CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

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

by

Yan Jin, Raymond E. Levitt, and Tore Christiansen

CIFE Working Paper
Number 24

November, 1993

Stanford University

If you would like to contact the authors please write to:

c/o CIFE, Civil Engineering,
Stanford University,
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

1994 International Conference on Simulation in Engineering Education, January 23-27, 1994, Tempe, Arizona.

THE VIRTUAL DESIGN TEAM: A COMPUTER SIMULATION FRAMEWORK FOR STUDYING ORGANIZATIONAL ASPECTS OF CONCURRENT DESIGN

Yan Jin, Raymond E. Levitt, and Tore Christiansen

Department of Civil Engineering
Stanford University
Stanford, CA 94305-4020

ABSTRACT

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. The goal of the Virtual Design Team (VDT) research project is to develop computerized analysis tools to support the systematic design of organization structures—including the communication tools that permit data, decisions and knowledge to be shared within and between organizations—for complex, project-oriented tasks. The Virtual Design Team is a computational discrete event simulation model incorporating qualitative reasoning concepts derived from artificial intelligence research. VDT explicitly incorporates information processing and communication models from organization theory that allow qualitative predictions of organizational performance. VDT's behavior has been validated extensively for internal consistency. Its behavior also compares well with theoretical predictions about, and the observed behavior of, real project teams of petrochemical refinery, offshore oil systems, and power plant construction. In this paper we describe the theoretical basis for, and implementation of, VDT and explain how it can be used to teach future project managers about the design of organization structures.

THE GOAL OF VDT

The Virtual Design Team is a computational discrete event simulation model incorporating qualitative reasoning concepts derived from artificial intelligence research. The goal of the Virtual Design Team research project is to develop computerized analysis tools to support the systematic design of organization structures—including the communication tools that permit data, decisions and knowledge to be shared within and between organizations—for complex, project-oriented tasks.

THE OVERVIEW OF VDT

VDT explicitly incorporates information processing and communication models from organization theory that allow qualitative predictions of organizational performance. The inputs to VDT are: a description of the design task, including project policy, and the subtasks that comprise it, including sequential dependencies between subtasks; a description of the actors in the design team and of their organizational structure; and a listing of 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 subtasks, 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.

The simulation model can serve as a facility to formulate and test specific conjectures regarding the qualitative effect on project cost and duration of changes in the organization structure of the team, or in the communications tools available to participants. Engineering disciplines have long had mathematical models and, more recently, numerical computational models, to support analysis and optimization of physical systems. This work provides initial evidence that symbolic computer modeling can be used to express and test social science theories applied to real world organizations and the communication tools that they employ.

ORGANIZATION THEORY CONCEPTS IN VDT

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, 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 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 tasks, communications, actors, tools, and structures. Figure 1 illustrates our view of organizations implemented in VDT.

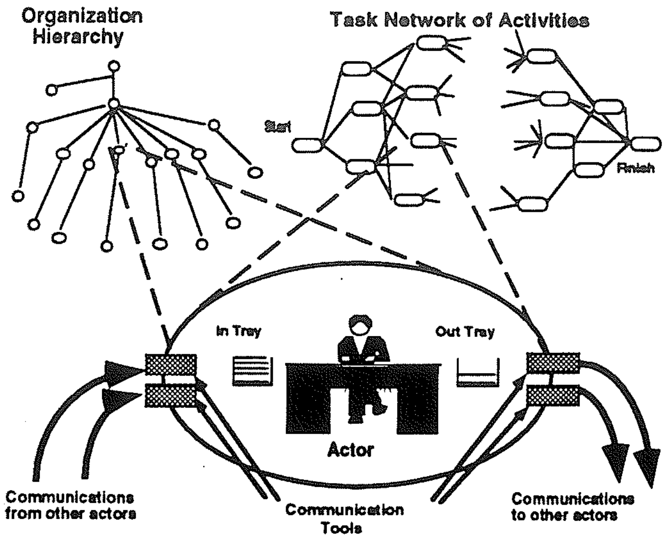

Figure 1: Overview of the Virtual Design Team.
VDT models the design task, actors, organization structure, communication tools, and project policy. The design task is broken down into a precedence network of activities. Actors are information processors with skills and attention allocation rules for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

Task

Our goal is to analyze engineering design teams carrying out routine designs. 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 of an amount of information defined as the magnitude of the activity, communication of information between and among design team participants, and craft requirements. Activities are also characterized by their complexity and uncertainty. The more complex and uncertain an activity is, the more likely the it is that exceptions (such as failures) may occur during task processing. The activities are modeled as being either reciprocally interdependent, i.e., a failure occurring in one activity may affect the processing of another activity, or sequentially interdependent, i.e., the output of a given task is the input for a succeeding task (Thompson 1967).

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, distribution list, ranking of natural idioms, variability of the associated task, and priority.

VDT represents four basic types of communications: design communications, exceptions, decisions and noise. Design communications carry the information required to perform the specified design activities. Exceptions (Galbraith 1977) are generated when there is a stochastic task failure or the need to review the task. Decisions are made by managers in response to exceptions referred to them by a subordinate. The decision is sent back to the subordinate who submitted the exception. 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.

Actors

Actors include managers and design subteams from various disciplines, such as electrical, process, 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).

Actors execute the following behaviors:

1. *Allocate attention*: Managers have limited time and attention to allocate to both routine activities and exceptions, and limited information with which to determine the importance of communications; that is, managers are boundedly rational (Simon 1976). Managers in the VDT have an "in-tray"—a metaphorical, dynamic queue of communications through which any attempt to communicate with the manager must pass. Managers must decide which communications in the in-tray should be addressed. According to our limited field observations (Cohen 92), design managers appear to evidence a characteristic pattern in the way that they pay attention to waiting messages. Items from each manager's in-tray are selected stochastically, based on these criteria, shown in Table 1.

2. *Process information*: Since the content of activities is not modeled in VDT, the time spent on information processing becomes the only measurement. VDT simulates the time to process a communication or a task based on the work volume, the match between the capabilities of an actor and the requirements of the design task, failures and the interruptions occurred during the processing.

Table 1: Manager's attention rules for selecting items in the in-tray

Attention Rule	Percentage of Time Used
Sending priority alone	50%
Length of time in in-tray, weighted by priority	20%
The item lying on top of the manager's desk.	20%
Random selection of items from the in-tray	10%

3. *Send communications to other actors*: VDT actors send communications to supervisors when there is an exception in task processing, to managers of predecessor, successor or reciprocal tasks when there is a stochastic need to exchange information, and to subordinates when a decision about an exception is made. They choose a tool for sending the communication based on: message priority, primary natural idioms in message, proximity of sender to recipient, and cost, as shown in Table 2.

Tools

Each communication is transmitted via 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 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.

Table: Probabilities for stochastic selection of communication tools

Message priority **Communication tool to be used**	*High*	*Medium*	*Low*
Meeting	35%	15%	10%
Phone	35%	30%	25%
Fax	10%	15%	10%
Mail	15%	35%	53%
E-Mail	3%	3%	2%
Video	2%	2%	0%
Totals:	100%	100%	100%

Organization Structure

Structure in VDT is defined by a set of organizational relationships among actors, and levels of authority of actors in specific roles. Organizational relationships among actors delimit the channels along which tools can be used to send communications. Relationships modeled in VDT include: *supervised-by* to implement hierarchical structure; *coordinates-with* to implement lateral relations among interdependent actors; and *socializes-with* to implement informal structure. A set of project-specific coordination policies assigns decision-making authority to actors in particular roles (e.g., *design manager*) for reviews and approvals. A centralized structure is implemented by policies that require these exceptions to be resolved by high level managers (e.g., the *design manager*); decentralized structures vest this authority in lower level managers (e.g., the *sub-team managers*).

RUNNING EXPERIMENTS IN VDT

VDT can simulate changes in different aspects of project team performance, given changes in organization structure, communication tool availability and project policy, such as centralized decision making and formalized communication. The implementation of VDT is based on discrete event simulation of the design process for a given set of product requirements. Given a description of the product to be designed, the design team organization, and the design process, the simulation produces predictions of the efficiency and effectiveness of the design process. Figure 2 illustrates an overview of the function of the VDT implementation.

The overview in Figure 2 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 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.

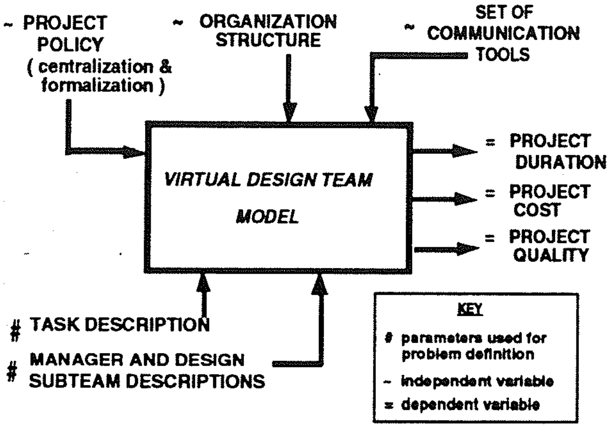

Figure 2: The function of the Virtual Design Team

VDT simulates changes in different aspects of project team performance, given changes in organization structure, communication tool availability and project policy.

VDT measures both efficiency and effectiveness as aspects of the performance of design teams. Measures of efficiency are obtained from the critical path duration and from the sum of all activity durations (the cost). Measures of effectiveness are obtained from considering how coordination items are dealt with during project execution. Coordination is modeled by *communication* and *verification*, and thus design process effectiveness is measured by the relative number of uncorrected exceptions (the verification quality) and the relative number of non-attended communications (the communication quality).

THE VDT SIMULATION ENVIRONMENT

VDT operationalizes Galbraith's information processing model of organizations by explicitly incorporating specific tasks and actors with attention allocation capabilities, and by looking into coordination issues at the micro-level in terms of explicit interaction among team participants. The VDT simulation environment can be characterized by a number of objects representing tasks, projects, actors, and organizations, and the organizational processes that facilitate coordination among team participants. Figure 1 and Figure 2 have shown the overview of VDT implementation.

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 to the in-tray of an actor, to wait for processing. Actors allocate their attention to incoming tasks based on their attention rules described above. After a task item is selected from the in-tray, an actor calculates the time requirement for the task processing based on its processing speed and the work volume of the task, and then advances the clock based on the calculated duration. While processing a task item, an actor may be interrupted by an incoming communication from other actors. In this case, the actor may or may not choose to stop the current task processing to do the new task depending on the priority of the interrupting communication.

Exception processing and decision-making: After a task is finished an actor verifies the result to see if the task processing is successful or not. The verification failure probability of the activity determines the probability with which a task verification may fail. If a task fails by Monte Carlo simulation, 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 task should be reworked, corrected or ignored. Once a decision is made, it is sent back to the exception generator who 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: Coordinations in VDT are composed of exception-decision processes described above and communications among actors, including formal meetings and informal information exchanges. Meeting schedules are set up 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 may choose to attend or not 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 projects coordination quality. The higher the frequency of nonattendance, the worse coordination quality is.

The OPDL Language

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 his/her 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 description of organizations and projects.

An OPDL program is composed of a number of sentences describing activities, projects, actors, teams, and operations for manipulating the process of simulation. Figure 3 presents the syntax of OPDL. In an OPDL program, the letters between the character "%" and the end of a line are comments. Figure 4 illustrates part of an example OPDL program. Our current work tries to extend the OPDL to describe not only the objects but also the processes such as information processing, communication, collaboration, and process management. Collaboration is under way with groups at MIT, the University of Hawaii, the University of Michigan and Carnegie Mellon University to develop a standard language for organizational modeling.

```
<program>    ::= <sentence>*
<sentence>   ::= <operation> | <definition>
<operation>  ::= (<operator> <operation-argument>*)

<operator>   ::= Include | Load | Save | Monitor | Run

<operation-argument> ::= <filename> | <application-name> |
                         <object-name> | <variable>
<definition>  ::= (<definition-name> <object-name>
                                   <att-val-pair>*)

<definition-name>  ::= Application | Actor  | Activity  |
                       Project | Organization | BehaviorMatrix

<att-val-pair>     ::= <keyword> <value>*

<keyword>          ::= :<predefined-alphanumeric-string>

<value>            ::= <alphanumeric-string> | <number> |
                       <variable> | (<value>*)
<filename>         ::= <alphanumeric-string>.opd |
                       <alphanumeric-string>

<application-name> ::= <alphanumeric-string>
<object-name>      ::= <alphanumeric-string>
<variable>         ::= ?<alphanumeric-string>
```

Figure 3: A BNF description of OPDL

```
(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

This is part of a program describing a project of building design. Comments after each line explain possible values separated by "/", and default values denoted within [].

Developing An Input File

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 a input file of VDT simulation environment. This model uses Quality Function Deployment (QFD) (Hauser and Clausing 1988) to derive interactions between requirements and engineering solutions, dependence between design activities in an activity precedence network, and relations between members of the project team, and consequently 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 93). Figure 5 shows an overview of this model.

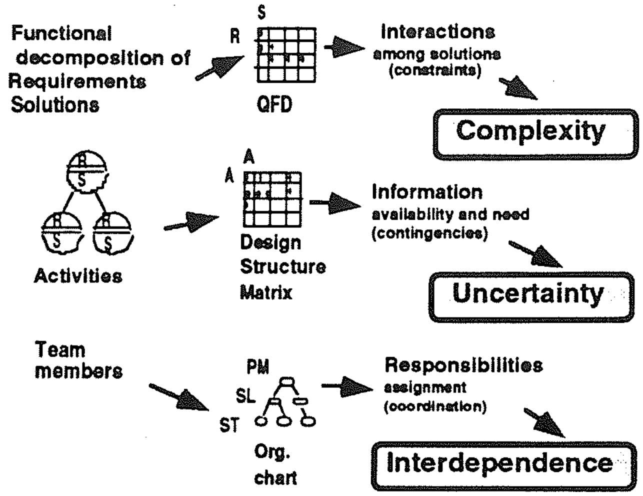

Figure 5: A model of coordination load for design teams
This model uses Quality Function Deployment (QFD) (Hauser and Clausing 1988) to derive interactions between requirements and engineering solutions, dependence between design activities, and relations between members of the project team.

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 spend waiting for decision etc., and progress data of activities, such as work completed, amount of rework, etc.

SOFTWARE AND HARDWARE ENVIRONMENT

VDT is implemented as a symbolic model using object-oriented programming techniques. It has a set of objects with attributes and behaviors to define the design project and its subtasks, the actors who perform the information processing tasks that comprise design, and the communication tools available to each actor. Organizational structure emerges from the authority provided to each actor and the channels along which the actor can send and receive communications. The model is formal in that it includes the basic concepts of, and predicts behavior based on, a set of widely accepted theories. The model was implemented on a Sun Microsystems IPX Sparcstation using Prokappa from IntelliCorp, and the SIMLIB discrete event simulation system we developed on top of Prokappa.

QUESTIONS TO BE ANSWERED IN THE SIMULATION CLASSES

VDT will be employed in a class to be taught by the second author, commencing in January, 1994. Students in this class will be taught the theory that is modeled in VDT (a first course in Organizational Behavior is a prerequisite for this class). They will then use VDT to model and simulate a real concurrent design team, and to explore a set of questions that the team's managers might like to study. This could be to consider alternative organization structures, introduction of new communication tools (e.g., video conferencing) or changes in the task (work-process re-engineering). We believe that students can gain new insights from this type of class, by modeling phenomena that, up to now, have only been described in "word theories."

At the same time our research should benefit in two ways: (1) we expect to gain valuable data to calibrate VDT; and (2) we have been required to provide robust and intuitive interfaces both for describing and "editing" project models as inputs to VDT, and for exploring the enormous volume of potential output data generated by this type of simulation tool in useful ways.

REFERENCES

Christiansen, T. 1992. Modeling Efficiency and Effectiveness of Coordination in Engineering Design Teams. *PhD dissertation, Stanford University, Department of Civil Engineering.*

Cohen, G.P. (1992). The virtual design team: An information processing model of coordination in project design teams. *PhD dissertation, Stanford University, Department of Civil Engineering.*

Galbraith, Jay. (1977). *Organization design.* Reading, MA: Addison-Wesley.

Hauser, J. and Clausing, D. 1988. *The house of quality* Harvard Business Review, May-June 1988

March, J.G., & Simon, H.A. (1958). *Organizations.* New York, NY: Wiley.

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

Simon, H.A. (1976). *Administrative behavior: A study of decision-making processes in administrative organization.* New York, NY: Free Press.

Thompson, J.D. (1987). *Organizations in action.* New York, NY: McGraw-Hill.