Guest Editorial: Special Issue on Collaborative Engineering

Ram D. Sriram, National Institute of Standards and Technology, Gaithersburg, MD 20899, sriram@nist.gov
Simon Szykman, National Coordination Office for Networking and Information Technology Research and Development, Arlington, VA 22230, szykman@itrd.gov
Delcie Durham, National Science Foundation, Arlington, VA 22230, ddurham@nsf.gov

Design of complex engineering systems is increasingly becoming a collaborative task among designers or design teams that are physically, geographically, and temporally distributed. The complexity of modern products means that a single designer or design team can no longer manage the complete product development effort. Developing products without sufficient expertise in a broad set of disciplines can result in extended product development cycles, higher development costs, and quality problems. On the other hand, ensuring comprehensive technical proficiency in a world where trends are toward more multidisciplinary design can become a costly undertaking for a company.

Driven by such issues, companies are increasingly staffing only their core competencies in-house and depending on other firms to provide the complementary design knowledge and design effort needed for a complete product. Designers are no longer merely exchanging geometric data, but also more general knowledge about design and the product development process, including specifications, design rules, constraints, and rationale. Furthermore, this exchange of knowledge more and more often crosses corporate boundaries. As design becomes increasingly knowledge-intensive and collaborative, the need for computational frameworks to support a collaborative product development environment becomes more critical. There are several research and development issues that need to be addressed for the effective realization of such a collaborative design environment. These issues, along with relevant papers from this special issue, are discussed below.

Architectural Frameworks. Software frameworks are complex software applications, or software systems composed of an integration of multiple applications, designed to provide an infrastructure for addressing a given problem. Key issues associated with development of software frameworks include traditional systems engineering principles, including requirements gathering, functional analysis, architecture design, detailed design, validation, etc. Concerns of interest in the development of architecture frameworks include the traditional importance of cost, quality, schedule, and functionality, as well as key needs for usability, interoperability, modularity, and effective integration into a collaborative product development process. Examples architecture frameworks include blackboard-based systems, service oriented architectures, and support for multi-media interactions.

Shared Representations. The representation of product information that supports sharing forms the fundamental cornerstone of any collaborative engineering (or collaborative product development) application. This product information should include not only the geometric data corresponding to the physical parts and their relationships but it should include non-geometric information such as material characteristics, function, behavior, and design intent as well. The current focus on product families supports this shared representation requirement for distributed design. Support should be provided for dealing with multiple levels of functional abstraction, geometric representation, constraints and for generating multiple functional views through the entire product life cycle. In the recent past, various groups within industry, academia, and government have been developing sharable and reusable models known as ontologies. All ontologies consist of a vocabulary along with some specification of the meaning or semantics of the terminology within the vocabulary. In doing so, ontologies aid in sharing information and facilitating interoperability by providing a common vocabulary with a shared semantics.

Engineering Repositories and Long Term Knowledge Management. Engineering repositories are the electronic substitute for and successor of the traditional file cabinets where information on past designs is stored [Szykman et al. 2000]. Design repositories store descriptions of past designs in a form suitable for browsing and retrieval for direct use and reuse in the active design process. Various knowledge-based extensions to design repositories can be made to document general knowledge about design and the product development process, design rationale, design rules, ontologies and taxonomies, catalogs of successful designs, etc. Manufacturing repositories contain manufacturing process knowledge and can aid the designer avoid costly downstream errors. Currently, proprietary repositories are maintained in ad-hoc fashion by individual organizations. Hence, in order to fully leverage the shared representations described above, there is a need for the development of standardized knowledge capture, storage, and retrieval methods, including case-based reasoning. These knowledge stores can further aid in the automated generation of new engineering knowledge [Sriram 1997].

Constraint Management. In a large class of engineering problems, much of the associated knowledge can be represented by a set of mathematical relations between symbols or real-valued quantities. Maintaining these relationships, which can be viewed as constraints, and performing inferences on these relationships is the realm of constraint management or satisfaction techniques. The general problem of constraint satisfaction involves the following: given a set of variables and a set of constraints, find a set of assignments for each variable consistent with the set of constraints. The variables may be discrete or continuous. Constraint solvers can be broadly classified into numerical constraint solvers, and symbolic constraint solvers. In general, the constraint satisfaction problem is NP-complete. Various heuristic algorithms and techniques that exploit certain characteristics of a domain, e.g., geometry, have been developed to reduce the computational complexity.

Coordination and Transaction Management. Collaborative engineering environments necessitate a powerful and flexible transaction management framework to temporally coordinate the concurrent activities of multiple users. The key requirements for supporting coordination in a collaborative engineering environment are [Sriram 1992]: 1) the transaction framework should be dynamic, so as to support highly interactive, iterative, and interleaved engineering transactions; 2) locking mechanisms, including appropriate security controls, must be interactive and flexible enough to allow multiple transactions to proceed concurrently without having to wait indefinitely for other transactions to complete; 3) extensive records of the clients/agents and their activities must be maintained, to include the data used, dependencies between clients and data, and time-stamping of data and operations; 4) the version management system should keep track of the design evolution, in addition to supporting version merging; 5) nested transactions should be supported, in order to allow hierarchical subdivision of design effort and provide smaller units of control for various transaction management facilities; 6) nesting of database partitioning should be supported, where each partition serves one or more divisions of the design activity; and 7) long duration transactions, which are very different from the short duration transactions supported by traditional database management systems, should be supported as well. Design decision-making assignments and records must be implicitly captured within the transaction management framework.

Negotiation and Conflict Mitigation. Conflicts due to differing perspectives among designers or among participants in different stages of the product realization process often lead to expensive designs, delays in product development, or undesirable compromises in the final product. One reason for conflicts is the lack of information that designers have about other designers' or stakeholder objectives, and reasons for rejecting or accepting a given alternative. Other reasons, such as self-enrichment and the need to achieve an upper hand contribute to conflicts. Hence, effective negotiation and conflict management techniques are crucial for mitigating conflicts. Taxonomies of design conflicts and design rationale encoding would greatly facilitate conflict mitigation strategies. Additionally, techniques developed in economics, mathematics, and political science should provide knowledge that can be used to enhance the effectiveness of negotiation approaches.

Organizational Issues. Collaborative engineering (CE) activities can be classified into minute (task-level), small (project-level), large (program-level), and mega (enterprise-wide) [Carter and Baker 1992]. Minute CE involves two to three individuals working on a small unit (e.g., a power supply) and requires knowledge in a single domain, small CE involves a group working in a single domain (e.g., mechanism design), large CE involves multiple disciplines (e.g., structural, mechanical, thermal, etc.), and mega CE crosses company boundaries and involves considerable outsourcing. In all these activities, managing engineering requirements is of paramount importance. Efficient allocation of tasks (task decomposition), smooth workflow management, and effective product development supply chain management could result in considerable benefits to the collaborative design enterprise. Finally, there is a need to conduct empirical studies of designers performing real world collaborative design, along with assessing the effectiveness of various tools and techniques.

Virtual Reality and Collaborative Design Issues. While traditional CAD (computer-aided design) systems provide comprehensive tools for generating geometric forms, which encourages designers to come up with a form first and think about function later (i.e., *form-to-function transformation*, and knowledge-based design systems focus on the symbolic aspects of design first and later map the symbolic structure to a geometric model, in virtual reality-based systems (also called immersive CAD), the human being becomes part of the design by using various immersive environments, including haptic, visual, and speech interfaces. In distributed design, the application of virtual reality methods that allow exploration of the design space and the visualization of alternatives by stakeholders across the organization and throughout the supply chain could accelerate the design process with benefit to the overall time to market.

Decision-based Design Issues. Decision-based design addresses the cognitive "structuring" of a problem; the drive for innovation where the existing "structure" or solution space is ill-defined or insufficient; the need to reduce complexity by mapping to what we know; and the consistent use of decision technologies to optimize the decision-making capabilities within the design space we have created. The principles and methods developed from game theory, decision and risk analysis, and utility models are incorporated into the design process for complex systems, where consistency is necessary at all levels of decision-making in order to resolve conflict and intransivity, and to facilitate negotiation and optimization.

Papers in this special issue

This issue contains nine full length papers and three technical notes. Below we discuss how these papers address the collaborative engineering research issues described earlier. Some of these papers address more than one issue as we note in our discussions.

Su et al.'s paper "Scalable Vector Graphics –based Multi-level Graphics Representation for Engineering Rich-Content Exchange in Mobile Collaboration Computing Environments" deals with both the development of an architectural framework for supporting multimedia information on mobile platforms and the representation of multimedia information. The proposed architecture uses a multi-level decomposition method that hierarchically sub-divides graphics content, which is in the scalable vector graphics (SVG) format, and distributes the information at the appropriate level needed. This obviates the need for transferring all the information. The current prototype deals with two dimensional graphics and plans are underway to extended to it other multimedia content, e.g., video and audio. *Category: Architecture and Shared representation*

Many of the techniques in vogue today for building ontologies are fairly ad-hoc. Nanda et al., in their paper "A Methodology for Product Family Ontology Development, using Formal Concept Analysis and Web Ontology Language," describe a methodology for rational construction of product ontologies. They utilize a method called Formal Concept Analysis (FCA), which has been successfully applied in computational linguistics. They use FCA to identify the common features of various engineering objects and then classify them accordingly into a concept lattice. This concept lattice is mapped into a OWL (web ontology language) representation, making it amenable to semantic web applications. *Category: Shared representation.*

Features have been the subject of study for over a decade [Shah 1995]. Chen et al. extend earlier work in this area ([Bronsvoort and Jansen 1994], [Gorti et al. 1998], [Szykman et al. 2001]) to develop a model for representing features that includes both geometry and non-geometric information. In their paper "Extended Feature Association in a Unified Modeling Scheme," they discuss how their model aids in sharing different views of a product, while maintaining the integrity of relationships between various artifacts in a collaborative environment. A related approach, where shared features and relationships are stored in a table is described by Bouikini et al. in their technical note "A Product Feature Evolution Validation Model for Engineering Change Management." While Chen et al's paper focuses on various relationships between objects, Wang and Nnaji focus on a formal representation of a feature-based representation augmenting a feature-based model by including design intent. An implementation of their model is provided in RDF (resource description framework). Another related piece is the technical note "Design Modification in a Collaborative Assembly Design Environment" by Lu et al. This work describes the development of an hierarchical assembly model that associates client-ids with leaf nodes, enabling any changes in the part (leaf-node) to be propagated to the appropriate designer. *Category: Shared representation.*

For many problems in computer-aided design, the spatial relationships between objects result in sets of non-linear equations. Geometric operations, such as moving, applied to these objects involves the solution of these equations. Algebraic techniques, such as the Newton-Raphson method, are computationally very expensive. Geometric reasoning techniques, which reason symbolically about geometry, provide efficient solution methods for such problems. The geometric constraint satisfaction problem (GCSP) can be stated as follows [Kramer 92]: "Given a collection of geometric bodies, and constraints that describe how the bodies interact with their neighbors, where must all the bodies be located in space so as to satisfy all constraints simultaneously?" Jaon-Arinyo, et al.'s paper "Constraint-based Techniques to Support Collaborative Design" addresses the GCSP in a collaborative design environment. A framework that involves a geometry-based master model that supports client views and change propagation from clients is presented. The paper discusses theoretical underpinnings of such an approach. *Category: Constraint management.*

In order to support security for nested transactions, which allow hierarchical subdivision of design effort and provide smaller units of control for various transaction management facilities, Wang et al. propose a role-based access mechanism in their paper "Intellectual Property Protection in Collaborative Design Through Information Modeling and Sharing." Data is shared between applications through UL-PML (Universal Linkage-Product Markup Language), an XML-based data modeling scheme that the authors have developed, which has many of the features discussed earlier. As data is shared at different levels of granularity, the advantages are similar to the SVG-based approach described by Su et al., while providing appropriate levels of security. *Category: Coordination and transaction management.*

Most of the papers on computer supported collaborative engineering in the engineering literature focus on technical aspects of the problem. However, the social and economical considerations should be taken into account for any successful implementation. Lu et al. in their paper “A Socio-Technical Framework for Collaborative Product Development” focus on the social and technical aspects of collaborative engineering. The framework proposed by Lu et al. is very comprehensive and includes eight stages: 1) defining a base line process, 2) identifying various stakeholders, 3) proposing a product information structure (which they call a concept structure), 4) establishing perspectives of various stakeholders (e.g., thermal, mechanical, etc.), 5) building perspective model state diagrams (PMSDs) for each concept in their concept structure, 6) performing perspective analysis on PMSDs, 7) conducting conflict management based on the results of perspective analysis, and 8) obtaining a shared reality as a final outcome. One of the novel features of their framework is the conflict management system, which consists of two parts: conflict classification and conflict resolution. Conflict classification organizes conflicts into a taxonomy of conflicts, which is used to classify conflicts, after which appropriate conflict resolution strategies are applied. Lu et al. demonstrate their framework for facility planning and development in the construction domain. *Category: Architecture and Negotiation/conflict management.*

Rather than developing and offering large number of unrelated products, manufacturers find it advantageous to deal with families of products, where product variety is obtained by variations of a generic product platform. Having access to manufacturing process knowledge during the development of the generic product platform and during the generation of product variations will result in superior quality designs. Alizon et al. in their paper “Reuse of Manufacturing Knowledge to Facilitate Platform-based Product Realization” describe a three stage process for retrieval and use of process knowledge in repositories. The first step involves a search to find the appropriate cases in the repository. This search may produce several alternatives, which are then ranked. Finally, ranked cases are mapped to the current situation, which could involve selecting either snippets or a complete case. The authors illustrate their method with examples of assembly line design for an air conditioner module in automobile production. *Category: Engineering repositories and long term knowledge management.*

Empirical studies are very important to studying and understanding engineering design. A substantial move towards observational studies, however, has yet to take hold in the engineering disciplines in any significant fashion. The paper “Empirical Studies on Inter-Organizational Collaborative Product Development in Asia Pacific Region” by Chu et al. addresses this issue by conducting six case studies in Taiwan. Their studies indicate that the companies have yet to effectively utilize software tools for collaboration. *Category: Organizational issues.*

The initial stages of any product development involves the generation of various specifications or requirements, which evolve through the entire product life cycle. Several approaches have been taken to mapping customer requirements to engineering requirements and engineering requirements to design parameters. In the technical note “ERMM: An Engineering Requirements Management Method,” Becker and Wang use a matrix-based method, similar to the Design Structure Matrix to capture relationships and associations between various engineering components. They illustrate their technique by representing requirements for an automotive A-pillar design. *Category: Organizational issues.*

Looking Forward

There are several issues for distributed collaborative design that are not addressed in this issue, but are topics for further consideration and concern. One issue that relates to the effects on globalization on design is the concern for sustainable product development. The need to consider design for reuse, remanufacture, and recycle are at the forefront of numerous industry product development plans, as information concerning products, their production and product take-back requirements extend the need for accessible design repositories, product information structures, and new collaboration tools to support concurrent lifecycle assessment analysis within the design process. Gloabalization also necessitates a design for supply chain approach. Current research on platform families, on exploring the interrelationships between enterprise product planning and engineering product development using multilevel optimization; and on an ontology-based semantic approach for real-time collaboration and communication all hold promise for future advances in collaborative product development.

Disclaimer

The summary of the papers in this guest editorial does not imply recommendation or endorsement by NIST, NSF, and NITRD, nor does it imply that any materials or equipment identified are necessarily the best available for the purpose.

References

[Boonsvoort and Jansen 1994] Bronsvoort, W. and Jansen, F., "Multi-View Feature Modelling for Design and Assembly," In *Advances in Feature-based Manufacturing*, Shah, J., Mäntylä, and Nau, D. (editors), pp. 315-330, Elsevier Science B.V.

[Carter and Baker 1992] Carter, D. and Baker, B., *Concurrent Engineering: The Product Development Environment for the 1990s*, Addison Wesley Publishing Company.

[Gorti 1998] Gorti, S., Gupta, A., Kim, G.J., Sriram, R.D., Wong, A., "An Object-Oriented Representation for Product and Design Processes," Journal of Computer-Aided Design, Vol. 30, No. 7, pp. 489–501.

[Kramer 1992] Kramer, G., *Solving Geometric Constraint Systems*, MIT Press.

[Shah 1995] Shah, J. J. and Mäntylä, M., *Parametric and Feature-based CAD/CAM: Concepts, Techniques, and Applications*, John Wiley and Sons, Inc.

[Sriram 1992] Sriram, D., Ahmed, S., and Logcher, R., "A Transaction Management Framework for Collaborative Engineering," *Engineering With Computers*, Vol. 8, pp. 213-232.

[Sriram 1997] Sriram, D., *Intelligent Systems for Engineering: A Knowledge-based Approach*, Springer Verlag.

[Szykman et al. 2000] Szykman, S., C. Bochenek, J. W. Racz, J. Senfaute, and R. D. Sriram, "Design Repositories: Next-Generation Engineering Design Databases," *IEEE Intelligent Systems*, Vol. 15, No. 3, pp. 48-55.

Szykman et al. 2001] Szykman, S., Fenves, S., Keirouz, W., and Shooter, S., "A Foundation for Interoperability in Next-Generation Product Development Systems," *Journal of Computer-Aided Design*, Vol 33, pp. 545-559.