CONCURRENT ENGINEERING: Research and Applications

Modeling and Simulation in Enterprise Integration—A Framework and an Application in the Offshore Oil Industry

L. C. Christensen and T. R. Christiansen*

DNV Research, Veritasveien 1, 1322 Høvik, NORWAY

Y. Jin, J. Kunz and R. E. Levitt

Stanford University, Civil Engineering Department, Stanford, CA 94305, USA

Received 8 January 1996; accepted in revised form 10 June 1996

Abstract: This paper presents a framework and methodology that make systematic use of model definition, formalization, and analysis to enhance enterprise integration in engineering projects. Our definition of enterprise models is based on a philosophy for thinking about enterprise in terms of "an *Organization*, carrying out a set of *Processes* to create one or more *Products* which satisfy predefined *Objectives*." This framework, which we have denoted the OPPO, illustrates our belief that enterprise models must give a complete and correct description of relevant aspects of reality in order to address their stated purpose. Based on the OPPO framework and an information processing view of project enterprise, we formalize our models using a methodology for describing coordination requirements. This is based on a set of interaction matrices to develop so-called "houses of complexity, uncertainty and interdependence" that describe important dependencies between project requirements (objectives), deliverables (products), activities (process), and team members (organization). We use these dependencies as a basis for deriving measures of the coordination which must take place between various project participants during project execution. We can then use these coordination requirements as input to analysis, in the Virtual Design Team (VDT) object oriented discrete event simulation environment. The simulation results can be used for systematically assessing predicted effects of proposed changes. The framework, methodology, and analysis is illustrated by an example from engineering design of subsea modules for oil and gas production in offshore field development. The results from simulation give various performance measures, including critical path duration, work volume (a substitute for cost), and process performance in coordination (error handling and communication attendance).

Key Words: Engineering management, Process modeling, Quality Function Deployment, Information models, Monte-Carlo simulation.

1. Introduction

The objective of this paper is to describe an approach to enhancing enterprise integration by modeling of engineering projects, and a methodology for analyzing enterprise models to predict probable effects of proposed changes in the project organization. We illustrate our description with an example taken from offshore field development in the Norwegian part of the North Sea.

Our motivation from modeling and analyzing project enterprise is the increasing pressure for improved performance facing engineering projects in almost every industry today, and the belief that "there is a general lack of knowledge and appreciation of the capabilities of integrating systems to improve the productivity and economic return of enterprises of all types" [20]. Improvement requires a clear understanding of the current situation, a vision of the desired situation, and a strategy for achieving the necessary change [7]. It is our belief that these requirements may be addressed by enterprise modeling and analysis.

Given our objective, we first describe our framework for thinking about project enterprise in Section 2. Next, we describe our methodology for building formal enterprise models, and an associated enterprise modeling toolkit in Section 3. Then, in Section 4, we briefly describe a discrete event simulator, The Virtual Design Team-VDT [11], which uses our enterprise model of engineering design projects as input, and produces a set of predicted performance measures resulting from proposed changes in the execution of the project task or organization of the project team. All of these sections are illustrated by a project to design subsea modules for offshore oil production.

It is our intention to give an overview of our research activity, rather than provide full detail on all aspects. Further explanation can be found in References [1–3,11,12].

2. A Framework for Modeling Project Enterprise

Reality is often complicated and confusing, and insight is rarely achieved without considerable simplification. In order to improve project enterprise we must understand it,

*Author to whom correspondence should be addressed.

1063-293X/96/03 0247-13 $10.00/0

and thus the challenge in enterprise modeling is to make the necessary simplifications, without losing elements that are essential to representation and reasoning. Also, since enterprise modeling takes time and costs money, the model must be developed with a justifiable purpose in mind. In other words, enterprise models should be developed in the spirit of giving a ***complete*** and ***correct*** description according to model purpose.

Our framework for thinking about enterprise is in terms of "an *organization,* carrying out some (set of) *process(es)* to create *products* which satisfy predefined *objectives*." That is, we wish to highlight the four pillars on which purposeful human action rests: why we act (the objective), what is the result of action (the products), when and how we act (the process), and with whom we interact (the organization). Project enterprise is thus characterized by an assigned team of people, working together for a planned period of time to deliver according to specification, and thus achieve a stated purpose.

2.1 The Statfjord Subsea Satellites Engineering Design Project

Engineering design projects in offshore field development are undertaken to develop design drawings and procurement specifications for complex and costly installation components. Typical design objects include fixed and floating structures, drilling and processing systems, and control and automation systems.

The Statfjord Subsea Satellites (SSS) project involves development of a set of subsea production units for the Statfjord Field. A total of eight subsea production units were installed at various locations, in order to increase the total amount of production, as field characteristics changed during production from three existing platforms. The subsea units, which produce and preprocess crude oil, were connected to an existing platform for further processing and transportation. The total SSS project involved several hundred thousand person-hours, of which the engineering design contract constituted somewhere around 30 thousand person-hours, carried out over fifteen months by an engineering design team varying between ten and twenty-five persons.

Figure 1 shows an illustration of one of the Statfjord Subsea Satellites. We shall use the SSS project throughout this paper as an example application to illustrate our framework and methodology for enterprise engineering.

We believe that ***model completeness*** is addressed by a holistic description of why project enterprise is undertaken, how and by whom it is undertaken, and what the result of the enterprise is. In the next subsections, we illustrate the modeling of the SSS project.

2.2 Describing Objectives and Products

To describe *why* projects are carried out, we model the project requirements (objectives). Similarly, to describe *what* is the result of project execution, we model solution

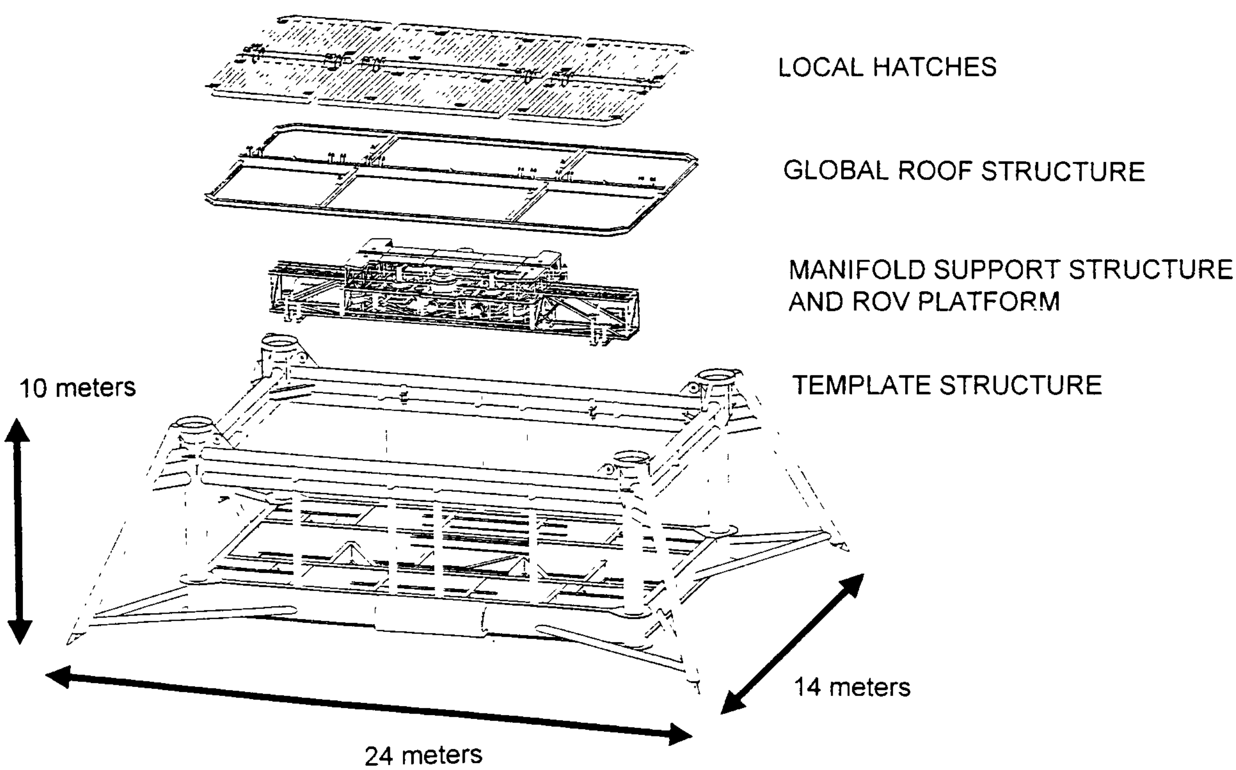


Figure 1. The project deliverable from the Statfjord Subsea Satellite (SSS) project.

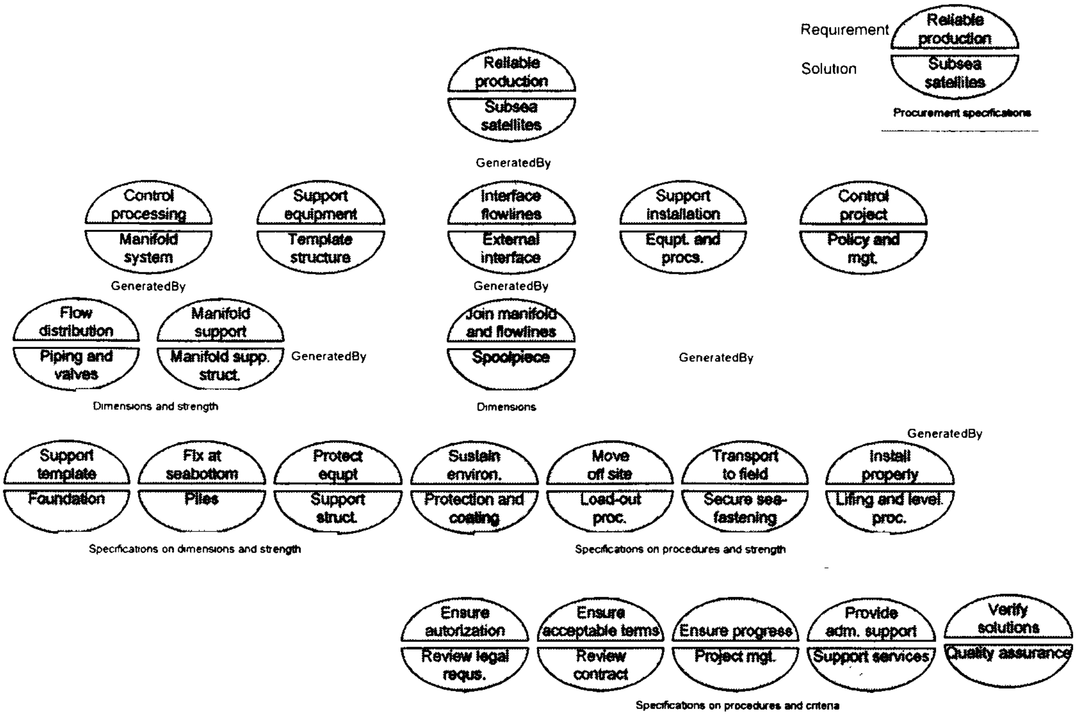


Figure 2. Requirements-solution breakdown structure for the SSS project.

deliverables (products). To represent objectives and products we use functional decomposition, and the FUTS technique [19], in which a top level requirement is met by a corresponding top level solution. This solution generates a set of lower level requirements, which in turn are met by more detailed solutions (which in turn generates new detailed requirements, and so on, until a suitable level of detailing for describing procurable specifications is reached). Figure 2 illustrates the requirements-solution breakdown structure for the subsea satellites in the SSS project.

FUTS modeling of requirements and solutions may be thought of in terms of required and realized project results. The difference between them determines project performance, which includes measures of duration and cost (project efficiency), as well as the behavior and characteristics of project deliverables (project effectiveness or quality).

2.3 Describing the Process

The question of *how* relates to planning and execution of the project (the process dimension). In the same manner that the difference between objectives and product deliverables defines product preformance, the difference between planning and execution defines process performance. We describe processes in terms of *activities* related by precedence (predecessor successor) relations, showing the logic of project execution.

Figure 3 shows a simplified project plan for engineering design of the SSS project. Note the high degree of concurrency caused by tight scheduling of activities to carry out Engineering, Procurement, and Construction (EPC) one full year faster than was typical in past projects. Note also that the figure only shows the engineering activities for the first of a total of eight modules. Engineering analysis for this first module involved a major part of the work, and the work for the remaining seven modules was performed as a "production line," in direct continuation of the activities shown.

2.4 Describing the Organization

The question of *who* relates to responsibilities of and relations between project team participants (the organizational dimension). We describe organizations by their formal policies and structural relations, and *actors* in terms of their abilities (craft, skill, and experience) and preferences. The difference between project policies and personal preferences of the project team participants will determine how planned action is translated into actual behavior. This determines organizational performance, and influences and constrains process and product performance.

Figure 4 illustrates the engineering design project team organization and their relations to other project team members. Note how dual reporting is caused for sub-team leaders (lead engineers) by two different organizational hierarchies, the functional and project lines of command and control. This is typical for engineering project organizations, and is often referred to as a matrix structure [6]. In the SSS project, the matrix was dominated by the

No	Name	Responsible		Volume
1	Review authority req s	Project mgr		80 hours
2	Review contract	Project mgr		80 hours
3	Template sys eng	Lead struct.		602 hours
4	Manifold sys eng	Lead struct.		155 hours
5	Load case description	Lead struct		80 hours
6	Load-out analysis	Lead struct		80 hours
7	Transportation analysis	Lead struct		80 hours
8	Seafastening analysis	Lead struct		80 hours
9	Lift and level analysis	Lead struct		200 hours
10	Lifting frame design	Lead struct		380 hours
11	In-place analysis	Lead struct		120 hours
12	Earthquake analysis	Lead struct		80 hours
13	Snag analysis	Lead struct		120 hours
14	Dropped object analysis	Lead struct		252 hours
15	Foundation design	Lead struct		40 hours
16	Pile-in-place analysis	Lead struct		136 hours
17	Pile drivability analysis	Lead struct		80 hours
18	Unpiled stability analysis	Lead struct		80 hours
19	Routing of manifold design	Lead manif		637 hours
20	Pipe stress analysis	Lead manif.		793 hours
21	Manifold supp struct analysis	Lead struct		2481 hours
22	Piping specifications	Lead manif		300 hours
23	Piping corr /erosion analysis	Lead manif		107 hours
24	Cathodic protection design	Lead struct		466 hours
25	Material coating and spec	Lead struct		320 hours
26	design part of DFI resume	Project mgr		160 hours
27	Databook design	Project mgr.		360 hours
28	Drafting clash check	Lead struct		160 hours
29	Piping drafting	Draft sup		40 hours
30	Structural drafting	Draft sup		1420 hours
31	Misc drafting	Draft sup		460 hours
32	Pipeline spoolpiece	Lead struct		490 hours
	Project management	Project mgr		2201 hours
	Project services	Project eng		2390 hours
	Quality assurance	Project eng		330 hours
	Milestone 1	Project mgr		
	Milestone 2	Project mgr		
	Finish	Project mgr		
	Total			16210 hours

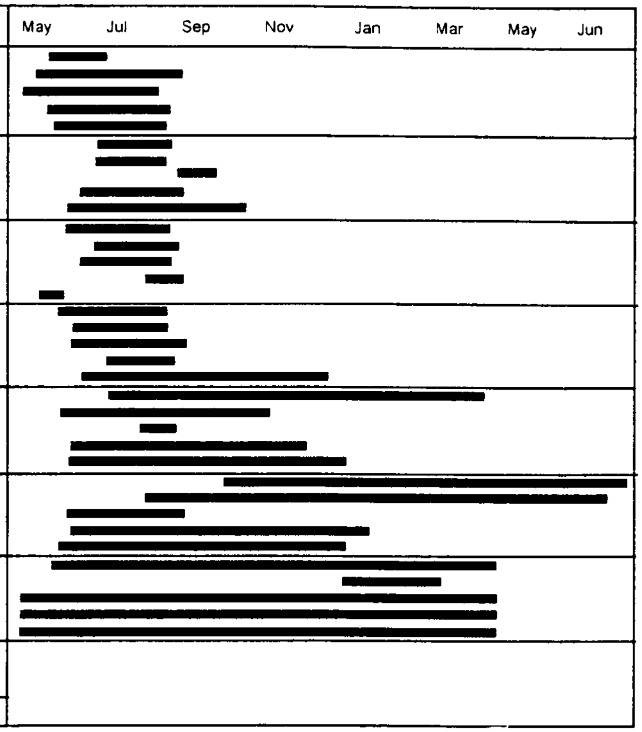


Figure 3. Project plan for engineering design phase of the SSS project.

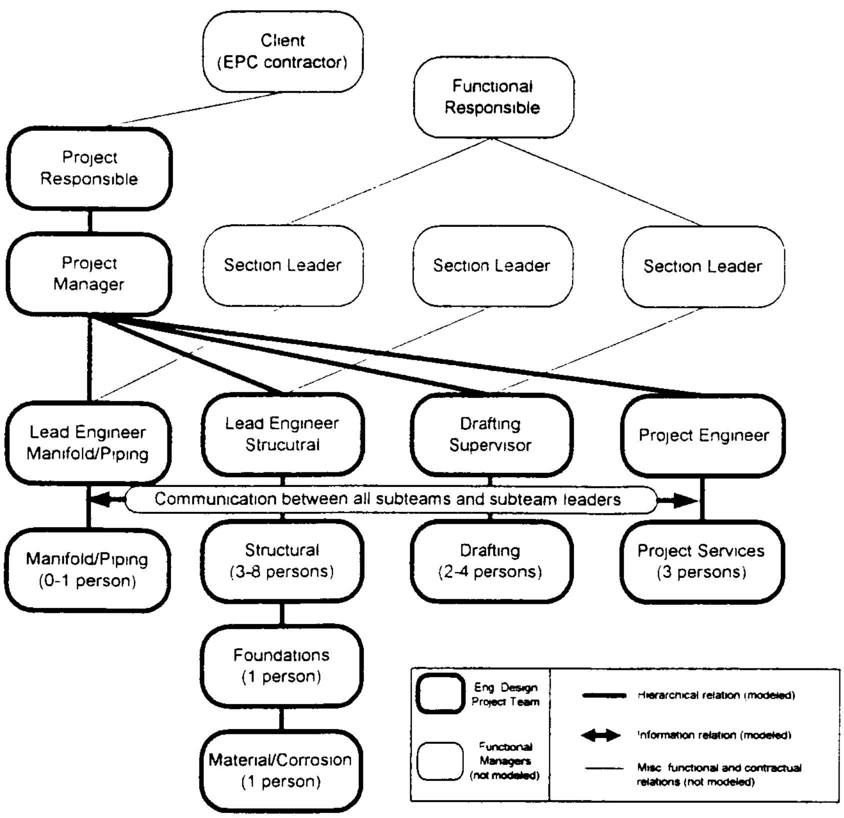


Figure 4. Engineering design team organization for the SSS project.

project manager, who controlled budgets, schedules, performance evaluation, and incentive structures. This is the hallmark of a so-called strong matrix. Moreover, the engineering design team was physically co-located, which enabled them to rely on informal communication as their means for handling intense coordination requirements.

2.5 Explaining Project Performance

In the above, we have pointed out how performance may be thought of as a measure of the fit between planned and actual project execution, and how this can be applied to products, processes, and organizations. This relates to a view of ***model correctness*** as the ability of the model to capture differences between ideal and real situations (i.e., planned action vs. actual behavior). Below, we define action in enterprise using two very different views of causal logic, and compare their different predictions.

Purposeful engineering professionals like to believe that they inhabit a rational world, where enterprise is explained according to "a logic of intention" [13]. In this ***normative*** view, causality is explained starting from objectives, which define some set of required products. One or more processes are devised to create the products, and a suitable organization is designed to carry out the processes (which create the products, which satisfy the objective). Thus, project enterprise is seen as a rational means for achieving fulfillment of stated objectives.

A natural systems view of enterprise [15] explains causality according to "a logic of implication" [13]. In this ***descriptive*** view, enterprise is defined by some set of individuals (the organization). Between organizational members, there exists a mix of rational and irrational relations which determine what processes the organization can (and cannot) carry out. The possible processes determine a set of possible products, for which objectives are devised to explain (and defend) the existence of the organization. Thus, enterprise is seen as a-posteriori justification of action.

In reality, neither of the above models explain human enterprise in full. Reality is filled with bounded rationality [16], and causality can be viewed by an ***interpretative*** logic of both intention and implication. That is, a logic where causality is explained according to the chosen interpretation of reality. Note how the word made up by the first letter of our four enterprise dimensions, *OPPO*, is a palindrome. This symmetry symbolizes the way in which distance from enterprise, whether in time, space, or function, precludes accurate determination of causal mechanisms. That is, the difference between cause and effect becomes difficult to observe, but all the more important to describe.

Figure 5 illustrates the different accounts for causal logic in the SSS project. In an intentional explanation, the SSS project was carried out in order to increase the recoverable reserves (life cycle profit) from the Statfjord Field. This was met by installing a set of subsea production units to operate a set of additional wells. Thus, the SSS project plan specified design, building, and installation of eight subsea modules. Finally, the SSS project team was set up in order to carry out the project.

Conversely, an implied explanation starts from a set of engineering organizations looking for work [14]. The SSS project was organized in terms of three main contractors, with a number of subcontractors to each (lots of work for everyone). The process involved segregated work packages, and separate subprocesses. Because of the resulting communication problems, a large number of variation orders (VOs) were generated to handle required changes due to confusion and misunderstanding between different dependent organizations. Consequently, project duration became the critical variable, and successful installation of the modules on time

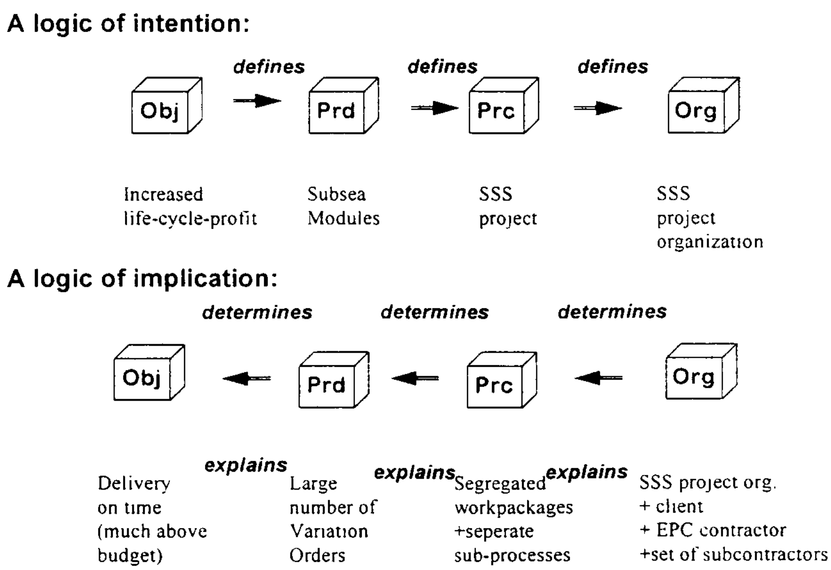


Figure 5. Different accounts for causal logic in the SSS project.

became the number one priority objective, delivered at a cost 50 percent above the original budget estimate.

2.6 Summarizing the Complete OPPO Modeling Framework

In Figure 6, we summarize the above framework for enterprise modeling [4]. Starting from the overall objective, we use functional decomposition to derive a structured set of requirements as a "desired model." Given the set of requirements and associated product solutions, and the various types of dependencies between them, we define a sequence of activities to produce the desired deliverable. Between these activities, various types of precedence relations will determine the required control mechanisms throughout the process. By assigning responsibility for the various activities to project team participants (actors), we define a set of related actor dependencies, which determine communication needs. The execution of the project plan by the project team will produce the deliverable (realized solution), which may be compared to the desired solution. Thus, project performance can be explained as the result of definition, planning, and assignment of dependencies (*coordination requirements*), and subsequent handling of these dependencies in execution (*coordination mechanisms*).

3. A Methodology for Formalizing Enterprise Models

In order to use enterprise models as the basis for predicting probable effects of proposed changes in projects, we must turn our modeling framework into a formal *methodology* for developing executable models.

3.1 An Information Processing Model of Coordination in Engineering Design

An important aspect of our model is to study how the match between load, structure, and capability of actors determines the capacity of project teams to handle coordination requirements arising between actors that are dependent on each other for producing, consuming, and sharing information to carry out the project plan.

In analogy with physical structures, we view the project team organization as a structure of relations, and take an information processing view of project execution as a flow of tasks being processed by information processors [8]. Both physical and organizational structures consist of elements with given material properties, connected by nodes in a given configuration. Both are subject to loading from their environment, and have capacity to meet this load determined by their material properties and configuration. As noted earlier, the match between required and realized behavior under load determines the performance of the structure.

Figure 7 illustrates our view of project team members as information processing entities (from [5]), constrained by a set of hierarchical and communicational relations, and subjected to load determined by their responsibility for tasks in the project plan.

Coordination capacity is determined by actors' handling of coordination load, and depends on the skill, experience, policies, and preferences of project team members, as well as a set of coordination mechanisms for handling dependencies between project team members. This defines the information processing behavior of the various actors, in terms of decision making about attention allocation and participation [13].

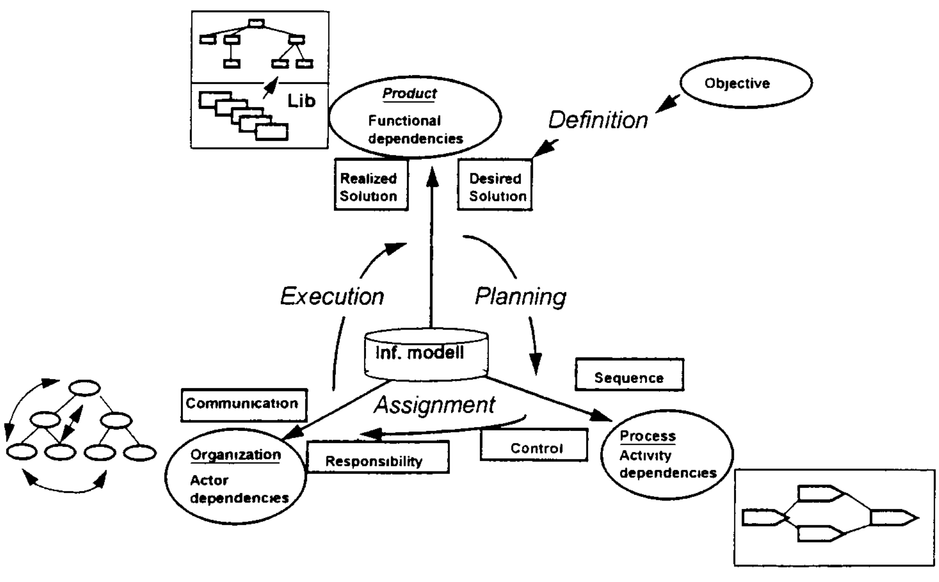


Figure 6. A summary overview of the OPPO enterprise modeling framework.

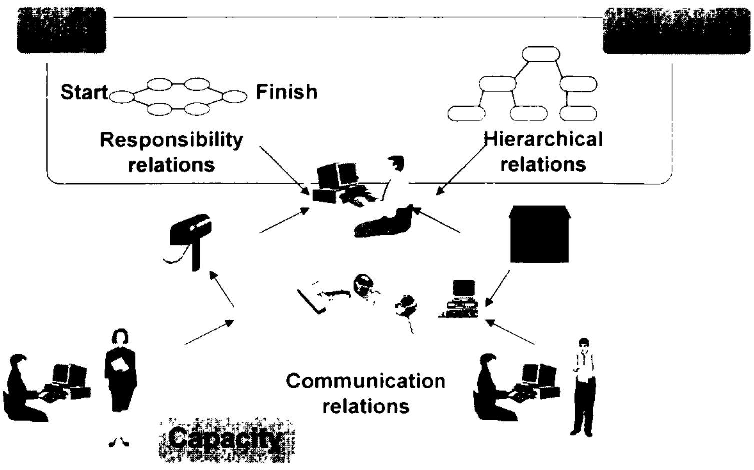


Figure 7. Engineering design teams as information processing entities.

3.2 A Formal Model of Coordination Load, Structure, and Capacity

To account for the dual nature of causality, our methodology includes constructs to describe the differences between planned action and actual behavior. Thus, our *load model* attempts to define and operationalize coordination measures which influence how projects are actually carried out.

To implement the coordination *load model,* we use a set of matrix tools based on Quality Function Deployment (QFD) [10], to calculate project specific measures of *complexity* of various parts of the project task, *uncertainty* of various parts of the project plan, and *interdependence* between various members of the project team.

We use these measures to quantify the probabilities of failure in producing solutions and satisfying requirements, the required communication frequency, and the required participation by project team members [3]. These are, in our view, a crucial part of describing how work processes are carried out in the real world by organizations consisting of boundedly rational (human) actors [16].

Figure 8 illustrates our methodology for translating requirements and corresponding solutions, together with activity plans and organization charts for the project team,

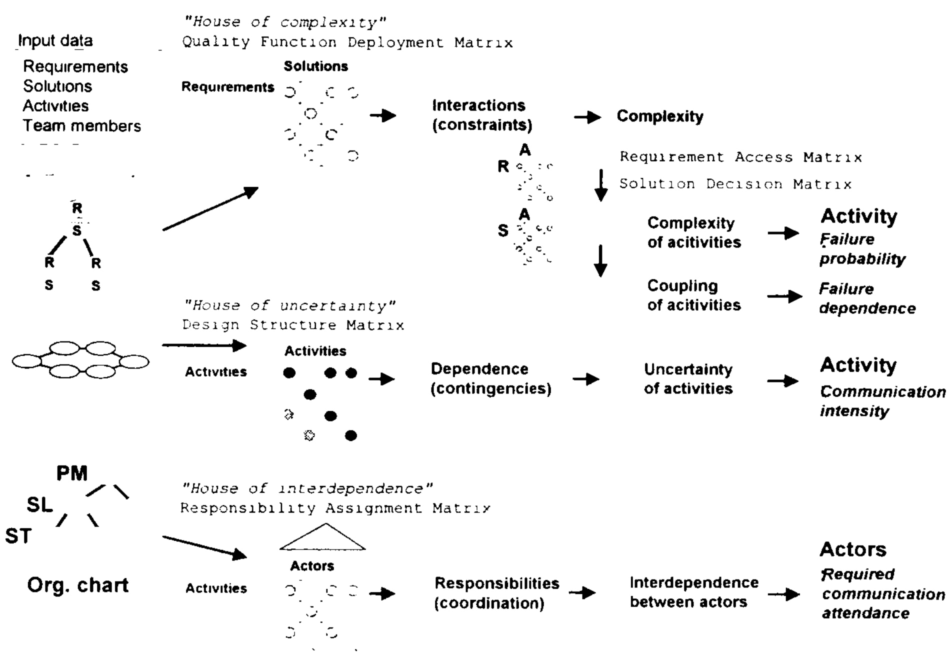


Figure 8. Overview of the load modeling methodology.

into measures of failure probability, communication intensity, and required attendance.

3.3 The House of Complexity

To define coordination load due to physical and functional constraints in the project task, we describe the various interactions between project requirements and solutions in a QFD interdependence matrix [10]. We can use the resulting *house of complexity* to derive a relative distribution of the complexity of requirements and solutions in the project. Using Herbert Simon's [17] notion of complexity as "the number of constraints an actor must simultaneously keep in mind while working," we count the number of interactions between requirements and solutions to get complexity measures. The more requirements a given solution must contribute to satisfying, the more complex is the solution. Similarly, the number of solutions that contribute to a given requirement gives a measure of the complexity of the requirement.

Figure 9 shows the house of complexity for the SSS project. We see how the requirement for 'Control processing' (R1) interacts with the solution 'External interface' (S3), giving the value 1 for the interaction between the two. Note also, that the interaction between a requirement and its own solution is weighted by the number of sub-solutions. Thus, for example, the interaction between R1 and S1 (see Figure 2) has the value 2, since 'Manifold system' (S1) gives rise to two lower level solutions (S1.1 and S1.2).

The resulting *solution complexity* may be viewed as a measure of the probability that actors producing a given solution will make errors when carrying out their work. Conversely, even if all solutions contributing to satisfy a requirement are in order, the customer may still not be satisfied. *Requirement complexity* is a measure of the probability of not satisfying a given requirement.

3.4 The House of Uncertainty

To define coordination load due to informational contingencies between project activities, we describe the production of, and need for, information between activities in the same type of interdependence matrix. Placing the project activities both along the rows and columns of the matrix, and using the Design Structure Matrix (DSM) technique [9], $a_{i,j}$ means that activity j produces information which is needed by activity i. If we order the activity matrix according to order of execution, we see that any $a_{i,j}$ where j is larger than i (i.e., which lies to the right of the midline diagonal) represents information which is not available when it is needed. Summing all $a_{i,j}$s where j is greater than i, gives a relative distribution of uncertainty for the various activities.

We use Galbraith's [8] notion of uncertainty as "a result of differences between the information which is needed to carry out a task and that which is available at the time the task is carried out." The more information is needed, but not available, the more uncertain is the task. Thus, the *house of uncertainty* gives the relative distribution of uncertainty of activities. Assuming that uncertainty gives rise to information needs, we may use this uncertainty distribution as an indication of the required communication intensity of actors who are responsible for various activities. That is, actors who are responsible for activities with high uncertainty need to communicate frequently with actors responsible for activities which deliver delayed information.

As noted by Gebala [9], DSM may be used to optimize the design sequence by LU-decomposing the activity plan as far as possible to bet a process with minimal uncertainty. However, we have not yet investigated the associated implications for project performance.

3.5 The House of Interdependence

Finally, we relate actor's responsibility for activities with the information flow between activities. The resulting *house of interdependence* shows which actors are responsible for given activities, and which actors need information produced by those activities. We use this matrix to indicate the type of interdependence between actors. That is, whether they are, in James Thompson's terms [18], pooled, sequentially or reciprocally interdependent. Thus, the tool illustrates the required participation in communication and information exchange during project execution. Figure 10 shows the house of interdependence for the SSS project.

For example, the Project Manager is reciprocally interdependent with the Manifold Lead, since they both are responsible for activities from which the other needs information (A1 and A4). Similarly, the Structures and Drafting Leads are only sequentially interdependent, since the Structures Lead needs information from the Drafting Lead's activities (A30 and A31), but not vice versa. The Drafting and Support Leads do not need to share specific information, and only have a pooled interdependence.

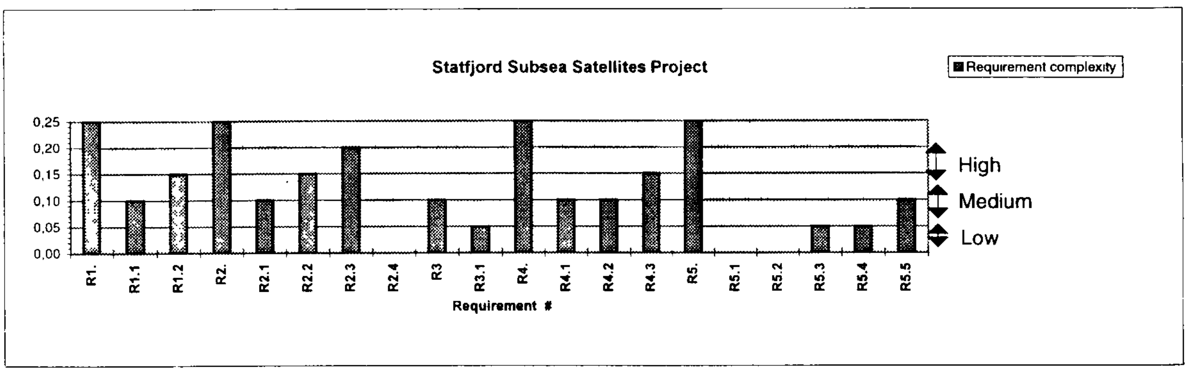
4. Simulating Information Processing in Project Enterprise

Given the above information processing model of engineering design and the models of coordination load and capacity in projects, we use them to simulate project execution as information processing of tasks according to activities in the project plan.

4.1 The Virtual Design Team (VDT) Discrete Event Simulator

VDT is the result of an ongoing project at Stanford University [12], with the aim of using computer simulation to investigate project team organization. VDT simulates project execution to get estimates of efficiency and process

		S1 Manifold system	S1.1 Piping and valves	S1.2 M. support structure	S2 Template structure	S2.1 Foundation	S2.2 Piles	S2.3 T. support structure	S2.4 Protect and coting	S3 External intefaces	S3.1 Spool piece	S4 Equpt. and procedure	S4.1 Load-out procedure	S4.2 Seafastening	S4.3 Lift. and level. proc.	S5 Policy and mngt.	S5.1 Review autorithy req.	S5.2 Review contract	S5.3 Management	S5.4 Support services	S5.5 Quality assurance	Requirement complex
R1.	**Control processing**	2			1					1		1										0,25
R1.1	Flow distribution		0	1											1							0,10
R1.2	Manifold support			0					1				1	1								0,15
R2.	**Support equipment**				4							1										0,25
R2.1	Support template					0	1	1														0,10
R2.2	Fix at sea-bottom					1	0	1	1													0,15
R2.3	Protect equipment					1	1	0					1	1								0,20
R2.4	Sustain environment								0													0,00
R3	**Interface flowlines**	1								1												0,10
R3.1	Join manifold and flowlines							1			0											0,05
R4.	**Support installation**	1			1							3										0,25
R4.1	Handle at construction site			1				1					0									0,10
R4.2	Transport safely to field			1				1						0								0,10
R4.3	Install property		1	1				1							0							0,15
R5.	**Control project**															5						0,25
R5.1	Ensure autorization																0					0,00
R5.2	Ensure acceptable terms																	0				0,00
R5.3	Ensure progress																		0	1		0,05
R5.4	Provide adm support																		1	0		0,05
R5.5	Verify solution																1	1			0	0,10
	Solution complexity	0,20	0,05	0,20	0,30	0,10	0,10	0,30	0,10	0,10	0,00	0,25	0,10	0,10	0,05	0,25	0,05	0,05	0,05	0,05	0,00	



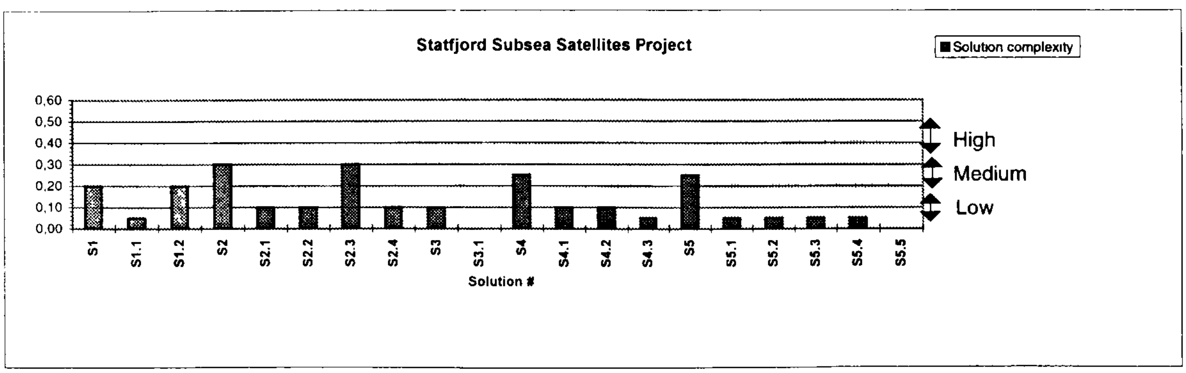


Figure 9. The House of Complexity for the SSS project.

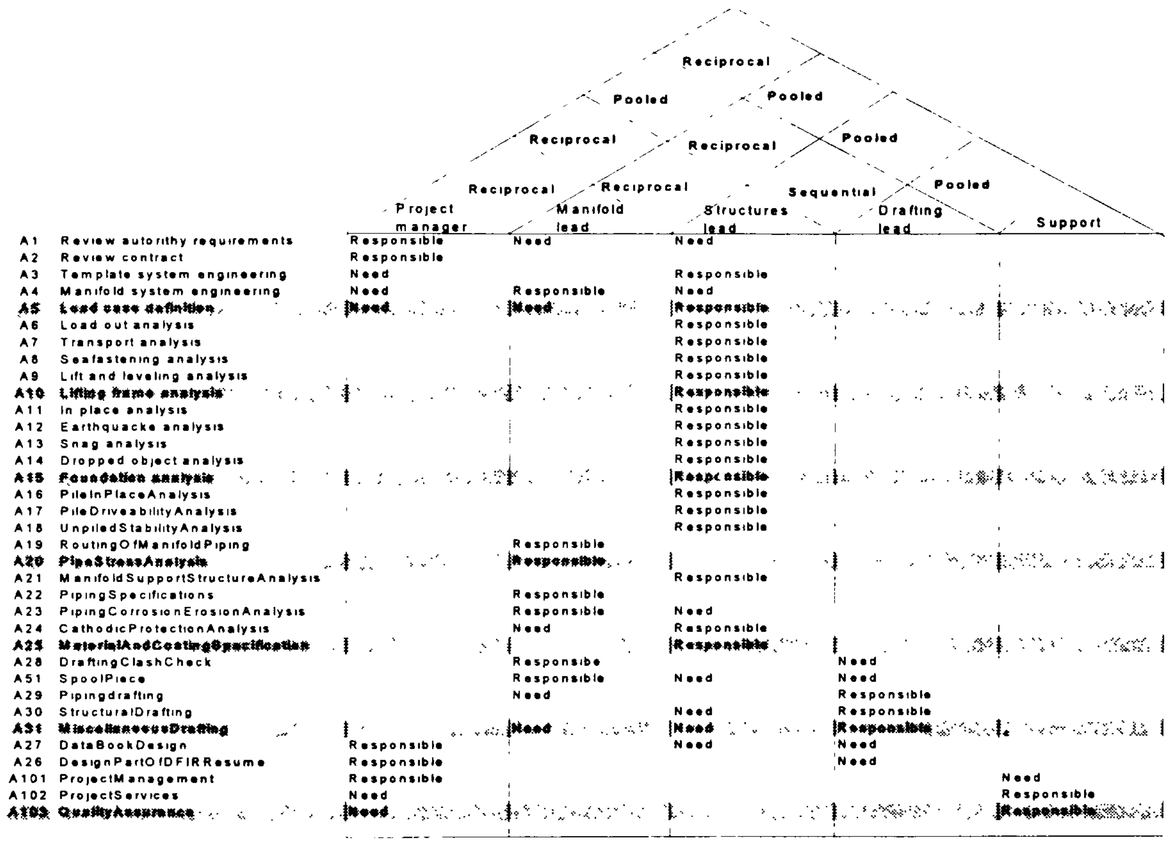


Figure 10. The house of Interdependence for the SSS project.

quality, using a set of stochastic (random number) process elements to account for uncertainty in human decision making. Thus, by varying requirements, deliverables, plan, team, policies, and preferences, we can obtain predictions for the probable effect of proposed changes. VDT is implemented as an object oriented discrete event simulator where each processor (actor) carries out work (activities) according to allocated responsibilities using a set of communication tools. The processing speed is determined by their skill, experience, and preferences, and the project policies for decision making in given situations.

Figure 11 gives an overview of input and output for analyzing information flow in the VDT. The input consists of a description of the load from the environment, the capacity

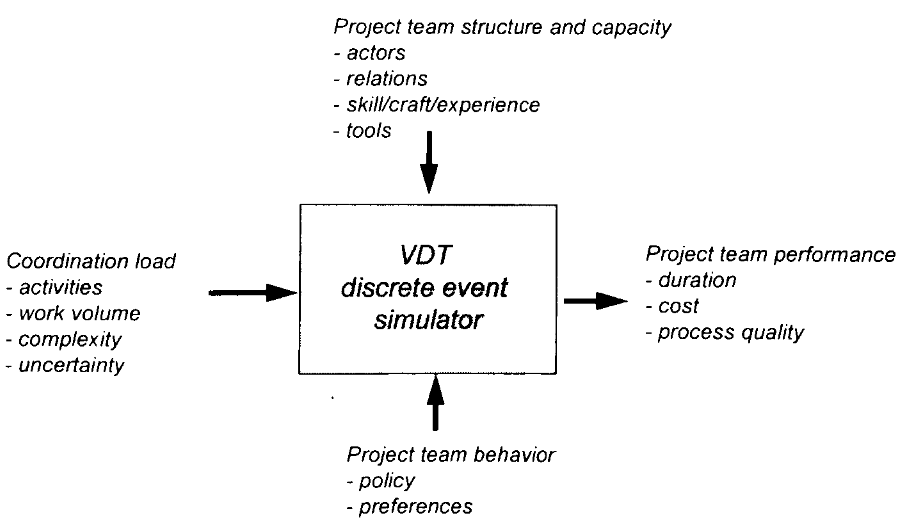


Figure 11. Information flow in the VDT simulation.

of the project team, and behavior of the team. The load is described in terms of activities' work volume, failure probability, and communication intensity. Team capacity is described in terms of skill, craft specialization, and experience, and team behavior is determined by a combination of policies (what should be done in given situations) and preferences (what actors are inclined to do in those same situations). The output from the simulation is project performance, defined in terms of the critical path duration, work volume (a substitute for project cost), and process performance in coordination (error handling and communication attendance) [3].

Below we give examples of results from simulation of the SSS project in VDT, as obtained from the mean of a series of simulation runs with different random seeds for stochastic process elements.

4.2 Performance Predictions from Simulation

Figure 12 outlines how complexity affects project performance, and shows simulation results for project duration, cost, and verification quality, as a function of centralization in the SSS project. Centralization in this context relates to "how high up in the hierarchy" decisions about exception handling "must travel" before reaching an actor with the authority to make a decision. The results illustrate how a change in coordination policy (higher or lower value than the one used in the SSS project) is likely to affect performance. The simulation predictions are compared with predictions from contingency theory [18] and predictions from the project manager (who planned and led the execution of the project).

For *duration*, the expected model behavior from contingency theory is based on the assumption that project managers have a more global view of different parts of the project, and thus will tend to prefer rework since they understand the potentially detrimental effect of ignoring failures in one activity on a number of dependent activities. Project team members, on the other hand, will often engage in local suboptimization of performance by ignoring and quick-fixing failures. In addition, decisions from managers will be delayed by other items in their "in-tray" (the manager's agenda). These effects cause higher centralization to give both more rework and more waiting time, and thus longer duration. Similar explanations for the cost and verification quality predictions can be found in References [3] and [11].

The simulation results indicate that there is no universally "best" centralization policy for the SSS project. The most suitable policy depends on whether efficiency or quality has the highest priority, in which case, one should choose a decentralized or centralized policy, respectively. Thus, the model behavior reflects the contingent nature of perfor-

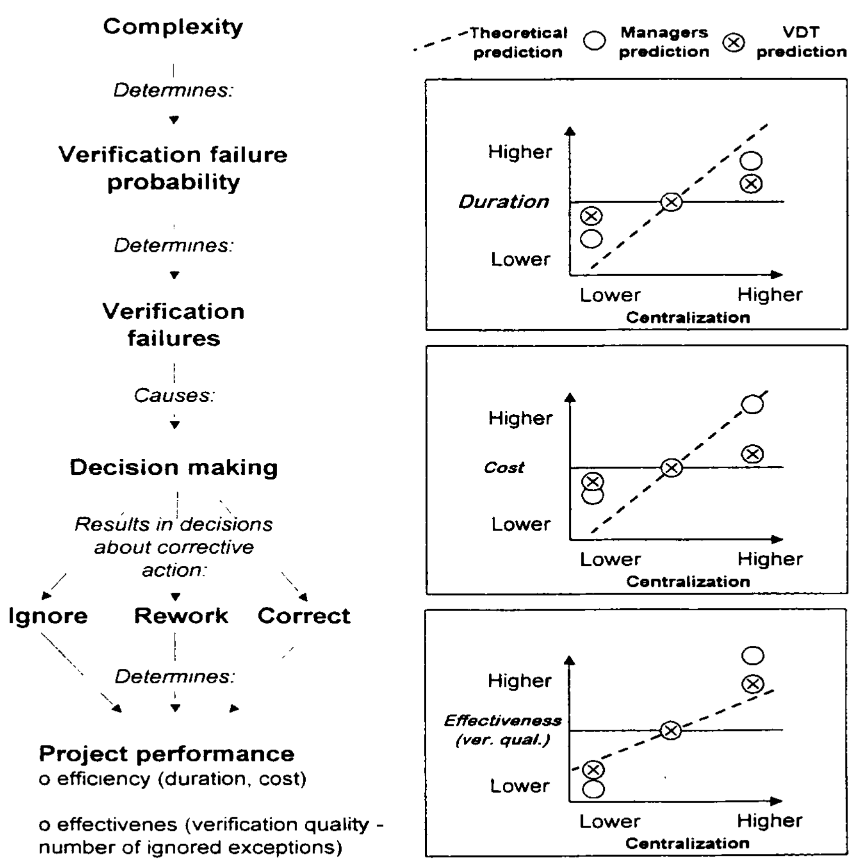


Figure 12. Project performance as a function of centralization in the SSS project.

mance [18], and gives predictions of the effect of centralization which are both qualitatively and quantitatively in agreement with theory and real world experience.

Similar studies on the effect of formalization of communication policies [3] also agree with theoretical and real world predictions, and give the same type of contingent behavior.

All of the simulation results indicate order of magnitude of qualitative change as selected input variables are altered. Because the VDT model includes a set of stochastic elements, several simulations are necessary to obtain statistically stable results (mean and standard deviation). The simulation results are stable in the sense that changes in input variables produce a consistently larger change in output variables than the standard deviation of the mean of those same output variables. This is an aspect of model correctness.

Both sets of results illustrate how project performance is contingent on proper organization, and how differences in performance may be expected for different policies applied to different project teams. We claim that this model behavior gives insight and has explanatory power, as a result of model completeness and correctness.

5. Future Research

In ongoing work, we are studying how performance is a function of changes in coordination requirements: the choice of different product solutions, division of the project into work packages (activities), and assignment of responsibility between project team members. Other possible studies include the performance effect of skills, experiences and preferences of the project team. This extensibility is an aspect of model completeness. Given that our model is similarly "correct" for these aspects of project enterprise, we may use it to study the various trade-offs between alternative ways to plan, man, and execute projects.

We are currently integrating a set of tools for modeling, formalizing, and analyzing enterprise. The resulting Enterprise Development Toolkit (EDT) will form an integrated system for modeling, analysis, and evaluation, and will enable us to systematically and efficiently investigate enterprise design and opportunities for improvement.

We are also starting to investigate *the implementation* of proposed organizational change in actual projects, to understand what is needed to make enterprise engineering and integration work in practice, and thus bring real improvement to engineering design projects.

Acknowledgements

The background for this work was carried out as part of the VDT project at Stanford University, and additional work was carried out in DNV Research, the research company within Det Norske Veritas ship classification society. The VDT project at Stanford was supported by National Science Foundation, Division of Information, Robotics and Intelligent System, Grant Number IRI-9725441 and by Seed Research Grants from the Center for Integrated Facilities Engineering, Stanford University. Project manager Dan Kyrre Stangeby from DNV's engineering services supplied information and knowledge about engineering projects in general, and the SSS project in particular.

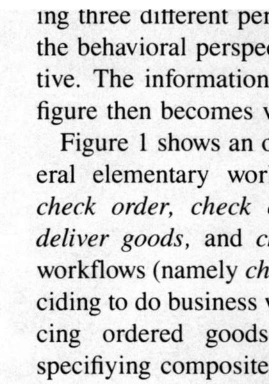
References

1. Christensen, L. C., T. R. Christiansen and T. G. Syvertsen. 1994. "What is Needed to Model Human Enterprise–Elements of a Unified Enterprise Engineering Model," presented at TIMS Cluster on BPR, Anchorage, *DNV Research Report No. 94-2020,* Høvik, Norway.
2. Christensen, L. C. and T. R. Christiansen. 1994. "QFD and Process Modeling for Organizational Analysis," presented at CE'94 Workshop, Sofia Antipolis, France, Oct. 1994, Published as *DNV Research Report 94-2034,* Høvik, Norway.
3. Christiansen, T. R. 1993. "Modeling Efficiency and Effectiveness of Coordination in Engineering Design Teams," Ph.D. dissertation, Civil Engineering Department, Stanford University, published as *DNV Research Report No. 93-2063,* Høvik, Norway.
4. Christiansen, T. R. and J. Thomsen. 1994. "CAESAR–an Architecture for Enterprise Modeling in the AEC Industry," presented at CIB W78, Helsinki, Finland, 1994, published as *DNV Research Report No. 94-2019,* Høvik, Norway.
5. Cohen, G. P. 1992. "The Virtual Design Team: An Object Oriented Model of Information Sharing in Project Design Teams," unpublished Ph.D. dissertation, Civil Engineering Department, Stanford University.
6. Davis, S. M. and P. R. Lawrence. 1977. *Matrix.* Reading, MA: Addison-Wesley.
7. Davenport, T. H. 1993. *Process Innovation–Reengineering Work through Information Technology.* Boston, MA: HBS Press.
8. Galbraith, J. 1973. *Designing Complex Organizations.* Reading, MA: Addison-Wesley.
9. Gebala, D. and S. D. Eppinger. 1991. "Methods for Analyzing Design Procedures," *Proc. of Third Intnl. ASME Conf. on Design Theory and Methodol.*, Miami, FL, September.
10. Hauser, J. R. and D. Clausing. 1988. "The House of Quality," *Harvard Business Review,* Boston, May.
11. Jin, Y., R. E. Levitt, T. R. Christiansen and J. Kunz. 1995. "The Virtual Design Team: A Computer Simulation Framework for Studying Organizational Aspects of Concurrent Design," *Simulation,* 64(3):160–174.
12. Levitt, R. E., Y. Jin, G. A. Oralkan, J. Kunz and T. R. Christiansen. 1995. "Computational Enterprise Models: Toward Analysis Tools for Designing Organizations," *CIFE Working Paper No. 36,* Stanford Univ.
13. March, J. G. 1988. *Decisions and Organizations.* Oxford, UK, Basil Blackwell.
14. March, J. G. and J. P. Olsen. 1986. "Garbage Can Models of Decision Making," *Ambiguity and Command: Organizational Perspectives on Military Decision Making,* J. G. March and

A. Weissinger-Baylon, eds., Cambridge, MA: Ballinger, pp. 11–35.

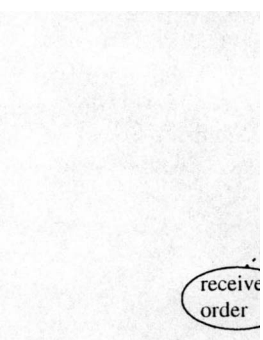
15. Scott, W. R. 1987. *Organizations: Rational, Natural and Open Systems.* Englewood Cliffs, NJ: Prentice-Hall Inc.
16. Simon, H. A. 1958. *Administrative Behavior.* New York: Macmillan.
17. Simon, H. A. 1969. *The Sciences of the Artificial.* Cambridge, MA: MIT Press.
18. Thompson, J. 1967. *Organizations in Action: Social Science Bases in Administrative Theory,* New York: McGraw-Hill.
19. Willems, P. 1988. "A Functional Network for Product Modeling," *PLI-88-16, IBBC-TNO,* Netherlands.
20. Williams, T. J. et al. 1993. "Architectures for Integrating Manufacturing Activities and Enterprises," in *Information Infrastructure Systems for Manufacturing,* H. Yoshikawa and J. Gossenaerts, eds., North-Holland, Amsterdam.

Lars Christian Christensen



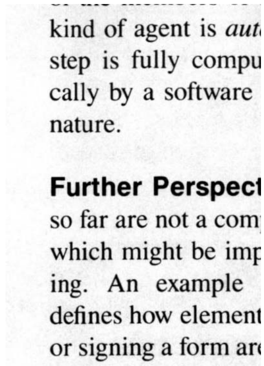
Lars Christian Christensen is a Doctoral Student in the Department of Civil Engineering at the Norwegian Institute of Technology, currently visiting the Department of Civil Engineering at Stanford University, where he is working on developing computer tools to model and analyze project enterprise. He is educated in Civil Engineering from the Norwegian Institute of Technology and has worked for many years in the AEC industry.

Tore Reider Christiansen



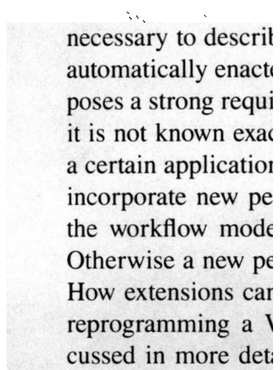
Tore Reider Christiansen is working as Principal Research Engineer, in the Department for Information Systems of DNV Research, the strategic research company of Det Norske Veritas ship classification society. He is educated in Aeronautical engineering from the University of London and the Massachusetts Institute of Technology, and obtained his Ph.D. at Stanford University, Department of Civil Engineering in 1993, working in the VDT project team. His current work involves enterprise modeling and the use computer tools to improve the efficiency and effectiveness of project organizations.

Yan Jin



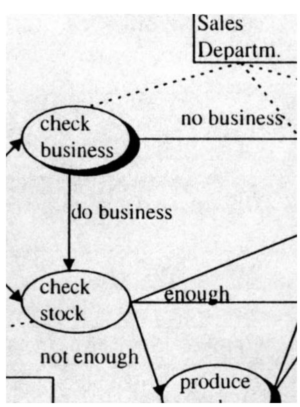
Yan Jin is Research Associate at Department of Civil Engineering, Stanford University. He earned his Ph.D. degree in Naval Engineering from the University of Tokyo in 1988. Since then Dr. Jin has been doing research on knowledge-based planning systems, distributed problem solving, organizational modeling, and their application to marine traffic control, collaborative design and concurrent engineering. His work on the Virtual Design Team integrates his current interests in organizational modeling and coordination science.

John Kunz



John Kunz is Senior Research Associate at the Center for Integrated Facility Engineering in the Department of Civil Engineering at Stanford University, where he teaches and does research on symbolic (non-numeric) modeling in engineering. He served as the Intellicorp Chief Knowledge System Engineer and Director of Manufacturing Applications, and has led AI systems application development in diverse areas including design, plant control and project management. The Virtual Design Team research has extended his interests in modeling from products and processes into social systems.

Raymond Elliot Levitt



Raymond Elliot Levitt is Professor of Civil Engineering in Stanford's Construction Management Program and Associate Director of the Center for Facility Engineering. Dr. Levitt earned MS and Ph.D. degrees in Civil Engineering at Stanford, and a BSCE at the University of Witwatersrand. He was on MIT's Civil Engineering faculty from 1975 to 1980, before moving on to Stanford. The Virtual Design research, in which Dr. Levitt is currently engaged integrates his organization theory and artificial intelligence interests.