Support System for Different-Time Different-Place Collaboration for Concurrent Engineering

Eswaran Subrahmanian, Robert Coyne, Suresh Konda, Sean Levy, Richard Martin, Ira Monarch, Yoram Reich, Arthur Westerberg

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

In the ***Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Concurrent Engineering (WET ICE)***
(Morgantown, West Virginia, April, 1993)
IEEE Computer Society Press, pp. 187-191, 1993

Abstract

To be effective in practice, concurrent engineering requires access to and organization of knowledge accumulated over time, product versions, and customers. More important, separate knowledge resources have to be shared and coordinated over space and time if successful design is to be accomplished. In this paper, we address the nature of communication in design, especially across disciplines, and the support systems that facilitate better communication. While a lot of research effort is being expended on same-time communications within a group, we consider, as well, the need for, and the problems associated with, different-time, different-place communication. We present these views in connection with an on-going development effort, *n*-dim.

1 Introduction

A large scale design project must coordinate expertise from many different disciplines. Empirical studies conducted by us and others indicate that *design is a continual negotiation of constraints, terminology and trade-offs for the creation of a shared understanding of the design process and product.*

Equally important, the companies involved in large-scale design and manufacturing are themselves many-faceted involving a number of different departments devoted to design engineering, engineering analysis, manufacturing, quality assurance, suppliers, subcontractors, procurement, legal, fiscal, marketing, customer service, and management. These departments plus the sub-departments in charge of various parts of design must manage their own information gathering and production. All of these have to be shared and coordinated over space and time if successful design is to be accomplished.

2 Communication in design

Communication among members of the design group requires that there be established processes of arriving at consensus on: the nomenclature to be shared by design participants, constraints on manufacturing, performance, disposal, etc., problem decomposition at every stage of artifact life-cycle, and design trade-offs negotiated amongst different company interests. Effective communication and coordination of work depends on such processes [1]. To reach consensus on a given issue, such processes should not, in general, be imposed from the outside but must, be, or have been, established by the design group itself. In order to support the processes for reaching consensus, the design environment must be conducive to: maintaining the thread of the discussion and facilitating the organization of the information exchanged among designers while keeping interruptions of the design process itself to a minimum. The problem becomes more complicated when design discussions are not restricted to *same-time same-place* collaborative settings. In the absence of face-to-face contacts, individual engineers must be able to participate in an asynchronous manner (different-time-different-place) through a variety of representational forms, media, and modes of communication while retaining the time sequence of the exchanges.

- Information comes in a variety of *representational forms* including sketch, picture, gesture, text (print and speech), graph, table;
- Information is exchanged in a number of *media* including paper and pencil, physical being, computer, video, film; and
- The representational forms exchanged in these media come in formal to informal *modes of communication*: reports, memos, e-mail, face-to-face.

Information capture and structuring over time depends on which representational forms, media, and modes of communication are used. For example, information techniques useful for text management are not useful for graphical information. On the other hand, organizational units or concepts extracted from textual information can be useful in classifying graphical information such as sketches, or drawings.

While the computer is the target medium for information management, care must be taken to transfer information from other media to it. Insisting that engineers use only the computer would be counterproductive. Hence, the results produced on other media must be transferred to or interlinked with the computer. This is not to reduce the computer into merely a recording medium – rather, interaction between designers is both captured in and facilitated by the computer.

3 Contexts, Threads, histories, and Perspectives

Much effective group work occurs in same time and place using the direct physical presence of the participants to immediately clarify misunderstandings. Availability of high-speed communications and video technologies has relaxed the constraints on the participants to be in the same place while retaining much of the context of face-to-face communication [2]. However, same-time collaborations are limited in the number of participants that can effectively contribute and lack viable mechanisms for capturing, organizing and accessing design history.

3.1 Synchronicity and Asynchronicity: Tradeoffs

Different-time and same-time collaboration are sometimes complementary in that one is strong where the other is weak. While in both asynchronous and synchronous communication, "conversation" tends to drift; i.e., the communicators tend to lose the overall context of the discussion, same-time-same-place communication provides more oppurtunities for quicker feedback without losing context. For example, e-mail users attempt to maintain part of the context by including the mail being responded to in their replies but a moments reflection reveals the difficulty of using such an approach in an extended conversation. However, while different-time collaboration has difficulties in maining access to shared context, it does make possible the participation of larger numbers of participants. Nevertheless, current technologies for both types of collaboration lack facilities for capturing and structuring design history as an integral part of the design process without additional overhead that participants are unwilling or unable to handle.

3.2 Gaps and Heterogeneity

When a designer joins (or re-joins) an on-going discussion, she needs to quickly be made cognizant of the current overall context of the discussion. This requires that a designer be able to search for information and context making use of both the history as well as the content of the discussion.

In the general case of communications in concurrent engineering where engineers from quite different disciplinary, experiential, and organizational background need to work together, a significant problem lies in supporting both the reconciliation of different perspectives and the maintenance of each individual perspective. Otherwise a given individual designer may be unable to effectively access or understand the information generated collectively. There must be the simultaneous support for the generation and maintenance of both individual and collective view of the same underlying information objects.

In situations where issues are re-visited in subsequent meetings, the context of the previous discussions needs to be *re-created* to allow effective communication. This activity can benefit from facilities that assist in constructing and managing such contexts.

Many of these problems recur for the *single* designer "collaborating" with herself; i.e., a designer trying to re-construct the context of her own past design decisions. All the problems raised earlier apply even in this "degenerate" case of collaboration.

The importance of asynchronous, as well as synchronous, collaborative work has been recognized in several studies. The system, *n*-dim, described below significantly differs from others in providing mechanisms for maintaining context over time and aiding in the reconciliation of heterogeneous perspectives. Using *n*-dim people can include information objects in multiple models without replication; they can incorporate other design tools at varying levels of integration maintaining, at the least, the context within which the tool was used; users can derive models both inductively (generalizing from experience) and counter-inductively (based on an *ad hoc* generalization) because past cases are captured and organized; and engineers can create and have access to shared memory as part of their normal working patterns.

4 *n*-dim: A system for evolving concurrent engineering processes

n-dim is a computational environment built to facilitate and study collaborative design by enabling generalized modeling starting from the initiation of a design process and continuing throughout the life-cycle of the artifact [3, 4]. In the *n*-dim environement, generalized modeling is seen as an evolutionary process guided by a methodological emphasis on collaborative prototyping [5, 6]. Supporting such an evolutionary process requires that the *n*-dim environment address the problems of asynchronous communication, longitudnial integration, and multiple perspective reconciliation.

n-dim is implemented in a prototype-based object system called BOS, the Basic Object System [3]. Since it is prototype-based, there are no classes, *per se*; rather, any object is a potential prototype for another object.

4.1 How features of *n*-dim support concurrent engineering

There are four main features of *n*-dim that contribute to supporting concurrent engineering.

4.1.1 Flat Space

The Notion of *flat space* captures the fact that an individual object can be situated in multiple contexts. The space of objects in *n*-dim is conceptually flat; that is, objects do not contain other objects, *per se*. Instead, multiple structures can be imposed on this flat space by means of *models*. Models are are comprised of *links*, or relationships between objects (models themselves being objects). In this way, the same object may participate in many models.

The special link called `part` link is canonically represented as a box inside of a box; e.g., objects "inside" models. All other links are shown as directed and labeled lines. It must be stressed that due to the flat space of *n*-dim, a complex object like a model does *not* (in any physical sense) contain objects that appear inside it. This allows the same object to appear in multiple contexts or to be viewed from multiple perspectives. Things can be found *in the various context(s)* in which they are used (referred to, referenced by, placed in relationship to other things, etc.). Hence, multiple contexts that share common elements can be created with virtually zero overhead.[1] This means that different collaborators can reference and talk about the same object even though they see it from very different perspectives and constitute it in very different contexts, experimenting with changing: its material makeup, various of its characteristics and relationships, as well as its place in a structure or its role in a system. This enables one to represent structural or conceptual change while still refering to the same object or same concept undergoing the change.

One can *project* the structure of an *n*-dim model onto multiple *displays* that discriminate between possible views of that structure by fixing the characteristics of a given projection *in* its rendering.[2] Projections are models, as are renderings. The system merely interprets these models appropriately when needed.

4.1.2 Generalized Modeling

Generalized modeling allows people to operate on things and kinds of things interchangeably, and move freely from one level of abstraction to the other. Ev-

[1] The overhead in using objects in multiple models is close to zero since only the the part link needs to be stored. Thus, the storage of objects in multiple models is significantly reduced. Note however, that the architecture allows for objects to be stored in multiple locations if required for efficiency; it is simply that it does not *require* multiple storage.

[2] A rendering of a model can be something like a window presented to the user for interaction, a printed file, etc.

ery unpublished model is mutable, and can have operations and attributes defined on it without affecting any other model. This means that models (information structures) can serve as *prototypes:* potential progenitors of other models.[3] Any model can serve as the starting point for another model, in the sense that it can be copied and the copy modified in ways independent of the original (though not via the class/instance relationship). In a preliminary design phase the evolution of prototypes can be free-wheeling, but nevertheless constrained and manageable. Exploration of ideas can be structured without having to commit to a taxonomic structure that prevents consideration of important ramifications.

Models can also serve as *modeling languages*, that is paradigms for structuring information. Modeling languages are represented as models themselves (i.e., as things to be designed, negotiated, etc.). In this sense, models can act as types for the creation of other models. Modeling languages are codifications of inter-relationships between information objects that are found useful throughout and beyond a current design situation. These languages can be used to codify formal and informal knowledge [7]. Languages for structuring formal knowledge can be used to code expressions, constraints, and production rules, that result in the specification of inference mechanisms. Informal languages can be used to annotate documents, drawings, and formal structures of knowledge. These languages provide the necessary facility for coding design knowledge, organizational procedures, task allocation, etc. The specification of a design can be increasingly standardized and automated.

Within a single project, a design history can be captured as a sequence of loosely constrained prototypes finally evolving into a modeling language that is the progenitor of a more tightly constrained and less varying sequence of prototypes, as, for example, in the designs of a mature product. Design histories can also be charted and organized for continued elaboration and refinement across projects, organizational entities, and cultural contexts.

Interruptions, gaps, and multiple perspectives in design communication require support for the construction of a shared memory that not only is maintained across times, places and individuals, but also can be structured and indexed for easy assessibility and continued effective use.

[3] We will use the terms "instance" and "prototype" somewhat interchangeably in what follows, since, in a prototype-based system, the two concepts coincide; the different connotations are useful in distinguishing various *uses* of a model, however.

4.1.3 Publishing

Publishing is a mechanism for making models formally exchangeable and persistent following the library as a metaphor. Hence, *traces* of the evolution of information and its structure over time can be found in the growing repository of published objects.

Design history in *n*-dim is treated differently than it is in traditional revision control systems. History is viewed here as a connected sequence of models created by designers. The maintenance of history is achieved by the publishing mechanism. Once a user publishes a model it is immediately available to anybody in the group or organization subject to access control. A published model is inalterable (persistent) but is copyable by any person in the group. Thus, *n*-dim both facilitates the act of collaboration and captures design history as a by-product. In short, design history is created and maintained as part of the process of creating good designs.

n-dim maintains a pedigree information in the form of Revision Models. Their primitives are such that most mechanisms proposed for maintaining different versions based on user, time-stamp etc., are directly derivable from the revision model of any model in the *n*-dim system. Hence, concurrency for asynchronous collaboration is not an issue. Evolution of ideas and negotiations on the development of the artifact is captured as a sequence of transformations to the models (including the creation of modeling languages and their instances) across time, product versions, and customers.

4.1.4 Text Processing, Indexing, Structuring, and Retrieval

Text processing or natural language processing (NLP) enables the discovery of terminological patterns implicit in large text corpora. The discovered patterns provide a basis for organizing, indexing, and retrieving information in various sub-domains – in effect helping to make explicit conceptual structure implicit in the co-occurence patterns of terminology in large domain-specific corpora. These conceptual structures support the creation of models by indicating the important features a model has to represent as well as the terminology that can be used to label these features. Since the links between these labels and their textual contexts can be maintained, contextualization and documentation of these models can be preserved and easily accessed.

Such context preserving, structuring, and documenting capabilities support: (1) fast introduction of team members to the various perspectives that are being brought to bear in the on-going design of artifacts and (2) the reconciliation of these multiple perspectives by crystallizing their differences and creating a graphic and indexing space in which these differences can be negotiated [1].

While NLP tools provide facilities for capturing textual context, *n*-dim's modeling facilities also allow for capturing other forms of context such as drawings, photographs, audio, and video. Future advances in various fields may allow the extraction of information from these forms as well.

4.2 Scaling-up *n*-dim

n-dim is designed to be portable, with a layered architecture that should scale up to deal with the intensive storage and other computational demands of real engineering tasks. Conceptually, a dynamic and flexible modeling environment is required to enable negotiation and consensus promotion among multiple perspectives. *n*-dim is different from systems that support specific domains – it is designed to be modified and evolved based on the needs of the designers in processes of product creation in their respective domains. This view extends the time honored rapid prototyping tradition of putting the end-user (the designers) in direct collaboration with the developer.

n-dim is also being designed to handle very large applications. To help achieve this, objects themselves are not stored in the database, but only the attributes necessary to search for objects (e.g. the intrinsic attributes). The objects themselves are stored locally in individual workspaces, or (possibly) in shared spaces if they have been published (and thus made immutable). This separation between the space in which objects are stored vs. the space in which attributes about them are kept for the purpose of search can be utilized to great advantage in scaling *n*-dim up for large (hundreds of thousands to millions of objects) applications.

Finally, *n*-dim is being designed to handle many organizational structures via configurations based on what in *n*-dim is called a *cell.* The collection of a set of database processes and workspace processes is called a cell. A broad range of cell configurations is possible, from a single, centralized database with several users clustered about it (a small work group), to a totally distributed configuration with both central and localized components for individual users (entire organization or sub-organization, or clusters of smaller work groups). Studies will be conducted for the appropriate configuration based on network and load balancing requirements. At some point, it should be possible to experiment with the use of high-speed networks (such as CMU's NECTAR or BBN's Butterfly) depending on the needs of the actual engineering application. These are issues for further research in the development and deployment of *n*-dim.

4.3 Integration of Multi-Media information and CAD/CAM tools

Integration of multi-media is required for both synchronous and asynchronous (same-time and different-time) concurrent engineering. Support for same-time collaboration is being actively pursued in the field. Support for different-time collaboration requires new approaches like on-line video and audio technologies.

The *n*-dim architecture actively aids in the integration of information from CAD/CAM tools. For example, a layout from a particular CAD tool can be related to appropriate information such as design decisions, test data, etc. However, the degree of data and tool integration will be based on the maturity of the tools and the level of integration desired. This can include (1) black-boxes where *n*-dim only allows for the management of inputs into and outputs from the tools in context to (2) the provision of mechanisms for aiding tool integration through neutral product models such as PDES/STEP. These latter can themselves be represented as *n*-dim models.

5 Summary & Conclusions

The position put forward in this paper is that communication in design is a complex process that takes place across time, space, and heterogeneous perspectives. While most research effort is directed towards same-time collaboration, we argue for the importance of different-time different-place collaboration especially in large and complex design situations as well.

Based on empirical studies (our own and those of others), we have identified the characteristics of collaborative environments and presented a system, *n*-dim, that addresses many of the issues, including asynchronous collaboration and reconciliation of multiple perspectives.

In discussing the *n*-dim system, we do not claim that it is the answer to all of concurrent engineering's problems. Rather our aim is to present an an environment which we are currently testing and refining in practice and which we have argued elsewhere addresses issues that must be addressed [1, 6, 7] if a more useful and usable design support system is to be developed.

References

[1] S. Konda, I. Monarch, P. Sargent, and E. Subrahmanian, "Shared memory in design: A unifying theme for research and practice," *Research in Engineering Design*, vol. 4, no. 1, pp. 23–42, 1992.

[2] S. Minneman and J. C. Tang, "VideoWhiteboard: Video shows to support remote collaboration," in *Proceedings of the Conference on Human Factors in Computing Systems*, (Reading, MA), pp. 315–322, Addison-Wesley, 1991.

[3] S. Levy, E. Subrahmanian, S. L. Konda, R. F. Coyne, A. W. Westerberg, and Y. Reich, "An overview of the n-dim environment," Tech. Rep. EDRC-05-65-93, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA, 1993.

[4] E. Subrahmanian, A. W. Westerberg, and G. Podnar, "Towards a shared information environment for engineering design," in *Computer-Aided Cooperative Product Development, MIT-JSME Workshop (Nov., 1989)* (D. Sriram, R. Logcher, and S. Hukuda, eds.), (Berlin), Springer-Verlag, 1991.

[5] R. Budde, K. Kautz, K. Kuhlenkamp, and H. Zullighoven, *Prototyping: An Approach To Evolutionary System Development.* Berlin: Springer-Verlag, 1992.

[6] Y. Reich, S. Konda, I. Monarch, and E. Subrahmanian, "Participation and design: An extended view," in *PDC'92: Proceedings of the Participatory Design Conference (Cambridge, MA)* (M. J. Muller, S. Kuhn, and J. A. Meskill, eds.), (Palo Alto, CA), pp. 63–71, Computer Professionals for Social Responsibility, 1992.

[7] E. Subrahmanian, S. L. Konda, S. N. Levy, Y. Reich, A. W. Westerberg, and I. A. Monarch, "Equations aren't enough: Informal modeling in design," *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, vol. 7, no. 4, pp. 257–274, 1993.