Computers in Industry 59 (2008) 855–862

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Computer supported collaborative design: Retrospective and perspective

Weiming Shen [a,*], Qi Hao [a], Weidong Li [b]

[a] National Research Council of Canada, 800 Collip Circle, London, Ontario N6G 4X8, Canada
[b] Faculty of Engineering and Computing, Coventry University, Priority Street, Coventry CV1 5FB, UK

ARTICLE INFO

Article history:
Available online 27 August 2008

Keywords:
Computer supported collaborative design
Computer supported cooperative work
Agents
Collaborative engineering
Concurrent engineering

ABSTRACT

Industry today requires new technologies to address increasingly complex product development and the high expectations of customers. Computer Supported Collaborative Design (CSCD) emerged in response to this requirement. With the rapid advancement of Internet and Web-based technologies, CSCD has been a very active R&D area in the past 15 years and has progressed dramatically. To achieve its full potential, more and more research and commercial CSCD systems have been recently developed. The depth and breadth of these applications and systems are far beyond the traditional definition of concurrent engineering. This paper presents a review of the R&D literature on CSCD, from the pre-CSCD technologies of the 1980s to today's state-of-the-art CSCD. Research challenges and opportunities on CSCD are also discussed and highlighted.

1. Introduction

Collaborative engineering is a new concept of optimizing engineering processes with objectives for better product quality, shorter lead-time, more competitive cost and higher customer satisfaction. Based on the rapid advancement of information and communication technologies, collaborative engineering has progressed dramatically. It has been widely applied to product design, manufacturing, construction, enterprise collaboration and supply chain management. Particularly, application of collaborative engineering to product design, which is usually called Computer Supported Collaborative Design (CSCD), becomes more promising. The depth and breadth of CSCD is far beyond the traditional definition of concurrent engineering [1]. CSCD is carried out not only among multidisciplinary product development teams within the same company, but also across the boundaries of companies and time zones, with increased numbers of customers and suppliers involved in the process. Based on first-hand research and industrial experience in this area, we will present the R&D overview of the field and discuss research challenges and opportunities for the future. The rest of this paper is organized as follows: Section 2 provides a brief overview of the CSCD field; Section 3 reviews the pre-CSCD technologies of the 1980s; Section 4 depicts the emergence of CSCD in the 1990s; Section 5 describes today's state-of-the-art CSCD; Section 6 discusses future research opportunities and challenges; Section 7 presents some concluding remarks.

2. Computer Supported Collaborative Design (CSCD)

Traditional product design systems use a sequential mode of design generation, which breaks a design task into a number of sub-tasks that can be sequentially executed in a predefined workflow. Recently, such a sequential design mode has been found to be brittle and inflexible. It often requires numerous iterations, which make design expensive and time-consuming, and also limit the number of design alternatives that can be examined. On the other hand, sequential design is usually practiced with a downstream-wise information flow. Information feedback from downstream operations (e.g., process planning and manufacturing at the shop floor) to the upstream design is usually performed by human interactions. It may also cause insufficient design evaluation/optimization and hence inefficient product development due to the absence of manufacturability checks at the design stage, based on available resources.

CSCD (also called Cooperative Design, Concurrent Design, or Interdisciplinary Design) is the process of designing a product through collaboration among multidisciplinary product developers associated with the entire product lifecycle. This includes those functions such as preliminary design, detailed design, manufacturing, assembly, testing, quality control, and product service as well as those from suppliers and customers [2]. An important objective of CSCD is to address the insufficient or even

* Corresponding author.
E-mail addresses: weiming.shen@nrc.gc.ca (W. Shen), qi.hao@nrc.gc.ca (Q. Hao), weidong.li@coventry.ac.uk (W. Li).

0166-3615/$ – see front matter. Crown Copyright © 2008 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2008.07.001

absent manufacturability checks concurrently by detecting and considering conflicts and constraints at earlier design stages. To support collaborative design, information and communication technologies are used to augment the capabilities of the individual specialists, and enhance the ability of collaborators to interact with each other and with computational resources. CSCD is not only compulsory for complex products such as the development of the Boeing 777 airplane, which involved 130,000 parts, 6800 internal people and more than 10,000 external people, but also quite helpful for many middle- or even small-size products such as tooling, and electronic products [3].

With the globalization of the manufacturing industry, CSCD is required to support distributed design. Members on a collaborative team often work in parallel and independently using different engineering tools distributed at separate locations, even across enterprise boundaries and across various time zones around the world. The resulting design process is then called distributed collaborative design [4,5].

Engineering design has some unique characteristics (e.g., diverse and complex forms of information, interdisciplinary collaboration, and heterogeneous software tools), which make interactions difficult to support. Traditional approaches to sharing design information among collaborators and their tools include the development of integrated tools and the establishment of common data standards. These approaches are not good at supporting effective collaborative design because of the highly distributed nature of the design teams and engineering tools as well as the complexity and dynamics of design environments. A successful implementation of CSCD needs: a series of new strategies, including an efficient communication strategy for a multidisciplinary group of people from the design and manufacturing departments to share and exchange ideas and comments; an integration strategy to link heterogeneous software tools in product design, analysis, simulation and manufacturing optimization to realize obstacle-free engineering information exchange and sharing; and, an interoperability strategy to manipulate downstream manufacturing applications as services to enable designers to evaluate manufacturability or assembleability as early as possible [6]. On the other hand, the objective of a design team has multiple facets, for example, optimizing the mechanical function of the product, minimizing the production or assembly costs, or ensuring that the product can be easily and economically serviced and maintained. Achieving global satisfaction, cooperative strategy, such as negotiation, optimization and trade-off, is an important research issue in CSCD.

3. CSCD eve: 1980s

This section provides an overview of the related research fields, including Computer Supported Cooperative Work (CSCW), Concurrent Engineering,[1] and Human–Computer Interaction (HCI), which triggered the emergence of CSCD.

3.1. Computer supported cooperative work

According to Schmidt and Bannon [7], the term CSCW was first used by Greif and Cashman in 1984 to describe the topic of an interdisciplinary workshop that they were organizing on how to support people in their work arrangements with computers [8]. Subsequently the term was abbreviated to CSCW. The definition of CSCW and the history of this research field are beyond the scope of this paper. Readers are suggested to consult a well-established journal called CSCW. In fact, many people simply refer to this area by the term of Groupware, though others consider this to be too narrow. Generally speaking, the term Groupware is widely used in commercial software products while CSCW is used more in the research community.

3.2. Concurrent engineering

The concept of concurrent engineering was initially proposed in the late 1980s as a potential means to minimize product development time. It was defined as "a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support" [9]. In a concurrent engineering environment, techniques, algorithms and software tools are connected to allow product designers and developers to interact with each other. With concurrent engineering, more time and money are usually spent in the initial design stage to ensure the overall optimization of concept selection. Product design changes can be reduced at the late stages, leading to better-engineered products with better total quality, time and cost competitiveness. There are a number of implementation strategies, from the parallelization of product lifecycle functions to the upfront consideration of DFX activities such as design for manufacturability, assembleability, serviceability, and recycleability, to the cooperation and coordination of product design teams with different expertise [8–10], all of which has laid a solid foundation for CSCD. To ensure the success of concurrent engineering, more emphasis is put on the establishment of teamwork culture between design and manufacturing teams, and the enhancement of quick and effective communication. Balamuralikrishna et al. [11] summarized concurrent engineering as three T's: tools, training and time. Tools refer to the communication facilities between the personnel in the multidisciplinary departments to address the information exchange that is obstructed by the complexity and wide range of specialized disciplinary areas and interdependent activities. Training provides a mechanism for employees to work collaboratively and concurrently, making the best use of the company's resources. Time means corporations need time to carefully investigate and plan concurrent engineering as it involves many complex software tools and information infrastructures. Many reported cases have shown that a hurried implementation of concurrent engineering usually has a high probability of backfiring. In industry, more companies have realized the great benefits of concurrent engineering. Honda Racing F1 Team's new development process is one good example of UGS PLM [12].

3.3. Human–computer interaction

Research on HCI was started as early as computers were invented. Myers [13] presented a brief HCI history in 1998. However, there is currently no widely agreed definition that covers a full range of topics that form the area of HCI, from computer graphics to ergonomics, and from virtual reality to digital human modelling.

Computer graphics was born from the use of CRT and pen devices very early in the history of computers. Work in computer graphics has continued to develop algorithms and hardware to allow the display and manipulation of ever more realistic-looking objects—which led to rapid developments of CAD/CAM tools in the 1980s.

There are many HCI related international conferences with the most widely recognized one being the HCI International Conference Series [14].

With its initial R&D focus on interaction between one user and one computer, HCI R&D was then extended to human–human interaction via networked computers, which is, in fact, the essence of CSCD.

[1] Concurrent Engineering and Collaborative Engineering are of the same acronym. To avoid confusion, full words are used in the paper.

3.4. Blackboard, DAI and software agents

The blackboard architecture was proposed in the HEARSAY project [15] as a means to organize and control large Artificial Intelligence (AI) systems. Its first version HEARSAY I was used for speech recognition based on the idea of cooperating independent acoustic, lexical, syntactic, and semantic knowledge sources.

The introduction of the Contract-Net is a milestone in the history of Distributed Artificial Intelligence (DAI). The Contract-Net protocol was developed by Smith [16] in 1980 and demonstrated on a distributed sensing system. The Contract-Net implemented a negotiation-based approach for allocating tasks dynamically to nodes in the network. When a node has to solve a problem, which is beyond its expertise, it broadcasts a task-announcement message that describes the task to be solved. Nodes that receive the message and wish to solve the problem then return a bid message. The node that issued the task-announcement message, called the *manager*, waits for bids for a certain period and then selects one (or more) bidder(s) to do the task, who is called the *contractor*. Thus, the choice of the contractor is done after the selection by the manager and by mutual agreement. To be able to function correctly, the system must include a high-level protocol that defines several types of messages with a structured content. Contract-Net has been widely used in various agent systems for negotiation among agents.

The Contract-Net protocol offered an early practical means for dealing with open systems from a software engineering point of view. Contrary to the blackboard approach, there is no shared memory where data and partial results are made available to the various knowledge sources. The Contract-Net approach has separate knowledge sources attached to distinct nodes in a computation network.

The actor model proposed by Hewitt [17] offers a model of computation for open systems at a finer grain than the Contract-Net approach. In the actor approach, problem solving is viewed as the result of the interaction of the activities of knowledge sources working independently and locally (with limited expertise and limited knowledge). Each node communicates with a limited number of other knowledge sources.

The concept of agents has evolved from the concepts of blackboard, Contract-Net, and actors. Separately, in applied fields such as manufacturing, object-oriented systems were developed with increasing intelligence being incorporated into the objects. What began as passive objects became 'active objects' or 'rule-based objects' or 'intelligent objects' and finally 'intelligent agent objects' as this stream of evolution merged with that of DAI [18]. All these technologies provide a good foundation for developing collaborative design systems. Under the CSCD context, an agent can be considered as a software system that communicates and cooperates with other software systems to solve a complex problem, which is beyond the capability of each individual software system [19].

4. CSCD emergence: 1990s

Modern design, particularly engineering design, is intrinsically multidisciplinary [20]. Various tools such as CAD/CAM/CAE (Computer-Aided Design/Computer-Aided Manufacturing/Computer-Aided Engineering) tools, developed and commercialized by different vendors without common specifications (or even with intentionally defined unique specifications for self-protection), do not address the needs of multidisciplinary design. On the other hand, large organizations like Boeing, Airbus and GM, must find a way to coordinate their research and development teams, which are geographically distributed around the world in an effective way to carry out new product developments within a very limited time frame. Technologies like CSCW and intelligent agents have been investigated to address this need, particularly to enhance communication, cooperation, and coordination among design team members as well as software tools. Some CSCW tools like groupware were directly used to facilitate communication among engineers and designers. Software agents were used to integrate different engineering tools. Examples of early applications of software agents in collaborative design include PACT [21], DIDE [22], and SiFAs [23].

With its emergence around 1993, the Web was quickly applied in the development of collaborative design systems, particularly for geographically distributed designers to share design documents. Along with the Web, a number of associated representation technologies have been developed, such as Hyper Text Mark-up Language (HTML), eXtensible Mark-up Language (XML), Virtual Reality Mark-up Language (VRML), to enable better cross-platform and cross-enterprise exchange of multi-media information and design models. In terms of system infrastructure, many early collaborative design systems were also developed using the Blackboard architecture and distributed object technologies like CORBA and COM/DCOM.

5. CSCD today

During the past 15 years, a large number of CSCD systems have been developed and reported, especially on the applications of CSCW, Web, software agents, and recently Web Services, Semantic Web and Computing Grids to collaborative design. A few CAD vendors and other software firms also started developing and promoting collaborative design tools, for example, AutoDesk's Inventor™ [24] and Buzzsaw™ [25], Streamline [26], ArchiCAD TeamWork™ [27], CoCreate's OneSpace Solution [28], Matrix PLM Platform [29], and UGS's PLM solutions [30]. The following subsections present a brief review of these researches, developments, and applications.

5.1. CSCW for collaborative design

CSCW has been a very active research field during the past two decades and the 20th anniversary edition of its international conference was held in November 2006. Design (particularly engineering product design and software design) has been one of the most important applications of CSCW technologies. The most widely used CSCW techniques in collaborative design systems include *groupware* techniques for facilitating communication among design team members and *context awareness* techniques for enhancing coordination among team members.

5.2. Web-based collaborative design

The Web was originally developed for information sharing within internationally dispersed teams and the dissemination of information by support groups. Proposed and developed early in the 1990s, the Web has quickly become a convenient media to publish and share information relevant to the design process, from concept generation and prototyping to virtual manufacturing and product realization. It has been adopted as the most popular implementation architecture of a collaborative product development (including design and manufacturing) tool. A CSCD system developed with the Web as a backbone will primarily provide: (1) access to catalogue and design information on components and sub-assemblies; (2) communication among multidisciplinary design team members (including customers, designers and production engineers) in multimedia formats; (3) authenticated access to design tools, services and documents. However, since the Web is still fast evolving, particularly with the development of Web services and Semantic Web technologies (see Section 5.7 for more discussions), many researchers and working groups in and outside W3C are working hard to improve the current Web infrastructure and supporting tools. Web-based infrastructure has been used in a number of collaborative product design systems. In most cases, the Web is primarily used by multidisciplinary team members as a medium to share design data/information/knowledge; while in some cases, it is integrated with other related technologies and is used for product data management and project management.

A comprehensive review of some Web-based tools and systems can be found in [4,31]. Most Web-based collaborative design systems are developed using Java and CORBA [32–34], some others are developed using Common Lisp (WWDL [35]), and Prolog (WebCADET [36]). In addition to HTML and Java Applets for developing client side user interfaces, ActiveX [37,38] and VRML [34,39] are widely used.

However, Web technology alone is not a complete solution to collaborative design systems, although it makes communication physically viable through a common network. In order to collaborate on a distributed design project, remote engineers and designers need active supports to coordinate their efforts. This coordination involves the translation of terminology among disciplines, locating/providing engineering analysis services, virtual prototyping services, and project management [40–42]. Web servers should not only be a repository of information but also provide intelligent services to help users to solve design problems. Such servers may be called software agents and will be discussed below.

5.3. Agent-based collaborative design

Application of software agents to collaborative design has been demonstrated by various research projects. PACT [21] might be one of the earliest successful projects in this area. The interesting aspects of PACT include its federation architecture using facilitators and wrappers for legacy system integration. SHARE [43] was concerned with developing open, heterogeneous, network-oriented environments for concurrent engineering, particularly for design information and data capturing and sharing through asynchronous communication. SiFAs [23] was intended to address the issues of patterns of interaction, communication, and conflict resolution using simple single-function agents. DIDE [22] was developed to study system openness, legacy systems integration, and distributed collaboration. ICM [44] developed a shared graphical modelling environment for collaborative design activities. Co-Designer [45] was intended to support localized design agents in the generation and management of conceptual design variants. Concept Database [46] described a strategic design support for version control, workflow management and information gathering. A-Design [47] presented a new design generation methodology, which combines aspects of multi-objective optimization, multi-agent systems, and automated design synthesis. It provided designers with a new search strategy for the conceptual stages of product design that incorporates agent collaboration with an adaptive selection of design alternatives. Some projects also addressed the issue of integration and collaboration among product design, process planning, and manufacturing scheduling [48–50].

In agent-based collaborative design systems, software agents are mostly used for supporting cooperation among designers, enhancing interoperability between traditional computational tools, or allowing better simulations (particularly distributed simulations). The book on "Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing" [51] provides a detailed discussion on issues in developing agent-based collaborative design systems and a review of several well-known projects or systems.

5.4. Integration of Web and agent technologies for collaborative design

Both the Web and agent technologies are very useful in implementing collaborative design systems. The attractiveness of the Web for propagating information makes it appropriate to integrate with agents for accessing and manipulating information automatically. The challenge is to build a Web-based environment that enables and supports seamless interactions between human designers, software agents, and Web servers using the available emerging technologies [52].

A Web-based collaborative design system usually uses a client/server architecture, in which the interaction between components is predefined. This kind of approach is insufficient to support dynamic collaborative design environments, where tasks are usually involving complex and non-deterministic interactions, producing results that might be ambiguous and incomplete. An agent-based collaborative design system is a loosely coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities [51]. Software agents in such systems are communicative, collaborative, autonomous (or semi-autonomous), reactive (or even proactive), and intelligent. Different system architectures have been proposed and used to implement agent-based systems. Some systems use approaches similar to the blackboard architecture or the client/server architecture, e.g., the Design Board approach in SiFAs [23]; the shared graphical modelling approach in ICM [44]; and the shared database approach [46]. Most systems use federated system architectures, e.g., a facilitator approach in PACT [21] and a mediator approach in ABCDE [47]. A few systems use the autonomous agent approach [22].

Although agent technology has been recognized as a promising approach for developing collaborative design systems, those agents that have so far been implemented in various prototype and industrial applications are actually not very "intelligent". In this view, agent applications in the Web-based collaborative design field are still facing many challenging questions. WebBlow [52] is an interesting attempt on the integration of the Web and software agents in implementing a multidisciplinary design optimization environment [53,54]. Before the emergence of Web Services, the concept of an active Web server was proposed to integrate the Web and agent technologies [55]. Since the active Web servers have very similar features of Web Services, it is natural for the further work to implement collaborative design systems using Web Services [56].

During the past 5–6 years, more and more collaborative design systems have been developed using Web Services as well as Semantic Web and Grid Computing techniques. It is now easy to find hundreds of publications on this topic, e.g., there were more than 30 related papers presented at CSCWD 2006 [57] and more than 40 at CSCWD 2007 [58].

5.5. New representation schemes for collaborative design

In a collaborative design process, product models need to be disseminated in a broader scope. Product models are the most important knowledge and properties of the product development companies, so companies are usually reluctant to share these models directly to avoid the leakage of the commercial secrets to competitors. This consideration makes it difficult to realize the full potential and benefits of collaboration. On the other hand, a product model is proprietary to a CAD system. In a collaborative design environment with multiple users, it is infeasible or uneconomical to install a CAD system for every user to view or manipulate the product model. To address these concerns, research efforts have been made to develop new representation schemes of product models based on VRML, including eXtensible 3D (X3D) [59], Web 3D (W3D) [60], Universal 3D (U3D) [61], JT [62], and OpenHSF [63]. These representation schemes retain the essential visualization information of proprietary product models to support display-based manipulations, such as rotation and zooming, annotation, and mark-up. Most of these schemes are open in formats and the features inside are neutral so that they have much broader acceptance than those of the proprietary product models. Major applications of these schemes for collaboration include customer surveys of product concepts and initial models, high-level project reviews among management, development and service departments, sales promotion, e-documents (e.g., Acrobat 3D), sharing catalogues, and visualization functions in Product Data Management/Product Lifecycle Management (PDM/PLM) systems. Since only the visualization information is included in these schemes, crucial design information is protected.

5.6. New visualization systems for collaborative design

In order to support the new representation schemes, some new visualization systems have been developed, e.g., Cimmetry Systems AutoVue [64], Actify SpinFire [65], SolidWorks eDrawing [66], RealityWave ConceptStation [67], and Autodesk Streamline [26]. The visualization-based platforms are cost-effective solutions to replace CAD systems to facilitate collaborative activities for various users. With new representation schemes and visualization systems, teams can collaborate more effectively, such as by taking on design discussions, reviewing new products, and conducting customer surveys to get design feedback as early as possible. This may overcome some drawbacks of proprietary CAD product models that hinder collaborative activities. A visualization-based collaborative system uses two-tier or three-tier client/server architecture. Java Applet and Microsoft ActiveX technologies are widely used for developing Web-based or specialized clients. Core functions or services are implemented in Java Servlet or Microsoft.Net ASP at the server side to provide system support and maintenance [68–70]. Recently, Java3D has been widely used to enable visualization-based manipulations of 3D objects and scenes, e.g., to build, render and control 3D objects for Web-based collaboration [71,72].

5.7. Product data management and product lifecycle management systems

Product Data Management (PDM) and Product Lifecycle Management (PLM) systems have been adopted by industry to facilitate engineering design. Such systems promised that the "right information" is provided to the "right person" in the "right time" according to the "right order". Mainstream solutions include UGS TeamCenter [73], PTC Windchill [74], ENOVIA VPLM, ENOVIA MatrixOne and ENOVIA SmarTeam [75]. Actually, the systems can be regarded as the system-level integrated implementation of the current collaborative technologies to support engineering design [76]. These systems have distinguished characteristics, but they share the following common functionalities:

- Team management: to map the structure of a product development team to a hierarchical structure of organizations.
- Product structure management: usually a Bill Of Materials (BOM) structure root, to represent the physical structure of a developed product at different levels, which generally contains assemblies, subassemblies and components.
- Workflow and process management: to allow an organization to automate procedures in which information, tasks, and documents are passed among participants.
- Design change management: to manage change information in design processes.
- Visualization-based collaborative workspace: to retain the visualization information of product models based on lightweight visualization schemes to support multiple users to manipulate the product models, such as rotation, measurement, annotation, and mark-up.
- Integration interfaces with CAD, shop floor execution systems, legacy enterprise information systems, and other partners on the product value chain.

From a research perspective, important issues need to be solved for the better application of PDM/PLM systems. The design process, along with the product itself, should be considered as a crucial component of an engineering enterprise's intellectual capital [77]. Five aspects of design processes have been studied, including support for design information transformations, support for design decision-making, modelling and representation of design processes, analysis of design processes, and synthesis of design processes. Qiu and Wong [78] developed a dynamic workflow mechanism to accommodate the changes during design by minimizing the repetitive execution of finished workflow nodes. This approach can address the data integrity issue by managing various workflow data such as node properties and scripts. Concurrency control is the foremost mechanism to organize synchronous activities to avoid conflicts. Locking is a primary means in managing concurrency control to prevent people from colliding, and three types of locking, i.e., non-optimistic, optimistic and visual, have been developed and used in various applications. Negotiation can formalize and implement the mediation and facilitation functions among people to handle conflicts. Some research projects [76,79,80] have been carried out to enhance PDM systems to support pre- and post-design stages. Huang et al. [81] developed a Web-based system to manage Engineering Changes (ECs).

5.8. Product data exchange

Standard for the Exchange of Product Model Data (STEP) is an ISO standard that provides a complete unambiguous, computer-interpretable definition of the physical and functional characteristics of a product throughout its lifecycle. It promises to ensure product data exchange among different computer systems and applications associated with the complete product lifecycle including design, manufacturing, utilization, and maintenance. In order to deploy the Web infrastructure effectively and efficiently in collaborative product development systems, it is better to integrate STEP into the Web infrastructure. Although some research and development groups are working actively in this area, e.g., XML Transactions for STEP by STEP Tools Inc. [82], more fundamental research and development work is still needed in this area.

5.9. CSCD related conferences

Collaborative engineering and particularly collaborative design has been a major topic for some major international conferences, including ASME International Design Engineering Technical Conferences [83] and the ISPE Concurrent Engineering Conference Series [84]. Among many others, a series of international workshops/conferences on Computer Supported Cooperative Work in Design (CSCWD) [85] has its focus in this area. It was started in 1996 and had its 12th edition in 2008.

6. CSCD tomorrow and research opportunities

CSCD has been an active R&D area for about 15 years. Some manufacturing and engineering companies have partially implemented in-house collaborative design systems. We expect a great future for CSCD and envision future collaborative product development systems as being:

- Fully integrated with all necessary software tools connected through the network covering the full product lifecycle from conceptual design, to detailed design (with detailed modelling, simulation and optimization), virtual prototyping, manufacturing, service and maintenance, and final disposal.
- Integrated with physical testing and validation systems for "hardware-in-the-loop" simulations during the new product development.
- Implemented as semi-automated interactive systems that involve human interventions.
- Operated on a Grid computing environment with automated computing load balancing, quick access and fast transfer of large volumes of engineering data.
- Secured with sophisticated security and privacy protection mechanisms.
- Able to allow users to choose favourite software tools according to their experience and preference.
- Able to provide different users (including engineers and managers, sales and services staff, as well as customers and suppliers) with different access privileges to the same product data/information.

To achieve this vision, a number of challenging issues have been identified for an academic research, further development, and wider deployment of collaborative design systems in industry. In fact, these challenges are also opportunities for the CSCD research community. Among others, the following areas are believed to be future research opportunities and challenges:

- Ontology and semantics based integration: one of the most difficult tasks in collaborative product design is to agree on the ontological commitments that enable knowledge-level communication among the distributed design parties. Another difficulty is the integration of the various available design tools. If the tool data and models are encapsulated, rather than using a standardized and unified approach, each tool will be free to use the most appropriate internal representations and models for its intended tasks. This is not a new research topic, but the progress in this area has not been satisfactory. The emergence of the Semantic Web makes progress in this area more likely to occur.
- Interoperability among product models: models help designers understand the nature of a design process by ignoring some of the not-so-important details. When deciding how to model a design process, determining the appropriate levels of abstraction is very critical for the model to be beneficial to its users. A key issue in collaborative design from a designer's perspective is how to bridge the multi-faceted models required to support a complex design project at various stages of the design process. The challenge is to use the relevant model for each task (the right abstraction and granularity) and to communicate the results in a suitable form to the various parties involved, whose needs are different and interests are diverse. One way to address this issue is through collaborative design process modelling which has been an active research topic recently, but significant efforts are still required.
- Product-centric design methodology: a product-centric design methodology is considered as a suitable approach for distributed collaborative product design. Featuring its self-learning ability, product-centric design fits well in a dynamically changing environment. Comprehensive care needs to be taken in modelling, collaboration, design and development issues in the whole product lifecycle. Fundamental research is still required in this area.
- Data/information/knowledge management: challenges in this area include knowledge discovery, support for natural language processing and information retrieval, the capturing of design intent in multimedia formats, dynamic knowledge management, self-learning, reasoning and knowledge reuse. Based on the current Web infrastructure, users are allowed to access server resources primarily through HTTP and FTP. Using appropriate protocols to access the right data at the right locations is essential in collaborative product development environments. This feature is particularly useful in large collaborative product design and engineering projects where access to large volumes of data at different locations is frequent.
- Collaborative intelligent user interfaces: human involvement in the collaborative engineering design processes is highly probable. Designers need to interact with a design system and negotiate with peers via a user interface. The challenge is to make intelligent interfaces available to all resources so that the designers will have more flexibility to do efficient and effective design. The interfaces should be integrated, expressive, goal oriented, cooperative, easy to use, and customizable;
- Distributed design project management: there must be some way of managing all the resources involved, including people, organizations, software tools, and equipment. Relevant research issues are collaborative workflow, conflict management, cost and task management, activity scheduling, and computing resource management. In an interesting experimental work, Hammond et al. [86] used a socio-technical theory as a framework to explore differences in engineering design team decision-making as a function of various media of communication. Their results indicate that design teams communicating via an electronic medium perceive an increase in mental workload and interact less frequently, but for a greater total amount of time. These results brought interesting implications and suggestions for the management of distributed design teams.
- Drag and drop functionality: drag and drop is a highly desired function in collaborative design using multiple computational tools. For example, a part designed under a CAD system may be moved to a CAE tool's graphical interface for analysis and simulation, and to a DFM tool's graphical interface for manufacturability analysis. It becomes more convenient if there is a drag and drop function that can copy or move a graphical object from one CAD/CAE system to another, particularly in a Web-based collaborative design environment. In fact, it is a type of communication between the two systems through the moving graphical object. The challenge is therefore to develop a common model or language for these related systems. A significant amount of research is needed to determine standard geometric representations for features that can be used by different CAD and simulation tools. It also requires research and development of drag and drop type standards similar to Object Linking and Embedding (OLE) which provides a protocol for organizing data in a standard format for exchange between different systems.
- Security and privacy: with the implementation and deployment of CSCD applications in industry, security and privacy issues become more and more important. The number of papers on this topic submitted to CSCWD conferences [85] has increased significantly during the past few years. This will continue, particularly with more practical techniques and applications.

- Software self-management and self-healing: since software self-management and self-healing have become an active research area, it would be natural to extend the research into CSCD systems.
- Social software and mass collaboration: social software approaches and Wiki-style collaboration tools may be developed and used for knowledge-intensive collaborative design systems [87,88].
- Cultural and social issues: with the globalization of the manufacturing industry and the development of worldwide production consortia, special attention is required with respect to cultural problems. Future collaborative design systems will need to integrate results from human sciences in order to address the cultural differences of designers and product users.

7. Concluding remarks

CSCD has been recognized not only by the academic research community but also by industry as a way to address the requirements resulting from increasingly complex product development and high customer expectation. With the significant development and advancement of Internet and Web-based technologies during the past 15 years, CSCD has progressed dramatically. To achieve its full potential and the vision of fully integrated collaborative product development systems, significant R&D efforts are still required. Some research challenges discussed in this paper may be addressed within next few years, while some of them may need a few decades to be thoroughly addressed.

References

[1] J. Hartley, Concurrent Engineering, Productivity Press, Cambridge, MA, 1992.
[2] E. Sprow, Chrysler's concurrent engineering challenge, Manufacturing Engineering 108 (4) (1992) 35–42.
[3] K.T. Ulrich, S.D. Eppinger, Product Design and Development, 3rd ed., McGraw-Hill, 2000.
[4] W. Shen, L. Wang, Web-based and agent-based approaches for collaborative product design: an overview, International Journal of Computer Applications in Technology 16 (2/3) (2003) 103–112.
[5] W.D. Li, S.K. Ong, A.Y.C. Nee, C.A. McMahon (Eds.), Collaborative Product Design and Manufacturing Methodologies and Applications, Springer, 2007.
[6] W.D. Li, S.K. Ong, A.Y.C. Nee, Integrated and Collaborative Product Development Environment—Technologies and Implementation, World Scientific Publisher, 2006.
[7] K. Schmidt, L. Bannon, Taking CSCW Seriously, Computer Supported Cooperative Work 1 (1/2) (1992) 7–40.
[8] I. Greif (Ed.), Computer-Supported Cooperative Work: A Book of Readings, Morgan Kaufmann Publishers, San Mateo, California, 1988.
[9] J. Turino, Managing Concurrent Engineering, Van Nostrand Reinhold, New York, 1992.
[10] B. Prasad, Concurrent Engineering Fundamentals: Integrated Product Development, vol. II, Prentice Hall, One Saddle River, NJ, 1997.
[11] R. Balamuralikrishna, R. Athinarayanan, X.S. Song, The relevance of concurrent engineering in industrial technology programs, Journal of Industrial Technology 16 (3) (2000) 1–5.
[12] UGS, PLM Solutions, http://www.ugs.com/CaseStudyWeb/dispatch/viewCaseStudy.html?id=147.
[13] B.A. Myers, A brief history of human computer interaction technology, ACM Interactions 5 (2) (1998) 44–54.
[14] HCI International. URL: http://www.hci-international.org/.
[15] R. Reddy, L. Erman, R. Fennel, R. Neely, The HEARSAY speech understanding system: an example of the recognition process, IEEE Transactions on Computers C-25 (1976) 427–431.
[16] R.G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Transactions on Computers C-29 (12) (1980) 1104–1113.
[17] C. Hewitt, Control structure as patterns of passing messages, in: P.H. Winston, R.H. Brown (Eds.), Artificial Intelligence: An MIT Perspective, MIT Press, 1979, pp. 435–465.
[18] D.H. Norrie, A.D. Kwok, Object-oriented distributed artificial intelligence, in: H. Maurer (Ed.), New Results and New Trends in Computer Science, LNCS 555, Springer-Verlag, 1991, pp. 225–242.
[19] W. Shen, Q. Hao, H. Yoon, D.H. Norrie, Applications of agent systems in intelligent manufacturing: an update review, International Journal of Advanced Engineering Informatics 20 (4) (2006) 415–431.
[20] T. Tomiyama, Collaborative product development in ill-structured problem domains, in: Proceedings of the 10th International Conference on CSCW in Design, 2006, pp. 15–20.
[21] M.R. Cutkosky, R.S. Engelmore, R.E. Fikes, M.R. Genesereth, T.R. Gruber, W.S. Mark, J.M. Tenenbaum, J.C. Weber, PACT: an experiment in integrating concurrent engineering systems, IEEE Computer 26 (1) (1993) 28–37.
[22] W. Shen, J.P. Barthès, DIDE: a multi-agent environment for engineering design, in: Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS'95), San Francisco, CA, (1995), pp. 344–351.
[23] D.C. Brown, B. Dunskus, D.L. Grecu, I. Berker, SINE: support for single function agents, in: Proceedings of Applications of AI in Engineering, Udine, Italy, 1995.
[24] Autodesk Inventor. http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=4246282.
[25] Autodesk Buzzsaw. http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2407898.
[26] Autodesk Streamline. http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2164339.
[27] Graphisoft ArchiCAD TeamWork™. http://www.graphisoft.com/products/archicad/teamwork/.
[28] CoCreate, OneSpace. URL: http://www.cocreate.com/.
[29] Matrix PLM Platform. URL: http://www.matrixone.com/matrixonesolutions/plm_platform.html.
[30] UGS, PLM Solutions. URL: http://www.ugs.com/solutions/.
[31] W.D. Li, Z. Qiu, State-of-the-art technologies and methodologies for collaborative product development systems, International Journal of Production Research 44 (13) (2006) 2525–2559.
[32] V. Jagannathan, G. Almasi, A. Suvaiala, Collaborative infrastructures using the WWW and CORBA-based environments, in: Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'96), 1996, pp. 292–297.
[33] G.Q. Huang, K.L. Mak, Web-based morphological charts for concept design in collaborative product development, Journal of Intelligent Manufacturing 10 (1999) 267–278.
[34] A. Wallis, Z. Haag, R. Foley, A multi-agent framework for distributed collaborative design, in: Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'98), 1998, pp. 282–287.
[35] Z. Zdrahal, J. Domingue, The World Wide Design Lab: an environment for distributed collaborative design, in: Proceedings of 1997 International Conference on Engineering Design, Tampere, Finland, 1997.
[36] N.H.M. Caldwell, P.A. Rodgers, WebCADET: facilitating distributed design support, in: Proceedings of the IEE Colloquium on Web-Based Knowledge Servers, London, UK, 1998, pp. 9/1–9/4.
[37] G.Q. Huang, S.W. Lee, K.L. Mak, Web-based product and process data modelling in concurrent 'design for X', Robotics and Computer-Integrated Manufacturing 15 (1) (1999) 53–63.
[38] G.Q. Huang, K.L. Mak, Design for manufacture and assembly on the Internet, Computer in Industry 38 (1) (1999) 17–30.
[39] Z. Zdrahal, J. Domingue, The world wide design lab: an environment for distributed collaborative design, in: Proceedings of the 1997 International Conference on Engineering Design, Tampere, (1997), pp. 19–21.
[40] R.H. Allen, S. Nidamarthi, S.P. Regalla, R.D. Sriram, Enhancing collaboration using an Internet integrated workbench, in: Proceedings of the 1999 ASME Design Engineering Technical Conference, Las Vegas, NV, 1999.
[41] X.Q. Liu, S. Raorane, M.C. Leu, A web-based intelligent collaborative system for engineering design, in: W.D. Li, S.K. Ong, A.Y.C. Nee, C.A. McMahon (Eds.), Collaborative Product Design and Manufacturing Methodologies and Applications, Springer, 2007, pp. 37–58.
[42] F. Mervyn, A. Senthil Kumar, A.Y.C. Nee, A 'plug-and-play' computing environment for collaborative product design and manufacturing across an extended enterprise, in: W.D. Li, S.K. Ong, A.Y.C. Nee, C.A. McMahon (Eds.), Collaborative Product Design and Manufacturing Methodologies and Applications, Springer, 2007, pp. 71–92.
[43] G. Toye, M. Cutkosky, L. Leifer, J. Tenenbaum, J. Glicksman, SHARE: a methodology and environment for collaborative product development, in: Proceedings of the 2nd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Society Press, 1993, pp. 33–47.
[44] R. Fruchter, K.A. Reiner, G. Toye, L.J. Leifer, Collaborative mechatronic system design, Concurrent Engineering: Research and Applications 4 (4) (1996) 401–412.
[45] M.J. Hague, A. Taleb-Bendiab, Tool for management of concurrent conceptual engineering design, Concurrent Engineering: Research and Applications 6 (2) (1998) 111–129.
[46] A. Varma, A. Dong, B. Chidambaram, A. Agogino, W. Wood, Web-based tool for engineering design, in: Proceedings of the AID'96 Workshop on Agents and Web-Based Design Environments, 1996.
[47] M.I. Campbell, J. Cagan, K. Kotovsky, A-Design: an agent-based approach to conceptual design in a dynamic environment, Research in Engineering Design 11 (1999) 172–192.
[48] S. Balasubramanian, F. Maturana, D.H. Norrie, Multi-agent planning and coordination for distributed concurrent engineering, International Journal of Cooperative Information Systems 5 (2/3) (1996) 153–179.
[49] W. Shen, F. Maturana, D.H. Norrie, MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing, Journal of Intelligent Manufacturing 11 (3) (2000) 237–251.
[50] M. Mahesh, S.K. Ong, A.Y.C. Nee, A web-based framework for distributed and collaborative manufacturing of engineering parts, in: W.D. Li, S.K. Ong, A.Y.C. Nee,

C.A. McMahon (Eds.), Collaborative Product Design and Manufacturing Methodologies and Applications, Springer, 2007, pp. 141–154.
[51] W. Shen, D.H. Norrie, J.P. Barthes, Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing, Taylor and Francis, London, UK, 2001.
[52] Y. Wang, W. Shen, H. Ghenniwa, WebBlow: a Web/agent-based multidisciplinary design optimization environment, Computers in Industry 52 (1) (2003) 17–28.
[53] W. Shen, H.H. Ghenniwa, Multidisciplinary design optimization: a framework for technology integration, in: Proceedings of the 1st International Conference on Multidisciplinary Design Optimization, London, ON, (2001), pp. 22–28.
[54] W. Shen, H.H. Ghenniwa, A distributed multidisciplinary design optimization framework based on web and agents, in: Proceedings of the 2002 ASME DETC/CIE Conference, September 29 to October 2, Montreal, Canada, 2002, DETC2002/CIE-34461.
[55] W. Shen, Web-based infrastructure for collaborative product design: an overview, in: Proceedings of the 5th International Conference on CSCW in Design, Hong Kong, (2000), pp. 239–244.
[56] W. Shen, Q. Hao, A service oriented framework for blow molded automotive parts design and optimization, in: Proceedings of the SAE 2004 Congress, Detroit, MI, 2004, SAE 2004-01-1244.
[57] W. Shen, et al. (6 Co-Eds.), in: Proceedings of the 10th International Conference on CSCW in Design, vol. 1/2, Nanjing, China, May 3–5, IEEE Press, 2006.
[58] W. Shen, et al. (7 Co-Eds.), in: Proceedings of the 11th International Conference on CSCW in Design, vol. 1/2, Melbourne, Australia, April 26–28, IEEE Press, 2007.
[59] X3D (eXtensible 3D). URL: http://www.x3d.com.
[60] W3D (Web 3D). URL: http://www.macromedia.com.
[61] U3D. URL: http://www.intel.com/technology/systems/u3d/.
[62] JT Open. URL: http://www.jtopen.com.
[63] OpenHSF. URL: http://www.openhsf.org.
[64] Cimmetry AutoVue, http://www.cimmetry.com/.
[65] Spinfire, http://autoweb.net/web2006/products-spinfire.shtml.
[66] Solidworks eDrawings, http://www.solidworks.com/edrawings.
[67] RealityWave ConceptStation, http://products.datamation.com/e-business/groupware/982097698.html.
[68] N. Shyamsundar, R. Gadh, Collaborative virtual prototyping of product assemblies over the Internet, Computer-Aided Design 34 (10) (2002) 755–768.
[69] L. Chen, Z.J. Song, L. Feng, Internet-based real-time collaborative assembly modelling via an e-assembly system: status and promise, Computer-Aided Design 36 (9) (2004) 835–847.
[70] S.S. Zhang, W. Shen, H. Ghenniwa, A review of Internet-based product information sharing and visualization, Computers in Industry 54 (1) (2004) 1–15.
[71] L. Wang, S. Lang, W. Shen, A Java3D Enabled Cyber Workspace, Communication of the ACM 45 (11) (2002) 45–49.
[72] W.D. Li, J.H. Fuh, Y.S. Wong, An Internet-enabled integrated system for co-design and concurrent engineering, Computers in Industry 55 (1) (2004) 87–103.
[73] UGS TeamCenter, http://wwwprd.ugs.com/products/teamcenter/.
[74] PTC Windchill, http://www.ptc.com/appserver/mkt/products/home.jsp?k=37.
[75] ENOVIA VPLM, MatrixOne and SmarTeam, http://www.3ds.com/products-solutions/plm-solutions/enovia/products/.
[76] X.W. Xu, T. Liu, A web-enabled PDM system in a collaborative design environment, Robotics and Computer Integrated Manufacturing 19 (4) (2003) 315–328.
[77] J.H. Panchal, M.G. Fernández, C.J.J. Paredis, J.K. Allen, F. Mistree, Leveraging design process related intellectual capital—a key to enhancing enterprise agility, in: W.D. Li, S.K. Ong, A.Y.C. Nee, C.A. McMahon (Eds.), Collaborative Product Design and Manufacturing Methodologies and Applications, Springer, 2007, pp. 211–243.
[78] Z.M. Qiu, Y.S. Wong, Dynamic workflow change in PDM systems, Computers in Industry 58 (5) (2007) 453–463.
[79] M. Saad, M.L. Maher, Shared understanding in computer-supported collaborative design, Computer-Aided Design 28 (3) (1996) 183–192.
[80] S.R. Fussell, R.E. Kraut, F.J. Lerch, W.L. Scherlis, M.W. McNally, J.J. Cadiz, Coordination, overload and team performance: effects of team communication strategies, in: Proceedings of the ACM Conference on Computer-Supported Cooperative Work, November 14–18, Seattle, Washington, USA, 1998, pp. 275–284.
[81] G.Q. Huang, W.Y. Yee, K.L. Mak, Development of a web-based system for engineering change management, Robotics and Computer Integrated Manufacturing 17 (3) (2001) 255–267.
[82] STEP Tools, XML Transaction for STEP. URL: http://www.steptools.com/projects/xml/.
[83] ASME International Design Engineering Technical Conferences, http://www.asme.org/events/.
[84] ISPE Concurrent Engineering Conference Series, http://www.ispe-org.net/page002.html.
[85] CSCWD Working Group and Conferences, http://www.cscwd.org/.
[86] J.M. Hammond, C.M. Harvey, R.J. Koubek, W.D. Compton, A. Darisipudi, Distributed collaborative design teams: media effects on design processes, International Journal of Human–Computer Interaction 18 (2) (2005) 145–165.
[87] D. Tapscott, A.D. Williams, Wikinomics: How Mass Collaboration Changes Everything, Penguin Group, New York, NY, 2007.
[88] D. Richards, Collaborative Knowledge Engineering: Socialising Expert Systems, in: Proceedings of the 11th International Conference on CSCW in Design, 2007, pp. 635–640.

Weiming Shen is a Senior Research Scientist at the National Research Council Canada. He is an Adjunct Professor in systems design engineering at the University of Waterloo and an Adjunct Professor in software engineering at the University of Western Ontario. He received his Bachelor (1983) and Master's (1986) degrees from Northern Jiaotong University, China and his PhD degree (1996) from the University of Technology of Compiègne, France. He has been working on intelligent software agents and their applications to collaborative engineering design and intelligent manufacturing for about 16 years. He has published two books and over 280 papers in scientific journals and international conferences, and co-edited five books and ten conference proceedings in the related areas. He is a Co-Chair of the Technical Committee on Computer Supported Cooperative Work in Design (CSCWD) with the IEEE SMC Society.

Qi Hao is a Research Scientist at the National Research Council Canada. She received her B.Sc., M.Sc. and PhD degrees from Northwestern Polytechnical University (China) in 1992, 1994 and 1997, respectively. She worked for 3 years at Northwestern Polytechnical University as an Associate Professor and 2 years in BUTONE Inc. (a well-known ERP provider in China) as a system architect. Her research interests include manufacturing/construction process planning and scheduling, enterprise resource planning (ERP), supply chain management, and applications of intelligent agents and Internet/Web-based technologies to these areas. She has co-authored one book and over 50 papers in the related areas.

Weidong Li is a Senior Lecturer at Coventry University, UK. He received his B.Sc., M.Sc. and PhD degrees from Xidian University (China) and National University of Singapore in 1992, 1995 and 2002, respectively. He worked at Singapore Institute of Manufacturing Technology, Bath University (UK) and Cranfield University (UK). His research interests are collaborative product design and manufacturing, computer-aided design, and advanced materials and manufacturing technologies. In the areas, he published two books and more than 60 research papers in international journals and conferences. He involved in various research and industrial projects in product design and manufacturing.