CRGP Working Paper
Submitted to Organization Science Special Issue on Organization Design
(William Starbuck and Roger Dunbar, Editors) 2005

Please send comments to ray.levitt@stanford.edu.
Do not reproduce or cite without the author's permission.

Organizational Design as "Virtual Adaptation":
Designing Project Organizations Based on Micro-Contingency Analysis[1]

Raymond E. Levitt[2]

Abstract

Using powerful and convenient analysis tools, engineers "adapt their product designs virtually"—i.e., they iteratively build models of potential product configurations virtually, in the computer; they analyze these models to predict their performance; they evaluate the predicted performance of each configuration against required or desired performance objectives; and they propose new configurations to address shortcomings of prior solutions. The critical distinction between systematic design *vs.* adaptation by trial and error is the availability of analysis tools that can make realistic enough predictions of performance to serve as "virtual experience" in an adaptive learning process. A "design" process thus depends on the availability of analysis tools that can model the behavior and interaction of micro-phenomena to predict specific macro-performance outcomes with acceptable accuracy. In a keynote talk to the NAACSOS 2003 conference, the author argued that computational modeling and simulation of organizations have "come of age" and can now be used as analysis tools for the design of organizations and larger social systems. This paper explains how analysis tools based on the 17-year Virtual Design Team (VDT) research project can now be used to support systematic organizational design for teams executing complex product development, software development and related knowledge work projects. We elaborate the micro-information-processing theory underlying VDT, describe how it was validated internally and externally, lay out the organization design methodology we developed to deploy it, and share the results of recent experience using this organization design approach to design real world project organizations.

Introduction

The organizational forms that managers create and "re-engineer" to achieve a wide range of public and private goals currently tend to be adapted from prior experience by trial and error, rather than being explicitly designed to meet a given set of goals and constraints in advance. For project organizations, both goals and metrics tend to be much clearer than the ambiguous and contested goals and metrics for entire enterprises (March & Simon 1958). Yet studies have shown that managers tend to adapt their organizational forms from prior experience even for well-defined projects, rather than explicitly designing them for each situation (Tatum 1983).

As shown in Figure 1, robust analysis capability is at the core of a true design process. Validated and calibrated analysis tools empower designers to predict the performance of

---

[1] This paper is based upon work supported by the Center for Integrated Facility Engineering (CIFE) and the National Science Foundation, under Grant No. IIS-9907403. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of CIFE or the National Science Foundation.

[2] Professor of Civil & Environmental Engineering; Professor, by Courtesy, of Medical Informatics; Director, Collaboratory for Research on Global Projects, Stanford University. Terman Engineering Center, Dept CEE, MC 4020, Stanford, CA 94305. Email: Ray.Levitt@stanford.edu

alternative configurations of proposed artifacts with high fidelity. This predictive power, in turn, allows a designer to adapt the configuration of an artifact iteratively, based upon the predicted success or failure of prior configurations that the designer has analyzed and evaluated. In this way, an analysis tool founded on a strong theoretical framework and implemented in a convenient mathematical model or a computer simulation tool, enables true design—i.e., it enables rapid, *a priori* "virtual adaptation" of an artifact, without ever constructing and testing a real instance of the artifact.

Adaptation is an effective way to enhance performance over time in stable environments through learning, but it is costly. Building and testing multiple, real-world artifacts consumes large amounts of time and effort; and failures of prototype artifacts can result in significant loss of property and lives along the way. Thus contemporary engineers would not conceive of an alternative to using analysis methods and tools for developing even the earliest prototypes of their real-world bridges, airplanes, or microprocessors.

So, after more than 50 years of research to develop a theory of organizations, why do managers still evolve their organizational forms by adaptation? The reason is that there are two basic requirements for using a true "design" process rather than costly trial and error adaptation. These requirements have both been difficult to satisfy in the case of organizations and other social systems.

Figure 1. A Formal, Double Loop, Design Process. A designer sets goals for the performance of an artifact, then cycles through the inner design loop, searching iteratively through design alternatives. In each cycle the designer proposes a candidate solution, analyzes it to predict its performance and compares its performance to the current set of goals for the artifact. When an alternative is found that achieves acceptable performance, the process terminates. If no candidate solution can be found whose goals satisfy the required set of goals, the designer enters the outer loop and evaluates whether a reduced set of goals can be found that is still attractive enough to pursue. If so, one or more of the goals are relaxed and the designer reenters the inner loop to search for a solution that meets the reduced set of goals. The design process terminates in success when an acceptable solution is found; or in failure, if no acceptable solution can be found without lowering goals below a threshold of acceptability. Note the close similarity between this view of design and the general model of adaptive search in organizations proposed by Cyert & March (1963; 1992).

First, "design" of an artifact like an organization is an intentional, goal-driven activity. This implies that there is a coherent and clearly articulated purpose for designing the artifact—i.e., there is an agreed upon set of individual or collective goals that the artifact is intended to achieve (Simon 1996). This is a challenge for many, if not most, organizations. As pointed out by March and Simon (1958) and by a host of subsequent organization theorists, individual stakeholders inside and outside a given organization typically have multiple, ambiguously articulated and conflicting sets of goals and sub-goals for the organization. Thus many organizational researchers have studied and modeled the processes involved in the development and evolution of organizational goals descriptively using "natural system" political frameworks such as power, bargaining and coalition formation. Few have viewed goals as outcome metrics in a prescriptive, model-based, "rational systems" approach (Scott, 2003).

Second, "design" requires that the designer be able to predict the performance of alternative configurations of an artifact in advance of creating it, by applying one or more "analysis" methods and tools, and assess this predicted performance against desired goals for the artifact. Starting with mathematical analysis models in the 1800s, and accelerated by computers since the 1950s, physical scientists and engineers, using interval and ratio measures of physical entities, have formalized, validated and calibrated predictive, model-based theories to underpin analysis methods and tools. Early mathematical modeling of physical systems often followed the path of setting up and solving systems of linear or differential equations whose predictions could be reconciled with the results of laboratory scale model experiments or field observations to validate and calibrate the theories. In contrast, the variables which organizational researchers must use to characterize the entities that comprise organization configurations—e.g., roles, skill levels, groupings, levels of centralization, means of coordination—tend to be nominal and ordinal variables, rather than interval or ratio measures. Mathematics and early computer languages provided excellent support for modeling and simulating interval and ratio variables, but were less amenable to formalizing and testing theories construed in terms of nominal and ordinal variables. Thus the vast majority of research on organizations to date has been descriptive, taxonomic and qualitative, rather than predictive, model-based, and quantitative[3].

Organizational researchers have not developed a robust body of theory to date that can underpin powerful analysis tools for organization design. To *advance the state of the art of organization design,* we need to develop a model-based micro-organizational theory that has strong predictive power. To *extend the state of practice of organization design,* the theory must be articulated in a way that has strong representational validity for managers. Two key representational and theoretical stepping stones toward such a theory are:

1. Recent advances in computer science modeling languages and tools for representing and reasoning about nonnumeric variables; and
2. The information processing view of organizations pioneered by the work of Herbert Simon and James March.

We discuss these twin points of departure for developing analysis tools next.

[3] There are, of course, numerous exceptions to this stark and oversimplified juxtaposition of qualitative, descriptive organizational research *vs.* quantitative, predictive physical science research, e.g., Cyert and March's (1963) *Behavioral Theory of the Firm* or the evolutionary organizational models of Hannan & Freeman (1977), and subsequent streams of mathematical and computational modeling by other researchers, e.g., (Levinthal & March 1981; Bonini 1967; March 1991; Carley & Svoboda 1996), over the past two decades. Nevertheless, this broad generalization holds for the vast majority of organizational research since the 1940s.

Recent Developments in Computational Languages and Tools

Starting in the late 1950s, computers that had been developed for code breaking and ballistic calculations in World War II emerged on the scene as tools for engineers. Their initial application was to automate tedious mathematical transformations such as solving sets of linear equations by matrix inversion, or calculating approximate numerical solutions for sets of differential equations. However, computer languages like FORTRAN, which excelled at modeling numerical arrays and manipulating sets of equations, provided little support for reasoning about nominal or ordinal variables in the social sciences. List processing languages like LISP and PROLOG, which can compare strings of characters or strings of words to perform pattern matching, forward and backward chaining logical inference over sets of rule, and other kinds of qualitative reasoning in near-natural language syntax, would address this need about 20 years later.

By the late 1970s, one could discern the forerunners of "agent-based" computational models of organizations in the "finite element" programs developed by engineering researchers to analyze the behavior of engineering structures, complex fluid flows, and a variety of other complex engineered systems. These agent-based programs model well-understood "canonical" physical micro-behavior of sub-elements of an engineering system and embed these behaviors in the computational agents. For example, in developing finite element models of structural systems, physical scientists and engineers embedded micro-behaviors based on well-understood material science physics in a set of physical agents termed "finite elements"—thousands or even millions of small triangular 2- or 3-dimensional spatial elements. The structural finite elements stress, strain and deform under various combinations of internal and external loads, while constrained to deform so that their edges remain in contact; and the resulting macro strains and deflections can be experimentally validated and calibrated against the observed macro-behavior of physical scale models and real structures. Mature analysis tools of this kind can now generate extremely accurate predictions of the emergent behavior of complex structural systems whose complexity and degrees of freedom far exceed the bounds of manual mathematical representation and solution.

Agent-based computational modeling and simulation is a natural and intuitive method for developing predictive, multi-level social science theory. Mature, validated, micro-social science models of micro-behaviors can be embedded in computational organizational micro-agents (individuals or small groups) as sets of canonical micro-behaviors. Organizational researchers developing analysis tools can then model the way in which these canonical agents behave and interact in their "virtual world"—which includes both other computational agents and relevant aspects of the task and/or environment—to generate meso- and macro-level outcomes that can be validated against macro empirical data. Figure 2 shows how agent based simulation links micro-theory and experience to macro-theory and experience to develop new multi-level, model-based organization theory.

The challenge in deploying powerful agent based computational models of organizations has been the limited affordances of traditional computing languages for modeling and reasoning about the nominal and ordinal variables required to describe many important attributes of organizations. The pivotal contribution to the social sciences of the languages and tools that resulted from artificial intelligence (AI) research of the 1970s the 1980s is the facility they provide to represent and reason rigorously about nonnumeric variables. Some of the key features of these languages and tools that allow nonnumeric reasoning include:

- List-processing functions in languages like LISP and PROLOG can compare and manipulate arbitrary strings of characters *(=words in natural language)* and strings of words *(=clauses that correspond to premises and conclusions of rules in natural* *language).* Thus, the primitive operators in such languages can compare and manipulate entire words and sentence clauses in natural (human) languages. These new list processing capabilities dramatically advanced the capability of computers at processing natural language text strings—the essence of qualitative reasoning.

- Powerful forward and backward chaining "inference engines" can automatically traverse long lists of rules to perform "If… Then…" deductions across sets of rules. Thus they can infer high-level inferences such as the appropriate level of organizational centralization, from low-level qualitative data about, for example, task complexity, worker skill levels and environmental uncertainty.
- Third, "object-oriented" development tools like IntelliCorp's *Knowledge Engineering Environment (KEE®)*—the forerunners of today's object-oriented languages like C^{++}, $C^{\#}$ and JAVA—allow easy definition of classes of objects like "tasks" or "workers" with generic micro-behaviors like information processing and communication. These generic task and worker agents can then be instantiated with specific data about each particular task or worker in an organization to model the variability of tasks or workers in an organizational setting with considerable realism, and with very little programming effort.

These three outputs from Artificial Intelligence research during the 1970s and 1980s significantly leveled the playing field for social scientists like the author, who seek to build analysis tools for organizations with comparable predictive power and representational validity to those built by physical scientists since the 1960s[4].

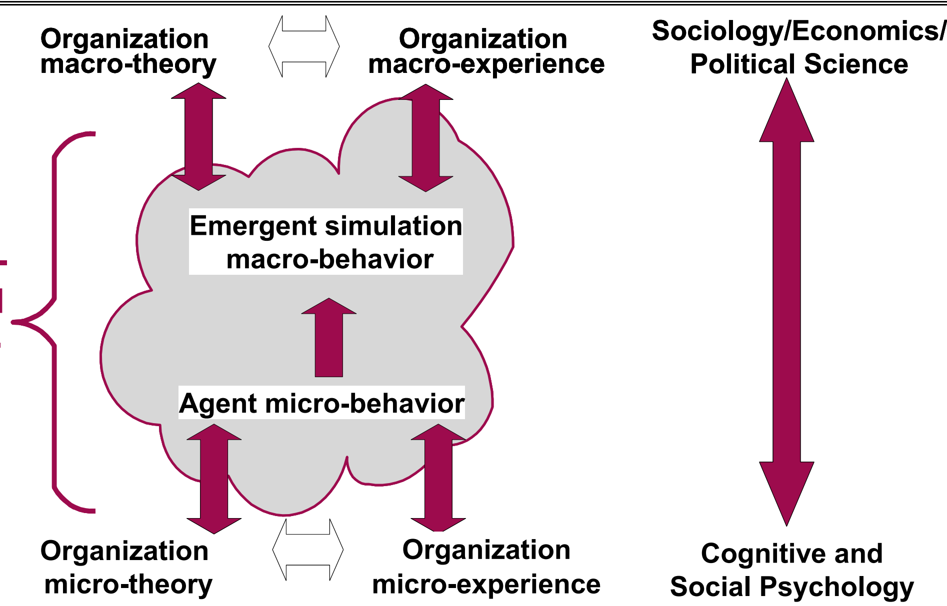

Figure 2. Agent-based Computational Modeling of Organizations. The figure shows how agent-based computational modeling and simulation builds new linkages between micro- and macro-social science theories and experience to create predictive analysis tools that can support organizational design.

[4] For a more detailed discussion of how artificial intelligence research can advance both qualitative and quantitative analysis for engineers and managers, see (Dym & Levitt 1991).

The well-known Garbage Can model of boundedly rational decision making in universities (Cohen *et al.*l 1972) used the FORTRAN computer language to develop an abstract, numerical model of actors, problems, and decision-making contexts in "organized anarchies"—universities and similar organizations with ambiguous goals and unclear rules for participation in decision-making. The Garbage Can model provides intriguing high-level, general theoretical insights about how organizational anarchies can actually make decisions in spite of unclear and contested goals, although some of the decisions they make might be purely symbolic. The authors perform an intriguing set of "theorem proving" or "intellective" experiments with this model by simulating idealized scenarios that represent large vs. small universities in good vs. bad economic times and predict performance outcomes for each case that can be compared to field data. However, the Garbage Can model does not attempt to predict the detailed performance of a specific educational institution, and hence provides only the most general kind of guidance for a university president attempting to improve the performance of her university in a particular context.

Among the first to exploit the emerging AI languages and tools for organizational modeling were Michael Masuch and Perry Lapotin (1989). Their AAISS agent-based model of an administrative organization performing simplified but real production and administrative tasks used some of the underlying concepts in the Garbage Can model, but extended them using the kinds of nonnumeric representation and reasoning languages and techniques described above. The level of representational validity that AAISS achieved for particular tasks and organization structures, and the predictive power that it could achieve in subtly varying contexts with these extensions showed just how far nonnumeric representation and reasoning could extend the representational validity and predictive power of numerically-based computational organization modeling. This inspired the approach taken for our Virtual Design Team research.

Theoretical Bases to Support Organization Design

Two major strands of organizational research provide the foundations for a theory strong enough to support analysis tools that can be used to design organizations:

1. The information-processing view of organizations provides the framework for an agent-based predictive model of micro-organizational behavior; and
2. Organizational contingency theory provides macro-level propositions relating structural form to context, based on empirical observation of organizations in different environments.

This section elaborates each of these bodies of theory and shows how they were used to support the Virtual Design Team organizational analysis framework.

The Information-Processing View of Organizations

The first theoretical requirement for building robust analysis tools to support the design of organizations is a body of micro-theory that lays out the kinds of micro-behaviors to be embedded in computational agents. The information-processing view of organizations describes how "boundedly rational" actors attempt to process information required to satisfy demands arising from a combination of assigned tasks and the need for constant environmental monitoring and response. This early "agent-based" framework for thinking about organizations was pioneered by Herbert Simon and James March in the 1950s, and extended by Galbraith (1973), Tushman and Nadler (1978) and Stinchcombe (1990), among many others.

Jay Galbraith, in particular, explicated the way in which boundedly rational actors could become overwhelmed by a combination of information-processing arising from their assigned direct tasks, combined with their need to process exceptions—situations in which the agent has insufficient information to complete the current task, and so must seek help to resolve the exception from a more knowledgeable agent. His observations were made in engineering organizations engaged in design and development of complex products such as aircraft and showed that middle managers in these settings could quickly become overloaded with information-processing when faced with a combination of concurrency and high levels of reciprocal interdependence[5]. In an influential paper, Galbraith (1974) proposed two generic organizational design strategies for addressing information overload:

1. Decentralizing decision-making to autonomous sub-projects, or increasing the amount of slack resources (e.g., bigger budgets, longer schedules and more design redundancy) could reduce the frequency of exceptions, and hence the demand for information-processing by managerial agents.
2. Alternatively, increasing the capacity of the vertical information system, or resorting to formal lateral communication and command structures—i.e., matrix structural forms—could increase the capacity of the organization to process exceptions.

Although qualitative in nature, Galbraith's work provided the first model-based theory that could predict organization failure arising from information overload, and provided managers with a set of general interventions for reducing information overload.

In his book, *Organization Design,*(Galbraith 1977) and in several subsequent books, Galbraith elaborated his information processing framework into a set of principles for designing five interacting aspects of organization design—strategy, structure, people, processes and rewards—simultaneously, albeit qualitatively. This work has provided a valuable set of design principles that managers and organizational consultants can now use to think about qualitative changes in the design of their organizations to address information overloads. Of course, information overloads represent only one kind of organizational failure. Organizations can fail because of intractable goal conflicts, failure to learn in fast-changing environments, and many other reasons. However, for many kinds of project teams engaged in fast-paced, complex product development, succumbing to information overload appears to be a primary failure mode that has resulted in delay and failure costs totaling billions of dollars and many lives. Thus the information-processing framework appears to provide an excellent first order theory to support analysis tools that can predict an important kind of failure for project teams.

The author's Virtual Design Team (VDT) computational organizational modeling and simulation framework described in this paper was directly inspired by Jay Galbraith's information-processing framework.

Organizational Contingency Theory

Organizational contingency theory provides many rules that can be used to select the overall macro-configuration of an organization, given its context, as elegantly summarized by Henry Mintzberg's (1979) five archetypal organizational configurations. Rich Burton and Borge Obel (Burton & Obel 1995;1998;2004) integrated literally thousands of studies from more than 50 years of contingency theory research on organizations into a coherent—albeit sometimes contradictory—set of "If…, Then…" propositions. These contingency propositions relate organization structure variables such as *Level of Vertical Centralization* to organizational context variables such as environmental *Complexity, Uncertainty* or *Equivocality.* To formalize these rules and make them usable by managers for diagnosing their organizations, they developed

[5] Reciprocal interdependence between two tasks arises when their goals interact so closely that the workers or groups responsible for the two tasks must negotiate directly with one another to reach a globally acceptable solution using a form of coordination that James Thompson (1967) termed mutual adjustment.

the *Organizational Consultant (OrgCon)* organizational diagnosis system. OrgCon exploits AI pattern matching and rule chaining methods to reason about the set of contingency theory propositions that Burton and Obel assembled and can identify misfits between aspects of an organization's context and its structure. They used judgment to assign higher weights to the findings with the greatest level of empirical validation, but allowed for weighing of conflicting conclusions. "Dueling propositions" are weighed using "certainty factors" for estimating the user's confidence in parameter values and evidence-weighing for inferring conclusions. This same "Dempster-Schaeffer" competing evidence approach was deployed in early medical expert system applications such as MYCIN (Buchanan & Shortliffe 1984).

A user of OrgCon performs a high-level diagnosis of an organization's fit with its context in a manner that can be viewed as "qualitative analysis". The current structure of an organization represents a candidate solution to test against the organization's context. OrgCon carries out a qualitative analysis of the performance of this organization and its context to predict three kinds of misfits: (1) misfits among context variables, (2) misfits between the overall context and the macro-structural configuration, and (3) internal misfits among details of the structural configuration.

OrgCon has been validated against a large set of empirical data from companies in Scandinavia in which a strong (inverse) correlation was found between the number and severity of organizational misfits of these companies and their long-term financial performance (Burton *et al.* 2002). However, OrgCon does not have the ability to predict specific costs, or even specific dimensions of organizational performance that will result from such misfits. Nevertheless, the book, *Strategic Organizational Diagnosis and Design*, and the OrgCon computer system provided with the book, represent a remarkable scholarly contribution to the field.

Building Blocks for a Computational Model of Organizational Performance

Although Galbraith never attempted to do this, his theory of information overload in Project teams begged to be quantified and operationalized in an agent-based computational model. The predictions of such a model could then be validated by comparing them to the predictions of the huge body of empirical contingency theory findings embedded in Burton & Obel's OrgCon framework for diagnosing project organizations. Over the past 18 years, the author and his students and colleagues have built on these two ideas to create the Virtual Design Team (VDT) organization design framework. VDT operationalized the bottom-up information-processing modeling framework articulated by Galbraith; extended and quantified Galbraith's framework based on ethnographic research in engineering organizations; and then validated its predictions against the predictions of contingency theory as well as our own empirical macro-observations of project teams.

In a keynote talk to the NAACSOS 2003 conference (Levitt 2004) the author argued that computational modeling and simulation of organizations have "come of age" in the early 21st century and can now be used as analysis tools for the design of organizations and larger social systems. The remainder of this paper explains how the Virtual Design Team approach and tools can now be used to support a systematic organizational design approach for teams executing complex product development, software development and related knowledge work tasks. The next section elaborates the micro-contingency information-processing theory underlying VDT, explains the organization design methodology that we developed for using VDT to analyze organizations, and shares the results of recent experience using this systematic model-based organization design approach for real world project organizations.

The Virtual Design Team (VDT) Organizational Analysis Framework

To enable organizational design, managers need access to analysis tools that can make credible predictions to guide a set of feasible managerial interventions in the structure and policies that comprise their organizational forms. Complementing the kind of top-down qualitative diagnosis that Organizational Consultant can provide for an *enterprise,* The Virtual Design Team (VDT) methodology and software (Levitt *et al.* 1999) performs a fine-grained and hybrid qualitative-quantitative analysis of the relationship between work process and organization structure at the level of a *project team.* Like finite element analysis tools in engineering, VDT formalizes, extends, operationalizes and quantifies organizational micro-contingency design principles as micro-behavior of organizational agents to predict performance outcomes at the level of a single project, or the level of a matrix organizational unit executing a portfolio of projects.

In the late 1980s, our research group concluded that attempts to model organizations computationally could benefit greatly from the use of nonnumeric or "symbolic" AI representation and reasoning techniques. Early attempts to do this had convinced us (Levitt & Kunz 1985) and others, e.g,. (Masuch & Lapotin 1989), that this was a fruitful line of research. Drawing from the rich information processing theory base, VDT employs symbolic (i.e., nonnumeric) representation and reasoning techniques from established research on artificial intelligence in conjunction with numerical discrete event simulation approaches previously used for modeling production workflows in factories or logistic flows through supply chains, to develop new kinds of hybrid qualitative/quantitative computational models of these theoretical phenomena. Once formalized through a computational model, the symbolic representation is "executable," meaning it can be exercised to emulate the dynamics of organizational behaviors.

We recognized from the outset that developing quantitative analysis tools for organization design was a significant challenge. In choosing the kinds of organizations that we would model, we picked project teams performing routine design or product development work. For this class of organizations, all work is knowledge work, so that the information processing abstraction of work in organizations (Galbraith 1974) applies well. Galbraith's "information demand, capacity and throughput" model can be viewed as an analog to Newton's Laws of Motion in physics, a simple and useful first order approximation to diagnose imbalances between information processing demand vs. capacity in a project organization performing relatively routine tasks.

Goals and means are both clear and relatively uncontested for routine project work, so we could finesse many of the most difficult "organizational chemistry" modeling problems inherent in the kinds of organizations that sociologists have frequently studied: mental health, educational and governmental organizations. By operationalizing and extending Galbraith's information processing abstraction in the Virtual Design Team (VDT) computational model, and starting our modeling work in the simplest corner of the space of all organizations, we were able to develop multiple versions of VDT (Cohen 1992; Christiansen 1993; Thomsen 1998) and validate their representation, reasoning and usefulness using the trajectory described in (Thomsen *et al.* 1999).

Organizations are inherently less understandable, analyzable and predictable than physical systems, and the behavior of people is non-deterministic and difficult to model at the individual level. But individual differences tend to average out when aggregated cross-sectionally and/or longitudinally. Economists have exploited this "mean field" approach in their models of production functions and supply-demand equilibrium for decades. Thus, one can augment a symbolic, nonnumeric model with aspects of mean field numerical representation when modeling aggregations of people in an organizational context (e.g., work groups, departments, firms). For instance, the probability of a given task incurring exceptions and requiring rework can be specified—organization wide—by a distribution; and the inherently stochastic decisions about attention of a worker to any particular task or event (e.g., to a new work task, communication, or request for assistance) can be modeled stochastically to approximate collective behavior. Once this has been done, specific organizational behaviors can be simulated hundreds or thousands of times using Monte Carlo simulation techniques to gain insight into which results are common and expected versus which are rare and exceptional.

Of course, applying numerical simulation techniques to organizations is nothing new (e.g., see Law and Kelton 1991). But this approach enables us to *integrate* the kinds of dynamic, qualitative behaviors emulated by symbolic models with quantitative aggregate dynamics generated through discrete-event simulation. It is through such integration of qualitative and quantitative models—bolstered by strong reliance upon well-established theory and commitment to empirical validation—that our approach diverges most clearly from extant research methods, and offers new capabilities for analyzing organizational performance (Nissen & Levitt 2004).

Information-Processing Micro-Behavior Underlying VDT

VDT models knowledge work through interactions of *tasks* to be performed; *actors* communicating with one another, and performing tasks; and an *organization structure* that defines actors' roles, and constrains their behaviors. Figure 3 illustrates this view of tasks, actors and organization structure. As suggested by the figure, we model the organization structure as a network of reporting relations, which can capture micro-behaviors such as managerial attention, span of control, and empowerment. We represent the task structure as a separate network of tasks, which can capture organizational attributes such as expected duration, complexity and required skills. Within the organization structure, we further model various *roles* (e.g., marketing analyst, design engineer, manager), which can capture organizational attributes such as skills possessed, levels of experience, and task familiarity. Within the task structure, we further model various sequencing constraints, interdependencies, and quality/rework loops, which can capture considerable variety in terms of how knowledge work is organized and performed.

Each actor within the intertwined organization and task structures has a queue of information tasks to be performed (e.g., assigned work tasks, messages from other actors, meetings to attend) and a queue of information outputs (e.g., completed work products, communications to other actors, or requests for assistance). Each actor processes tasks at a rate and with an error frequency that depend upon a qualitative match between: the actor's skill types and levels *vs.* the skill required for a given task; the relative priority of the task; the actor's work backlog (i.e., queue length); and how many interruptions divert the actor's attention from the task at hand. Actors' collective task performance is constrained further by the number of individual sequential and parallel tasks assigned to each actor, the "work volume" of those tasks, and both scheduled (e.g., work breaks, ends of shifts, weekends and holidays) and unscheduled downtime (e.g., awaiting managerial decisions, awaiting work or information inputs from others, or performing rework).

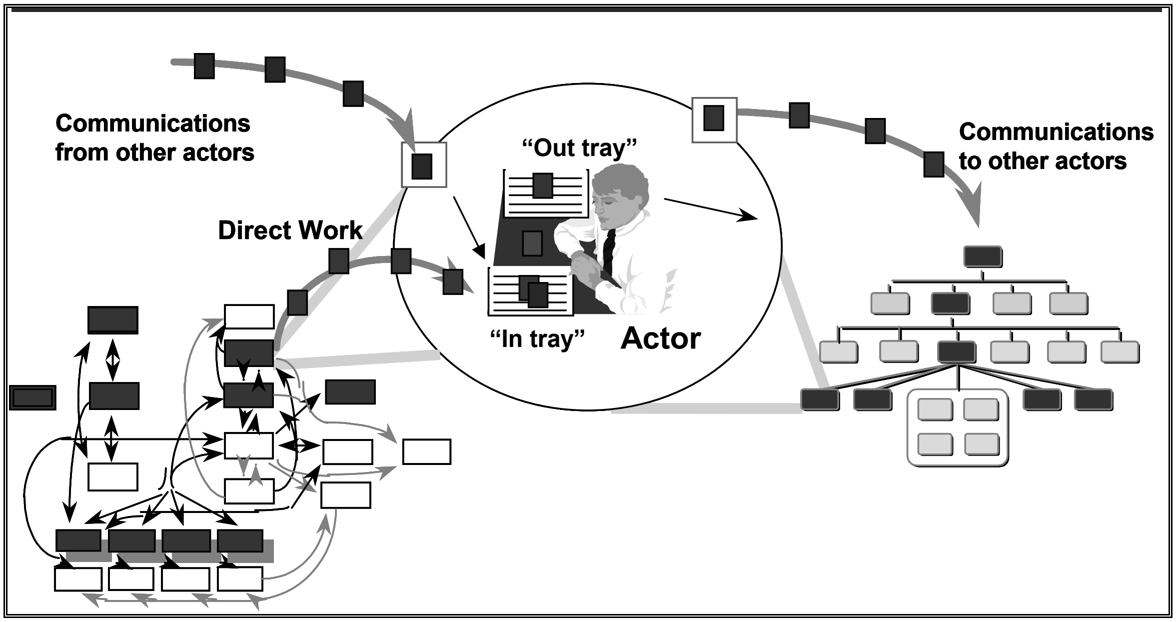

Figure 3 VDT's Information Processing View of Knowledge Work. VDT Actors process both direct tasks and communications from other actors as "information-processing sub-tasks." Based on Monte Carlo discrete event simulation, subtasks arrive constantly during the duration of the project and accumulate in actors' in-trays. When sub-tasks arrive faster than the actor can process them, the actor becomes backlogged, triggering delays and increased quality risks due to missed communications and meetings. Each sub-task is tagged with a specified work volume, required skill, arrival time and priority. Based on these tags, VDT actors stochastically select particular items from their in-trays to work on, experience exceptions in processing tasks, and determine which supervisory actors they will consult to help resolve exceptions that may arise. Unresolved exceptions, missed communications and missed meetings all increase the probability of exceptions in subsequent tasks.

Both primary work (e.g., planning, design, manufacturing) and coordination work (e.g., meetings, management, joint problem solving) are modeled in terms of *work volume* (measured in person-hours or person-days). Work volume is specified as full-time equivalent actors multiplied by time (FTE-Hours or FTE days) and represents the amount of information processing work associated with a task, a meeting, a communication, etc.

Thus, the VDT simulation engine employs both qualitative and quantitative reasoning. VDT alternates between qualitative pattern matching and numerical discrete event simulation. Results of the qualitative pattern matching adjust integer numerical variables such as numbers of missed meetings and real number variables such as error probabilities in the numerical Monte Carlo discrete event simulation part of the model.

VDT applies AI-style symbolic pattern matching—i.e., it reasons qualitatively, using pattern matching over nominal and ordinal variables, based on micro-behaviors derived from organization theory. In tandem, the discrete-event simulation engine steps a simulation clock forward in time using time steps as small as one minute to enable quantitative computation of work volume and elapsed time; and it tracks the number of missed communications, missed meetings, items of rework not completed, and other process quality metrics by task, actor and for the aggregate project team. Bridging between the qualitative and quantitative reasoning are a set of tables called "behavior files" which represent the results of our ethnographic studies of actor micro-behavior in project teams. Each behavior file is a small matrix with about three rows and three columns containing a set of numerical values in each cell of the matrix. The qualitative inference engine in VDT reasons about nominal and ordinal variables such as actor role (one of: *Subteam, Subteam Leader* or *Project Manager)* and level of centralization (one of: *Low, Medium* or *High)* to pick a row and column in the behavior file matrix; the VDT controller takes the numerical value from this row-column intersection in the behavior matrix and passes it to VDT's quantitative, discrete simulation engine where it is used to reset a task's exception probability, adjust an actor's processing speed, etc. An advantage of this representation is that many details of the actor micro-behaviors in VDT can be calibrated over time by simply using a spreadsheet or work processor to change the values of entries in the behavior files—enabling non-programmers to develop and extend the model[6]. Readers interested in additional details of the VDT model's implementation should see (Jin & Levitt 1996).

Quantitative simulation places a significant burden on the modeler in terms of validating the representation of a knowledge-work process. It requires hands-on fieldwork to study an organization in action, and to formalize and calibrate the information processing micro-behaviors of its participants. Our computational modeling environment benefits from extensive fieldwork in multiple domains—e.g., power plant construction and offshore drilling (Christiansen 1993); aerospace (Thomsen 1998); software development, (Nogueira 2000); healthcare (Cheng and Levitt 2001); and other domains. VDT has been used since 1996 to teach classes on organization design at Stanford University and at more than 30 other universities worldwide. Through a process of "back-casting"—attempting to predict known performance outcomes of a past project using only information available at the beginning of a project—students in these classes have developed VDT models of real-world projects and demonstrated dozens of times that "back-casted" project outcomes predicted by VDT correspond well to actual performance outcomes for those projects (Kunz *et al.* 1998).

Organization Design Methodology

Using the VDT organization design methodology, a manager develops models of the specific work process and organization for a given project by instantiating graphical templates (sometimes called "stencils" in the engineering software world) of generic projects, tasks, actors, and relationships between tasks and actors to create a realistic model of the project. We refer to this process of specializing and linking generic objects into a configuration that represents the real world project work process and organization as "emulating" the real project. The VDT organization design methodology for iteratively modeling and simulating candidate organizational forms proceeds through a series of steps that closely follows the generic design process presented in the Introduction.

1. Assess Project Goals and Tradeoffs

Project goals always involve a trade-off between the desired scope/quality of the product or service to be delivered, the time it takes to complete the project and the resources used to complete the project. The modeler begins by discussing the relative importance of schedule, scope and cost/resource goals of the project with the project sponsor. These goals will be the basis for diagnosing risks and evaluating potential managerial interventions, so it is critical to clarify them at the start of the organization design exercise. It is also important to be clear whose viewpoint the modeler is adopting. If the goals the modeler is attempting to meet are not

[6] Behavior files, as used in VDT, are similar in concept and function to "decision tables" used in many kinds of engineering analysis tools.

those of the person providing the information for this project, the modeler must be clear whose goals they are, e.g., the goals of the client, lead design professional, key end users, etc.

2. Model a Candidate Organizational Configuration

Modeling a candidate organizational configuration involves modeling: the work process to be executed in the project; the "actors" (individuals or sub-teams) who comprise the organization that will execute the work process; their reporting structure and decision making policies; and the assignment of tasks to actors in the organization.

2.1 Model Project Work Process

The modeler develops a precedence schedule of the project's tasks at a relatively abstract level of detail with between 30-60 tasks. The first step is to define about 5 key business milestones for the project. Then the modeler adds about 5-10 tasks that accomplish the work needed to reach each milestone. To complete the definition of the work process, the modeler adds task properties such as work volume, skill requirement, complexity and uncertainty levels to each task.

2.2 Model Project Organization

Next the modeler creates a set of actors, attributes and supervisory relationships to emulate the project's organization at a level of detail where there are about 10-20 "actors" (each actor is an individual manager or a sub-team of a single discipline, e.g., a team of 4 architects). The modeler defines a list of skills relevant to the organization, and sets each actor's skill levels to None, Low, Medium or High for each of the skills in this set. The level of team experience is set to High, Medium or Low; this affects the amount of explicit versus implicit coordination that will be needed. The project supervisory or "exception-handling" hierarchy through which problems are resolved and decisions made is modeled by creating a set of "Supervision" links from each manager to the actors supervised by the manager. Regularly scheduled project meetings are modeled by creating meetings with a start time, duration and frequency, and then graphically linking the actors that attend the meeting to the meeting icon with "attends meeting" links.

2.3 Model Assignment of Tasks to Actors

Next, the modeler assigns a primary responsible actor to each task in the work process. VDT's convention is that each task has only one primary responsible position (individual or group) in the organization. The modeler may need to iterate the schedule and organization a few times to get a good match between the grain-size of tasks vs. participants to get this match in abstraction between the model of the organization and the model of the work process. Additional "secondary" responsible positions (that contribute time to the assigned task when they are not working on their own primary tasks) can then be assigned.

2.4. Add Information Exchange and Failure Dependence Links between Tasks

To capture two kinds of task interdependence, the modeler inserts links between interdependent tasks to model "information dependence", based on the concept of reciprocal interdependence (Thompson 1967), and "failure dependence", in which an exception in one task triggers compensating rework for the "failure dependent" task, as discussed in (Levitt *et al.* 1999).

2.5. Add Organizational Decision-Making Policy Attributes

Next, the modeler sets the levels of Centralization, Formalization, Matrix Strength and Team Experience to Medium (nominal), Low (significantly lower than nominal) or High (significantly higher than nominal) for the project organization.

2.6 Calibrate Model of Organization and Work Process

The baseline "as-is" or "as-planned" work process and organization model of the project are now defined in terms of the elements, along with their attributes and relationships. The modeler now needs to calibrate the baseline model iteratively by running trial simulations and adjusting model input parameters, until s/he is satisfied that the baseline model of the project accurately represents the baseline assumptions made by managers for the real project. If the subject project is a first-of-a-kind effort for which there is no database of historical work volume or task duration data, this "model debugging" step may involve several rounds of discussions with the manager/s who provided the information about the project to verify that model values for task sequences, task work volumes, actor skills, etc., are reasonably accurate approximations.

3. Run Simulations to Predict Performance of Candidate Organization Design

Once the modeler is confident that the work process and organization are a reasonable representation of the subject project, s/he runs sets of simulations to predict significant actor backlogs, and to predict the schedule, cost and/or quality performance outcomes for the baseline case. Since VDT employs stochastic, Monte Carlo simulation, the modeler runs a suite of simulation trials for each configuration—typically 100 or more—and calculates the means and standard deviations for all significant performance metrics.

4. Evaluate Performance Risks for Current Configuration

The modeler now needs to evaluate whether the candidate solution will generate acceptable performance, by comparing the predicted performance of the baseline model to the project goals defined in **Step 1** above. If predicted performance is acceptable, the modeler can terminate the design process and declare success. If not, s/he proceeds through the following steps.

5. Iterate Through Alternative Organization Designs to Mitigate Risks

The modeler now begins to cycle through the inner loop of the design process in **Figure 1**, comprising **Steps 2-4** above, to explore a variety of possible managerial interventions that might mitigate the schedule and quality risks identified in the previous step. These can include changes in reporting structure, decision-making properties, actors' skill levels, size of sub-teams, task assignments, level of centralization, or other organizational parameters. The modeler represents each potential intervention as a separate scenario or "Case" in VDT, thereby maintaining a record of interventions that have been explored. The SimVision® commercial version of the VDT simulation engine can perform a 100-trial simulation of a relatively complex project containing scores of tasks and dozens of actors in just a few seconds. This allows for rapid exploration of alternatives during the inner-loop design process.

6. If No Acceptable Solution Can be Found, Consider Lowering Goals

Adding resources, lengthening the allowable schedule, or reducing the project scope all represent different ways of lowering goals or "aspirations" for the project. Following the outer loop of Figure 1, the modeler can experiment with lowering different goals, to see whether acceptable organizational configurations can be found that meet or exceed a reduced set of goals. Of course, lowering goals reduces the attractiveness of a project, so the sponsor must be engaged in any decisions to lower one or more project goals, in order to reaffirm that the project might still be attractive with these reduced goals.

7. Terminate Organization Design Process in Success or Failure

The VDT organization design process terminates in success when a candidate solution is found whose performance meets or exceeds the goals of the project sponsor. Or it terminates in failure when no solution can be found that meets the current set of goals for the project, and further lowering the goals would reduce the attractiveness of the project below some threshold value of attractiveness for the sponsor.

Validation of VDT Modeling Framework and Methodology

As shown in **Figure 2**, computational models of organizations can be viewed as providing a unique theoretical bridge between micro-organization theory and experience (formalized in theories of cognitive and social psychology) and macro organization theory and experience (formalized in theories of sociology, economics and political science). Computational emulation models of organizations like VDT embed well-accepted or "canonical" theories of micro-behavior in computational agents, instantiate a configuration of agents and tasks, and then explore the macro-outcome implications of this configuration. After they have been internally validated, organizational emulation models can be externally validated in two distinct ways—“intellective experiments” or “emulation experiments”—as shown in **Figure 4.**

- **Internal Validation:** Developers of computational models that attempt to emulate real-world behavior must first validate the micro-behavioral assumptions of their models, either by drawing on empirical micro-social science research findings, or by conducting their own ethnographies to describe and calibrate micro-behaviors. Once the behaviors have been captured and described, the implementation of the micro-behaviors in the model is internally validated (or “debugged”) using very simple "toy problems" for which predictions can be simulated manually, or with simple spreadsheet calculations.
- **Two Kinds of External Validation**: Second, the predictions of the model for specific idealized or real world configurations of work processes and organizations must be externally validated against the predictions of macro theory, via theorem-proving or "**intellective experiments**"; or they must be validated against macro experience via "**emulation experiments**". A third, newer form of external validation involves cross model docking in which pairs of models that address similar input and output variables are cross-validated by entering the same input data sets to both models and comparing their predicted outputs (Axtel *et al.* 1996). In the following section we discuss all three modes of external validation in more detail.

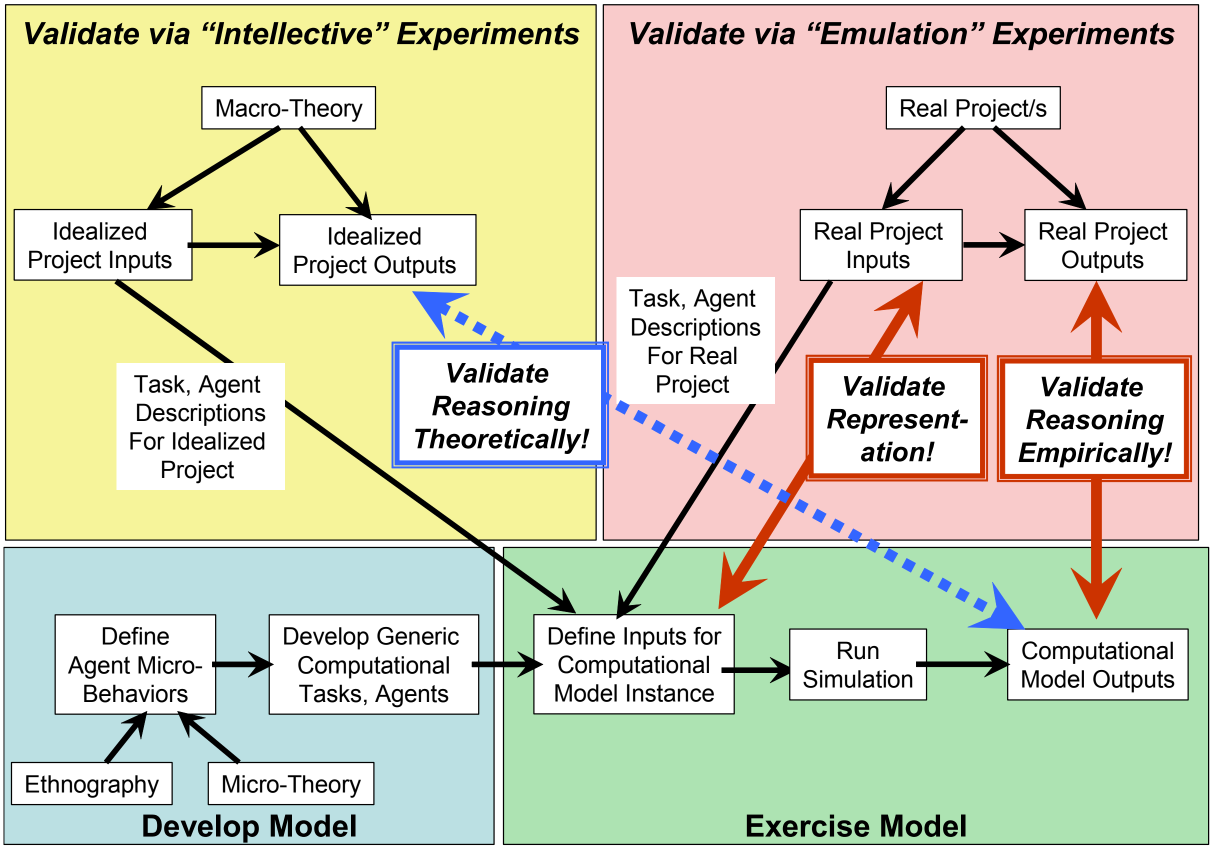

Figure 4. Two Kinds of External Validation for Computational Models of Organizations. In the VDT research program, we carried out internal validation continuously as part of model development each time new micro-behaviors were defined and embedded in the model. For external validation, we performed emulation experiments through multiple Ph.D. theses and class term projects. Initially we used historical "back-casting" experiments to calibrate and validate the model's performance retrospectively in what might be viewed as curve-fitting exercises. Then we began to conduct "forecasting" experiments to validate the VDT model predictions prospectively. (Cohen 1989; Christiansen 1991; Thomsen 1994). By 1996, managers at one of our test sites, Lockheed Martin, had developed sufficient confidence in VDT, following successful retrospective and prospective predictions for prior projects, that they began to use VDT predictions as a basis for intervening proactively in the design of their organizations. Validation of VDT through intellective experiments was initially viewed by our group as a more rigorous form of internal validation of the reasoning of VDT. Rigorous external validation of VDT through intellective experiments was carried out later, by us and others.

External Validation of VDT as an Emulator of Real Project Teams

To assess the extent to which VDT could emulate the performance of real-world project teams, we validated VDT in all of the ways shown in **Figure 5** of over a period of about six years. (Thomsen *et al.* 1998). First, we performed internal validation of VDT on toy problems—problems with one or two actors and one or two tasks that were simple enough for the researcher to simulate outcomes manually—to be sure that we had correctly incorporated the intended micro-behaviors in the agents. Next we developed a series of simple intellective experiments in which VDT model predictions were tested against the macro predictions of organizational contingency theory to test whether the overall model, which combines multiple kinds of micro-behaviors, was producing accurate macro performance predictions. For example, we modeled organizations engaged in highly uncertain tasks with both high and low centralization to check that low centralization would produce better performance outcomes, in line with predictions of organizational contingency theory, summarized in Burton and Obel's (1995;1998; 2004) OrgCon model.

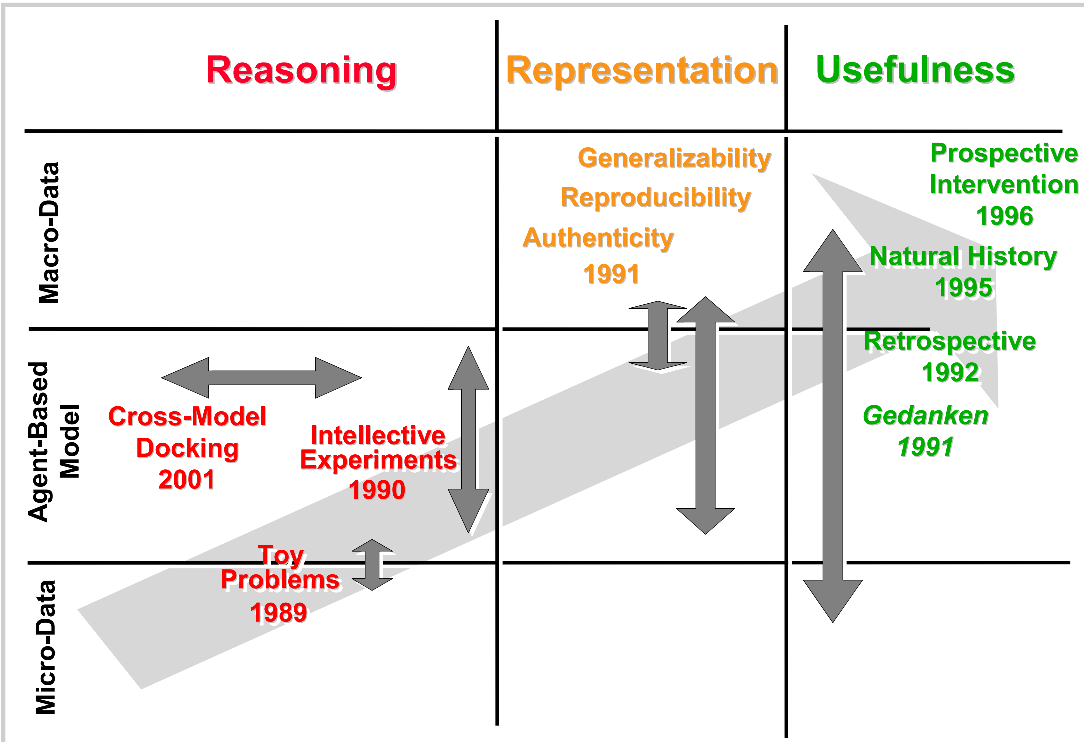

Figure 5. Validation Trajectory Used for VDT, and Proposed for Validating other Computational Emulation Models of Organizations *[Adapted from (Thomsen et al. 1999)]*

Next we tested the representational validity of the model by sending student teams out to model real project teams in which we gathered data from managers and discussed the results of model predictions with them. These experiments resulted in us renaming some of our model variables to better match the natural idiom used by project managers. We tested the reproducibility (inter-rater reliability) of the modeling framework by having multiple students model the same organization and work process and comparing the models that they produced. And we validated the generalizability of the modeling framework by using the framework to develop models in multiple engineering, software and human resource domains.

Finally, we validated the usefulness of VDT as an analysis tool by testing it on a series of real-world projects in industries ranging across power stations, offshore platforms, aerospace, semiconductors, biotechnology, theme parks and consumer products. We started with a series of back-casting experiments in which we modeled the initial conditions of a completed project and compared the VDT model predictions to the actual macro performance achieved on the project. Through a series of back-casting emulation experiments over five years, we calibrated the strengths of the various micro-behaviors in VDT to match the macro performance outcomes of these real-world projects. Next, we began to use VDT to make real-time, prospective predictions of performance outcomes for projects that were in the very early stages of conception and development.

In 1995-96, a group of VDT modelers from our research team made a spectacularly accurate prediction of exactly when and how the first Lockheed Launch Vehicle project organization would fail (Kunz 1998; Levitt *et al.* 1999). Following this successful prediction of a significant organizational failure, VDT had achieved a high enough level of external validity that a group of managers who had participated in the successful back-casting emulation experiments now had sufficient confidence in the model's predictions to start making proactive interventions in future projects, based on VDT predictions.

After almost eight years of development and testing, VDT had met the standard of being a "useful" analysis tool for designing project organizations!

External Validation of VDT through Intellective Experiments

Viewing VDT as a well-validated model of project-oriented knowledge work, researchers at Stanford and other universities began to use this computational modeling environment as a "virtual organizational test bench" to explore a variety of organizational questions starting in about 1997. The objectives of such "virtual organizational experiments" conducted with VDT to date include: understanding the effects of geographical distribution of team members on project performance (Wong and Burton 2000); replicating empirical findings of the classical Bavelas and Leavitt (Leavitt 1951) communication experiments (Carroll and Burton 2000); and exploring information processing and decision making behavior at the "edge of chaos" in organizations (Carroll & Burton 2000; Levitt *et al.* 2002).

Based on the success of these Intellective validation experiments, which we had not anticipated when we began to develop VDT as an emulator of real world organizations, we proposed an alternative trajectory for validating future computational emulation models in a more traditional theorem-proving mode, as shown in **Figure 6**.

External Validation of VDT through Cross-Model Docking Experiments

In cross-model docking experiments, two models are used to represent the same set of data that define an organization, work process and context; and the outcome predictions of the two models are compared with one another. A variant of model docking involves embedding one model within the other to create modeling and simulation and two different levels of analysis. Starting in about 2000, cross-model docking experiments were performed in which VDT was docked with: OrgCon (Burton & Obel 2004); with the ORGAHEAD model of Kathleen Carley and her collaborators (Louie *et al.* 2003); and with the BLANCHE knowledge network model of Professor Noshir Contractor and his colleagues (Pallazolo *et al.* 2002).

VDT's ability to predict the performance outcomes of alternative project organizations has thus been repeatedly validated externally against real-world project organizations, organizational contingency theory and other organizational simulation models. The following section describes a typical real-world organization design exercise using the VDT project organization design approach.

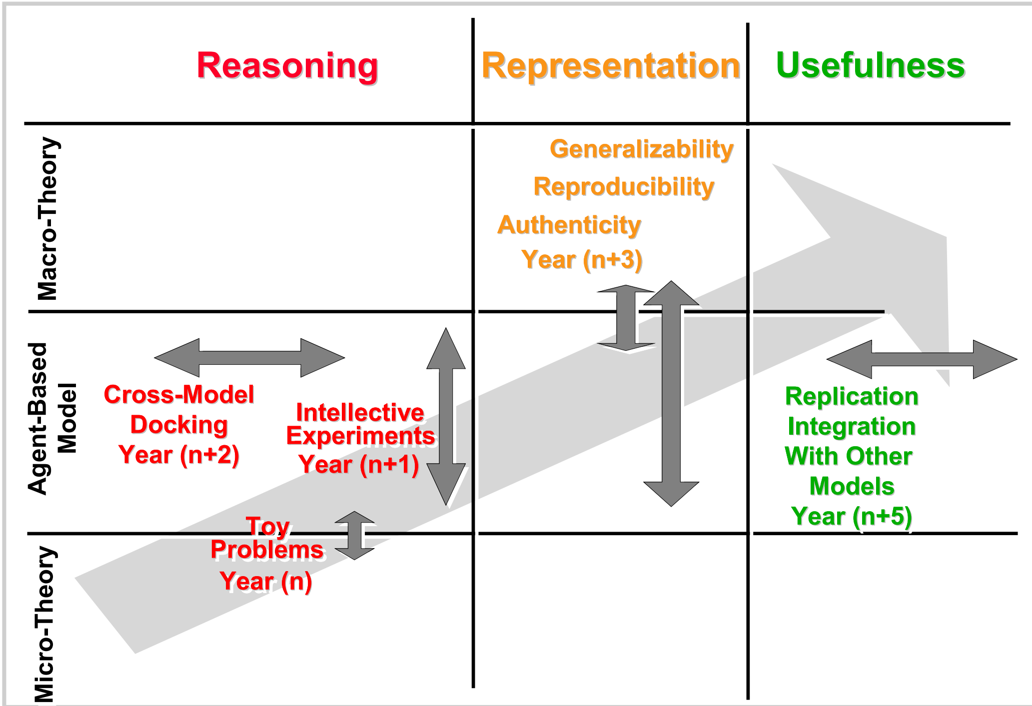

Figure 6. Trajectory for Validating Computational Organizational Models via "Theorem Proving "Intellective" Experiments *[Adapted from (Levitt 2004)]*

Applying the VDT Design Approach to Real Organizations

Starting in 1996, VDT was commercialized as SimVision, and began to be used to design real project organizations for a variety of fast track product development, software development and business process reengineering projects. The author took a leave of absence from Stanford University for 18 months to oversee the development of the commercial software, and to develop an organizational design practice methodology around the commercial implementation of VDT. This experience helped to refine and further validate the model through a series of back-casting validation experiments that individual clients performed to satisfy themselves that the model's predictions matched their past experience, before being willing to base managerial interventions on the model's predictions for future projects.

The following description of a project organization design process is typical of a number of such engagements that resulted in either changing organization designs to better meet a set of initial project goals, or in some cases reducing project goals—typically the planned scope of efforts such as software projects—in order to be able to accomplish them in an acceptable time with limited resources. We illustrate the application of our VDT project organization design methodology with a redacted case study of an organization design engagement for a project team attempting to design a custom chip for a personal digital assistant (PDA) in record time.

In January of 1998, a newly formed PDA company—hereafter called "Ace" to disguise its real name—was developing a new handheld personal organizer to compete with existing products in the marketplace. To meet its business plan, Ace needed to have about a dozen working prototypes of its initial product completed and ready to show at the next Comdex trade show in November of 1998. To meet this extremely aggressive development schedule, Adele Davidsen (not her real name), Ace's Senior VP of Product Development, determined that her project team needed to have the custom microprocessor for its new PDA completed by the third week in May. Ace's prior development efforts for application-specific integrated circuits (ASICs) of this complexity had all taken over eight months. Michael Levoy, the ASIC team project leader, and his key technical staff members were confident that they could get the ASIC developed by the end of May—i.e., in about 60% of the typical time for a chip of this complexity. However, this project was so critical to implementing Ace's strategy that Adele was not willing to "bet the company" on their intuitions. She wanted to carry out an independent analysis, to reassure herself and her managers that Michael's team could feasibly execute this aggressive development schedule before going ahead with the planned development.

We will step through the process that Michael's project team followed in designing the Ace project work process and organization. The goals of this process were: (1) to provide Adele with the independent verification of the team's estimate that she wanted; and, (2) to redesign the work process and organization as needed to increase the likelihood that the team could deliver a high-quality product on this aggressive schedule.

Step 1: Model Baseline Work Process and Organization

Michael's project team, working with an organization design consultant, first defined and documented their baseline assumptions about the work process and organization for developing this custom chip. The first step was to identify critical business milestones for this project, beginning with the final completion milestone. In this case, the completion milestone was *Fabricate, Test and Deliver.* Two intermediate milestones were *Logic Release* and *Layout Release.* Once the team had defined these milestones, they defined five to ten activities that would enable each milestone and estimated the direct work volume and the type of skill required to perform each task. The team had no trouble developing this high-level plan. They next defined the sequences of these tasks and milestones, to understand which tasks they were planning to perform in parallel, versus in sequence.

The team's workflow model is shown in the lower part of Figure 7 (hexagons show the milestones, rectangular boxes show the tasks, and the left-right solid arrows show the task precedence links). Up to this point, the workflow model is similar to that used by traditional project scheduling tools such as Microsoft Project®. However, to capture the coordination work load that would arise from this extremely concurrent schedule, Michael's team augmented its task precedence model with two additional kinds of relationships between tasks: *Communication Links* and *Rework Links.* Communication Links (dash-dot two-way arrows between tasks) show that two tasks have tight technical interdependency, and will need to be tightly coordinated if they are executed in parallel.[7] Rework Links between two tasks (one-way, dashed arrows) indicate that any exception significant enough to require rework in the first task will trigger compensating rework in the second, "rework-dependent" task.

This complete workflow model captures the "total project effort"—that is, all of the information that needs to be processed to execute the project. This includes the *direct work,* plus all of the required *communication work* to coordinate interdependent tasks, the *supervision work* to answer workers' questions, and an allowance for the *rework* that will propagate between parallel tasks as changes or errors occur. This intuitive, graphical task and organization model, drawn

---

[7] The type of interdependency shown by the green arrows is what James Thompson (1967) termed "reciprocal interdependency" in his classic monograph, *Organizations in Action*.

on a whiteboard or the computer, helps each individual identify the work that must be performed, and it helps the team understand how different groups and tasks share precedence, coordination, supervision and rework interdependence throughout the project.

Having modeled the "total effort" for the project, Michael's team now modeled the project organization's information processing capacity to execute the total effort for his project. First Michael and the consultant modeled the participants in the project team. For each *Position* in the organization (staffed by an individual or a subteam of several persons with similar skills), Michael described the total number of full-time equivalent persons, and listed the set of project-relevant skills and skill levels that each person or subteam possessed (at a *high, medium* or *low* level). For example, the Chip Architect position was allocated one full time equivalent person (1 FTE), who had high *Logic Design* skill, medium *Floorplanning* skill and medium *Design Coordination* skill. Michael then described the team's reporting structure. The *Chip Architect*, the *Foundry Lead,* the *Marketing Team* and the *Test Engineering Team* reported to Michael, who filled the position of Project Lead. The Chip Architect and Foundry Lead each had two subteams reporting to them. Meetings also take time and need to be factored into a total effort analysis. Michael identified all regularly scheduled project meetings that he would expect to hold (the three rhomboid boxes at the upper left) and the team members who needed to attend these meetings (light, dashed arrows from participants to the meetings). The Organization design organization model is shown in the upper half of Figure 7 (above the horizontal line) in which person icons represent the positions and the rectilinear solid lines describe the project reporting hierarchy.

The team then linked the workflow model to the organization model. They assigned each task to one, and only one, responsible position. The curved, solid downward arrows from actors to tasks in Figure 7 show these assignments of responsible positions to tasks.

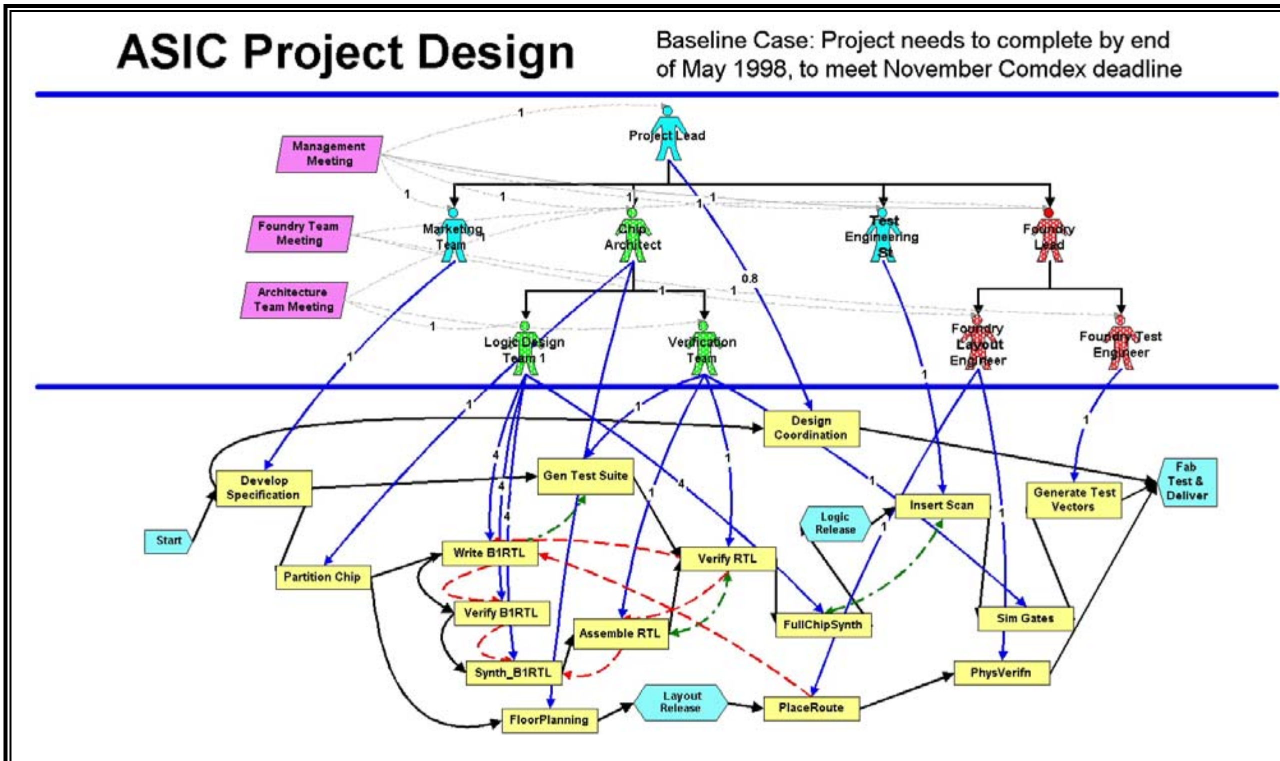

Figure 7. Baseline Organization design Model for ASIC Project

Finally, Michael described his company's decision-making policies for projects. He assigned *a medium* value to *Centralization* (how high up in the organization decisions get made); a *low* value to *Formalization* (to what extent team members would wait for meetings to coordinate, versus initiate ad-hoc queries to one another); a *high* value for *Matrix Strength* (the degree to which different disciplines on this project would be collocated versus located in separate functional groups); and a *medium* value for *Team Experience* (the extent to which members of this team had previously worked together). The consultant explained that each of these parameters affected the micro-decision-making behavior of workers, subteam leaders and project managers, and so would have an impact on which parts of his project team, if any, would become backlogged while executing this ASIC project.

The baseline project model was now complete. Adele and Michael could now simulate Michael's team executing the project, and assess the realistic delay and process quality risks associated with the proposed work plan and organization for the project.

Step 2: Simulate Project to Assess Risks for Baseline Case

Simulating the baseline plan for the PDA ASIC team produced a nasty surprise for Michael and Adele. When the total effort to accomplish this ASIC project was taken into account, the completion date would likely be in mid-September, even though a traditional Critical Path Method (CPM) analysis, which only considers direct work, had predicted it could be done by the end of May, as required. Figure 8 showed that overlapping logic design, layout and fabrication tasks as planned would trigger large amounts of coordination and rework for many activities—comparable to or exceeding the amount of direct work for some tasks like *Synth_B1RTL* (= Synthesize Block 1 Register Transfer Logic) and *Verify RTL.*

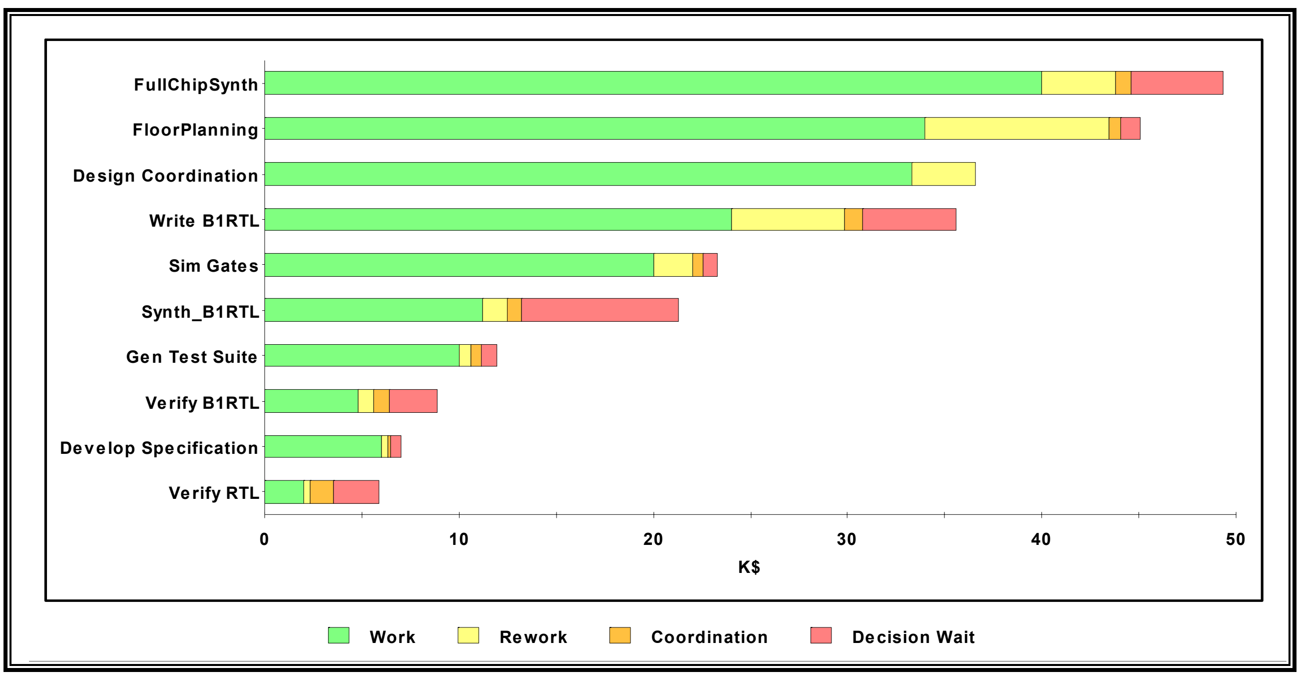

Figure 8. Total effort for tasks in the ASIC project

The VDT simulation predicted that the Chip Architect would become the worst bottleneck in this project team, falling behind by as much as eighteen working days during the middle of the project from a combination of doing his own work, coordinating with team members responsible for interdependent tasks, answering questions from his subordinates when they needed assistance, and attending project meetings. Worse, the simulation indicated that several tasks were likely to have severe Process Quality Risks, as backlogged team participants focused on catching up their own work and missed coordinating with interdependent colleagues or attending meetings. **Figure 9** shows that, for the *Generate Test Suite* task performed by the *Verification Team* and the *Write B1RTL* task performed by the *Logic Design Team,* there was a risk that more than 50 percent of work-related communications would be missed! The consultant informed the team that communication quality risks above 50 percent had been correlated with serious failures in the past (including the cable harness system for the launch vehicle described previously), and would thus become issues to be addressed in a project redesign.

Step 3: Search for Alternative Management Interventions

Faced with convincing evidence that their initial project schedule was unrealistic, Michael and his team began to think hard about the kinds of proactive interventions that they could make to mitigate these organizational risks and increase the project's likelihood of success.

Michael and his direct reports proceeded to iterate through a number of possible interventions by having the consultant model each intervention and "flight simulate" it (i.e. model and simulate the results of each intervention in just a few minutes) to predict its effect on project performance. They found that adding additional capacity to the *Chip Architect* position would shorten the project schedule by about three weeks. The *Verification Team* and *Logic Design Team* became the worst predicted bottlenecks in the project. Adding additional capacity to these two positions could further reduce the schedule, but the predicted project completion was still early August, more than two months later than its "3rd week in May" deadline.

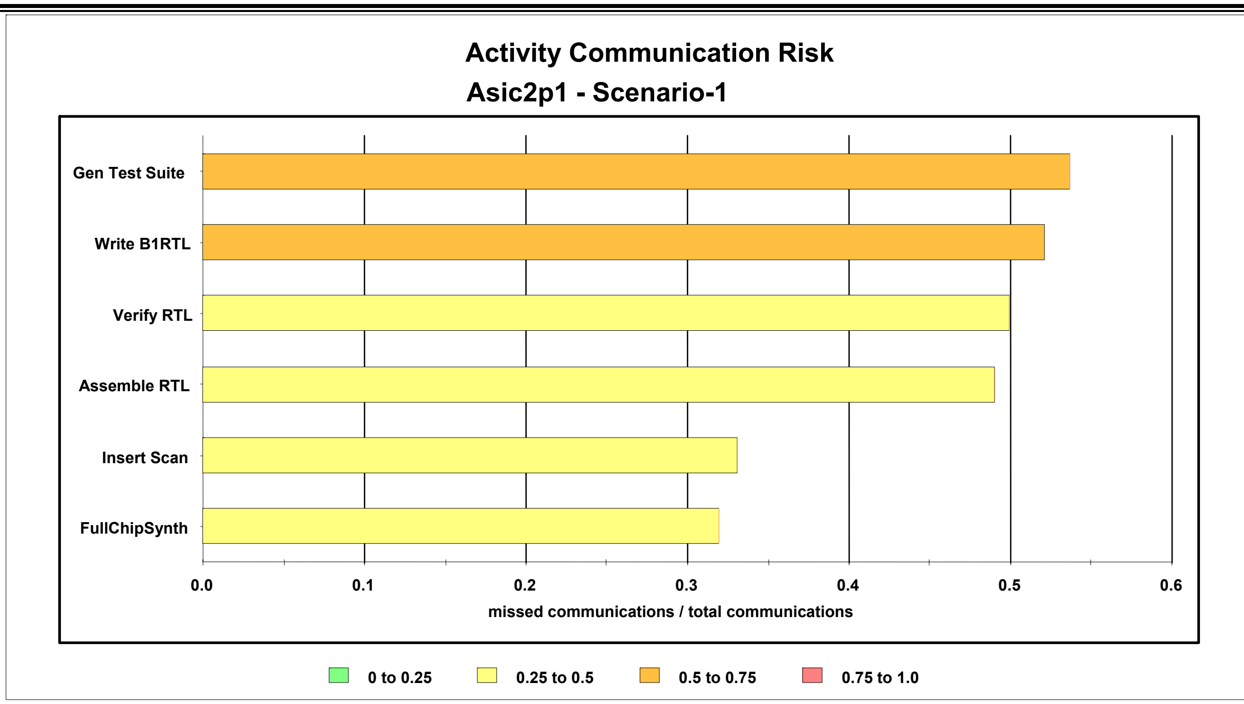

Figure 9. Predicted Communication Risk for ASIC Project Tasks

In *The Mythical Man-Month,* a classic study of the project that developed the operating system for an early IBM mainframe computer, Brooks (1975) explains why “throwing more bodies” at a delayed knowledge work project can be a futile way to try and accelerate it. Some tasks are not easily divisible, and new team members consume the time of existing team members in being brought up to speed, so that adding further staffing can sometimes actually further delay the project. Michael’s team found that three additional full-time equivalent staff was all that the team could productively absorb, yielding an early August completion, still more than two months behind the target date.

Michael’s team next explored the possibility of reassigning some tasks from the most backlogged positions to other team members who had the appropriate skills for those tasks but were less backlogged. Reassigning three such tasks took one more week out of the project duration.

The team now began to look at other kinds of interventions. They modeled and simulated the effect of changing skill levels for particular positions. In particular, the skill level of the Verification Team was lower than they would have liked, so they considered outsourcing the verification activities to a consulting firm that could provide high skilled ASIC verification engineers. They learned that this could save another three weeks on the project and improve the Communication Risk significantly, at only nominal additional cost.

Changing decision making policies did not help much in this case. Changing the level of Centralization from *Medium* to *Low* would accelerate the project by a few days, but would also increase Communication Risk unacceptably for some key tasks.

Step 4: Reduce Goals for the Project

So the team had exhausted the organization and work process changes that its leaders felt were feasible to implement, and was still looking at a mid-July completion vs. the target date of late May. The consultant had seen this situation in numerous previous Organization design sessions. She proceeded to ask Michael and his subteam leaders whether there was any way to lower the technical goals for the project by simplifying the design of the ASIC, without compromising its ability to meet critical business requirements. After a painful round of discussions among the team leaders, they concluded that they could reduce the clock speed of the chip. The speed of the chip that they were planning to use significantly exceeded its usage demands in this device. A slower—but still technically adequate—clock speed would greatly shorten and simplify chip layout and floor planning tasks.

With this technical change, assuming that three additional full-time equivalent staff were added to the team and verification tasks were outsourced, the simulation indicated that the team could meet its completion milestone in the third week of May with acceptably low process quality risks.

Through this systematic and repeatable organization design process, Michael’s team identified a feasible plan to meet the aggressive delivery milestone for its project, with a slight rescoping of its technical requirements, a minimum of additional staffing, and a small amount of focused outsourcing. The project actually completed in the second week of May 1998, and the prototype for the product was subsequently presented at the November 1998 Comdex trade show.

Social Benefits of Organization Design

We have shown how Organization Design, carried out with the managers of a company’s critical strategic projects, allows them to predict risks of schedule delays and quality failures early on, so they can proactively intervene to mitigate these risks. Equally or more importantly, experience in carrying out project organization design revealed that the very act of engaging in modeling and simulation could help to transform decision-making about organization design—previously viewed as a purely intuitive realm of management—from **ego-based conflicts** about competing alternatives into **fact-based discussions** about which set of implications was a better match with the goals of a particular project. This experience replicated Michael Schrage's findings about the value of performing simulations of all kinds in organizations, as reported in his book, *Serious Play"* (Schrage 2000).

We next discuss some valuable additional benefits that accrue to a project team when the entire management team engages in project organization design collectively, as a social process.

Develop shared mental models

A computer component manufacturer built an organization and work process model at the launch of a major new product development initiative. In one day, the major stakeholders—including the VP, who was the project sponsor, the project manager, and key functional leaders—specified the project business milestones, the major tasks to meet those milestones and the staffing. By the end of the day, simulation results showed a major staff imbalance: some tasks were overstaffed; others understaffed. One engineer showed visible relief when her management observed that she was going to be a bottleneck and that she would need to be freed of some ongoing responsibilities.

The project team fixed most problems quickly. The VP committed to resolve a few others. At the end of a day and a half, all of the team stakeholders reported that they understood the business drivers of the project, the milestones, the plan, and the structure of the team. They felt that they understood the project better after the 1-1/2 day modeling and analysis exercise than they normally did after a month or two of work on a new project. They said that they believed that they could work successfully to meet the scope-schedule-budget objectives, and they publicly committed to doing so. The presence and shared understanding of the risk-owning VP project sponsor motivated the team to complete the project successfully.

Transform ego-based conflicts into fact-based discussions

In the late 1990s, a global consumer products company was developing an innovative new consumer product that involved aspects of its paper goods technology used in making products like paper towels and diapers, together with its chemicals technology used in making products like toothpaste and shampoo. The expertise for these two areas came from different parts of the company with quite different cultures about the need for meticulous and lengthy consumer testing to discover all possible safety concerns (on the "wet" chemicals side), vs. the need to get innovative products to market rapidly to scoop competitors and gain early market share, and then fix any product bugs in follow-on products (on the "dry" paper goods side).

Working with members of the two groups separately failed to reconcile the desired work processes of the two sub-teams. The team leader finally decided that the only way to reconcile these differences was to convene a joint organization design session involving sixteen senior managers drawn from the two teams in a conference room equipped with a computer projector to display the project model and charts of its predictions. When we assembled the team in this room, a heated debate erupted. Two of the paper goods managers argued passionately that the team should freeze product specifications based on the results of early, small-scale consumer tests, and immediately place orders for the custom, "unit operations" manufacturing equipment (which required almost eighteen-months lead time) in parallel with ongoing consumer testing. The chemical group managers argued equally emphatically that the product involved unproven technology and might cause unanticipated injuries to customers, and so should be more extensively consumer-tested before committing to specifications and ordering equipment.

The consultants called for a truce, and proceeded to display and simulate the cautious vs. fast-track approaches advocated by the two groups with the entire team watching and participating. To our surprise, this "project flight simulation" process was transformational. Conducting this exercise as a social process gave all team members present the ability to view graphical representations of two different work process and organizational models so they could collectively understand the assumptions inherent in each. And their ability to get virtually instantaneous graphical feedback on the implications of each approach immediately changed the tone of the meeting from confrontational and argumentative to constructive and problem-focused. In about two hours, the project team was able to clarify their assumptions and debate the costs and risks *vs.* the revenue, ROI and market share benefits of the two approaches. The fast-track approach yielded an earlier product release date, at the cost of increased coordination effort for the team, and increased risk of costly late-cycle changes to the manufacturing equipment. The more cautious, sequential approach yielded significantly lower expected costs for the manufacturing equipment through reduced risk of rework, and lower coordination costs from less aggressive fast-tracking of tasks, but delayed the product release date by four months. The group discussed the value of four months of early market revenue, and the long-term increase in market share that they would enjoy from being first to market with this product. The team's review of the assumptions in the fast-track work process model convinced all of them that any needed changes to the product to address safety concerns could still be addressed through late cycle rework—albeit it costly—to the manufacturing equipment. Understanding this, the "chemical" team members became convinced that the risks of additional coordination effort and any costs for possible late changes to the manufacturing equipment were a small price to pay for the benefits of earlier time-to-market using the fast-track approach.

Schrage's (2000) book, *Serious Play,* describes and illustrates exactly these kinds of benefits for a wide range of business modeling and simulation approaches and tools. So, it is perhaps not surprising that the same benefits would show up for simulations of project teams executing fast-track work processes.

Discussion

The VDT framework has severe limitations that restrict its applicability to a narrow range of organizations engaged in relatively routine, albeit complex and highly concurrent, project work. In concluding, we discuss these limitations and some of our past and ongoing research aimed at addressing these limitations.

Limitations of VDT Framework

First, VDT requires that the work process be routine enough that all tasks, their sequence and technical interdependence can be predefined. Moreover, the framework requires that any exceptions that arise in the course of executing the project can be modeled as simply adding work volume to be predefined set of activities, without changing the set of activities with their sequence. In parallel it requires that the organization be defined with a fixed set of actors, with fixed levels of skill, organized in a fixed reporting structure, and with fixed assignments of actors to tasks. Although these constraints might seem very limiting, it turns out that they are satisfied in many kinds of product development, software development and business process development efforts.

As stated above, we started in the "easiest corner of the space of organizations" and by 1996 were able to begin analyzing the performance of routine project organizations with some precision as part of a formal organization design process (Cohen 1992; Christiansen 1993). Over the next 10 years, we gradually extended the original prototype VDT framework (which we call VDT-2) by a series of extensions to the model's representation and reasoning to address progressively more dynamic tasks executed by more flexible organizations.

The extensions to VDT have gradually moved it from an "information flow physics" model of routine project teams to a model of less routine project and service/maintenance work that includes some aspects of "organizational chemistry". Successive researchers in our group have added representation and reasoning to model: non-routine tasks and flexible organizations such as found in health-care delivery and other kinds of service/maintenance organizations (Fridsma & Thomsen 1998; Cheng & Levitt 2001); goal conflict among project participants (Thomsen 1998); and the use of social networks and knowledge networks for handling exceptions (Palazollo *et al.* 2003; Buettner 1995; Lambert *et al.* 2001). Figure 10 summarizes the trajectory of the VDT research program over the past 17 years.

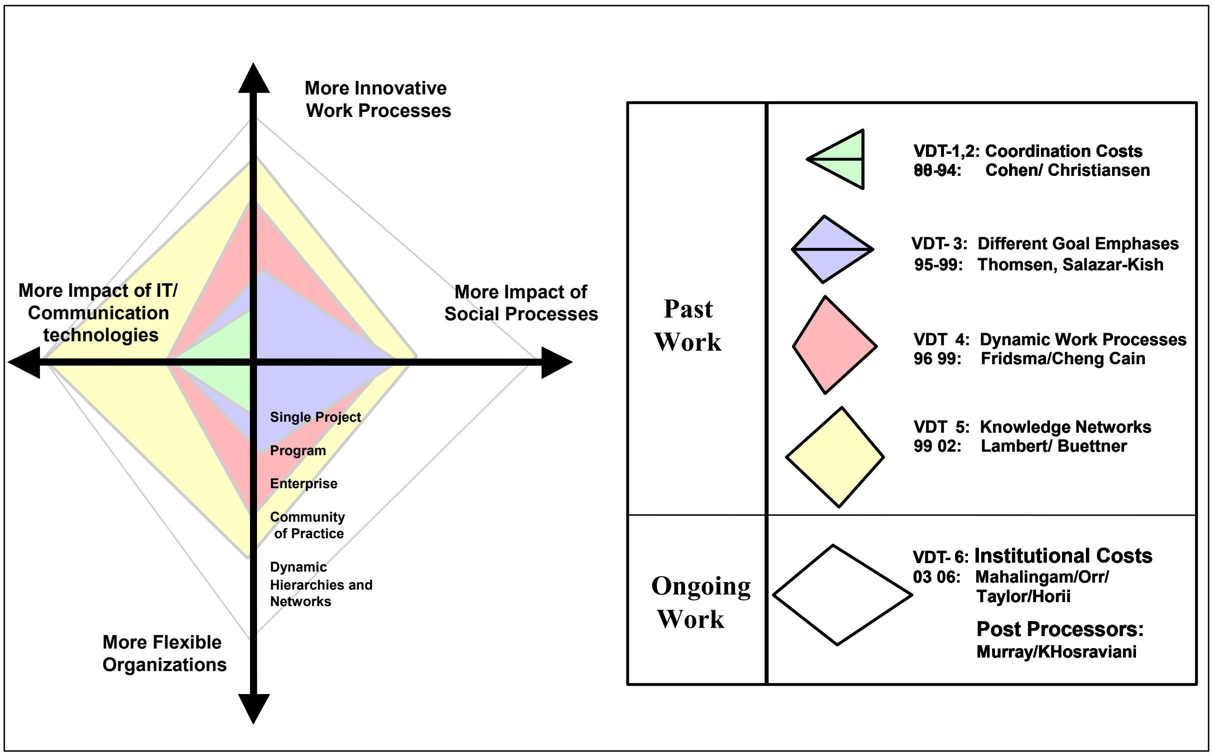

Figure 10. Trajectory of VDT Research Program. The development of VDT began in 1988. By 1996, VDT-2 had been calibrated and validated to the point that it was beginning to be used for real organization design. Since then, a succession of VDT researchers has progressively extended the model framework by adding representation and reasoning to address additional flexibility and complexity along each of the four axes in this graph.

Our current research is extending VDT to: model actors from multiple cultures engaged in global projects (Horii 2004); to model knowledge flows in project teams (Nissen & Levitt 2004); and to develop organization design optimizers that exploit "evolutionary computing" techniques such as genetic programming to "evolve" an initial baseline organization design into a more optimal project organization for a given work process and context (Khosraviani 2004a, 2004b). The latter line of work offers an exciting potential new way to validate contingency theory by evolving the organizational structure that best satisfies fitness criteria for survival in a given task and environmental context.

VDT represents a limited, but significant, first step in developing quantitative, model-based analysis tools for designing organizations. It took us eight years to validate and calibrate VDT to a level that could provide useful support to practitioners of organizational design for the simplest possible kind of organization—product development teams with well-defined tasks, static skill sets for actors, a fixed organization structure for the team, and clear and congruent goals for all actors. The extensions described above are in various stages of development and initial results from these extension efforts are encouraging. However, a great deal of work remains to be done to achieve levels of external validation for any of them that come close to those already achieved for the original single project VDT-2.

Moreover, there is a very real boundary of complexity that limits advancements beyond the state-of-the-art for organizational analysis models. Adding additional representation and reasoning capabilities to an already relatively complex model like VDT exponentially increases the difficulty of validating and calibrating the many possible interactions between its various micro-behaviors. Thus, our current efforts to introduce features like knowledge flow will require that we eliminate some of the extant information flow and coordination features of the model, to prevent VDT-X from becoming just as opaque and intractable as real world organizations.

Ongoing developments in computer simulation languages and toolkits have eliminated a great deal of the duration, tedium and difficulty involved in developing computational analysis models to support organizational design. Thus, the major barrier to further progress in developing and validating theory and tools to support analysis of a wider range of organizations has become the creativity and ingenuity of computational organizational modelers, rather than the power and ease-of-use of modeling languages and tools. It is truly an exciting time to be conducting research on organization design!

References

Axtel, R., Axelrod R., Epstein, J.M., & Cohen M.D. (1996). Aligning Simulation Models: A Case Study and Results. *Computational and Mathematical Organization Theory.* ***1****:2,* 123-141.

Bonini, C. P., (1967). *Simulation of Information and Decision Systems in the Firm, (A Ford Foundation Prize Dissertation),* Chicago, Markham Pub. Co.

Brooks, F.P., (1975). *The mythical man-month : essays on software engineering,* Addison-Wesley, Reading, MA.

Buchanan, B.G. and E.H. Shortliffe, (1984). *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project,* Addison-Wesley, Reading, MA.

Buettner, R.R., (1995). "Simulating the Impact of CKM on Project Performance," *Proceedings of the CASOS 2002 Conference,* CMU, Pittsburgh, PA.

Burton, R.M. and B. Obel, (1995, 1998, 2004). *Strategic Organizational Diagnosis and Design, 3rd Edition,* New York, Kluwer.

Burton, R.M., Lauridsen, J. and Obel, B., "Return on Assets Loss from Situational and Contingency Misfits," *Management Science* 48:11 (2002), pp. 1461-1485.

Carley, K.M. and D.M. Svoboda, (1996). Modeling organizational adaptation as a simulated annealing process. *Sociological Methods and Research,****25****:1,* 138-168.

Carley, K.M. and Lin, Z., "A Theoretical Study of Organizational Performance under Information Distortion," *Management Science* 43:7 (1997), pp. 976-997.

Carley, K.M., "Organizational Learning and Personnel Turnover," *Organization Science* 3:1 (1992).

Carroll, T., and R.M. Burton, (2000). "Organizations and Complexity: Searching for the Edge of Chaos," *Computational & Mathematical Organization Theory* ***6****:4*, pp. 319-337.

Cheng, C.H., and R.E., Levitt, (2001). "Contextually changing behavior in medical organizations" *Proceedings of the 2001 Annual Symposium of the American Medical Informatics Association,* Washington, DC.

Christiansen, T.R., (1993). *Modeling Efficiency and Effectiveness of Coordination in Engineering Design Teams* doctoral dissertation, Department of Civil and Environmental Engineering, Stanford University.

Cohen, G.P., (1992). *The Virtual Design Team: An Object-Oriented Model of Information Sharing in Project Teams* doctoral dissertation, Department of Civil Engineering, Stanford University.

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

Cyert, R.M., and J.G. March, (1963 and 1992). *A Behavioral Theory of the Firm, 2nd Edition,* Cambridge, MA: Blackwell Business.

Dym, C.L., and R.E. Levitt, (1991). *Knowledge-Based Systems in Engineering.* New York, McGraw-Hill.

Fridsma, D.B., and J. Thomsen, (1998). "Modeling Medical Protocols for Organizational Simulation: An Information-Processing Approach," *Computational and Mathematical Organization Theory,* ***4****:1,* 71-95.

Galbraith, J.R., (1974) "Organization Design: An Information Processing Approach", *Interfaces 4 (5):* 28—36.

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

Hannan, Michael T. and John H. Freeman, (1977). "The Population Ecology of Organizations," *American Journal of Sociology 82:* 929-964.

Jin, Y. and R.E. Levitt, (1996). "The Virtual Design Team: A Computational Model of Project Organizations,"*Journal of Computational and Mathematical Organization Theory* ***2****,3*; 171-195

KHosraviani, Bijan and Raymond E. Levitt, (2004a), "Organization Design Using Genetic Programming," *Proceedings of 2004 North American Association for Computational Social and Organizational Science (NAACSOS) Conference,* Pittsburgh, PA, June 27-29, 2004

KHosraviani, Bijan, Raymond E. Levitt and John. R. Koza (2004b), "Organization Design Optimization Using Genetic Programming", In Keijzer, Maarten (editor). *Proceedings of 2004 Genetic and Evolutionary Computation Conference (GECCO-2004).* International Society of Genetic and Evolutionary Computation, Seattle, WA, July 2004.

Kunz, J.C., R.E. Levitt and Y. Jin, (1998). "The Virtual Design Team: A Computational Simulation Model of Project Organizations," *Communications of the Association for Computing Machinery* ***41****:11* (1998), pp. 84-92. Jin, Yan, and Raymond E. Levitt.

Lambert, M., J.C. Kunz and R.E. Levitt, (2001). "Hierarchies and Transactive Memory Systems: Crafting a Model of Flexible Exception Handling," *CASOS 2001 Conference,* CMU, Pittsburgh, PA., 79-83.

Law, A.M. and Kelton, D., *Simulation Modeling and Analysis* (Second Edition) New York, NY: McGraw-Hill (1991).

Leavitt, H., (1951), "Some Effects of Communication Patterns on Group Performance," *Journal of Abnormal and Social Psychology* ***46****:* 38-50.

Levinthal, D. and J.G. March. (1981) "A Model of Adaptive Organizational Search". *Journal of Economic Behavior and Organizations, 2:* 307-333.

Levitt, R.E., and J.C. Kunz, (1985). "Using Knowledge of Construction and Project Management for Automated Schedule Updating," *Project Management Journal, 16 (5):* 57-76.

Levitt, R.E., G.P. Cohen, J.C. Kunz, C.I. Nass, T. Christiansen, and Y. Jin, (1994). “The ‘Virtual Design Team’: Simulating How Organization Structure and Information Processing Tools Affect Team Performance,” in Carley, K.M. and M.J. Prietula, editors, *Computational Organization Theory,* Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ,.

Levitt, R.E., Thomsen, J., Christiansen, T.R., Kunz, J.C., Jin, Y. and Nass, C., (1999). “Simulating Project Work Processes and Organizations: Toward a Micro-Contingency Theory of Organizational Design,” *Management Science 45 (11):* 1479-1495.

Levitt, R. E., M. Fyall, P. Bjornsson, W. Hewlett, III, T. Casebolt, (2002). “When Information Flow in Project Organizations Becomes Turbulent: Toward an Organizational Reynolds Number,” *Proceedings of the Conference on Agent-Based-Modeling in the Social Sciences,* UCLA Lake Arrowhead Conference Center, May 2002, p3.

Levitt, R.E., (2004). “Computational Modeling of Organizations Comes of Age", *Journal of Computational and Mathematical Organizational Theory, 10 (2):* 127-145.

Louie, M.A. , K.M. Carley, L. Haghshenass, J.C. Kunz, and R.E. Levitt, (2003). “Model Comparisons: Docking ORGAHEAD and SimVision. NAACSOS Conference Proceedings, CMU, Pittsburgh, PA.

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

March, J.G., “Exploration and Exploitation in Organizational Learning,” *Organization Science* 2:1 (1991), pp 71-87.

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

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

Nissen, M.E. and Levitt, R.E., (2004). “Agent-Based Modeling of Knowledge Dynamics,” *Knowledge Management Research & Practice 2 (3):* 169-183.

Nogueira, J.C., *A Formal Model for Risk Assessment in Software Projects* doctoral dissertation, Department of Computer Science, Naval Postgraduate School (2000).

Palazollo, E.T, A. Ghate, R. Dandi, A. Mahalingam, N. Contractor and R. Levitt, (2002). Modeling 21st Century Project Teams: Docking Workflow and Knowledge Network Computational Models, *Proceedings of the CASOS 2002 Conference,* CMU, Pittsburgh, PA.

Schrage, Michael, (2000). *Serious play: how the world's best companies simulate to innovate,* Harvard Business School Press, Boston, MA.

Scott, W.R., *Organizations: Rational, Natural, and Open Systems* (Fifth Edition) Upper Saddle River, NJ: Prentice-Hall (2003).

Simon, Herbert A., *The Sciences of the Artificial,* MIT Press, Cambridge, MA, (1996).

Stinchcombe, Arthur L. (1990). *Information and Organizations,* Univ. of California Press, Berkeley, CA.

Tatum C.B., Decision Making in Structuring Construction Project Organizations. *Ph.D. Dissertation. Department of Civil and Environmental Engineering. Stanford University* (1983).

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

Thomsen, J, (1998). *The Virtual Team Alliance (VTA): Modeling the Effects of Goal Incongruency in Semi-Routine, Fast-Paced Project Organizations* doctoral dissertation, Department of Civil and Environmental Engineering, Stanford University.

Thomsen, J., R.E. Levitt, J.C. Kunz, C.I. Nass, D.B. Fridsma, (1999). “A Trajectory for Validating Computational Emulation Models of Organizations”, *Computational & Mathematical Organization Theory, **5**:4,* 385-401.

Tushman, M., and D. Nadler (1978), "Information Processing as an Integrating Concept in Organizational Design," *Academy of Management Review*, 3, 613-624.

Wong, S.S. and , R.M. Burton, (2000) ."Virtual Teams: What are their Characteristics, and Impact on Team Performance?," *Computational & Mathematical Organization Theory* ***6****:4*, pp. 339-360.