SIMULATION OF DISTRIBUTED PRODUCT DEVELOPMENT WITH DIVERSE COORDINATION BEHAVIOR

Bryan Moser
United Technologies Research Center
East Hartford, Connecticut

Fumihiko Kimura   Hiromasa Suzuki
Department of Precision Machinery Engineering
University of Tokyo, Japan

KEYWORDS
global design project, activity model, coordination distance, product development simulation

ABSTRACT
Managers, by using models of product development activity, can experiment with various project approaches before committing to real people, resources and process. These activity models are evaluated through simulation to forecast emergent project cost, time and quality. Models of distributed projects require representation of the various local coordination behaviors. A simulation method for product development activity which incorporates diverse coordination behavior is presented.

INTRODUCTION

The increasingly global economy has led to the prevalence of globally distributed product development projects. These projects create new challenges for the project manager. Typically, local projects benefit from established patterns of work and low coordination costs. Globally distributed projects have coordination requirements which both are less understood and have a larger impact on total project results. Inherently, the work and coordination styles of global projects are diverse.

This paper is part of research on methods and tools for support of managers of product development projects. The view is of the project manager as designer; that is, the project manager must design the activity for product development. The total activity includes the integration of product, process, and organizational structures (Figure 1).

There is no assumption in the model as to the order of assignment of products, people and resources to tasks. As these elements are assigned to the task structure over time, evaluation of the project becomes more and more meaningful. (Moser et.al. ,1997).

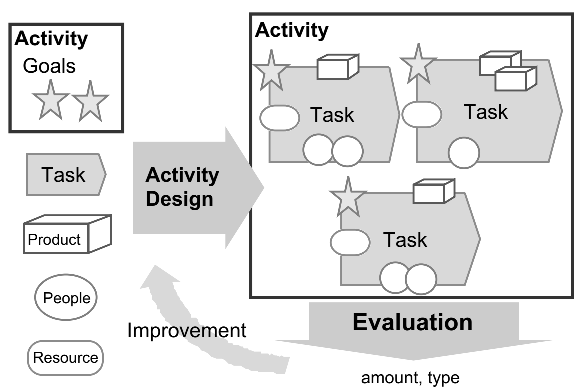

FIG. 1: DESIGN OF PRODUCT DEVELOPMENT ACTIVITY

BACKGROUND

Traditional Models of Product Development

Most models of work projects are based on tasks and sequential dependence. Beginning with early efforts by Taylor and others to improve industrial output through standardization of process and reduction in variation, management has meant control through decomposition of work. In this traditional view, work is broken down into tasks and an optimized process is defined by task sequence. Resources of similar type are interchangeable and assigned to tasks as late as possible. The ideal process can be planned ahead of time in a scientific manner (Meiksins, p. 73; Zuboff, pp. 42-46) Indeed, the role of people as human resources does not differ significantly in these models from other resources. One might describe these assumptions simply as *"Observe everything, Select one way, Always do it the same way, and Eliminate uncertainty."*

Since Taylor, project management methods continued to emerge, driven by the growth in size and complexity of industrial and military projects. Examples include the PERT method, Critical Path Method, and GANTT charts, which have origins now stretching back half a century. While these tools have been extended, they still share the same assumption that process can be optimized before the project begins.

For projects well understood *before* beginning and the option to assign resources freely *after* a process has been selected, perhaps the traditional methods are sufficient. However, product development projects are naturally the contrary. Still, the basic assumptions about decomposition of work are also found in the most popular project management tools used by companies today. Consider a typical case in which Microsoft Project GANTT charts are repeatedly revised only to be fully descriptive after a design project is complete. Use of the tool may be better than nothing, but the model in the tool doesn't seem to reflect the ongoing phenomenon in the real activity.

Emerging Models of Product Development

A new class of tools is emerging which include task dependency, organizational behavior and uncertainty and evaluate the project through simulation. The basic assumptions about work decomposition do not hold for complex, dependent, and diverse projects. It is now recognized that in large projects -- typical in industrial companies -- fixed task sequence and the elimination of uncertainty are not only difficult, but also an inaccurate description of the natural course of product development. Indeed, in some cases increased concurrency of tasks and process iteration lead to improved project cost, time and quality.

Pugh recognized the mismatch between uncertainty and creativity inherent in product development and existing project management technique. He argued that models of *total activity*, based on observation of design, would better serve project managers. Pugh's concept selection method is a procedure which promotes qualitative comparison of design alternatives in a structured way. (Pugh, 1996). However he did not go as far as to formally represent dependence and uncertainty in dynamic way in his methods.

Task Dependence. Two tasks are considered dependent if the work in the tasks is related. The progress of one task *depends* on the goals, shared resources, people, products, or knowledge of the other task. Unanticipated dependency can cause project delay, increased iteration and cost, reduced quality, or failure to meet basic project requirements. Thus inclusion and management of work dependencies in a project model is critical. (Malone, 1992) defines coordination as the management of these dependencies.

By recognizing tasks dependencies in a project model, the chance (or risk) of project variation due to uncertainty can be estimated, The most well know approach which includes the impact of task dependencies is the Design Structure Matrix (DSM), A more recent extension of DSM includes probability of task iteration due to dependence (Eppinger, 1994).

Organizational Behavior. From contingency theory in management science, organizations have been observed to follow patterns of behavior within limited rationality and dependent on key factors in the organizations structure and environment. Such contingency theory has been included in simulation of organizations from Cohen's 1972 Garbage Can Model to more recent *Organizational Consultant,* a rule-based expert system. (Baligh et.al, 1996)

Another established view from organization theory is the information and communication processing role of organizations. In the 1970's Galbraith described the exception handling behavior of organizations with established channels of communication and limits in information processing capacity. (Jin and Levitt, 1996). More recently Yates and Orlikowski have observed that communication behavior in groups tends to follow patterns of usage by *genre* -- a combination of media type and communication purpose. (Yates and Orlikowski, 1992).

Simulation-based Approaches. Simulations of product development projects represent participants in a project as agents. These software agents (or actors) are modeled with both work behavior and coordination behavior. Work behavior enables the agent to complete the skill-based work within an agent's own domain (i.e. task). Coordination behavior allows the agent to respond and interact with other agents. A simplified yet typical simulation event loop is shown in Figure 2. The event loop begins with the agent observing the state of the project.

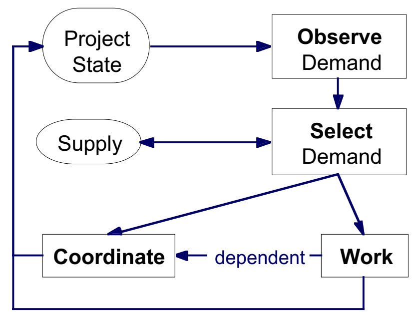

FIG. 2: TYPICAL SIMULATION EVENT LOOP.

Participation in the project implies that there are demands on the agent, both in response to direct contract work within a task and through dependency with work in other tasks. While specific models characterize types of demand differently, demand to work, to communicate, and to transfer results are most common. The agent makes a selection amongst these demands based on availability of its own supply and utility. The selected demand is attempted. If the selection is work which is dependent on work, information, or resources from a different agent, additional effort may be required within the event loop to coordinate across that dependency.

RESEARCH PROBLEM

If models of complex yet local projects are improved with the handling of uncertainty, dependence and organizational behavior, how should one model globally distributed projects?

By definition, global projects are diverse and uncertain. It is perhaps a semantic argument that a globally distributed team over a long period might develop its own work culture and thus efficiencies of a local project team. However, two points suggest this is not likely. First, some of the increase in coordination and uncertainty of global distribution is inherent. Second, it is typical for teams members to simultaneously participate in other local projects.

By investigating and incorporating the effects of these phenomena, the goal of this research is to provide a more accurate and insightful analysis of globally distributed projects. Two simulation-based models of product development that have inspired this work are the Virtual Design Team (VDT) and the Design Information Flow Simulation (DiFS). This paper focuses on two problems:

- the integration of exception handling found in VDT with the task dependencies in DiFS.
- the inclusion of coordination behavior in the simulation to forecast coordination work

Before addressing these problems, the DiFS and VDT simulation approaches are described in more detail.

Design Information Flow Simulation

DiFS was developed by Andrew Christian at M.I.T. as part of his doctorate work under the guidance of Warren Seering. As its name implies, DiFS characterizes the design process as the generation, dependence and flow of information. The DiFS event loop is similar to that shown in

Figure 2. Christian incorporated a detailed planning behavior in each agent for the selection of work on a daily basis. He also introduced a rich representation of task dependencies. (Christian, 1995.)

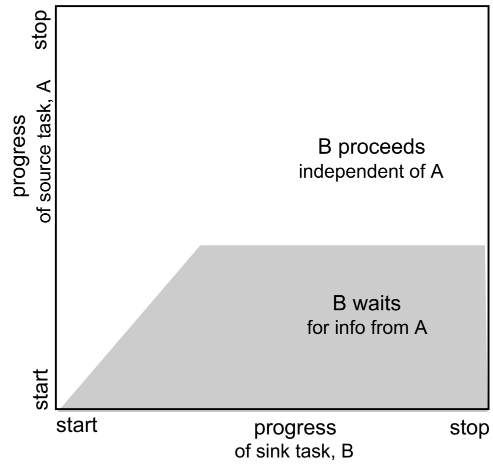

FIG. 3: TASK DEPENDENCE IN **DIFS**.

Figure 3 shows a dependency between two tasks, A and B, as it would be represented in DiFS. It is best to describe this dependency as *when information from A is needed in B* during the progress of the two tasks. Thus the lower right shaded area is a period when the sink task, B, cannot proceed because information from the source task A is required. The shape of the shaded area suggest that A must proceed at least in parallel to B during early progress, after which B can proceed without constraint due to A.

In addition to *when* function shown in Figure 3, the DiFS dependency also captures the scope and depth of information from source task required by the sink task. Note that if a dependency in the opposite flow from B to A exists, it is represented separately.

Virtual Design Team

VDT, originally developed in doctoral work by Geoff Cohen at Stanford, has been continuously developed by Y. Jin (now at U.S.C.), R. Levitt and J. Kunz. of Stanford. Given the depth and complexity of the VDT approach, there is much that could be described here. However, this paper focuses on the central innovation of VDT: the inclusion of exception handling in the model and simulation.

The simulation event loop in VDT begins with observation of the project state and selection of a task (Figure 4). In VDT, "inbox" and "outbox" metaphors are used to describe the demands from each agent (referred to as an "actor").

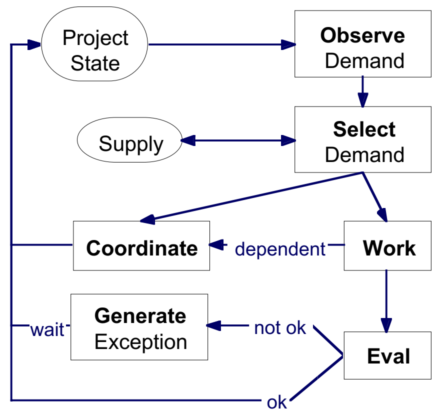

FIG. 4: SIMULATION EVENT LOOP IN **VDT**.

Next, VDT relies on assumptions from G. Cohen's research for the selection step. Incoming demands are selected based on a probabilistic choice, reflecting the priority and duration of the request in the inbox. As in DiFS, if work is dependent on other agent's work, coordination in the form of an information request occurs. VDT simulates the path of communication across the organization structure, so with the transfer, the information request can cause work across the organization.

It is after work has been attempted that VDT's most unique characteristic can be seen. Work success is evaluated based on a probabilistic chance of work failure. If the work fails, an agent generates an exception -- a signal to the organization that expected performance is not being met. The agent waits (until a time-out) for a decision from another agent whether to rework, correct, or ignore the exception. The choice of decision maker is also derived from the structure and contingency behavior of the organization. Through the inclusion of organization structure and behavior, VDT allows the analysis of project coordination quality.

APPROACH

Activity Model Object Language

The approach by this research team is embodied in the Activity Model Object Language (AMOL), a modeling and simulation method. The purpose of this paper is to explain the simulation method used in AMOL and how it relates to AMOL and VDT.

AMOL characterizes product development as both generation of information and transfer of people, products and resources. The inclusion of availability and transfer costs leads to a realistic evaluation of project cost and time. Discrete-event based simulation is employed to capture the diverse work and coordination behavior found in globally distributed projects, In the simulation, people provide skill to satisfy work demand. The work reduces uncertainty in products, utilizes resources, and generates information.

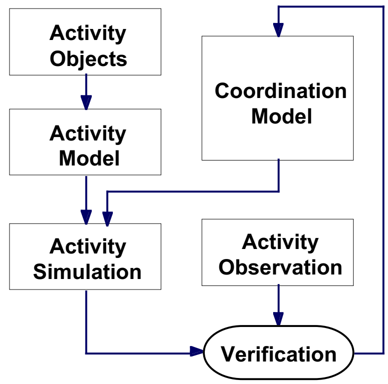

FIG. 5: RESEARCH APPROACH.

The general approach in research using AMOL is shown in Figure 5. The activity model is extended with a model of coordination. The significance of the included coordination behavior is judged by comparing the results of simulation with data from an observed product development project.

<u>Integration of Exception Handling</u>

The key contribution of VDT is exception handling, yet task dependencies are limited to precedence and reciprocity constraints. Exception handling in VDT and task dependency in DiFS are integrated by recognizing that contingency behavior is implicit in DiFS as project delay. In the integrated approach, violation of work demand both from dependencies and contracts generates an exception. The exception triggers the change from nominal to exceptional work behavior, leading to delay, cost increase and/or quality loss.

A DiFS-derived task dependency as used in AMOL is shown in Figure 6. First, the dependency is represented in both directions, from A to B and from B to A. Second, the shaded area depicts not only waiting, but the general zone of exceptional activity. Thus the open area -- when the two tasks can operate independently -- is a zone of nominal activity.

Crossing the exception boundary into a zone of exceptional activity triggers the organization's exception handling behavior as in VDT. Finally, also note that the shape of the shaded area is not arbitrary. For that reason in AMOL the breakpoints in each stage of the dependency's shape are referred to as milestones in the task.

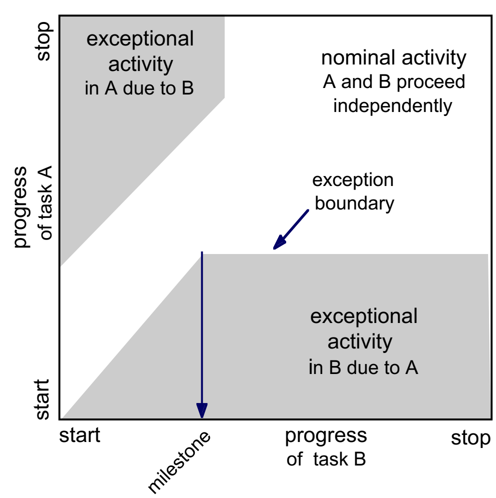

FIG. 6: **DIFS-**LIKE TASK DEPENDENCY IN **AMOL**.

With this approach the richer task descriptions introduced by Christian in DiFS and the inclusion of exception handling as in VDT have been integrated in AMOL. It is also plausible to represent the zone of exceptional activity as a fuzzy rather than crisp space. This research team is currently considering an approach based on the Method of Imprecision (Antonsson and Otto, 1995).

Inclusion of Coordination Behavior

Even with the integration of some characteristics of VDT and DiFS, there is still a requirement to introduce the diverse coordination and work behavior found in globally distributed projects.

Coordination is included in the simulation as a type of work supply and demand. Contracts and dependencies are generators of work demand, while availability and behavior are sources of work supply. It is likely that the local contract which defines work inside an agent's task is in local terms. Thus handling of these demands should not differ significantly from the event loop in VDT.

However, the ability of a participant to supply coordination to a participant who is not local requires an additional step in the event loop. The simulation event loop as used in AMOL is shown in Figure 7.

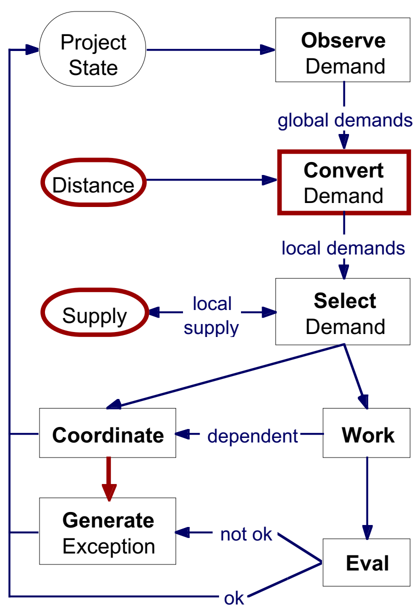

FIG. 7: SIMULATION EVENT LOOP IN **AMOL**.

AMOL begins with the observation of the project state by agents. Given the intellectual property boundaries and delays in communication common in global projects, already the agents will have a different view of the project. (A local view of project state is also possible in DiFS). Next the agent converts the global demands into local terms. The local supply of coordination can satisfy the demand at some time, cost and quality based on the coordination distance between the supply and demand.

Once the demands have been converted into local terms, the simulation event loop proceeds as in VDT. Additionally, the use of DiFS-like task dependencies allows a generation of exception not only from failed work but also from coordination. If the exceptional activity zone has been penetrated to some amount, then coordination work can result in an exception.

APPLICATION

Activity Model Object Language (AMOL)

AMOL is implemented as a set of libraries in C++. Libraries include objects for modeling, simulating, scheduling, and visualizing activity models of product development. The simulation object allows for the toggle of exception handling, rich dependencies, and coordination distances so as to calibrate against existing techniques.

Case

The method is currently being validated in an industrial project. The project is a partner-based design of a new product, with multiple partners on several continents. The progress of design drawings, design rework, and partner communication is available as real data to validate against the predictive analysis of the tool.

CONCLUSION

The simulation method in AMOL integrates the strengths of two related approaches -- the exception handling of VDT and rich task dependencies of DiFS. AMOL newly simulates coordination work in a product development project by using coordination distances to convert global demand to local demand. It is expected that globally distributed development projects will benefit from this method given their inherent diversity.

Future work is focused in two areas. First calibration of the technique against both the industrial case described above and with results found in DiFS and VDT. Second research on inheritance of task dependencies directly from product, resource, and people models is proceeding.

ACKNOWLEDGEMENTS

Special thanks go to Prof. Warren Seering at MIT and Profs. Kunz and Levitt at Stanford for sharing their tools and methods. Ralph Wood at UTRC has been an encouraging supporter. Kazumichi Kuroda and Kenichi Sanesige have helped with the details and in time became experts at AMOL, DiFS and VDT.

REFERENCES

Antonsson, E. K. and Otto, K.N. (1995), “Imprecision in Engineering Design”, *Transactions of the ASME,* Vol. 117, June, pp. 25-32

Baligh H., Burton R., Obel B., (1996), “Organizational Consultant: Creating a Usable Theory for Organizational Design”, *Management Science,Å@*Vol. 42., No. 12, December

Christian, Andrew D., (1995), *Simulation of Information Flow in Design*, Ph.D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, September

Eppinger, S. et. al., (1994). “A Model-based Method for Organizing Tasks in Product Development”, *Research in Engineering Design*, June.

Jin Y. and Levitt R.,(1996) “The Virtual Design Team: A Computational Model of Project Organizations”, *Computational and Mathematical Organization Theory* Vol.2, No.3 , pp. 171-196

Malone T. and Crowston, K. (1994), “The Interdisciplinary Study of Coordination”, *ACM Computing Surveys,* March, Vol. 26, No. 1, pp. 87-119

Meiksins P. and Smith C., (1996), *Engineering Labour: Technical Workers in Comparative Perspective*. Verso, London.

Moser B., Mori K., Suzuki H., and Kimura F., (1997) Global Product Development based on Activity Models with Coordination Distance Features”, *Proceedings of the 29th International Seminar on Manufacturing Systems*, Osaka, Japan, May 11-13, pp. 161-166

Pugh, Stuart, (1996) *Creating Innovative Products Using Total Design: The Living Legacy of Stuart Pugh*, Addison-Wesley.

Yates, J. and Orlikowski, W. (1992). “Genres of Organizational Communication: A Structurational Approach to Studying Communication and Media”, *Academy of Management Review*, April.

Zuboff, Shoshana. (1988). *In the Age of the Smart Machine*, Basic Books, New York.