predictions of organizational performance. VDT's behavior has been validated extensively for internal consistency. We have also validated that its behavior compares well with theoretical predictions about, and the observed behavior of, real design project teams for petrochemical refinery, offshore oil systems, and power plant construction projects.

Keywords Organization design, simulation concurrent & engineering coordination.

1. Introduction

Concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including 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 (Carter, and Baker, 1992). In a typical concurrent design project, multiple design activities are carried out in parallel. Designers responsible for the activities usually need information from each other, and they must make proper decisions when there is any conflict among their designs. To make concurrent design successful one needs adequate design tools, an integrated framework, and a well-organized design team.

Research on concurrent engineering to date has focused on developing communication infrastructure, design tools, and product data models. 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 (Karandikar, 1993). A successful implementation of concurrent engineering requires that these issues be identified and solved up front.

Our research on the Virtual Design Team attempts to develop computerized analysis tools to support the systematic design of organization structures for complex, project-oriented concurrent design tasks. We have developed a computer software system, called VDT, through operationalizing and extending extant organization theory. VDT is a computational discrete event simulation system. It explicitly incorporates information processing and communication models from organization theory. The input to VDT is the descriptions of design activities, actors (i.e., designers, managers and subteams), 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 activities (a surrogate for total labor cost of design), the duration to complete the entire design project, and verification and coordination quality.

2. Simulating Organizational Behavior of Design Teams

Design of artifacts to meet human needs—whether physical artifacts such as buildings, or social artifacts such as business organizations—is a ubiquitous human activity. It can be broken down into the following generic steps: *requirements 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.

Engineering disciplines have long had mathematical models and, more recently, numerical computational models, to support analysis and optimization of physical systems. In contrast, the use of computers to support analysis in the design of social systems has been very limited because most organizational behaviors of interest to scientists or managers can only be represented as discrete, nominal, or ordinal variables, leading to a mismatch between the qualitative content of organization theories and quantitative analysis capabilities of the traditional computer techniques. Fortunately, the advances in symbolic programming and artificial intelligence have opened the way to use computers to model non-numerical variables. Recently, there has been a small but growing move toward computational organization design in the organization research community (Masuch & LaPotin, 1989; Carley et al., 1992; Carley & Prietula, 1994; Ingemar, 1994).

2.1 Approaches to Organization Simulation

From a computational modeling perspective, there are several ways to simulate organizational behavior of an engineering project or firm, namely, mathematical, heuristics-based, and model-based.

In the mathematical approach to organizational simulation, an investigator models organizations using mathematical equations. A specific organization situation is represented by setting values of the parameters of the equations. This approach has been explored since the early 1960s and influenced by the contributions of the cybernetics, general systems, and system analysis movements; organizational formalisms; and the efforts from dynamic systems. Cyert and March's *A Behavior Theory of the Firm* (Cyert & March, 1963) is a classic example of this kind of simulation model.

The *heuristics-based* approach attempts to bring together the causal links between the parameters used to describe an organization in a comprehensive way. When a specific organizational situation is given to a heuristics-based simulation system, the system can infer how the organization will behave, and what kind of organization structure may improve the organization's behavior, using a set of predefined rules. The theoretical basis for this kind of system is the contingency view of organizational theory (Thompson, 1967; Galbraith, 1977). From an organization design perspective, contingency theory suggests that an appropriate organizational structure is contingent or dependent upon such factors as size, strategy, technology, technologies, and leadership. There is a large literature of contingency theory, and the heuristics-based approach tries to integrate the different views into a comprehensive one to support organization design. For example, ORGCON is an expert system that helps a designer analyze an organization and its structure (Burton & Obel, 1993). Given the contingency factors such as size, technology and strategy of the organization, ORGCON suggests desirable organization structures.

Both mathematical and heuristic approaches treat the aggregate organization as the unit of analysis in their models. From an organization design point of view, the problem with these approaches is that they can make only predictions about aggregate behaviors of organizations, treating environmental constraints and contingencies like point loads at the center of mass of an organization. They do not consider organizational components.

To investigate the contingency of organization design upon micro-level (e.g., actor level, technologies or tool level) factors, one needs to focus on the actors in organizations. The model-based approach described below addresses this micro-level contingency focus.

2.2 Simulating Actions of and Interactions among Actors

The basic premise of the *model-based approach* is that "organizations don't make decisions, people do." Actors involved in an organization are the engine that drives behavior of the organization. From an organization modeling perspective, this approach allows investigation of how the effect of organizational structures is contingent upon such micro-level factors as distribution of individuals' capabilities, use of communication tools, complexity of certain specific tasks, and task and actor interactions. The theoretical basis of this approach is related to the growing awareness among organizational theorists that people, tasks, and the sociostructural situations defining people's interaction networks do matter (Carley & Prietula, 1994). On the other hand, the advances in logic and symbolic programming, artificial intelligence, and network analysis provide important methodologies for the approach. In a model-based organization model, the organization structure constrains, or empowers, the behavior of actors. Organizational behavior emerges from the actions of, and the interactions among, the actors.

Figure 1 shows an abstract view of relationships among organizational constraints (e.g., organization structure), actors' behaviors and organizational performance. Since the actors of an organization are the engine that produces the organization's behavior, the actor model becomes a crucial part of an organization model. Depending on what issues an organization model addresses, one may model actors at different behavioral levels, as shown in Figure 1 (we assume that a model that adopts the lower level behaviors also includes all the higher level ones). At the first (top) level, an actor's behavior is directly controlled by a set of external attributes of which the values are specified by simulation designers. An actor's behavior captured at this level is "linear" in the sense that the actor's certain behavior (e.g., rate of making mistakes) at a certain time (e.g., now) does not affect, nor is affected by, any of the actor's other behavior (e.g., preference to talk or not to talk to others) at any other time (e.g., tomorrow). Although the behavior of a single actor is relatively simple at this level, in the absence of simulation, the impact of organization design on the organization performance will not be straightforward if many actors depend on each other in some complex way.

As shown in Figure 1, the indirect behavior of an actor can be defined as dynamic change of some behavior or behaviors of the actor caused by some other direct or indirect behaviors of the actor. For example, an actor that prefers to go to meetings may automatically reduce its probability of making coordination mistakes. This kind of behavior is viewed "non-linear" in the sense that an actor's certain behavior can be influenced by some other behaviors dynamically. Introducing the indirect actor behaviors provides the constructs to model trade-off between actor behaviors and consequently makes it possible to simulate the indirect impact of certain organization constraints.

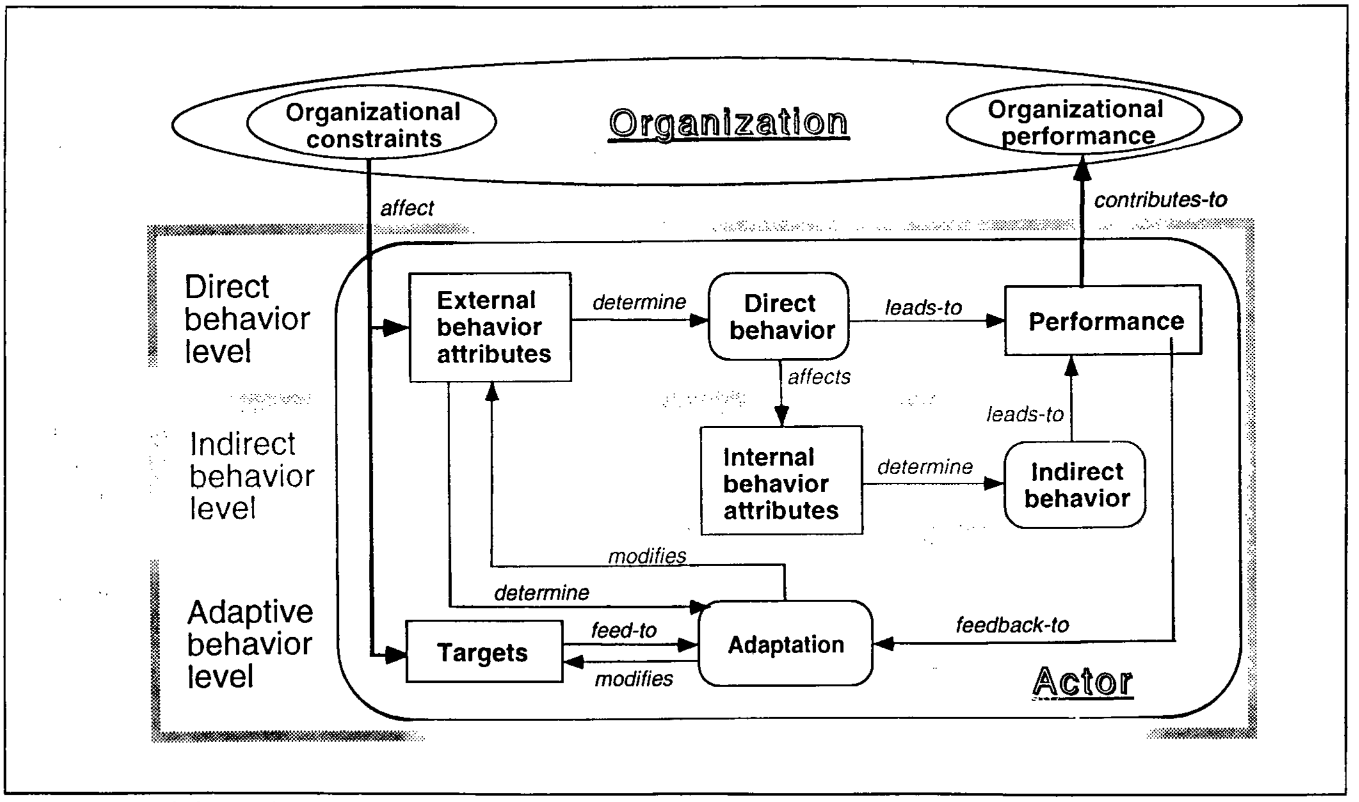


Figure 1. Actor behavior levels for organization simulation.

The actor model at the third (bottom) level includes not only the interactions among the behaviors of different types but also those among behaviors at different times. In this model, an actor not only behaves but observes the performance of its behaviors and adapts itself based on the difference between its observed performance and target. For example, if an actor finds that going to meetings reduces its coordination mistakes, it may adapt its not-prefer-meeting behavior into prefer-meeting behavior. If going to too many meetings takes too much working time, the actor may change its prefer-meetings behavior back. An actor may also choose to lower or raise its target to reduce the target-performance difference. The adaptive model of actors incorporates actors' experience-based learning behavior and provides the basic constructs required to address the impact of organizational reward systems and organizational change.

To build an organization simulation model, the behaviors illustrated in Figure 1 must be grounded on some specific context. In our study, this context is defined by a set of design tasks (e.g., building design) and design environment (e.g., communication tools). From a computational point of view, modeling actor behaviors in a certain context may involve complex descriptions and time-consuming computation.

2.3 Dealing with Complexities

Human organizations involve complex intelligent actors who work on simple or complex tasks. Complexity is a crucial issue for organization modeling and simulation. 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 (such as Blocksworld) 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. If the purpose is to generate some general organization theory, a simple hypothetical task may be good enough. If the purpose is to develop an analysis tool for engineering organization design, then the corresponding specific and complex engineering domain must be considered.

One way to deal with task complexity is abstraction. Generally speaking, less abstract (or close to real) task descriptions often set up requirements and constraints of the tasks. The actors working on the tasks must infer detailed actions to accomplish the tasks through knowledge-based reasoning or problem solving. More abstract task descriptions, on the other hand, describe tasks in terms of time and resource requirements and the detail of "how" is out of scope. Abstract task models can be applied to routine tasks for which an activity precedence network can be predefined, required time and resources can be pre-specified, and how actors perform their tasks is not important.

Actor Complexity: Human actors are complex, and it is difficult to construct a model that sufficiently, coherently and mechanistically describes their behavior. From our experience of organization modeling we found that there are two ways to reduce the complexity. The first is to choose an adequate behavior level as shown in Figure 1. The lower level models are more complex than the higher level ones. If the purpose of simulation is to analyze how coordination among actors working on relatively routine tasks may impact on organizational performance, then a level two model should be appropriate. If the purpose is to simulate interplay between technologies and learning organization, then the adaptive model will be used.

The second way 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 (see Section 4.1) to treating actors as intelligent agents (Jin & 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 may be useful, though it is complex. Our experience 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 results.

3. Organizational Issues in Concurrent Design

As described above, many decisions for building an organization simulation model depend on the purpose of simulation. To present the VDT model, we first discuss the organizational issues involved in concurrent design.

Successful implementation of concurrent design depends on project requirements, team organization, task arrangements, and technology (or tools) available to the team. In concurrent design, the activities are carried out in parallel. As a result, coordination among the actors who are responsible for the parallel activities becomes a critical engineering design process. Better team organization facilitates communication and information sharing among team members and leads to more 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 meetings? Should team members meet formally or talk to each other informally whenever necessary?

Technology or tools: What tools should be used for communication? Is it helpful to introduce new communication tools such as voice mail, e-mail, and video conference? Is it necessary to introduce new CAD tools? What attributes of tools make them useful?

Task arrangement: How should tasks be arranged—more concurrently or more sequentially? What are the predictable consequences 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 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 obvious. Organization theory can provide aggregated and qualitative answers, but not detailed prescriptions. Concurrent design management experts—of whom there are few—address the questions based on their experience. Our simulation model of concurrent design attempts to predict how changes in organization design may impact on team behavior and performance at the level of detail addressed by the above questions.

4. Organization Theory Concepts in VDT

4.1 An Information Processing Model of Design Teams

We have chosen to model concurrent design teams that work on project-oriented and routine design tasks. This choice allows high level abstraction in modeling both task and actor behavior contents. The basic premise of the VDT model is that organizations are fundamen-

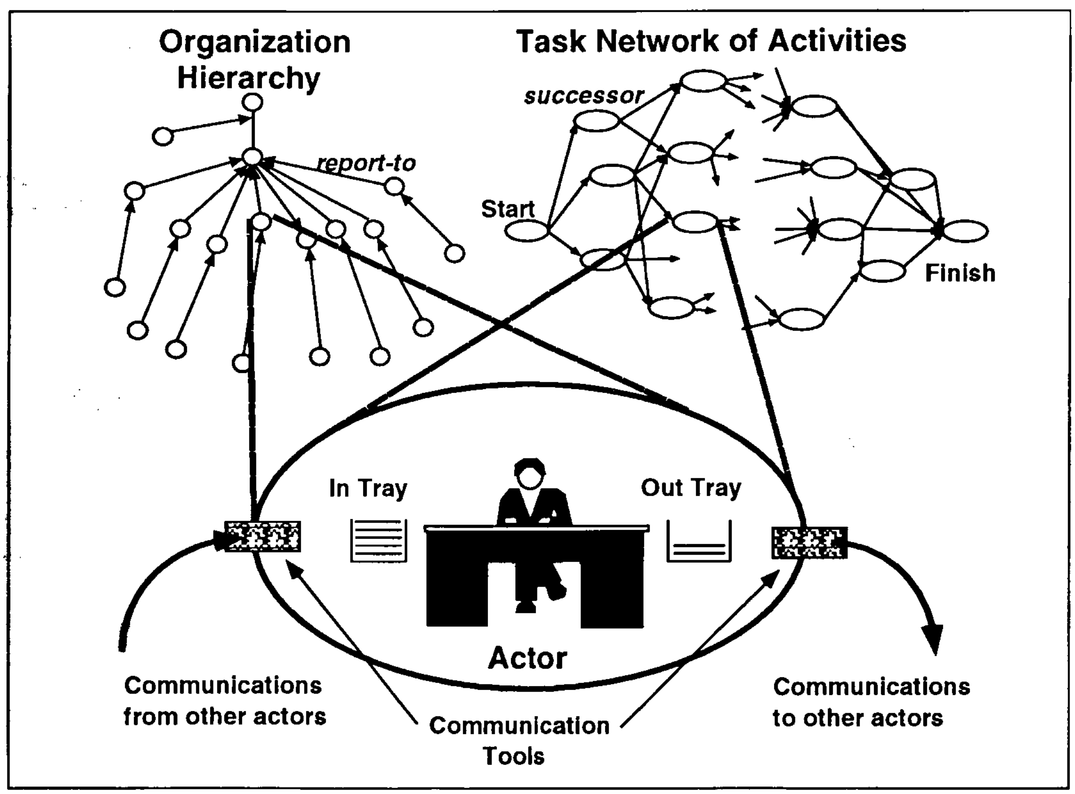


Figure 2. Overview of the Virtual Design Team.

tally 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, (1958), Simon (1976), and 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 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). To capture these characteristics and constraints, VDT employs explicit descriptions of activities, communications, actors, communications tools, and organization structures. Figure 2 illustrates the way that organizations are implemented in VDT.

4.2 Activities and Subtasks

Our goal is to analyze engineering design teams carrying out routine design tasks. We, therefore, view the task of the design team as the completion of a set of pre-determined activities. These activities consist of the design, review, and approval of a series of components or sub-systems of the artifact to be designed. For instance, in the case of a refinery, the activities include chemical process design, piping design, and structural design. Each activity involves processing an amount of information defined as the magnitude of the activity, communication of information among design team participants, and craft requirements. 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.

In life and in VDT, actors allocate their attention and time among their activities and the multiple communications they have received. In order to model as little detail as possible about activities but still predict performance, we model information processing requirement of activities in terms of *work volume* and *work type*. Work volume is the time needed for an actor with average skill to accomplish the activity. Work type is the specialized skill or "craft" an actor must possess to carry out the activity effectively.

Contingency theorists have characterized tasks in terms of *complexity* and *uncertainty* (Galbraith, 1977; Thompson, 1967). 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 responsible actors.

An activity in VDT is composed of a number of subtasks. A subtask contains a certain amount of work volume. A subtask is a work item that can be passed into an actor's in-tray, picked up and processed by the actor. A subtask is the minimum amount of work that can fail. For example, a structural design activity may contain 20 one-day subtasks. The design actor responsible for the activity verifies its design at the end of each day. The result of verification can be "failed" based on the activity's verfication failure probability attribute. VDT determines an activity's *verification failure probability* based on the activity's complexity and the match between the activity's type (e.g., civil engineering work) and responsible actor's skill (e.g., civil engineering).

Interactions among actors affect organizational performance of concurrent design teams. Dependency relationships among activities require responsible actors to interact and coordinate with each other. The more concurrent the design activities are, the more coordination will be required among actors. Radical concurrency may actually cause the design to take longer due to the overwhelming requirement for coordination among actors. A key goal of the VDT simulation is to let the simulation infer coordination requirements, generate coordination tasks dynamically, 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 includes information exchange *communication intensity* (CMI) as an attribute of activities. The value of the attribute is derived from each activity's uncertainty level.

4.3 Communications

A communication in VDT is an elementary packet of information sent from one actor through a specified channel to another actor, using a single communication tool. Completion of each activity involves processing the number of communications specified by the activity's magnitude. Each communication has attributes of: time stamp, author, recipient, work volume, type, distribution list, ranking of natural idioms, variability of the associated task, and priority.

Communication work volume indicates how much time it will take to process the communication. 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 subtask described in Section 4.2. 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 subtask failure, it generates a failure exception communication and sends the communication, together with the failed subtask, to a decision-maker for a decision on how to deal with the failure.

Decision: When a decision-making actor receives a failure exception, it will take time to process the exception and make a decision stochastically whether the failed task should be reworked, or not. When a decision is made, the decision-maker then 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 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 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 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 actor's in-tray.

4.4 Actors

Actors include managers and design subteams from various disciplines, such as electrical, and mechanical engineering. The actor description includes role characteristics, such as position in the team hierarchy; authority for design, approval and coordination tasks; and allowed communication patterns (either strictly hierarchical or allowing peer-to-peer contact). The actor description also includes individual attributes, such as craft and skill (e.g., high skill in mechanical engineering); task experience (high, medium or low) and the natural idioms of communications that the actor processes most effectively (e.g., words, schematics, plans). Because we are modeling institutionalized actors working on routine design tasks, the VDT actor model is an abstract indirect behavior model, as shown in Figure 1.

In VDT, an actor selects one item from its in-tray and then performs several actions. The series of actions for one item selection is called an "action cycle". Different action cycles may go through different actions depending on what item is selected, as shown in Figure 3. Generally, actors execute the following actions during an action cycle.

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 (92), based on his observations, was that actors use priority about 50 percent 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 percent of the time; the most recent item in the in-tray or LIFO is used 20 percent of the time; and the actor chooses items randomly 10 percent of the time.

Process information: After selecting a subtask or communication item from the in-tray, an actor calculates and consumes the time required to process it based on the actor's information processing speed and the work volume of the subtask. During the time when an actor is processing a subtask or communication, an incoming communication from other actors may arrive at any time. The actor applies an attention allocation rule

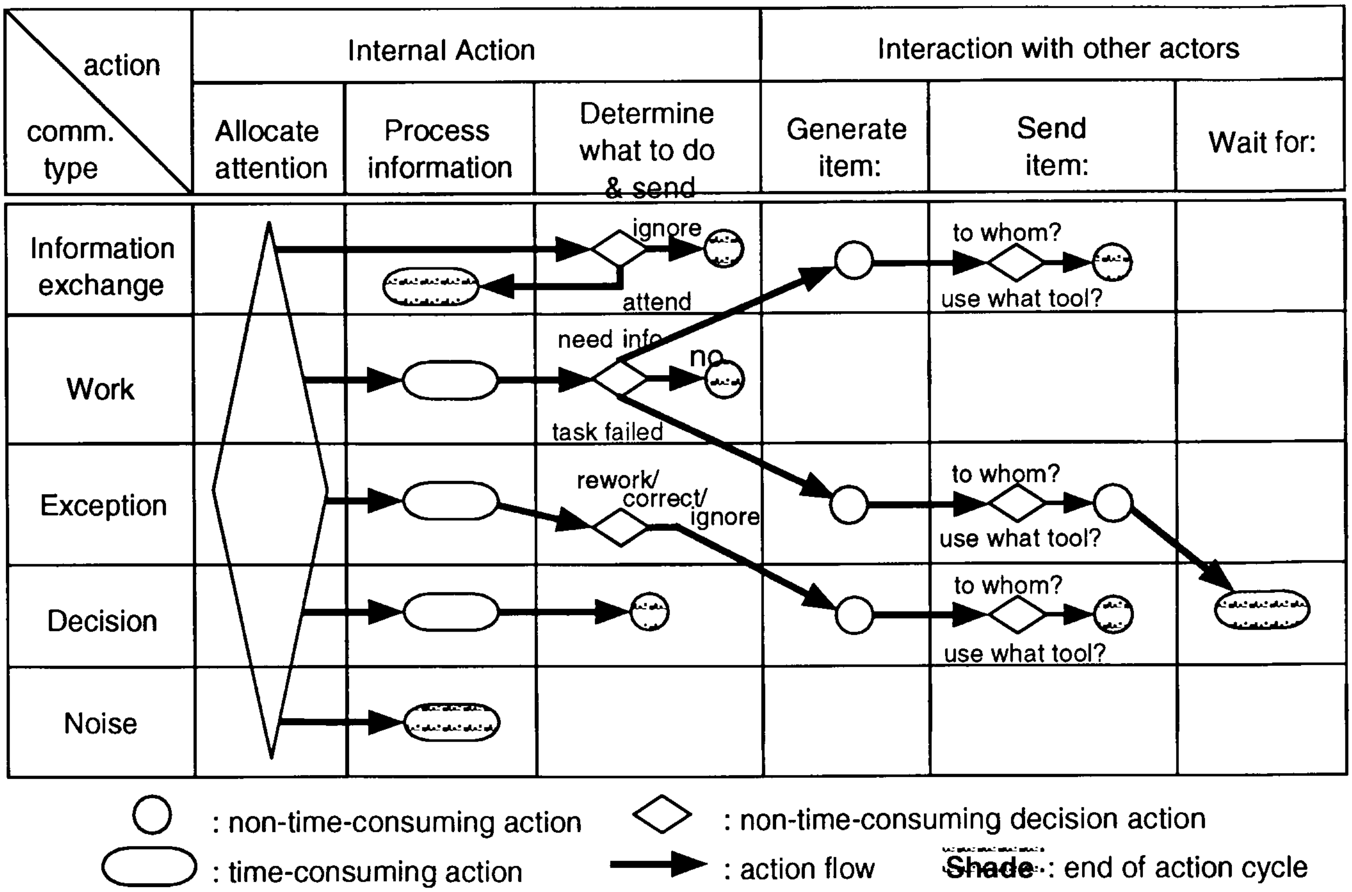


Figure 3. Actor behaviors modeled in VDT.

stochastically to determine whether to stop processing the current subtask to attend to the new subtask or communication.

Determine what to do and send communications: When finishing a subtask item, an actor verifies the subtask and decides stochastically whether there is a subtask failure and whether information from others is needed. If a need for information exchange with others is identified, the actor generates an information exchange item and sends it to the relevant actors. If a task failure is identified, the actor generates a subtask exception item. After determining the exception recipient based on the organization structure (i.e., who reports to whom) and decision-making policy (i.e., what decisions should be made at which level of organization hierarchy), the actor sends the exception communication to the decision-maker, and then awaits the decision. If the actor does not receive a decision from the decision-maker within a certain amount of time, the actor will assume "delegation by default" and make a decision by itself. When a manager finishes processing an exception item, it decides whether the failed task should be completely reworked, partially corrected, or ignored. The decision is then sent to the exception reporting actor.

4.5 Tools

Each communication is transmitted via a communication tool selected by an actor. The VDT framework represents tool attributes that are theorized to affect both the choice of tool and the results of that choice. The adoption and behavior of tools is then defined in terms of the relationships among the tool variables and the characteristics of the task, actors and 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; 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.

4.6 Organization Structure

One of the fundamental questions for organizational modeling is to determine what changes when an organization's structure changes, and how the change affects the organization's performance? Organization performance in VDT emerges from simulated actions of, and interactions among, actors. In VDT, organization structure affects organizational performance by enforcing behavioral constraints on individual actors.

VDT differentiates formal control structure and information communication structure.

A *formal control structure* is a hierarchy of *reporting-to* (or *supervised*) relationships between actors and has a certain level of *centralization. Reporting-to* links guide actors to determine with whom they should communicate when a task fails. 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, most decisions are made by project managers. Thus when an engineer actor detects an exception, the actor reports the exception to the sub-team leader. The sub-team 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 sub-team leaders or even by engineers themselves. Therefore, in decentralized organizations, fewer communications are sent to and processed by high-level managers, reducing both the need for communication and the need for information processing.

An informal communication structure is defined by *coordinate-with* relationships among actors and has a certain level of organization *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 inter-actor information exchange. Organization "matrix strength" (Davis & Lawrence, 1977) – the level of multiple control over organization members-also affects the strength of communication structure. Since actors in "weak matrix" organizations are often not co-located, they tend to use informal inter-actor communications less often, relying on formal meetings. We also call matrix strength *organization culture* since it reflects actors' informal social relationships.

5. The VDT Simulation Environment

VDT operationalizes Galbraith's information processing model of organizations (Galbraith, 1977) by explicitly representing attention allocation capabilities of actors and task requirement of activities. The VDT simulation environment manipulates the discrete events associated with the generations of subtasks and communications and the start and finish of activities, as shown in Figure 2. 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 Microsystems 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.

5.1 How VDT Works

System Initialization: A simulation starts from system initialization. Based on user inputs, the initialization process sets up initial values of the key intermediate variables including actor processing speed, activity verification failure probability (VFP), and activity communication intensity. For example, the user specifies the actors' ability (craft, skill and task-experience) and the craft requirements of the activities for which the actors are responsible. Based on the degree of match between these two input values, VDT determines the processing speed of the actor. Similarly, activity verification failure probability is determined based on each activity's complexity, its responsible actor's capability, and the degree of match between the activity's craft requirement and the actor's craft. Communication intensity is decided based on the activity's uncertainty.

Attention allocation and task processing: Tasks including design tasks and communications arrive at the in-tray of an actor and await processing. Actors allocate their attention to incoming subtasks based on their attention rules described above. After a subtask item is selected from the in-tray, an actor calculates the time requirement for the subtask processing based on its processing speed and the work volume of the subtask;

SIMLIB then advances the clock based on the calculated duration. While processing a subtask item, an actor may be interrupted by an incoming communication from other actors. The actor chooses whether to stop the current subtask processing to do the new task depending on the priority of the interrupting communication.

Exception processing and decision-making: After a subtask is finished an actor verifies the result probabilistically to see if the task processing succeeds. The verification failure probability of the activity determines the probability with which a task verification may fail. If a subtask fails, then the responsible actor generates an exception. After determining who should make the decision about the exception (based on the project policy on centralization), the actor sends the exception to the decision maker and then waits for a decision. Upon receiving the exception, the decision-making actor decides whether the failed subtask should be reworked, corrected or ignored. Once a decision is made, it is sent back to the exception generator which then follows the decision to rework, correct or ignore the failed task. If the exception generator does not get a decision back from the decision-maker within a given time for any reason, it will then follow the "delegation by default" rule, ignoring the failure and continuing to work on the next task.

Meeting and information exchange: Communications among actors include formal meetings and informal information exchanges. Meeting schedules are set up deterministically based on input data. There can be multiple meetings in a single project and different meetings may have different participants. Information exchange intensity is derived from the activity uncertainty, and from interdependencies among the activities. When an actor receives a meeting notice or an information exchange communication, it chooses whether to attend the meeting or information exchange. Actors' preference among meetings vs. information exchange depends on the project culture (weak vs. strong matrix). Although attending meetings and information exchange takes time, nonattendance to both meetings and information exchange will negatively affect the project's coordination quality. The higher the frequency of nonattendance, the worse coordination quality is.

```
(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]
)
```

Figure 4. Part of an OPDL program.

5.2 The OPDL Language and Graphical User 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 organizational behavior and performance of teams working on engineering projects. Using OPDL, a user can program project activities, project policy, actors and organizations. VDT reads OPDL files and then simulates the project's performance. OPDL is not only an interface to VDT but has been designed as a more general language for formal description of organizations and projects. Figure 4 shows part of an OPDL program.

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

5.3 From Real Project to VDT - The Load Model

VDT's activities or tasks are described in terms of complexity, uncertainty and interdependency. Therefore, in order to simulate a real engineering project in VDT, one must derive these task properties from the real project data. In our research, we developed a coordination load model that describes real projects in VDT terms and maps the real project into an input file in the VDT simulation environment (Christiansen, 1993). This model uses Quality Function Deployment (QFD) (Hauser & Clausing, 1988) and Design Structure Matrix (DSM) (Gebale & Eppinger, 1991) to derive interactions between requirements and engineering solutions, dependence among design activities in an activity precedence network, and relations between members of the project team. QFD is used to predict the required frequency and nature of verification and communication in the design process. A detailed description of the process of modeling coordination load can be found in (Christiansen, 1993). Figure 5 shows an overview of this model.

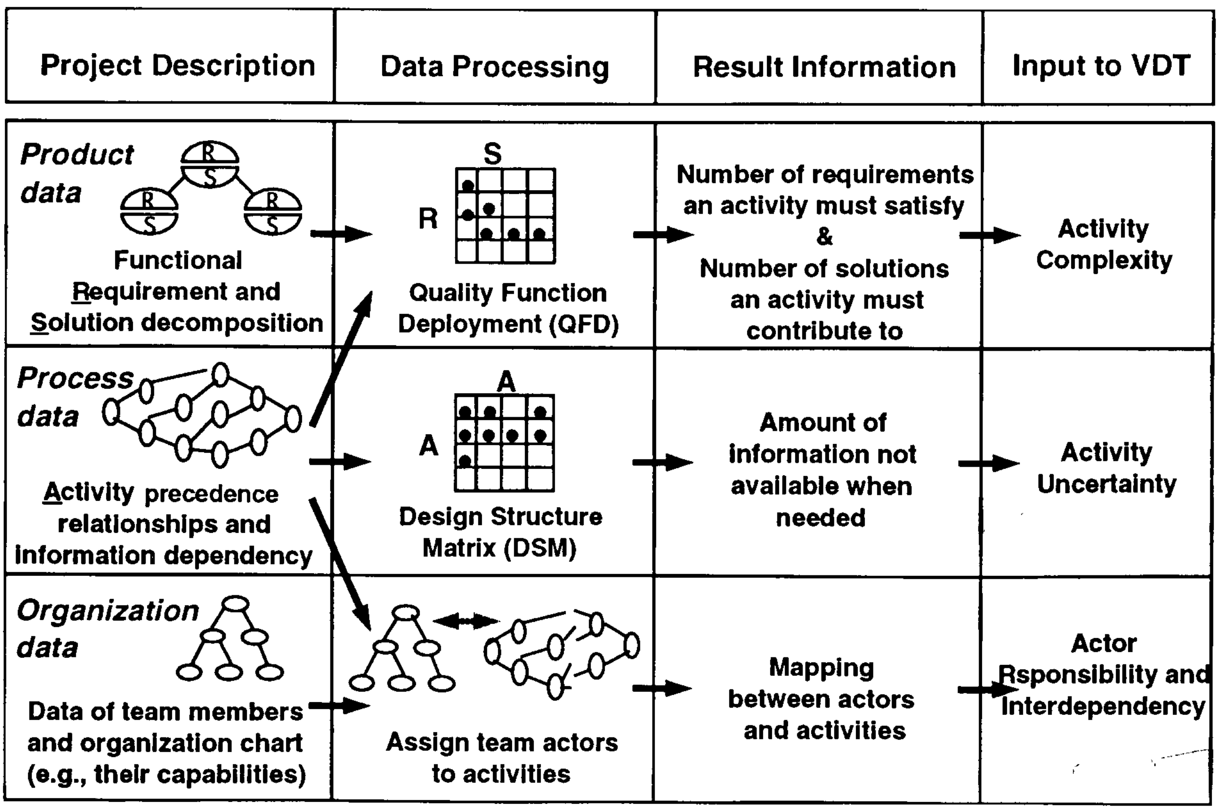


Figure 5. Process of transforming real project data descriptions into VDT inputs

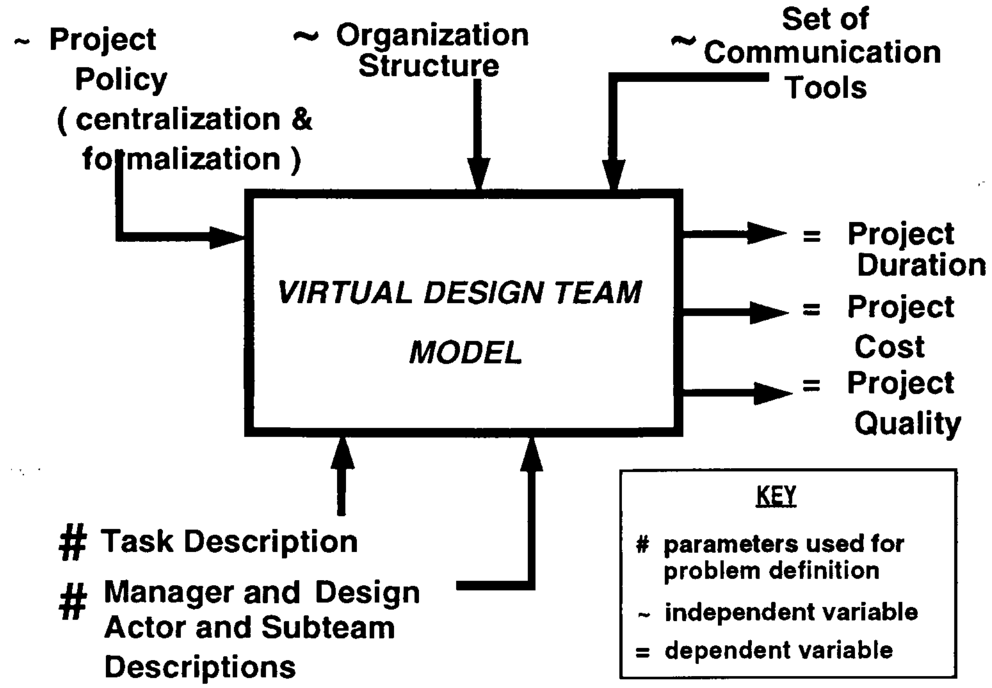


Figure 6. The function of the Virtual Design Team.

5.4 Output From VDT

The output from VDT includes project duration, total cost, and project quality measures such as verification quality (IgnoredExceptions / TotalExceptions), schedule quality ((ActualDuration - ScheduledDuration) / Scheduled Duration), coordination quality (NonAttended Communication / TotalCommunication), and budget quality (ReworkVolume / WorkVolume). Besides the performance results, VDT also records dynamic behavioral data of actors such as the number of items in an actor's in-tray at each time, time spent waiting for decision etc., and progress data of activities, such as work completed, amount of rework, etc.

6. VDT Validation

Figure 6 illustrates how the VDT produces a set of project level performance measures (dependent variables), using two dimensions of project decision making policy, organization structure, communication tools (independent variables), and a description of team actors, and project activities (state description). Project centralization policy determines the probability of how "high up in the hierarchy" decisions on how to deal with exceptions are made. For changes in centralization, the VDT simulation will give predictions about changes in project duration, cost and effectiveness of coordination (verification and communication quality). Similarly, project formalization determines the degree to which project communication is made up of formal meetings vs. informal information exchanges. Different types of project organizations, that is organizations with different "matrix strength", will give different priority to formal vs. informal communication. For a project with given matrix strength, the VDT simulation will predict attendance (due to decision making about whether or not to participate in communication), as a function of formalization. Communication attendance is another aspect of the effectiveness of coordination (communication quality), which thus depends on the fit between the matrix strength of the organization and the formality of communication. The state description variables are set up to model the particular project under study and 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.

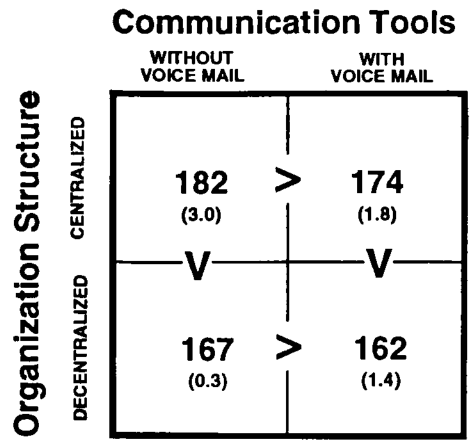


Figure 7. Impact of communication tools and organization structures on project duration (Cohen, 1992).

We validated the simulation model by carefully observing three separate industrial projects. In each case, we designed a set of experiments in which we varied one or two independent variables and fixed the others at typical values, usually "medium." To average the stochastic simulation behavior, we ran three to five simulations for each scenario with different random number seeds and took the mean values of dependent variables as the results of that scenario. Significance levels of results for each validation case were analyzed with standard statistical techniques.

Figure 7 shows the simulation results of change of duration of a three-year, petroleum refinery design project in response to the change in communication tools and organization structures (Cohen, 1992). 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 3 runs) of project duration, in working days. Standard deviation is the number shown in parentheses. The ">" indicates prediction of theory, e.g., that the mean project duration of a centralized project without voice mail will exceed that of a centralized organization with voice mail.

Figure 8 illustrates the VDT simulated effect of centralization on the duration of a subsea oil module engineering design project, together with the prediction from the project manager and the qualitative prediction from contingency theory. The expected behavior from the contingency theory is based on the assumption that higher level managers have a more global view and tend to make better (i.e., rework) decisions; but their delayed decisions lead to waiting subteams.

The prediction from the project manager matches the theoretical prediction, and the prediction from simulation is qualitatively consistent. The quantitative correlation between the simulation prediction (a total increase of 4 % in duration between lower and higher centralization) and the project manager's prediction (total increase of 17 %) is of the right scale, and thus acceptable.

7. Summary and Future Work

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 have not yet calibrated the quantitative predictions of the VDT simulation model.

We plan to extend VDT in three respects. First, we will continue to validate and calibrate VDT. We are offering a course at Stanford, CE251 – 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.

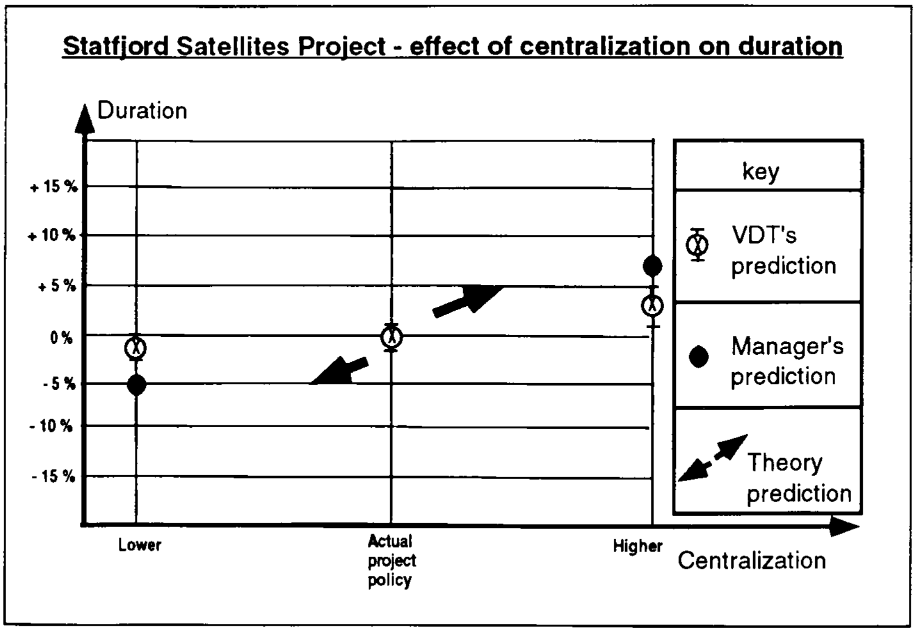


Figure 8. Impact of change of centralization on project duration (Christiansen, 1993).

Second, VDT has been developed in the facility engineering domain. We plan to use VDT to model project teams in other domains such as 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' direct and indirect behavior are shown in Figure 1. Our ongoing work tries to model adaptive behavior of actors. By doing so, we expect to be able to observe emergent organizational change in response to introduction of new technologies into a project team or an engineering firm. The current version of VDT can simulate only one organization at a time and the influence from other organizations is simulated in terms of noise. We plan to make VDT capable of explicitly simulating multiple projects so that we can study inter-organization issues using VDT.

References

Burton, M. R. and B. Obel. *Organizational Consultant - An Expert System Version 5.0.* Unpublished system manual. 1993.

Carley, K. and M. Prietula (Eds.), *Computational Organization Theory*, Lawrence Erlbaum Associates, NJ, 1994.

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

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

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

Cohen, G. P., *The Virtual Design Team: An Information Processing Model of the Design Team Management,* Unpublished PhD Thesis, Stanford University, December (1992).

Cyert, R. M., and J. G. March, *A Behavioral Theory of the Firm*, Prentice-Hall, Englewood Cliffs, NJ; 1963.

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

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

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

Hauser, J. & D. Clausing, "The Houseof Quality" *Harvard Business Review*, May-June (1988).

Ingemar, H. (Eds) *Proceedings of AAAI '94 Spring Symposium on Computational Organization Design*, Stanford, CA, 1994.

Jin, Y. and R.E. Levitt, "i-AGENTS: Modeling Organizational Problem Solving in Multiagent Teams", in *International Journal of Intelligent Systems in Accounting, Finance and Management*, Vol.2, No.4 pp.247-270, December 1993.

Karandikar, H.M., et al. "Assessing organizational readiness for implementing concurrent engineering practices and collaborative technologies." *Proceedings of Second Workshop on Enabling Technologies Infrastructure for Collaborative Enterprise*, IEEE Comput. Soc. Press, 1993. pp. 83-93.

Levitt, Raymond E., Yan Jin and Clive L. Dym, "Knowledge-Based Support for Management of Concurrent, Multidisciplinary Design," *Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, (5) 2, 1991, pp. 77-95.

March, J. G., and H. A. Simon, *Organizations*, John Wiley, NY; (1958).

March, J.G. *Decisions and Organizations*, Basil Blackwell; Oxford, UK, (1988).

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

Mintzberg, H., *Structuring in Fives: Designing Effective Organizations*, Prentice-Hall, Englewood Cliffs, NJ; (1983).

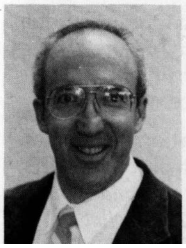
Mintzberg, H., *The Structuring of Organizations*, Prentice-Hall, Englewood Cliffs, NJ; (1979).

Scott, W. R., *Organizations: Rational, Natural, and Open Systems*, 3rd ed., Prentice Hall, NJ; (1992).

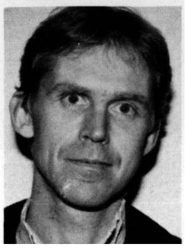
Simon, H. A., *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization*, Free Press, NY; (1976).

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

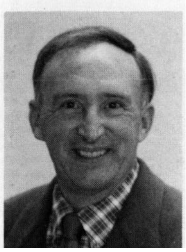
YAN JIN is a Research Associate at Department of Civil Engineering, Stanford University. He earned the PhD degree in Naval Engineering from University of Tokyo in 1988. Since then Dr. Jin has been doing research on knowledge based planning systems, distributed problem solving and multiagent systems, organization modeling, and their application 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 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 PhD 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. The Virtual Design Team research in which Dr. Levitt is currently engaged integrates his organization theory and artificial intelligence interests.



TORE CHRISTIANSEN is working as Principal Research Engineer in the Department for Information Systems of DNV Research, the strategic research company of Det Norske Veritas ship classification society. He studied Aeronautical Engineering at the University of London and Massachusetts Institute of Technology, and has experience as a developer and user of computational mechanics programs and computer aided engineering systems. He obtained his PhD at Stanford University, Department of Civil Engineering in 1993, working as part of the VDT project team. His thesis, and current interest lie in the area of using computer tools to understand and improve the efficiency and effectiveness of project organizations.



JOHN KUNZ is Senior Research Associate at the Center for Integrated Facility Engineering in the Stanford University Department of Civil Engineering, where he teaches and does research on non-numeric symbolic modeling in Engineering. He served as the IntelliCorp Chief Knowledge Systems Engineer and Director of Manufacturing Applications. He has led AI system applications development in diverse areas including experiment design, power plant control, petroleum exploration, project management, configuration and factory scheduling. The Virtual Design Team research has extended his interest in modeling from products and processes to include social systems.