Designing the Process Design Process

Arthur W. Westerberg[1 2], Eswaran Subrahmanian[2], Yoram Reich[3], Suresh Konda[2] and the *n*-dim group[2,4]

Abstract

We suggest that designing design processes is an ill-posed problem which must be tackled with great care and in an evolutionary fashion. We argue it is an important activity, however, as companies today use a small percentage of the intellectual capital they own when designing, suggesting there is room for significant improvement. We discuss who in industry and academia are currently involved with designing design processes. Based on empirical studies we and others have carried out, we have based our approach to study and support design processes on managing the information they generate and use. We are learning how to carry out studies more effectively with industrial partners, what features we need for managing information to study and improve design processes. We are even learning some general observations about the effect of different behavior of the group on its success at designing.

Introduction

In chemical engineering, the target of the process design process is the creation or modification of a flowsheet capable of manufacturing a desired chemical. In this paper we move up one level and look at making the design process itself the target of interest. In Fig. 1 we show typical information and control flows that can occur in a company and which are related to its design processes. Sales and/or research may initiate a design project. The design department may modify its prototypical design process when reviewing the nature of the proposed project -- leading to ‘feed-forward’ modifications. Reviewing the project as it occurs can lead to ‘feed-back’ modifications. Both of these operate with relatively short time delays. A company uses these mechanisms to improve the design process for the project at hand.

On this diagram we also show a company capturing and using history to improve its processes. However, very few firms have systematic procedures and mechanisms for capturing history, and, unfortunately, most such ad hoc attempts have led to information that is scattered and unorganized, making it difficult to review and reuse [Stewart, 1994]. A more systematic approach requires that a company set up a study team to look at its processes, to establish generic policies and protocols for design and to capture the lessons learned in anticipation that it may design this specific type of process again in the future. All of these are mechanisms to use experiences with design to improve future design processes.

If a company can capture its history in a useful form, it can form the basis for learning and reuse of its design processes. Over time it should be able to develop models to characterize and evaluate design process alternatives. We show such mechanisms as having long time delays. Just as with control loops, long delays can lead to unstable situations if adequate time is not allowed to assess the impact of changes before instituting more changes.

Design typically starts as an ill-posed problem. Our first activity is to establish the goals, tests, starting points and design space. Goals are typically many and in

1. Department of Chemical Engineering, Carnegie Mellon University
2. Engineering Design Research Center, Carnegie Mellon University
3. Dept. of Solid Mechanics, Materials and Structures, Faculty of Engineering, Tel Aviv University
4. The current *n*-dim group comprises: Douglas P. Cunningham, Allen H. Dutoit, Helen L. Granger, Suresh Konda, K. C. Marshall, Russell C. Milliken, Ira A. Monarch, J. Peter Neergaard, Robert H. Patrick, Yoram Reich, Eswaran Subrahmanian, Mark E. Thomas, Arthur W. Westerberg

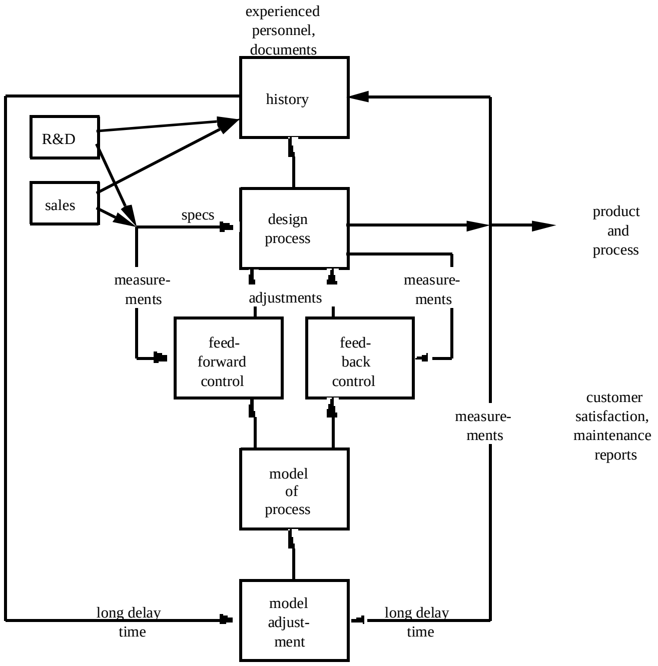

Fig. 1. The control and modification loops surrounding a design process

conflict -- e.g., for a chemical process we want the highest return on investment and we want the process to be safe and flexible. We use tests to evaluate and compare design alternatives as we enumerate them but before we build them. We carry out tests on proposed designs, not completed ones. Defining and implementing tests is usually very difficult. For processes, tests include flowsheet simulations, HAZOP studies, and committee reviews. Many require significant effort. We also need to define where we intend to start our design. Will it be based on an existing artifact -- such as when we design a new process by analyzing and modifying one or more existing ones or will we start from scratch? We need to be well aware of what has already been done before we establish our starting points. We also need to define the space of all alternatives from which we can select our design. Defining the design space is a continual process where we add and remove options as we learn about our problem and as we account for technological advances that change the space of available alternatives.

These same issues arise for designing process design processes. We need to set goals and tests. We need to define our design space of alternatives and our possible starting points. But what are our goals, tests, starting points and design space for design processes? Goals may include doing no damage to the morale of the company employees, keeping the organization effective on its current projects and creating faster yet more effective design processes. Without models of how our decisions impact the effectiveness of a proposed design process, we can only use our intuition to test likely impact of alternative processes on these goals. We almost certainly do not know the space of design process alternatives from which to design a new design process. Starting points can include or at least be aware of all design processes we and others have practiced and documented.

The space of alternatives is very large, with many promising approaches poorly explored, partly due to historical reasons as hierarchies are used as the primary organizational structures. An interesting approach is to start several independently operating design teams, each applying a different strategy. These teams communicate periodically only by sharing partial and complete solutions. Studies on asynchronous teams (A-teams) of computational agents by Talukdar and coworkers [Talukdar and Sousa, 1992; Talukdar et al, 1996] show that this strategy often outperforms any single strategy in searching for global optimal solutions to complex problems, an approach that may be very desirable if one does not want to be beaten soon by one’s competition. For this approach, there are relevant studies on computational and mathematical models of organizations [Huberman, 1995; Epstien and Axtell, 1996].

When reviewing design processes in industry, we find them to be very complex human activities -- even for routine products the company has designed for half a century. Fig. 2 shows the information flow for an electrical component manufacturing company we studied. The details on this figure are not important; the message is its complexity for what we might consider to be a relatively routine design process. The information flows are between the customer and the customer engineer and then between the customer engineer and the design department. Within the design department, we see a self-appointed ‘keeper’ of past designs, designers and technology specialists. We find information flowing from yester year’s customers in the form of problem reports and maintenance records.

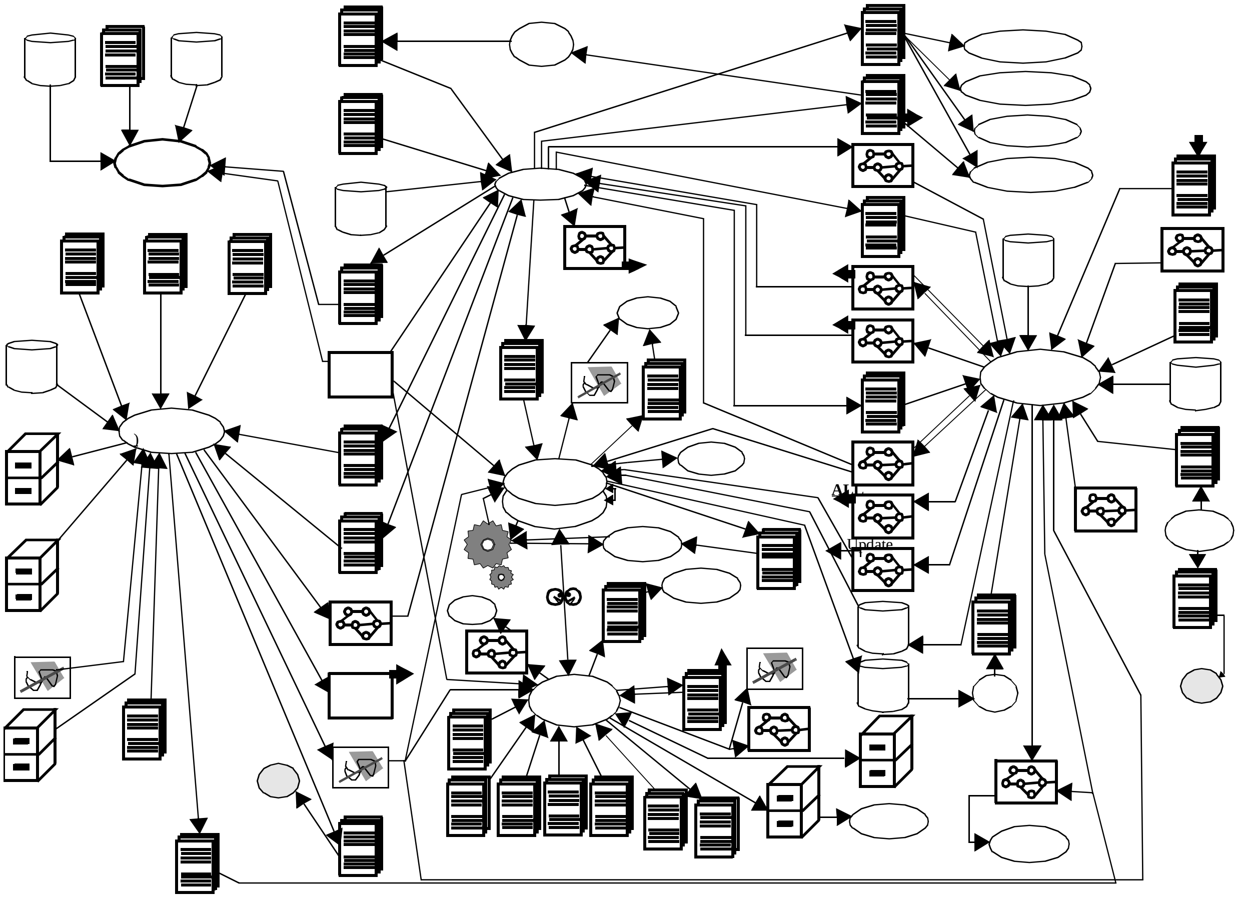

Fig. 2. Information Flow in an Organization Designing a Routine Electrical Product. Each oval is a human participant, the other icons indicate different types of information.

Design processes are further complicated when we examine them for globally dispersed companies. Such a company must organize how it intends to carry out designs: only at headquarters, led by headquarters but dispersed among several locations world wide, distributed by product (design pharmaceutical processes only in Europe, continuous petrochemical processes only in the US), and so forth. The interaction of design may be with a dispersed marketing and R&D activity in the company [Granstrand et al, 1993]. Added to the ever present and troublesome differences in 'domain' vocabularies of the design team participants (chemical engineers talking to mechanical engineers talking to sales talking to computer scientists), these groups face problems with language (English, French, German, Japanese) and culture, time zone differences, expensive face-to-face meetings and differences in regional knowledge (things mildew in the southern US).

The design processes carried out by a company are a significant part of its intellectual capital. Stewart [1994] estimates that only 20% of these intellectual assets of a company are captured and reused. If a company could use only 10% more, it would have increased reuse by 50%. We suggest that such an improvement would give that company a significant competitive edge. On the other side of this argument is the issue of information overload. Herb Simon poignantly notes that information has a cost -- the time to look at it, and time may well by our most limited resource. Thus just the capturing of history to make it available for future use is not sufficient. We see major efforts today in what are called 'data mining' tools -- i.e., tools that can automatically find patterns and relationships in data [Fayyad et al, 1996].

In this section we have discussed the nature of the problem of designing the design process and why we think it is important. Next in Section 3 we discuss who is currently concerned with capturing and improving our understanding of how design processes work, both in industry and academia. Section 4 describes what we are doing. We start by discussing the empirical studies we have carried out, largely with industrial partners. We then present our evolving hypotheses on how we should study design processes to improve them. We describe what we are learning using our approach. Section 5 gives a brief description of the computer support environment, *n*-dim, that we are creating based on our experiences.

Who currently worries about design processes?

There are many people worrying about improving design processes within companies. The most dramatic of these are people doing re-engineering. These people need a lot of courage as they often tear up the existing structure of an organization and propose implementing an entirely new one. Do they really know the possible outcomes that well? What is the likelihood they may destroy an organization [Strassmann, 1995]? The goal is laudable - better, more responsive organizations. Their problems with the affected participants are many, including all the stages of mourning: shock, denial, reluctant acceptance, recovery (a stage a company may not reach).

Teams of people are often established within a company to develop manuals for policies and procedures for 'best' design practice. These teams examine past projects and try to place order over their execution by defining the processes the company should use. To implement these processes, this team or a different team will define the types of documents to be produced and in which order, sign-off procedures, and so forth. Support tools include the use of document management systems and systems like PDM and Lotus Notes. The more routine the process, the more useful can be such manuals.

There are also people who are designing technical tools, support environments (e.g., the ASCEND environment to aid the creation of equation-based models [Piela et al, 1992-3]), and tool frameworks (e.g., the work by Director and coworkers [Jacome and Director, 1995]] on *Odyssey* and *Minerva* to support designers of VLSI circuits). Part of their charter is to anticipate how users can best interact with these systems. Also to populate frameworks and establish the coupling and ordering of tools, one needs a model of how designers design.

Computer Supported Collaborative Work (CSCW) considers the behavior of humans as central in processes such as designing. With several CSCW conferences since 1991 [CSCW-91 through 96], the literature in this area continues to grow rapidly. While initially technology driven, the field of co-operative work brings together social scientists, computer scientists and human computer interface specialists to address the development and incorporation of computer technology into the work place [Monarch et al, 1996].

There is much work on supporting different place same time group efforts -- such as several authors simultaneously editing a paper. Xerox's Colab project in the late eighties included a social component by developing technology for immediate and comprehensive communications among the geographically dispersed participants [Stefik et. al, 1987]. Working with wearable computers to connect repair personnel to the main office while performing maintenance work, the Human Computer Interaction Institute at CMU is studying the design of the equipment and the protocols critical for this type of co-operative work [Kraut et al, 1996].

Engineering research has also contributed to the understanding of engineering work, even though most of this work is not explicitly directed toward co-operative work. The literature reports several empirical studies in engineering design [Clark and Fujimoto, 1991; Hales, 1987; Kuffner and Ullman, 1991; Leifer, 1991; Subrahmanian 1992; Tang, 1989; Wilkins et al, 1989]. They range from comprehensive product development studies [Bucciarelli, 1984; Clark and Fujimoto, 1991; Hales, 1987] to studies of individuals and groups of designers **[**Goel and Pirolli, 1989**;** Minneman and Leifer, 1993]. These studies cover different ways to organize design, the evaluation of normative methods in design, group work around a table, information flow analysis, process-based analysis and task-related analysis for cooperating groups.

We started this section by asking who is designing design processes. The answer to the question to varying degrees is that we all are. As we build computational methods, tools and approaches to improve any aspect of human collective activity, we are engaged in designing the design process. Most models of organizational change describe moving from one state to another, where a structural or technological change initiates the move. However, experiences with changes that occur, especially when introducing groupware technology, provide an alternate model. Here a prescribed change is followed by a period of emergent and opportunistic change [Orlikowski and Hofman, 1997]. Researchers are still striving to understand the dynamics of change introduced to restructure organizations and processes along with embedding information technology. The INPRO group at the Norwegian Institute of Science and Technology (NTNU) in Trondheim is studying the juncture between technical and organizational issues in the support of process design and operation [www.inpro.unit.no].

In conclusion, we like to use an analogy from the field of chemical engineering. Prior to 1910, chemical engineers designed instances of specific processes, mimicking what they saw occurring in the lab. Then the concept of unit operations was introduced, where each type of equipment performed a limited set of functions (columns to separate, heat exchangers to exchange heat). We have since used unit operations to study chemical engineering and to design processes. We now see the emergence of a paradigm where we permit integrating several functions within a unit to design new processes. In the design of design processes we have evolved from artisans designing their own processes to companies using a few well understood organizational structures. We should expect to see an evolution in some organizations to highly integrated ever changing structures. Interesting is that these two paradigm shifts may be occurring together, possibly because of our much increased interest in understanding the process of designing.

What we are doing

The spectrum of our work embodies some of the areas of work we identified in the last section. We have an open minded approach to borrowing and adapting methods from other disciplines to both enhance our understanding of the problem and to devise approaches for embedding computational technologies and tools in the design process. While not organizational theorists, we have approached this problem of the design of design processes by examining information and knowledge management to design and development collaborative support tools. In achieving our goals, we have studied the tools and perspectives of a diverse set of disciplines. We also believe in “Learning by Doing” as an operative principle.

Empirical studies

Our own experience, which strongly supports the observations in the above literature, includes the following projects: the design of a process control system for power generation at Westinghouse, integration of materials and testing databases at Alcoa [Sargent et al, 1992], a study involving the four Engineering Research Centers at Carnegie Mellon, Delaware, Ohio State and Purdue collaborating to design an electrical connector [Wilkins at al, 1989], the design of transformers [Finger et al, 1993] and several year’s experience in a collaborative software design project carried out in an undergraduate computer science class at Carnegie Mellon [Dutoit, 1996].

Our working assumptions

In his classic paper, Bucciarelli [1984] observes that design is a social process, something with which we concur. Engelmore and Tenenbaum [1990] suggest that engineers spend about 85% of their time in meetings, on the phone, organizing information, and so forth, using only about 15% of their time to carry out technical computations. Owing to the diversity of backgrounds typically present, Bucciarelli states that a design team’s first activities are to negotiate the vocabulary its members will use, how it will make decisions, what decisions it will and has made, etc. We believe that support mechanisms must include, indeed maybe they should emphasize, aiding engineers with the activities they do for 85% of their time.

Our other hypotheses include: design is an evolution of the artifact description (e.g., the process flowsheet), of the information being gathered and organized, and of the design process itself; designers themselves should be enfranchised with the power to carry out this evolution without the attendant delays required when they cannot and/or are not allowed to modify their design support software; and finally, because we believe designers are always creating models of everything from organization charts to activity networks to flowsheet models, we can operationalize this support by providing a general modeling environment.

Our approach

In our approach to understand and improve design processes, we have been fortunate to form partnerships with industry. Our approach is a cautious one, leading to incremental changes whose effect we study. We first form a team with our industrial partner, carrying out all activities as a single entity [Reich et al, 1996a]. This integrated teaming engenders company “buy in” to the project. The team starts by selecting a target part of the company. It develops and uses questionnaires to identify information flows within this target area of the current design processes occurring, developing a diagram such as in Fig. 2 [Finger et al, 1993; Subrahmanian et al, 1993a].

The team then selects points where intervention could improve these processes. Particularly interesting points are those where failure or slowness now exists. In one case the goal of the team was to capture a complete history of the design data and its passing from tool to tool. The company kept only the current data for the final product in a design database. The company had no history of tool invocation and thus could not tie design failures occurring when the product was in use to the design decisions used to create the product.

The team next proposes scenarios to modify and hopefully improve the existing design process. After reviewing the scenarios with the designers, it uses them to specify a support tool. In the above case, the team elected to create a prototype system to quietly intercept ‘reads’ and ‘writes’ to the database as tools used and created data. The support could replay the complete process and tool invocation order from this captured history, allowing the designers and supervisors to review decisions and to branch very easily to alternative design paths, something they found extremely difficult to do with their current system.

We next bring these specifications back to CMU where we (rapidly) create [Dutoit et al, 1996; Reich et al, 1996b] a prototype support system, building on top of our *n*-dim [Levy et al, 1993] system. (We briefly outline the features of the *n*-dim system in the next section.) The designers then test the prototype system, giving their reactions to it. Modifications and hardening [Dutoit et al, 1996] form the last steps.

What we have learned and hope to learn

We are learning an approach to study design processes involving teams of geographically dispersed participants who have widely dissimilar backgrounds and what it takes to get industry to accept carrying out these studies. We have identified five methods we now use: (1) information flow-studies, (2) user participation, (3) prototyping, (4) testing by users (uncontrolled study) both in industry and in the classroom, (5) code maintenance and “hardening.”

We have focused on supporting the management of information. We are continually exposing extremely difficult technical issues that arises in information management such as capturing, interrelating, sharing, controlling access to, building repositories of, deleting and searching information by many collaborating participants. We must be able to work with multiple databases (with the best of all worlds we might be able to work with a single large distributed database such as MARI-

POSA as being developed by Stonebraker et.al. [1994]). Very interesting issues arise as to where to store information, how many copies to store, and so forth, to give us fast, reliable response as well as an approach that scales to immense collections of data.

We have learned, as have others, that participants need three types of persistence in their information (*personal information* to which the user can add, delete and generally "play," with no history kept; *shared information* with which members of the team can "play," but for which we maintain a revision history; and *published information* that no one is allowed to alter). We need search mechanisms to search over the structure of the information as well as the content (think of finding all pages on the World Wide Web that annotate pages describing the Ford Taurus). We need a history of who has had access and when, delivering this history in a manner that allows fast detection of who currently has access. We are learning various approaches to give designers the ability to add operations to their design environments without becoming C, C++ or Java programmers, and we are learning that this issue is a very difficult one on which to make "intellectual" progress.

We are learning the arguments we need to support our allowing anarchy by the designers in structuring and operating on their information. We see anarchy as the essential mechanism required for the end users to participate in the improvement of the design system through induction processes -- where they learn by creating and playing with many instances about what would be good general information structures and operations. We also now know that not all can be anarchy. Standards are needed to allow the attachment of powerful tools to design support environments. But anarchy is the first step to discovering and improving the standards. To this end we have proposed and strengthened our concept of modeling languages which allow us to impose standards. We quickly discovered the need for cardinality in these languages, but we needed to establish how to add it to models built of nodes and links.

We have formed and tested some interesting hypotheses on what improves design processes. An example is that those teams that negotiate early with other teams and among themselves to establish vocabulary and process will significantly outperform those teams that do not [Dutoit, 1996]. If valid this hypothesis suggests that a company should monitor teams to see if their members are carrying out these negotiations and let teams know when they are failing to interact as they should.

We have also hypothesized that, in doing design, engineers develop theories of products and that maintaining them is critical to design work [Reddy, 1996]. We have hypothesized that part of engaging in research and design projects is designing the process and that participatory design is good mechanism for executing such projects [Reich et al, 1996a].

The theories we can now pose and test are very context sensitive and "informal." This does not worry us since we know that much of engineering knowledge is informal and context sensitive [Subrahmanian et al, 1993b] and since we find these theories instrumental in building design tools and collaborating with industries.

We are working on the assumption that only as we gather more and more information about design processes will we be able to form those elusive design theories that everyone thought should be possible 15 to 20 years ago. Most of these earlier theories are based on little or no data coming from real design processes. Thus the first step has to be to gather good, usable data about how processes actually occur. Thus we have formed our belief that information management issues are central to our approach. Our reliable information will be that which we collect from the applications we build with *n*-dim. At first our theories will be very context sensitive, as noted just above. With time we will be able to create and test more general hypotheses about design processes. Only then will we be able to create tests that will allow us to predict the outcome of choosing among different design processes that are very different from the ones with which we have experience. And it may be only then that we really will be able to predict the outcome of re-engineering on a design organization.

The *n*-dim system

Reich et al [1996b] describe the essential ingredients of the *n*-dim system on which we build our design support environments. Earlier articles that describe aspects of *n*-dim are by Levy et al [1993], Robertson et al [1995], Westerberg et al [1995] and Dutoit et al [1996]. We created this system to provide a base of capabilities to any support environment we would chose to create. It supports rapid prototyping of these systems. We migrate code to C when we wish to harden it or speed it up.

The *n*-dim system gives users the ability to reference and display information stored anyway in a system of networked computers, be it a web page, a Framemaker document, a GIF file, an entry in a database, or a user created "atomic object" such as a "frame," a variable, or a string object. *n*-dim views these objects as living in a flat space. Models in *n*-dim are themselves objects. They contain reference pointers to any of the other objects. The user can add directed labeled links to models connecting any pair of pointers to express any desired relationship between two objects. The standard visualization of a model looks like a window on a PC where the pointers appear as labeled boxes in the window. Links appear as directed arrows between the boxes and have labels on them to express the relationship intended. The user can add operations to any model. An example is to add the operation that allows one to explore the local file system and pick an object one wishes to point at as one is constructing the model. We have found this form of modeling to be very expressive and have been able to build our applications very quickly on top of this representation scheme.

The user can create standard types of models by creating a modeling language (itself an *n*-dim model) that defines the types of objects and links that may be put into this type of model. We are adding cardinality to allow much more expressiveness in these languages. An operation attached to a modeling language is inherited by all instances created using that language. Users can play with languages to evolve useful types of models in the system. Creating modeling languages has proved to be very easy for end users. Creating new operations is yet a difficult activity as the user has to program in the underlying lower level language (Stitch). However, the ability to add operations to objects has given us a very powerful way to create expressive systems with minimal effort. As an example, we created an issue-based information system (IBIS [Kunz and Rittel, 1970]) called I-WEB [Coyne et al, 1994]) very quickly. We created a system to support "brain writing," a form of brainstorming over the Internet, in four hours, with most of the time spent in designing the operations. Only four operations with a total of 60 lines of code proved necessary. With this system, multiple users can place issues on a table (a public *n*-dim model) as text objects. After a prescribed time (say, 20 minutes), an operation distributes the issues amongst the participants. For another period of time they add comments to these issues and place them back on the table. A last cycle of adding comments completes the process. Of course more complex tools have taken us two months to design and implement. However, when these tools have proved too slow, we have quickly reorganized them to speed them up or to make them scale better for large problems -- in days not months.

The system supports a private workspace for each user. It also supports a public workspace where teams of user can collaborate to create and use models and a published workspace where one can place objects that are to be immutable. The system maintains a full history of the changes made to public models.

We are adding revision management and access control mechanisms to the base *n*-dim system. Our access mechanism will keep a history of who has had access and when.

We have integrated several legacy tools into *n*-dim. Thomas [1996] describes integrating the ASCEND equation-based modeling system into *n*-dim, creating a combined system that supports a team of modelers to collaborate when creating and debugging a complex model.

Conclusions

In order to design the process design process, we have to view it as any other design problem which is ill-posed. Product design roughly involves cycles of synthesis and analysis. Therefore, we need knowledge about synthesizing and analyzing process design processes. Unfortunately, in contrast to product design, analysis knowledge does not exist. In the process of collaborating with industry, we have learned about design and design processes. To better understand (and form theories) about designing process design, we believe we have to study design as performed by designers and as performed by us when we collaborate with industry. We have already formed some initial theories and continue to do so. We believe that we are using an "internally consistent" approach: we study our work as we study designers, we build tools that maintain design history for engineers and for our own study of design, and we design our tools to support our evolving theories about designing design processes.

References

Ashby, R. (1958). *Requisite variety and its implications for the control of complex systems*, Cybernetica, 1(2), 1-17.

Bucciarelli, L. L. (1984). *Reflective practice in engineering design*, Design Studies, 5(3), 185-190.

Clark, K. and Fujimoto, T. (1991). Product Development Performance, Harvard Business Press, Cambridge, MA.

CSCW [91-96], Proc. Computer Supported Co-operative Work Conference, 91 through 96, Assoc. Comput. Mach., New York.

Coyne, R., Dutoit, A., Uzmack, J., and O'Toole, K. (1994). *IWEB (Information WEB): Information management for software*. Technical Report EDRC-05-87-94, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.

Dutoit, A. (1996). The Role of Communication in Team-Based Software Engineering Projects, PhD thesis, Department of Electrical Engineering, Carnegie Mellon University, Pittsburgh, PA.

Dutoit, A., Levy, S. N., Cunningham, D., and Patrick, R. (1996). *The Basic Object System: Supporting a spectrum from prototypes to hardened code*. In Proc. OOPSLA '96, ACM, New York, NY, 104-121.

Engelmore, R. S. and Tenenbaum, J. M. (1990). *The engineer's associate: ISAT summer study report*., Unpublished.

Epstien, J., Axtell, R. (1996). *Growing Artificial Societies: Social Science from the Bottom Up,* Brookings Press and MIT Press.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). Advances in Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA.

Finger, S., Subrahmanian, E., and Gardner, E. (1993). *A case study in concurrent engineering for transformer design,* in Rosenburg, N. F. M. (ed), Proc. ICED-93 (The Hague), Heurista, Zurich, 1433-1440.

Goel, V, and Pirolli, P (Spring, 1989). *Motivating the Notion of Generic Design within Information Pro-*

cessing Theory: The design problem Space, AI Magazine, 18-47.

Granstrand, O., et. al. (1993). *Internationalization of R&D - A survey of some recent research*, Research Policy, 22, 413-430.

Hales, C. S. (1987). Analysis of The Engineering Design Process in an Industrial Context., Ph.D. thesis, Department of Engineering, University of Cambridge, Cambridge, UK.

Huberman, B. A., Hogg, T. (1995). *Communities of practice: Performance and Evolution*, Computational and Mathematical Organizational Theory, 1(1), 73-92.

Jacome, M. and S. Director (1995). *A Formal Basis for Design Process Planning*, Tech. Report, Engineering Design Research Center, Carnegie Mellon Univ., Pittsburgh, PA

Konda, S., Monarch, I., Sargent, P., and Subrahmanian, E. (1992). *Shared memory in design: A unifying theme for research and practice*, Research in Engineering Design, 4(1), 23-42.

Kraut R., Siegal, and M. Miller (1996). *Conversation and Collaboration*, Proc. ACM Conference on Computer Support of Collaborative Work**.**

Kuffner, T. A. and Ullman, D. G. (1991). *The information requests of mechanical design engineers*, Design Studies, 12(1), 42-50.

Kunz, W. and Rittel, H. (1970). *Issues as Elements of Information Systems. Center for Planning and Development Research*, Working Paper 131, Univ. Calif., Berkeley, CA.

Leifer, L. J. (1991). *Instrumenting the design process*, In Proc. ICED-91, Zurich, 301-310.

Minneman, S. L., and Leifer, L. (1993). *Group Engineering Design Practice: the Social Construction of Technical Reality*, Proc. International Conference on Engineering Design.

Levy, S., Subrahmanian, E., Konda, S. L., Coyne, R. F., Westerberg, A. W., and Reich, Y. (1993). *An overview of the n-dim environment*. Technical Report EDRC-05-65-93, Engineering Design Research Center, Carnegie Mellon Univ., Pittsburgh, PA.

Monarch, I. A., Konda, S. L., Levy, S. N., Reich, Y., Subrahmanian, E., and Ulrich, C. (1996). *Shared memory in design: theory and practice*, In Bowker, G., Gasser, L., Star, L. and Turner, W. (eds.), Social Science, Technical Systems, and Cooperative Work, Lawrence Erlbaum, Hillsdale, NJ.

Orlikowski, W., Debra Hofman, J. (Winter, 1997). *An Improvisational Model for Change Management: The case of Groupware Technologies*, Sloan Management Review.

Piela, P.C., McKelvey, R.D. and Westerberg, A.W. (1992-93). *An Introduction to ASCEND: Its Language and Interactive Environment*, J. Management Info. Sci., 9(3), 91-121.

Reddy, J. M. (1996). Design as Theory Building: Building and Reuse of Artifact Theories, Ph.D. Thesis, Dept. Civil Engineering, Carnegie Mellon Univ., Pittsburgh, PA.

Reich, Y., Konda, S. L., Levy, S. N., Monarch, I. A., and Subrahmanian, E. (1996a). *Varieties and issues of participation and design*, Design Studies, 17(2), 165-180.

Reich, Y., Konda, S., Subrahmanian, E., Cunningham, D., Dutoit, A., Patrick, R., Westerberg, A., and the *n*-dim group (1996b). *Being agile when building information infrastructures for agile enterprises*, submitted for publication.

Robertson, J.L., E. Subrahmanian, M.E. Thomas and A.W. Westerberg (1995). *Management of the Design Process: The Impact of Information Modeling*, in Biegler, L.T., and M.F. Doherty (eds), Fourth International Conf. on Foundations of Computer Aided Process Design Conference, AIChE Symp. Series No. 304, 91, CACHE Corp. and AIChE, 154-165.

Sargent, P., Subrahmanian, E., Downs, M., Greene, R., and Rishel, D. (1992). *Materials' information and conceptual data modeling*, in Barry, T. I. and Reynard, K. W. (eds), Computerization and Networking of Materials Databases: Third Volume, American Society For Testing and Materials, ASTM STP 1140.

Stefik, Mark, Foster, Gregg, Bobrow, Daniel G., Kahn, Kenneth, Lanning, Stan and Suchman, Lucy (Jan 1987). *Beyond the chalkboard: computer support for collaboration and problem solving in meetings*, Commun. ACM 30, 1, 32-4

Stewart, T. A. (Oct. 3, 1994). *Your Company's Most Valuable Asset: Intellectual Capital*, Fortune.

Stonebraker M., R. Devine, M. Kornacker, W. Litwin, A. Pfeffer, A. Sah, & C. Staelin (1994). *An Economic Paradigm for Query Processing and Data Migration in Mariposa*, in Proc. 3rd International Symposium Parallel and Distributed Info. Systems, Austin, TX. IEEE Computer Society Press, 58–67.

Strassman, P. A. (1995). The Politics of Information Management, The Information Economics Press, New Canaan, CT.

Subrahmanian, E. (1992). *Notes on empirical studies of engineering tasks and environments*, invited position paper, in NSF Workshop on Information Capture and Access in Engineering Design Environments, Ithaca, NY, 567-578.

Subrahmanian, E., Daleng, E., and Maeland, A. (1993a). *Information flow analysis of hydro power design.* Unpublished Internal Report, CMU-ABB.

Subrahmanian, E., Konda, S. L., Levy, S. N., Reich, Y., Westerberg, A. W., and Monarch, I. A. (1993b). *Equations aren't enough: Informal modeling in design*, Artificial Intelligence in Engineering Design, Analysis and Manufacturing, 7(4), 257-274.

Talukdar, S., and de Souza, P. (1992). *Scale Efficient Organizations*, in Proc. IEEE Int'l. Conf. on Systems, Man and Cybernetics, Chicago, IL.

Talukdar, S., Baerentzen, L., Gove, A., and de Souza, P. (1996), *Asynchronous Teams: Organizations for Algorithmic Cooperation*, EDRC Tech. Report 18-56-

96, Carnegie Mellon Univ, Pittsburgh, PA.

Tang, J. (1989). *Listing, drawing, and gesturing in design*, Technical Report SSL-89-3, Xerox Palo Alto Research Center, Palo Alto, CA.

Thomas, M. E. (1996). Tools and Information Management in Engineering Design, Ph.D. Thesis, Dept. Chem, Engng, Carnegie Mellon Univ., Pittsburgh, PA. Available as Tech Rep. EDRC 02-38-96.

Wilkins, D. J., Henshaw, J. M., Munson-Mcgee, S. H., Solberg, J. J., Heim, J. A., Moore, J., Westerberg, A., Subrahmanian, E., Gursoz, L., Miller, R. A., and Glozer, G. (1989). *CINERG: A design discovery experiment*, in NSF Engineering Research Conference, College of Eng'ng, Univ. Mass., Amherst, MA, 161-182.

Westerberg, A.W., and the *n*-dim group (July, 1995). *Distributed and collaborative computer-aided environments in process engineering design*, Proc. Intelligent System for Process Engineering Conf., Snowmass, CO.