COMPUTER-AIDED DESIGN

Computer-Aided Design 34 (2002) 981–996

www.elsevier.com/locate/cad

Collaborative conceptual design—state of the art and future trends

Lihui Wang*, Weiming Shen, Helen Xie, Joseph Neelamkavil, Ajit Pardasani

Integrated Manufacturing Technologies Institute, National Research Council of Canada, 800 Collip Circle, London, Ontario, Canada N6G 4X8

Received 21 December 2000; revised 21 July 2001; accepted 30 July 2001

Abstract

This paper presents a state of the art review of existing research, projects, and applications in the domain of collaborative conceptual design, based on the Internet and Web technologies. The purpose of the review is to understand the needs for conceptual engineering design, to clarify the current conceptual design practice, to classify the available technologies, and to study the future trend in this area. The emphasis of this paper is to briefly outline the methodologies, architectures, and tools developed for the projects reviewed in this paper. It also uncovers approaches to conflict resolution and team/project management, as they are vital to a successful engineering design in a collaborative environment. More than 80 journal and conference papers and about 20 projects are reviewed based on the primary focus mentioned above. The selected research works are further categorised into several areas based on the application domain, design theory, and the technology used for implementation. The selected research projects and applications are basically for, but not limited to, the collaborative conceptual design. Crown Copyright © 2002 Published by Elsevier Science Ltd. All rights reserved.

Keywords: Conceptual design; Design collaboration; Conflict resolution; Project management; Distributed AI

1. Introduction

Conceptual design is perhaps the most crucial task in an engineering product development cycle. According to Wang et al. [1], conceptual design is a very important task in computer-aided design (CAD), but it is also very difficult to accomplish. Computers have been used extensively in areas such as simulations, analysis, and optimisation, but there are relatively few applications at the conceptual design stage. This is because knowledge of the design requirements and constraints during this early phase of a product's life cycle is usually imprecise and incomplete, making it difficult to utilise computer-based systems or prototypes [2]. A design concept, by its 'soft' nature, is often difficult to capture, visualise or communicate electronically among a multidisciplinary design team, especially when the team is geographically dispersed. Conceptual design issues at stake are highly interdisciplinary, and often involve collaboration from customers, designers, and engineers. Not only is the conceptual design becoming more and more central in meeting the increasingly specialised demands of customers, it can also have a powerful impact on manufacturing productivity and product quality, as many manufacturing processes (e.g. moulding, casting, or machining) are indirectly determined at this stage. As shown in Fig. 1, the impact of design decisions is initially very high, and declines steeply as the design matures [3]. Great opportunity exists at the preliminary design stage. The concept generated at this stage affects the basic shape generation and material selection of the product concerned. In the subsequent phase of detailed design, it becomes extremely difficult, or even impossible to compensate or to correct the shortcomings of a poor design concept formulated at the conceptual design phase [2,4]. Today, as experienced by many industries, not only the resources and equipment, but also the knowledge and expertise are geographically distributed. The demands for shorter time-to-market and designing a product right-the-first-time, however, are increasing to keep companies competitive in the customer-centric market. Experiencing a significant paradigm shift, the conceptual design needs to adopt a more pragmatic and aggressive approach—*through collaboration, supported by artificial intelligence, and fuelled by information technologies*.

The encouraging news is that this goal is achievable. The Internet provides instant access to a wealth of design information, ranging from part library to 3D product model data. The ability to access this information from anywhere makes the Internet an extension of the designer's reference library. The Internet becomes a unique infrastructure for resource integration, data sharing, and design collaboration. The

* Corresponding author. Tel.: +1-519-430-7084; fax: +1-519-430-7064.
E-mail address: lihui.wang@nrc.ca (L. Wang).

PII: S0010-4485(01)00157-9

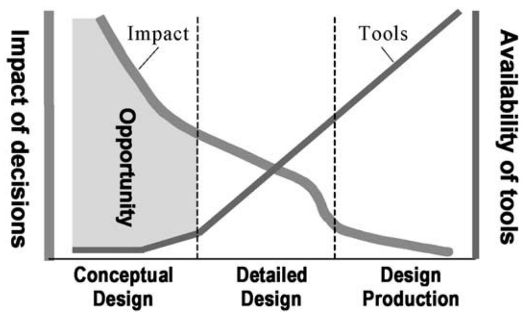

Fig. 1. Opportunity in early design stage.

popularity of the Internet is largely due to the influence of the World Wide Web proposed in 1989, which has made the Internet accessible and available to mass population. Powered by the ever-improving information technologies, such as Java, search engines, email, HTML (Hyper Text Markup Language), XML (eXtensible Markup Language), and RMI (Remote Method Invocation), the Web provides another familiar interface and gives us a common 'look and feel' to information exchange. As the use of the Internet and Web spreads, and because of globalisation, the paradigm of the design activity is changing drastically. Specifically, there is an ever-increasing need for the continuous collaboration among geographically distributed design teams. The collaborative conceptual design process is physically enabled by the Internet and Web technologies, and functionally supported by the technologies in the domain of artificial intelligence, such as agent technology, knowledge management, knowledge-based systems, and so on. These enabling technologies serve as the wheels of the collaborative design vehicle to move forward.

As an extended version of our paper presented at CSCWD 2000 [5], this paper is to report on the needs and requirements for conceptual engineering design, to clarify the current situation of conceptual design practice, to classify the available methodologies, architectures, tools, and technologies, and finally to identify the future trend in this area. More than 80 journal and conference papers and about 20 projects are selected and reviewed based on the primary focus mentioned above. The selected research works are further categorised into several areas based on the application domain, design theory, and the technology used for implementation. The selected research projects and applications are basically for, but not limited to, the collaborative conceptual design.

2. Collaborative conceptual design

2.1. Conceptual design

Conceptual design commences with high-level descriptions of requirements and proceeds with a high level description of a solution [6]. Conceptual design is that phase in the product design cycle, when the basic solution path is laid down through the elaboration of a solution principle [4]. It involves formulation of abstract ideas with approximate concrete representations [7]. The early or conceptual stage of the design process is dominated by the generation of ideas, which are subsequently evaluated against general requirements' criteria. There follows a process whereby additional data are incorporated allowing decisions to be made between competing alternatives as more tangible evidence of function is derived [8].

The conceptual design is crucial, particularly, when designing new and innovative products, or when generating a completely new design for an existing product. It is common knowledge that the majority of the product cost is committed by the end of the conceptual design phase [9,10]. At this phase, information is very fuzzy and incomplete, which makes the design process quite difficult and challenging. It also renders a problem for representing the designed product. Several representations have been proposed for this phase—bond graphs [11], the sketching of abstractions [12] to name a few. How to capture user's intent at this stage is challenging. Qin et al. look at this interesting research problem of capturing user's sketching intentions and automatically generating the corresponding 2D geometric primitives [13]. When it is possible, the 2D objects are projected into 3D models.

Most common techniques used in the conceptual design include problem solving strategies, genetic algorithms, case-based reasoning, and agent technology. Wang et al. applied a DAER (design-analysis-evaluation-redesign) model for conceptual design, combining numerical calculation with symbolic reasoning [1]. Hague et al. [14] acknowledged the fact with the help of machine learning, that product developers must, at an early design stage, take into account all the life-cycle concerns such as manufacturing, reliability, marketing and distribution, to achieve highest return on investments. This requirement was realised, partially, by Co-Designer [15], using agents and machine learning techniques such as rote learning and parameter adjustment learning. Santillan-Gutierrez and Wright [16] use genetic algorithm (GA) for locating groups of promising solutions, aimed at helping designers during the end of the conceptual design stage and dealing with often vague and imprecise information. Most efforts concentrated on a specific type of design problems; they have limitations to extend to commercial applications. The result has been that engineering specification is not the driver for design generation. Rather, designers generate design based on what they are most familiar with. Unfortunately, an optimal design is not likely generated in current design practice.

Usually, the conceptual design phase starts with clarified engineering specifications. It is followed by the establishment of function structures, by the search for appropriate working principles and their combination, and by the evaluation of concept variants against technical and economic criteria. By the end of the conceptual design phase, a decision is made on the solution principle.

There have been a number of methods and techniques for establishing function structure. Function block diagram is used to describe overall function based on the flow of energy, material and signals and to express the relationship between inputs and outputs [4]. Functional Analysis Systematic Technique (FAST) diagrams are used to form a hierarchical tree to represent functional relationships [17,18]. Functional flow charts and functional logic diagrams are other examples [19]. A human-oriented approach focuses on providing a constructive modelling environment to enhance the creativity of human designers. Huang and Mak [20] have developed a web-based FAST diagram editor. Using the editor, a designer is able to create a sub-function or a function at a desired level and add details as necessary. The defined function structures are used for generating concept. Umeda et al. proposed Function-Behaviour-State (FBS) diagram [21] to represent a design object hierarchically. The FBS diagram represents a function as an association of function symbols and behaviours, rather than just either of them. It distinguishes subjective parts of a design object (function symbols and function-behaviour relationships) and objective parts (behaviours and states).

The fundamental issue in creating a representation of function structures is developing a formal method for how individual isolated systems behave and, further, how the connected sum of systems behave. Campbell et al. developed functional representation [22] based on qualitative physics [23], bond graphs [24,25], functional block diagrams [4], and more specifically on works done by Welch and Dixon [26] as well as Schmidt and Cagan [27]. In their representation, ports or points of connectivity with other components describe the isolated systems. Information about how the isolated systems are constrained at their ports, how energy and signals are transformed between ports, and how energy variables within the system related to others is also described. Recently, Al-Hakim et al. proposed the incorporation of reliability with functional perspectives, using graph theory to represent a product and the relationships between its components [28]. With this representation, it is easy to visualise energy flow between components and, thus, trace any loss of functionality. It also allows one easily to take into consideration of various constraints such as cost for further design refinement. For smooth integration with downstream applications of product development, Brunetti and Golob suggested a feature-based representation scheme [29] for capturing product semantics handled in the conceptual design phase. As information carriers to the downstream applications, features are used to model the relationships between requirements, functional descriptions and physical solutions of a product.

Generating concepts involve several tasks: search for working principles; combine working principles; select suitable combinations; and firm up into principle solution variants [4]. Human-oriented approach focuses on assisting designers to achieve these tasks, while computer-oriented approach focuses on generating concepts automatically. The common challenges for these approaches are supporting the combination of working principles for a suitable solution. To overcome the problem of combinatorial explosion, Cartmell [30] suggests a simple but modified version that involves evaluating the individual solutions to identify the optimum ones and then combining only those to obtain an overall solution. However, the compatibility between solutions corresponding to different sub-functions may not be easily identified at this stage. A compromise is possible by a double strategy in which the solutions are initially evaluated and then filtered out to reduce their number, thus pruning the extent of the combinatorial explosion [9]. The filtered solutions are then combined using a morphological matrix to obtain a number of concept variants that can then be advanced to a stage where an evaluation can reveal an optimum solution. In efforts to achieve and generating a solution principle without human interaction, Sieger and Salmi [31] proposed an expert system approach to guide the selection and coupling processes. This approach is good at the scope of its knowledge base. However, it is difficult to extend the scope of knowledge base for generic conceptual design.

During the next stage, the design evaluation and verification are of major importance for product development. Conceptual design activity is not complete unless we can evaluate and verify that the design concepts satisfy the necessary functional requirements. The evaluation of variants involves the following steps: identifying evaluation criteria, weighting the evaluation criteria, assessing values for each variant, and determining overall value. The large number of variants has to be reduced to a single concept, or just a few, to be pursued further. This decision incurs a big responsibility and can only be made after careful evaluation.

Web-based morphological concept assessor [20] assists designers to perform the evaluation. It narrows down the feasible alternative concepts from a fairly large number to a more manageable number for further investigation. The basic mechanism is a morphological evaluation chart. Sieger and Salmi [31] measure design performance through simulation. Simulation can leverage the existing modularity offered by hierarchy. Each node of the system structure is paired with an engine that performs the simulation. Each node can be modelled using as abstract state representation, differential equations, existing tools, or some hybrid combination. The simulation engine utilises discrete event system specification as its formalism. Deng et al. [32] proposed a generic constraint-based approach targeting at how functional design verification can be carried out automatically. This approach is based on a functional design model through the use of constraint propagation and dynamic design verification based on graph. Design verification is achieved by identifying input and output design variables, developing a variable dependency graph, propagating constraints over the graph, and checking the values of the design variable against these constraints.

2.2. Distributed collaborative design

When a product is designed through the collective and joint efforts of many designers, the design process may be called Collaborative Design (it may also be called Co-operative Design, Concurrent Design and Interdisciplinary Design). This would include those functions as disparate as design, manufacturing, assembly, test, quality and purchasing as well as those from suppliers and customers [33]. The objectives of such a collaborative design team might include optimising the mechanical function of the product, minimising the production or assembly costs, or ensuring that the product can easily and economically be serviced and maintained [34]. Since a collaborative design team often works in parallel and independently with different engineering tools distributed in separate locations, even across various time zones around the world, the resulting design process may then be called distributed collaborative design.

Traditional design systems have used a sequential model for design generation, which breaks the design task into subtasks that are serially executed in a predefined pattern. Recently, researchers found that sequential design is brittle and inflexible and often requires numerous iterations, which make the design expensive and time-consuming, and also limit the number of design alternatives that can be tried out. On the other hand, sequential design is usually practised with downstream information flow. Information feedback from low-level manufacturing activities (e.g. process planning or shop production) to the high-level design is usually performed by human interactions. It may cause an inefficient design (and hence inefficient product development), due to the absence of manufacture-ability checks at the design stage. Collaborative design tries to address these problems concurrently by considering constraints and detecting conflicts early in the conceptual design stage. Wang [35] introduced a unique combination of machining-feature, agent technology, and function block to tackle and facilitate concurrent design problems with due considerations of downstream constraints under distributed environment.

To support collaborative design, computer technology must not only augment the capabilities of the individual specialists, but must also enhance the ability of collaborators to interact with each other and with computational resources. However, engineering design has to address several complex characteristics (e.g. diverse and complex forms of information, interdisciplinary collaboration, heterogeneous software tools etc.) and these make interaction difficult to support. Traditional approaches to sharing design information among collaborators and their tools include the development of integrated sets of tools and the establishment of data standards. These approaches are becoming insufficient to support collaborative design practices, because of the highly distributed nature of the design teams, diversity of the engineering tools and the complexity and dynamics of the design environments. A number of emerging technologies including distributed objects, agents and the Internet and Web technologies have been proposed to implement collaborative design systems.

3. Approaches for collaborative conceptual design

3.1. Web-based collaborative design

The ability of the Web for designers to combine multimedia to publish information relevant to the spectrum of the design process, from concept generation and prototyping to product realisation and virtual manufacturing, motivated the adoption of the Web as a design collaboration tool. It is now playing increasingly important roles in developing collaborative product development systems. A collaborative design system developed with the Web as a backbone would primarily provide: (1) access to catalogue and design information on components and sub-assemblies; (2) communication among multidisciplinary design team members in multimedia formats; and (3) authenticated access to design tools, services and documents.

The Web is used by the design team members as a medium to share data, information and knowledge [36,37], and in some cases for product data management and project management by integrating the Web with appropriate technologies [38]. In some other cases, the Web may only be used to monitor the design process and to check the status of the working system [39]. A number of frameworks have been proposed for Web-based collaborative design systems [38,40–42], but most of them are still under proof-of-the-concept prototype development stage.

Recently, a commercial software suite called ipTeam [http://www.NexPrise.com/] becomes available from NexPrise Inc. for collaborative product development. It was derived from the DARPA-sponsored AIMS (Agile Infrastructure for Manufacturing Systems) program. However, ipTeam is primarily for virtual enterprise and supply chain management instead of conceptual design.

Some researchers have developed Web-based tools or systems based on standalone applications. For example, Web-based DFX tools were developed by Huang and co-workers [20,43] and WebCADET by Rodgers and co-workers [44,45].

Most Web-based collaborative design systems are developed using Java, whereas few others are developed by the use of Common Lisp (e.g. WWDL by Zdrahal and Domingue [46]), or Prolog (e.g. WebCADET by Rodgers et al. [44]). HTML and Java Applets are widely used today for developing the client side user interfaces, in addition to ActiveX and VRML. Huang and co-workers used ActiveX

Table 1
Summary of projects/systems on collaborative product design

Name of project/system	R&D group	Key features	Implementation technologies
CPD	Roy et al., KBEL, Syracuse Univ.	Shared product design web pages; shared geometric models in VRML; shared database; multi-server architecture	Web, HTML, VRML, RDBMS, CAD tools
DFX Shell	Huang et al., Univ. of Hong Kong	Web-based deployment of DFX tools using ActiveX	Web, HTML, ActiveX
DOME	Pahng et al., KIST, Korea	Use distributed object technology; multi-server architecture	Web, CORBA, Java, HTML
ipTeam	NexPrise Inc.	Primarily a virtual enterprise integration system with a suite of tools for supporting collaborative product development	Web, E-mails, Multimedia, and much more
KA Framework	Sony System Design Corp.	A collaborative design system architecture from KA point of view; an interesting approach for tacit knowledge capture and sharing	Web, CORBA, OODB, Distributed DB, STEP
Schemebuilder	Bracewell et al., Lancaster Univ.	Scheme based knowledge representation, and sharing via the Web	Web, HTML, CLIPS, Matlab, CAD tools
WebCADET	Rodgers et al., Univ. of Cambridge	Web based deployment of CADET as a decision support system for evaluating conceptual designs	Pro-Web server toolkit, Prolog
Web-based Morphological Chart	Huang et al., Univ. of Hong Kong	Web-based collaborative environment using morphological chart	Web, HTML, ActiveX
WWDL	Zdrahal et al., KMI, Open Univ.	Tadzebao metaphor for guiding designers around ongoing design dialogues; Distributed CBR using agents	Web, HTML, Java, Lisp, LispWeb, CBR tool, Agents

to develop Web-based DFX tools [43,47] and Morphological Chart based collaborative conceptual design system [20]. A number of Web-based design systems use VRML as a neutral representation in their geometric models [48–50]. However, VRML can only be used to display the geometric models but with no editing capability, though it allows designers to put some annotations and comments on the design [50]. Table 1 summarises several web-based collaborative design systems or tools.

Web-based collaborative design systems use the client/server architecture. In order to support collaboration, Web-based design servers need to communicate the structure of the design representation so that users can pose queries about formal design concepts such as rationale and purpose, or the causality between physical and functional elements. To facilitate a viable design environment, Web servers must also engage users in a dialog-like interaction that encompasses a range of activities, such as geometric and semantic product modelling, design representation, user-interaction and design browsing and retrieval. However, the Web technology itself cannot satisfy these requirements. In other words, information access is not the only outstanding problem. In order to collaborate on a distributed project, remote engineers and designers need active help to coordinate their efforts. This coordination involves translation of terminology among disciplines, locating/providing generic analysis services (e.g. finite element analysis), prototyping services, and product management. To the degree that Web servers are not mere repositories of information, but engage users in active dialogue with each other while providing such remote services in order to solve design problems, such servers may be called agents. Agent technology may provide support to enhance the performance of collaborative design systems (see Section 3.2).

3.2. Agent-based collaborative design

Agent technology has been used to develop collaborative design systems even before the Web-based technology. In fact, a number of earlier projects on agent-based collaborative design started when the Web was not yet available, e.g. PACT [37] and DIDE [39,51].

The case for using agents in industry is well founded by Parunak [52]. Noting that the current trends are towards increased product complexity and diversity as well as increased product variety over the time, Parunak has analysed where agent technology can best be used in design and operation activities. His answer has been 'agents are best suited for applications that are modular, decentralised, changeable, ill-structured, and complex'. The reasons often given for adopting an agent approach are linked to their being pro-active object systems and to the simplification of the architecture of the software systems. The real gain obtained from an agent approach, however, often comes from a better description of the real world by focusing on objects rather than functions. When used appropriately, this leads to the desired modularity allowing flexible simulation and to better response and improved software reusability. In addition, the fact that agents can cope with a dynamically changing world by performing dynamic linking, allows them to handle ill-structured or rapidly changing situations in a more economical way [53].

In agent-based collaborative design systems, agents have mostly been used for supporting co-operation among designers, providing a semantic glue between traditional tools, or for allowing better simulations. An earlier review of multi-agent collaborative design systems can be found in Ref. [54]. The book on 'Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing' [53] provides a detailed discussion on issues in developing agent-based collaborative design systems and a review of significant, related projects or systems. Here, we give an overview of the current status in this area, particularly when agent technology is being integrated with other emerging technologies, such as the Web, CSCW (Computer Supported Collaborative Work), Groupware, and Knowledge Engineering, to develop collaborative product development systems.

The use of agents in design has been demonstrated by various research projects. PACT [37] 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 [36] 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 [55] was intended to address the issues of patterns of interaction, communication, and conflict resolution using single function agents. DIDE [39] was developed to study system openness, legacy problem integration, and geographically distributed collaboration. ICM [56] provides a shared graphical modelling environment for collaborative design. Co-Designer [15] was intended to support localised design agents in the generation and management of conceptual design variants. Concept Database [57] provides strategic design support for version control, workflow management and information gathering. A-Design [22] is a new design generation methodology, which combines aspects of multi-objective optimisation, multi-agent systems, and automated design synthesis. It provides designers with a new search strategy for the conceptual stages of engineering design that incorporates agent collaboration with an adaptive selection of designs.

Similar to the Web-based design systems, agent-based systems also provide a collaborative environment for the sharing of design information, data, and knowledge among distributed design team members. In fact, much of the research work done in building agent-based collaborative design systems has also focused on sharing information and data among agents. However, this could easily be achieved using the Web technology.

Unlike the Web-based design systems using the client/server architecture, an agent-based design system is 'a loosely coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities'. Agents in such systems are communicative, collaborative, autonomous (or semi-autonomous), reactive, and intelligent. Different approaches have been proposed in the literature for system organisation. Most systems use a federation approach utilising facilitators, mediators, brokers, and other types of middle agents. Some systems use approaches similar to the blackboard architecture or client/server architecture, e.g. the Design Board approach in SiFAs [55]; the shared graphical modelling approach in ICM [56]; and shared database approach by Varma et al. [57].

The Autonomous Agent approach used in DIDE [39] is different. Although various definitions have been proposed for autonomous agents, an autonomous agent should normally have the following characteristics: (1) it is not controlled or managed by any other software agents or human beings; (2) it can communicate/interact directly with any other agents in the system and also with other external systems; (3) it has knowledge about other agents and its environment; and (4) it has its own goals and an associated set of motivations.

Although agent technology has been recognised as a promising approach for collaborative design systems, those agents that have so far been implemented in various prototype and industrial applications are not actually very 'intelligent'.

Both agent technology and Web technology are very useful in developing collaborative design systems. The attractiveness of the Web for propagating information makes it attractive to the use of agents for accessing and manipulating information automatically. The challenge is how to build the Web environment that will make the designer/agent/server interaction successful through the integration of related emerging technologies.

Table 2 summarises several agent-based collaborative design systems or tools.

4. Tools for collaborative conceptual design

It has become obvious from the literature survey that the objective of the conceptual design phase is to generate multiple design options meeting functional specifications and to select an optimal and feasible solution. The key to the conceptual design activity is the rapid and reliable evaluations of several design options and, this requires participation of people from multidisciplinary backgrounds.

One interesting observation that has emerged from our survey is that there exist many commercial CAD tools to support detail designs. Within the conceptual design phase, there are more tools available to support the later phase of the conceptual design than its earlier phase (Fig. 2); the later phase is the boundary between the conceptual design and the detail design. In the mechanical domain, the component shape is decided at the later phase. Commercial tools that support conceptual design, if exists, belong to this phase. Most research tools (being developed in universities and/or research laboratories) support this phase.

Table 2
Summary of projects/systems on agent-based collaborative product design

Name of project/system	Group	Key features	Implementation Technologies
A-Design	Campbell et al., CMU	Two tier representation; Multi-objective optimisation; evaluation-based iterative algorithm	Agents, Internet, Lisp
Concept Database	Varma et al., UC Berkeley	Provide strategic support for version control, workflow management, and information gathering	Agents, Internet/Web, Relational Database
Co-Designer	Hague et al., CSCE, U of Derby	Localised design agents with high degree of authority for decision-making based on the rich downstream product life cycle information	Agents, Internet/Web
DIDE	Shen et al., UTC	Autonomous agents approach; wrapper for legacy system integration; Web for system monitoring	Agents, Internet/Web, ELM, Lisp, MOSS
ICM	Fruchter et al., Stanford	A shared graphical modelling; an iterative communication approach	Agents, Internet/Web, AutoCAD, ProKappa
Madefast	Madefast Consortium	No formal top-down management structure and no central authority; Web for posting, access, sharing of design info & data; synchronous & asynchronous communication	Internet/Web, Agents, CSCW, HyperMail, and much more
RAPPID	Parunak et al., ERIM	characteristic agents, marketplace approach, set-based design	Agents, Internet
PACT	Cutkosky et al., CDR, Stanford	Federation architecture using facilitators; wrapper for legacy system integration	Agents, Internet, KQML, Lisp
SHARE	Toye et al., Stanford	Federation architecture, Asynchronous communication using e-mails; Web-based tools for information capturing & sharing	Agents, Internet/Web, KQML, NoteMail, ServiceMail, and more
SiFAs	Brown et al., WPI	Single function agents with minimal responsibility	Agents, Internet, CLIPS

As shown in Fig. 2, the earlier phase of the conceptual design itself can be split into two stages—the first stage in which fuzzy customer requirements are mapped to functional specifications and the second stage where a design team tries to develop multiple alternative design solutions from functional specifications. No automation tools (commercial or research) yet exist to address this first stage, whereas, several research tools are being developed to satisfy the needs of the second stage.

The software tools classification described in this section are not confined to a particular domain and cover a wide spectrum, from somewhat domain independent to highly domain specific tools. We classify a tool as being domain independent if the tool can be configured to support conceptual design of multiple domains (e.g. mechanical or mechatronics). While there exist many domain dependent tools to support conceptual design, there are only a few that are actually domain independent.

Domain dependent tools are hard-wired with engineering knowledge and formulas, and they cannot be altered. These tools take some major product attributes as inputs and generate multiple options for conceptual design and then perform various evaluations suggesting optimal solution. For example, RaDEO [58], a conceptual design tool developed by Rockwell Palo Alto Lab, claims to have a highly flexible and responsive design environment to

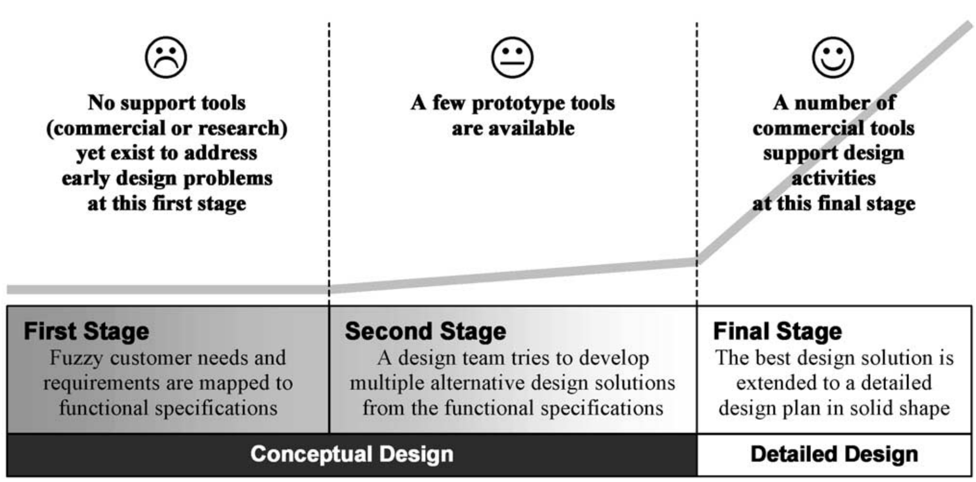

Fig. 2. Availability of design tools.

evaluate multiple design alternatives. Several commercially available engineering analysis tools have been integrated into RaDEO. The methodology embedded in RaDEO is based on a combined use of constraint management technology, symbolic mathematics, and equation solving capability. Other examples include the Ship Design [59] experimental system in which design operations are classified into generating system structures, configuring spatial arrangement, determining system attributes, and selecting system components.

The domain independent tools may be subdivided into three categories: drawing tools, design data repositories, and knowledge-based evaluation tools.

Drawing tools are usually shape forming tools to provide support for drawing in 2D, 3D or in the virtual reality, and may provide simulation support for a particular domain (i.e. motion simulation and animation). Most tools in this category are commercially available and are useful for industrial designers and mechanical engineers. Though the names of some commercial tools are listed in this document based on vendor fact sheets, we have not evaluated these tools and hence, no attempt is made to describe them.

Design data repositories are used during the conceptual design phase to get knowledge and information that is helpful to designers either in reusing previously developed concepts or in supporting the validation of a concept. The responsibility to access, interpret and process the information in a meaningful manner rests upon the designer. Morphological charts, electronic catalogues, data handbooks, design case-bases, and concept databases are some of the most prominent aids in this category. Morphological Chart is one of the formal design tools enabling collaborative product development, and an effective technique for conceptual design of products, processes, and systems. Huang and Mak [20] together have developed a Web-based DFX analysis and design tool that makes extensive use of morphological charts. Concept Database [57], on the other hand, provides 'smart navigation' through a hypermedia database of linked design concepts. Its goal is to provide design teams with easy access to information with regard to life cycle design issues for competing design concepts and access to relevant past designs and other information databases of the enterprise.

Knowledge-based evaluation tools are treated as domain independent as they can be populated with design knowledge of multiple domains. Since these tools are driven by the knowledge, the framework of the tool and the richness of the knowledge base are the limiting factors to the scope of their application. Schemebuilder [60] is an engineering knowledge management tool for storing knowledge about the past solutions to enable effective design reuse. WebCADET [44,61] supports designers by providing them with feedback about alternative solutions by searching through design knowledge. The WebCADET is a domain independent tool that supports knowledge from multiple domains. In addition to the above domain in-/dependent tools, the following design frameworks have the potential to become functional tools in future. ICEDMP [62] is knowledge and web-based evaluation framework that accesses product data and enables the user to perform producibility evaluations (DFM, DFA, etc.). A suite of tools consisting of Pro/Engineer, parts library, feature conversion program, etc. are integrated into this system. CODSAS [9] is a domain neutral prototype tool that makes use of a high-level design language called DPPL in order to support the conceptual design process. ICM [56] is a software prototype that enables sharing and capture of multi-criteria design proposals, design semantics, critique, explanation, and change notifications. The key technical concept of ICM is that a graphical design environment can also serve as the central interface among designers and as the gateway to tools/services in support of interdisciplinary design. CPD [42] consists of several design service modules provided by servers residing on geographically separated machines. Each designer creates his model from his personal design station using conventional CAD package; it is converted to VRML based models by invoking remote translation services via Internet services. NODES [63] is a prototype that models knowledge of design objects and their associated numerical relations. It is an interactive modelling system that enables the designer to build, manipulate and analyse a model of the design artefact by providing feedback on the model. EDM [31] is a window-based design aid to guide individual designers and/or teams, and it provides a mechanism for integration of product development processes. It tracks design participant activities and uses performance metrics to assess the state of the design. It also provides an abstract representation of the product being designed. SHARE [36] is an open, heterogeneous, network-oriented environment for concurrent product development, enabling engineers to participate in a distributed team using their own tools and databases. It helps a design team to achieve a shared understanding of their designs and design process using agent-based computational tools and services. DOME [40] is an open, web and object-based framework for distributed, collaborative, and integrated design. It allows designers to define mathematical models and interconnect them to form larger system models. These models are encapsulated in the system modules and distributed over the network, collectively forming a distributed model for a collaborative, multidisciplinary, and concurrent design evaluation. Co-Designer [15] is a framework that supports localised design agents in the generation and management of conceptual engineering design variants within a distributed concurrent engineering design environment. It empowers individual local design agents to the highest degree such that through collaboration those 'experts' can make localised decisions. WWDL [46] is a prototype that enables designers to participate in synchronous and asynchronous design dialogues over the Web. It has an ontology editor that allows designers to collaboratively construct an explicit shared conceptualisation, which

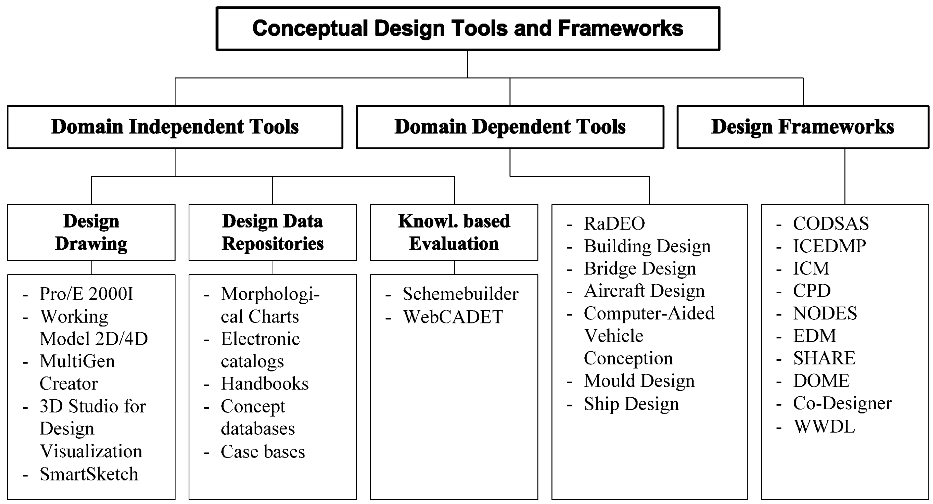

Fig. 3. Conceptual design tools and frameworks.

facilitates design re-use. Fig. 3 classifies the available tools and frameworks into several categories for quick reference.

5. Managing collaborative conceptual design

5.1. Conflict resolution in collaborative design

As mentioned in the previous sections, a collaborative design is a collection of the co-operated efforts undertaken by a team of designers and other specialists. Each team member works on different parts of the design, and works from different perspectives and towards satisfying different functional criteria. Each member must assess the impacts of his/her decisions on others, and notify the affected parties promptly. However, complex activities such as conflict resolution are still facilitated mostly through negotiations and face-to-face meetings. In most cases, the creative negotiation activities such as generating new solutions, preventing and detecting conflicts are still left to the human experts. To provide supportive environments for collaboration, design systems must provide participants in the collaboration with facilities for information sharing, task coordination, and conflict resolution. Particularly, conflicts and disputes arise regularly in decision-making process during collaborative design, such as goal selection, proposal exchanging, task co-ordination, role-playing, and allocation of the limited resources, etc. Unmanaged or improperly managed conflicts affect not only the productivity of the whole design team but also hamper achieving the design goal. Conflict resolution consists of at least five steps: conflict detection, conflict identification, negotiation team formation, solution generation, and solution evaluation.

As one type of supportive design environment, Appelt and Busbach developed a Web-based BSCW (Basic Support for Cooperative Work) system [64]. The basic idea of the BSCW is a shared workspace located on some computers for coordinating and organising collaborative design works. The shared workspaces do not use predefined coordination and cooperation models, but follow the idea that CSCW (Computer Supported Cooperative Work) applications should *inform* rather than *constrain* [65]. The shared workspaces can be regarded as a medium for communication, conflict resolution and management. The potential conflicts during information sharing and data exchange can therefore be largely avoided by using the shared workspaces.

Major conflicts in design process often stem from specification conflicts among the constituent design tasks. Coupled tasks represent conflicts in the flow of information of a design process. Resolving the specification conflicts early on in the development process is critical for successful collaborative design. This type of conflicts is closely related to task coordination, and occurs when the relationships and dependencies between some design tasks are violated (i.e. when tasks that are inherently sequential are performed concurrently). According to Yassine and Falkenburg [66], the specification conflicts can be resolved by de-coupling the conflicting tasks, if the specification of one task can be changed to absorb a certain percentage of any possible variation in the output of the other task. The result of an efficient task co-ordination or specification management can increase both the concurrency of a design and the productivity of its design team.

Wallis et al. introduced a multi-agent framework for distributed collaborative design [67], which uses a *deontic logic* based formalism to facilitate distributed conflict management. 'Rather than being a rare and avoidable occurrence, conflicts play a central role in design cooperation' [68]. Due to the diverse knowledge and viewpoints of the agents in the system, conflict may arise at any time, often as a result of differing priorities. Where agents are in conflict, direct negotiation takes place between the conflicting agents. Conflict resolution can be done based initially on priorities assigned to each agent, and where this fails

Table 3
Summary of several systems and tools on conflict resolution

Name of system/tool	Research group	Key features	Application domain
BSCW	Appelt & Busbach, GMD, Germany	Using *shared workspaces* as platform for collaboration; Extension of standard Web server; Acting as information repository with version/access-right control.	Environment for conflict resolution
Multi-agent Framework	Wallis *et al*., GCU, UK	Using deontic logic based formalism for conflict management; Conflict resolution based on priorities; Human has control of conflicts with equal priorities.	Conflict management between agents
CONCENSUS	Cooper & Taleb-Bendiab, MMU, UK	Behaviour and autonomy are controlled by control profiles; *Decision tree* keeps all decisions; Implemented as a multi-agent system	Conflict negotiation between human and/or software agents
NegotiationLens	Adelson, Rutgers U., USA	Providing method and process for conflict negotiation; Benefit to all parties; Based on individual interests, resources, and goals. Mutually acceptable solutions.	Conflict resolution & management between human designers

because of equal priorities, the final decision is passed back to the human users. It is desirable to provide the users with a detailed context of the conflict, in order to prevent a cascading generation of additional conflicts.

CONCENSUS (CONCurrent Engineering Negotiation SUpport System) is a prototype developed by Cooper and Taleb-Bendiab [69] for the support of multi-party negotiation. It is a multi-agent system for conflict negotiation between human and/or software agents, and enables design team members to participate in an iterative process of exchanging proposals, rejections, supporting arguments and compromise until a consensus is reached. In many situations, the best (and in some cases, perhaps the only) solution to a conflict may itself create additional conflicts, which also have to be resolved before a final and complete solution can be agreed upon. Without adequate control, a simple scenario of one conflict between two agents can quickly develop into a situation requiring the involvement of many agents to resolve complex multiple conflicts. CONCENSUS uses control profiles, containing control gates and preferences, to specify the desired behaviour and level of automation for a specific negotiation scenario.

NegotiationLens [70] is a tool to facilitate conflict negotiation intended to produce gain for all parties (human designers). It helps attain good negotiation results by moving the diverse groups typically involved in a large design project, away from an adversarial stance and back into a collaborative relationship. This is accomplished by creating a context in which the groups present well-reasoned and fair considerations. The fairness of the views enables each party to be heard, and thus to feel respected. NegotiationLens allows joint construction of solutions that are more beneficial than the unilateral solutions each party initially brought to the table and supports a commitment to implementation. The primary focus of conflict resolution in NegotiationLens is kept on individual interests, resources, and negotiation goal.

Table 3 summarises and compares all research approaches reviewed in this subsection.

5.2. Team and project management in collaborative design

Collaborative engineering processes involve many people and teams within an organisation who must coordinate their activities based on information flow, available resources, and various other constraints. The challenge is how to handle the tremendous complexity involved in planning and executing large numbers of interconnected and dynamic development tasks. One answer is the development of a process model, which will help people capture, represent and evaluate a wide variety of events. The implementation of this model can substantially reduce the process cycle-time. There are several established representations for process modelling. Park and Cutkosky [71] have given a review of these models. A directed graph (or digraph) is the simplest way to represent processes. It is quite effective when the number of nodes and edges is small.

One of the familiar representation schemes is the Project Evaluation and Review Technique (PERT) developed in 1950s for estimating the project completion time and for identifying the critical path of tasks within the project. It is a standard representation scheme available in most project management applications. However, it has limitations in representation of iteration loops and simultaneous interactions. Another method, the Structured Analysis and Design Technique (SADT) was originally developed by Ross [72], and later adopted in the IDEF0 representation and tools. These modelling techniques have been used to document large, complex processes. A more recent technique, the Design Structure Matrix (DSM) is capable of representing complicated dependencies among numerous project entities and provides an elegant way to detect and manipulate iteration loops [73,74]. Still another methodology, the Petri Net and its derivatives, applied most commonly for computer systems and manufacturing processes, provide methods for assessing process feasibility and performance [75].

A major issue in process modelling is the existence of information cycles. Cycles represent conflicts in the flow of information within the design process. That is, there is no obvious order in which the design tasks can be performed. Tearing down the process is often required to determine an execution order for the design tasks. Yassine et al. proposed an enhanced structural model based on the DSM to break the information cycle [76]. The sensitivity of a task to one of its predecessors expresses how sensitive a task is to the changes in the output of its predecessor. The variability is the possible deviation of an estimate, at the time of assessment, from the actual value. The two dimensional variable format of the DSM provides a better reflection of the design structure and allows one to develop an improved tearing process.

The important aspects of process management are its implementation issues. The DSM model technique has proven to be an effective tool for planning and managing product development projects. However, currently the DSM related information is obtained through extensive interviews and cross-functional meetings. The data acquisition approach is logistically difficult to carry out and is therefore time-consuming. The dynamic nature of product development leads to frequent changes to the process. The accommodation of such changes to maintain model accuracy is also quite a laborious effort. To address these limitations of DSM in large projects, Sabbaghian et al. proposed a distributed and asynchronous modelling approach [77]. They implemented this approach through a Web-based prototype system, with efforts on efficiently engaging a large number of participants in the modelling activities and proving a distributed and user-friendly access to very large models. Integrating information received at different times from a large group of dispersed individuals is a major challenge in this approach.

To address process modelling issues in larger-scale complex engineering projects, Park and Cutkosky proposed Design Roadmap (DR) [71], yet another process modelling tool. It primarily focuses on integration and co-ordination of multiple, interacting processes. The underlying data structure of the DR is a bipartite graph consisting of two primary node types: tasks and features. Arcs, which represent three different types of relationships, connect the nodes. These are precedence, abstraction, and constraint links. The DR framework is applicable to a wide range of problems. It has been used to develop and document engineering processes, to construct functional diagrams of integrated systems, to create process templates, and to manage collaborative projects. The ultimate application of the DR framework is in the development of process templates and process components that encapsulate expertise.

6. Future opportunities and challenges

A framework for enabling collaborative design should allow a designer to access his/her favourite tools from hypermedia workspace. However, today's Web technology supports only limited coordination through provision of shared information space. To fully participate in a collaborative design, designers need to be able to, not only exchange data but also to negotiate their design intent governing the design generation. This negotiation requires a task-oriented view of the design project, rather than just the data-oriented view provided by the Web. Fortunately, the tools capable of supporting a task-oriented view can be implemented on top of the Web infrastructure using AI techniques and the latest information technologies, such as Java servlets, JSP (JavaServer pages), EJB (Enterprise JavaBeans), intelligent search engines, XML (extensible mark-up language), VRML (virtual reality modelling language), Java 3D, and RMI (remote method invocation). In addition, client side scripting, applets, and ActiveX controls often make significant contributions to the execution of design rules. Some efforts have been devoted already to address these problems. For example, Jagannathan et al. [78], Pahng et al. [40], and Wallis et al. [67] use CORBA and Java to facilitate collaborative design activities. DeRoure et al. [79] use proxies and an open and extensible message protocol to establish an open architecture based on the existing Web infrastructure for communication. Maher et al. [80] studied how designers communicate design semantics and whether more or less design semantics are captured in a collaborative session. Li and Hopper [81] proposed a framework for synchronous and asynchronous collaboration. It is clear that challenges in these areas will remain as a research opportunity.

Currently, design concept generation, evaluation, and reuse are problematic. Advanced AI-based design tools will support increased efficiency in multidisciplinary design environment. AI plays a pivotal role in meeting major challenges posed by the collaborative conceptual design. It can learn new concepts; reason and draw useful conclusions about a design problem; understand natural languages of designers; and perceive and comprehend a visual scene. By enhancing its intelligent behaviours, AI can make a design system integrated, goal oriented, expressive, cooperative, and customisable [82]. AI-based user interfaces can collaborate with designers to meet their specific needs, such as capturing design intent and morphological inputs. These interfaces will handle multiple modalities including natural language, gestures, graphics, and animation and will be able to employ whichever modality best suits a particular user request. The interfaces may be implemented using agent technology for distributed collaboration. Such interfaces will operate as intelligent agents, allowing designers to state what they want accomplished and automatically determining the actions required to satisfy these needs and when to perform them.

To be useful, a collaborative conceptual design system must have intelligent indexing functionality and provide convenient access to all kinds of information. This is simply because the system may contain information on a multitude of diverse subjects, and data represented in a wide variety of forms, including various natural languages, digital and video images, audio, geometric model, and database relations. Extracting and identifying multimedia content and indexing it with symbolic descriptions is a challenge that enables fast and flexible retrieval of answers to designers' queries. Knowledge capture, representation, and reasoning methods will enable data translation services to convert design information from one format to another. Agent architectures may provide the basis for constructing specialised software agents to act as subject-specific brokers, tracking the creation of new design, noting updates to existing repositories, and answering queries in their targeted area.

Among the variety of design methods, such as case-based design, rule-based design, model-based design, and feature-based design, etc., biological design concept throws new light on the design domain. First proposed by Ueda for dynamic problems in manufacturing, the biological approach [83] aimed at dealing with dynamic changes in external and internal environments in a product's life cycle from planning to disposal, based on biologically-inspired ideas, such as self-recognition, self-growth, self-organisation, self-recovery, adaptation and evolution [84]. Similar to biological organisms being capable of adapting themselves to environmental changes and sustaining their own life by showing such functions, product-centric biological design can generate a design plan in the same manner. The functions of organisms are displayed by expressing two types of biological information, which are the genetic information evolving through generation (*DNA-type*) and individually achieved information during one's lifetime (*BN-type*). For a new product design, user requirements, design intent, and goal correspond to the DNA-type information, while the BN-type of data can be achieved gradually during the design process via interactions with the environment (resources and human designers). The biologically inspired feature of the design concept fits well in the dynamically changing and distributed design environment.

In summary, the following areas have been identified as future research opportunities and challenges:

1. *System Architecture for Web-Based Collaborative Conceptual Design*. The architecture of a collaborative conceptual design system needs to be carefully formulated to make full use of the Web features to capture fuzzy information and facilitate early concept generation. From client's perspective, these features include client-side scripting, applets, ActiveX controls, and plug-ins. On the server side, a web application is featured by the extensive use of CGI (common gateway interface), Java Servlets, ASP (active server pages), JSP, and EJB. Special attention should be paid to deal with the limitations of today's Web server, and to make effective use of resources at both the client and server sides.
2. *Collaborative Conceptual Design Modelling and Data Sharing*. Models help dispersed designers understand the nature of an early design concept. The incomplete initial design concept is generally difficult to communicate unless it is modelled in a mutually understandable way. The decisions on how to model and what to model have an enormous effect on the understanding of a problem and the shape of its solution. As a web application, a conceptual design system together with the design model is shared among a collaborative design team over the Internet. Efficient design data sharing is largely determined by effective design concept modelling. For example, the latest Java 3D API allows a concept model to be downloaded and rendered on a client machine while remaining alive by controlling its behaviour through message passing. SGDL (solid geometry design logic) language, on the other hand, allows a user to describe a complex conceptual shape in a mathematical way, and manipulate the shape through a set of control points [http://www.sgdl.com]. When deciding how to model a design concept, determining the appropriate levels of abstraction and detail are critical to be beneficial to the users of the model. Attention should be paid to the selection of a modelling language.
3. *Product-Centric Design Methodology*. Design methodology is the kernel of a design system. It drives and directs the design processes and activities from concept generation to geometry creation. A product-centric design methodology such as the biological design concept is considered a suitable approach for the distributed collaborative conceptual design, in terms of information sharing and conflict reduction. Featured by its self-learning ability, product-centric design fits well in a dynamically changing environment. Once a design seed (initial needs, user requirements, and design intent, etc.) is sowed, the conceptual design system will fertilise it towards a product concept based on its own intention. Designer's interactions, on the other hand, will enhance the characteristics and behaviours of the product.
4. *Conceptual Design Selection*. In generating a design concept, challenges have been identified in combinatorial exploration of working principles to achieve an overall function. The research opportunities will be in analysis of compatibility of working principles in terms of technical feasibility, economical effectiveness, and environmental factors, etc. The compatibility model is then combined with a morphological matrix of atomic functions and working principles to provide a basis for searching and optimising a design concept.
5. *Knowledge Management in Collaborative Environments*. Valuable knowledge is generally stored in designer's brain as 'human intelligence' in the form of tacit and experiential knowledge. The knowledge, once identified and captured, needs to be managed properly to be able to represent, index, store, retrieve, modify, validate, and learn from them so that they can be applied and re-used later in new concept generation. Work on this research

seeks to discover convenient, efficient, and appropriate methods, techniques, and mechanisms for capturing knowledge from various resources, for representing knowledge in computer readable and retrievable format, for sharing among collaborative team members, for learning from past design experience, and for reuse in new concept generation. Challenges in this area include knowledge discovery, self-learning, natural language processing, dynamic knowledge management, design intent capturing in multimedia formats, and effective knowledge reuse.

6. *Intelligent Web-based Users Interface*. With increasing number and kinds of services and resources available electronically, such as on-line part catalogue, multimedia libraries, and shared design tools, designers will be overwhelmed by information explosion, unless the user access becomes simple and effective. Designers need also to interact with a design system and negotiate with peers via the interface during collaborative design. The interface becomes a user's portal for concept generation, data sharing, and task coordination. The challenge here is to make the intelligent interfaces available to all resources so that the designers will have more flexibility to do efficient and effective design. The interfaces should satisfy the following interrelated criteria: integrated, expressive, goal oriented, co-operative, ease of use, and customisable.
7. *Distributed Design Project Management*. Creating, maintaining, and closing a design project that is being executed in a distributed environment poses many difficult problems apart from those issues of design, evaluation, and optimisation. There must be some ways of managing all the resources involved, including people, organisations, software tools, and equipment. Managing distributed collaborative design projects requires suitable mechanisms for coordinating independent design activities and project plans. Relevant issues are conflict management, cost management, task management, and activity scheduling. An effective and systematic approach remains to be developed to meet the challenge in a distributed environment.
8. *Implementation of Virtual Design Studio*. As the name suggests, Virtual Design Studio will be an integrated collaborative design environment, allowing a distributed design team working together in harmony, as if they are in the same office. With the product-centric design methodology as a kernel, this prototype system will likely be implemented with intelligent interfaces, powerful search engine and inference engine, advanced knowledge management tools, and other modules facilitating resources integration, design collaboration, information sharing, and communication. Required tools and technologies need to be selected after careful study and with caution.

7. Conclusions

This paper presents detailed literature reviews of the existing research projects and applications dealing with the collaborative conceptual design. From in-depth study, eight research areas have been identified as future opportunities and challenges.

Knowledge management in design has been cited as an important area of research for the future. The challenge here is to capture and re-use the existing designs, help them to adapt to new requirements, and maintain the design knowledge as corporate asset. Both Web-based and agent-based approaches are identified as dominant and enabling technologies for the implementation of distributed collaborative design systems, while conflict resolution and distributed team/project management are the other important issues that influences the achievement of a successful engineering design. Numerous tools and technologies useful for the conceptual design are selected and classified. The selected research projects and applications are predominantly for, but not limited to, the collaborative conceptual design.

One of the most difficult tasks in collaborative design is agreeing on the ontological commitments that enable knowledge-level communication among the distributed design modules. Developing a shared ontology is difficult because it must bridge the differences in abstractions and views. Another difficulty is the integration of the various available design tools. If the tool data and models are encapsulated, rather than using a standardised and unified approach, each tool will be free to use the most appropriate internal representations and models for its intended task.

Finally, a key issue in concurrent design from a designer's perspective is how to bridge the multitude of models required to support a complex design 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.

References

[1] Wang Q, Rao M, Zhou J. Intelligent systems for conceptual design of mechanical products. In: Mital A, Anand S, editors. Handbook of expert systems applications in manufacturing: structures and rules, New York: Chapman & Hall, 1994.

[2] Hsu W, Liu B. Conceptual design: issues and challenges. Computer-Aided Design 2000;32:849–50.

[3] IMTR, Integrated Manufacturing Technology Roadmapping Initiative, http://imtr.ornl.gov/, May 1999.

[4] Pahl G, Beitz W. Engineering design—a systematic approach. Springer-Verlag, London, 1996.

[5] Wang L, Shen W, Xie H, Neelamkavil J, Pardasani A. Collaborative conceptual design: a state-of-the-art survey. In Proceedings of Computer Supported Cooperative Work in Design, Hong Kong, Nov. 29–Dec. 1, 2000. p. 204–9.

[6] McNeil T, Gero JS, Warren J. Understanding conceptual electronic design using protocol analysis. Research in Engineering Design 1998;10:129–40.

[7] Takala T. Design transactions and retrospective planning tools for conceptual design. In: Akman V, ten Hagen PJW, Veerkamp PJ,

editors. Intelligent CAD systems II, Springer Verlag, 1989. p. 262–72.

[8] Baker K, Ball L, Culverhouse P, Dennis I, Evans J, Jajodzinski P, Pearce P, Scoththern D, Venner G. A psychologically based intelligent design aid. In: ten Hagen PJW, Veerkamp PJ, editors. Intelligent CAD systems III, Springer Verlag, Heidelberg, 1991. p. 21–39.

[9] Al-Salka MA, Cartmell MP, Hardy SJ. A framework for a generalised computer-based support environment for conceptual engineering design. Journal of Engineering Design 1998;9(1):57–88.

[10] Sieger DB, Salmi RE. Conceptual design tool for the systematic design method. Concurrent Product Design and Environmentally Conscious Manufacturing 1997;DE-Vol. 94/MED-Vol. 5:145–52.

[11] Finger S, Rinderle JR. A transformational approach to mechanical design using a Bond Graph grammar, ASME Design Theory and Methodology, Montreal, Canada, 1989.

[12] Mukherjee A, Liu CR. Representation of function–form relationship for the conceptual design of stamped metal parts. Research in Engineering Design 1995(7):253–69.

[13] Qin SF, Wright DK, Jordanov IN. From on-line sketching to 2D and 3D geometry: a system based on fuzzy knowledge. Computer-Aided Design 2000;32:851–66.

[14] Hague MJ, Taleb-Bendiab A, Brandish MJ. An adaptive machine learning system for computer supported conceptual engineering design, AI System Support for Conceptual Design. In: Sharpe J, editor. Proceedings of the 1995 Lancaster International Workshop on Engineering Design, London, UK: Springer, 1996.

[15] Hague MJ, Taleb-Bendiab A. Tool for the management of concurrent conceptual engineering design. Concurrent Engineering: Research and Applications 1998;6(2):111–29.

[16] Santillan-Gutierrez SD, Wright IC. Solution clustering with genetic algorithms and DFA: an experimental approach, AI system support for conceptual design. In: Sharpe J, editor. Proceedings of the 1995 Lancaster International Workshop on Engineering Design, London, UK: Springer, 1996.

[17] Miles LD. Techniques of value analysis. 2nd ed. New York: McGraw Hill, 1965.

[18] Raven AD. Profit improvement by value analysis, value engineering and purchase price analysis. London: Cassell, 1971.

[19] Sturges RH, O'Shaughnessy K, Reed RG. A systematic approach to conceptual design. Concurrent Engineering: Research and Applications 1993;1:93–105.

[20] Huang GQ, Mak KL. Web-based morphological charts for concept design in collaborative product development. Journal of Intelligent Manufacturing 1999(10):267–78.

[21] Umeda Y, Ishii M, Yoshioka M, Shimomura Y, Tomiyama T. Supporting conceptual design based on the function–behaviour-state modeller. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 1996;10(4):275–88.

[22] Campbell MI, Cagan J, Kotovsky K. A-Design: an agent-based approach to conceptual design in a dynamic environment. Research in Engineering Design 1999(11):172–92.

[23] Forbus KD. In: Shrobe H, editor. Qualitative physics: past, present, and future, exploring artificial intelligence, Morgan Kaufmann, San Francisco, 1988. p. 239–96.

[24] Paynter HM. Analysis and design of engineering systems. Cambridge, MA: MIT Press, 1961.

[25] Ulrich K, Seering W. Synthesis of schematic descriptions in mechanical design. Research in Engineering Design 1989;1:3–18.

[26] Welch RV, Dixon J. Guiding conceptual design through behaviour reasoning. Research in Engineering Design 1994;6:169–88.

[27] Schmidt LC, Cagan J. Recursive annealing: a computational model for machine design. Research in Engineering Design 1995;7:102–25.

[28] Al-Hakim L, Kusiak A, Mathew J. A graph-theoretic approach to conceptual design with functional perspectives. Computer-Aided Design 2000;32:867–75.

[29] Brunetti G, Golob B. A feature-based approach towards an integrated product model including conceptual design information. Computer-Aided Design 2000;32:877–87.

[30] Cartmell MP. Lecture notes on systematic design, department of engineering, University of Aberdeen, UK, 1987.

[31] Sieger DB, Salmi RE. Knowledge representation tool for conceptual development of product designs. Proceedings of the IEEE International Conference on Robotics and Automation 1997:1936–41.

[32] Deng YM, Britton GA, Tor SB. Constraint-based functional design verification for conceptual design. Computer-Aided Design 2000;32:889–99.

[33] Sprow E. Chrysler's concurrent engineering challenge. Manufacturing Engineering 1992;108(4):35–42.

[34] Hartley J. Concurrent engineering. Cambridge, MA: Productivity Press, 1992.

[35] Wang L. An approach to collaborative design and intelligent manufacturing. In Proceedings of International Joint Conference of SCI'99 (Systemics, Cybernetics and Informatics) and ISAS'99 (Information Systems Analysis and Synthesis), vol.7 (Industrial Systems), 1999. p. 431–7.

[36] Toye G, Cutkosky M, Leifer L, Tenenbaum J. Glicksman J. SHARE: A methodology and environment for collaborative product development. In Proceeding of Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Society Press, 1993. p. 33–47.

[37] Cutkosky MR, Engelmore RS, Fikes RE, Genesereth MR, Gruber TR, Mark WS, Tenenbaum JM, Weber JC. PACT: An experiment in integrating concurrent engineering systems. IEEE Computer 1993;26(1):28–37.

[38] Numata J. Knowledge amplification: an information system for engineering management, Sony's Innovation in Management Series, vol.17. Japan: Sony Corporation, 1996.

[39] Shen W, Barthes JP. An experimental environment for exchanging engineering design knowledge by cognitive agents. In: Mantyla M, Finger S, Tomiyama T, editors. Knowledge intensive CAD-2, Chapman & Hall, London, 1996. p. 19–38.

[40] Pahng GDF, Bae S, Wallace D. A web-based collaborative design modelling environment, Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'98), 1998. p. 161–7.

[41] Huang GQ, Mak KL. Web-based collaborative conceptual design. Journal of Engineering Design 1999;10(2):183–94.

[42] Roy U, Kodkani SS. Product modelling within the framework of the World Wide Web. IIE Transactions 1999;31(7):667–77.

[43] Huang GQ, Lee SW, Mak KL. Web-based product and process data modelling in concurrent 'design for X'. Robotics and Computer-Integrated manufacturing 1999;15(1):53–63.

[44] Rodgers PA, Huxor AP, Caldwell NHM. Design support using distributed Web-based AI tools. Research in Engineering Design 1999(11):31–44.

[45] Caldwell NHM, Rodgers PA. WebCADET: facilitating distributed design support. London, UK: IEE Colloquium on Web-based Knowledge Servers, 1998.

[46] Zdrahal Z, Domingue J. The World Wide Design Lab: an environment for distributed collaborative design. In Proceedings of International Conference on Engineering Design, Tampere, Aug. 19–21, 1997.

[47] Huang GQ, Mak KL. Design for manufacture and assembly on the Internet. Computer in Industry 1999;38(1):17–30.

[48] Allen R.H, Nidamarthi S, Regalla SP, Sriram RD. Enhancing collaboration using an Internet integrated workbench. In Proceedings of DETC99—ASME Design Engineering Technical Conference, Las Vegas, NV, 1999.

[49] Roy U, Bharadwaj B, Kodkani SS, Cargian M. Product development in a collaborative design environment. Concurrent Engineering: Research and Applications 1997;5(4):347–65.

[50] Klein M. Capturing geometry rationale for collaborative design. In Proceedings of the IEEE Workshops on Enabling Technologies

Infrastructure for Collaborative Enterprises (WET ICE'97), 1997. p. 24–8.
[51] Shen W, Barthes JP. A distributed architecture for design environment using asynchronous cognitive agents. In Proceedings of Second Singapore International Conference on Intelligent Systems, Singapore, 1994. p. B327–34.
[52] Parunak HVD. What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC, Cooperative information agents II: learning, mobility and electronic commerce for information discovery on the Internet. In: Klusch M, Weiss G, editors. Second International Workshop, CIA'98, Paris, France: Springer, 1998. p. 1–18.
[53] Shen W, Norrie DH. Multi-agent systems for concurrent intelligent design and manufacturing. London, UK: Taylor & Francis, 2000.
[54] Lander SE. Issues in multiagent design systems. IEEE Expert 1997;12(2):18–26.
[55] Brown DC, Dunskus B, Grecu DL, Berker I. SINE: support for single function agents. In Proceedings of Applications of AI in Engineering, Udine, Italy, 1995.
[56] Fruchter R, Reiner KA, Toye G, Leifer LJ. Collaborative mechatronic system design. Concurrent Engineering: Research and Applications 1996;4(4):401–12.
[57] Varma A, Dong A, Chidambaram B, Agogino A, Wood W. Web-based tool for engineering design, Working Paper, 1999.
[58] Reddy SY, Fertig KW, Smith DE. Constraint management methodology for conceptual design trade-off studies. In Proceedings of ASME Design Engineering Technical Conferences and Computers in Engineering Conference, Irvine California, 1996.
[59] Fujita K, Akagi S. Agent-based distributed design system architecture for basic ship design. Concurrent Engineering: Research and Applications 1999;7(2):83–93.
[60] Bracewell RH, Sharpe JEE. Functional descriptions used in computer support for qualitative scheme generation—'Schemebuilder'. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 1996;10(4):333–45.
[61] Rodgers JL, Salas AO. Toward a more flexible Web-based framework for multidisciplinary design. Advances in Engineering Software 1999;30(7):439–44.
[62] Chen KH, Chen SJ, Lin L, Changchien SW. An integrated graphical user interface (GUI) for concurrent engineering design of mechanical parts. Computer Integrated Manufacturing System ;11(1– 1998;2:91–112.
[63] Duffy AHB, Persidis A, MacCallum KJ. NODES: a numerical and object based modelling system for conceptual engineering design. International Journal on Knowledge based systems 1996;9:183–206.
[64] Appelt W, Busbach U. The BSCW system: a WWW-based application to support cooperation of distributed groups. In Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'96), 1996. p. 304–9.
[65] Beck E, Bellotti V. Informed opportunism as strategy: Supporting coordination in collaborative writing. In Proceedings of the Third European Conference on Computer Supported Cooperative Work, 1993. p. 233–48.
[66] Yassine A, Falkenburg DR. A framework for design process specifications management. Journal of Engineering Design 1999; 10(3):223–34.
[67] Wallis A, Haag Z, Foley R. 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. p. 282–7.
[68] Klein M. Detecting and resolving conflicts among co-operating human and machine based design agents. The International Journal for Artificial Intelligence in Engineering 1992;7(2):93–104.
[69] Cooper S, Taleb-Bendiab A. CONCENSUS: multi-party negotiation support for conflict resolution in concurrent engineering design. Journal of Intelligent Manufacturing 1998(9):155–9.
[70] Adelson B. Developing strategic alliances: A framework for collaborative negotiation in design. Research in Engineering Design 1999(11):133–44.
[71] Park H, Cutkosky MR. Framework for modelling dependencies in collaborative engineering processes. Research in Engineering Design 1999(11):84–102.
[72] Ross DT. Structured Analysis (SA): a language for communicating ideas. IEEE Trans. Software Engineering 1977;3(1):16–34.
[73] Steward DV. The Design Structure System: a method for managing the design of complex Systems. IEEE Trans. on Engineering Management 1981(28):71–4.
[74] Gebala DA, Eppinger SD. Methods for analysing design procedures. In: Stauffers LA, editor. Design theory and methodology, Miami: ASME, 1991. p. 227–32.
[75] Peterson JL. Petri net theory and the modelling of systems. Englewood Cliffs, NJ: Prentice-Hall, 1981.
[76] Yassine A, Falkenburg D, Chelst K. Engineering design management: an information structure approach. International Journal of Production Research 1999;37(13):2957–75.
[77] Sabbaghian N, Eppinger S, Murman E. Product development process capture and display using Web-based technologies. In Proceedings of the IEEE Conference on Systems, Man and Cybernetics, 1998. p. 2664–9.
[78] Jagannathan V, Almasi G, Suvaiala A. 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. p. 292–7.
[79] DeRoure D, Hall W, Reich S, Pikrakis A, Hill G, Stairmand M. An open architecture for supporting collaboration on the Web In Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises, 1998. p. 90–5.
[80] Maher ML, Cicognani A, Simoff S. An experimental study of computer mediated collaborative design. In Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'96), 1996. p. 268–73.
[81] Li SF, Hopper A. A framework to integrate synchronous and asynchronous collaboration. In Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'98), 1998. p. 96–101.
[82] Weld DS, Marks J, Bobrow DG. The role of intelligent systems in the national information infrastructure. AI Magazine 1995;Fall:45–64.
[83] Ueda K. A concept for bionic manufacturing systems based on DNA-type information. In Proceedings of IFIP 8th International PROLAMAT Conference, 1992. p. 853–63.
[84] Ueda K, Vaario J, Ohkura K. Modelling of biological manufacturing systems for dynamic reconfiguration. Annals of the CIRP 1997;46(1):343–6.

Dr Lihui Wang is a research scientist at Integrated Manufacturing Technologies Institute at National Research Council of Canada. He received PhD and MEng from Kobe University, Japan in 1993 and 1990, respectively, and BSc from Beijing University of Printing, Chian in 1982. He has worked for 2 years at Kobe University and 2 years at TUT, a national university in Japan, as an Assistant Professor before joining NRC. Dr Wang has more than 40 papers published in journals and conference proceedings. Challenging at future potentials in design and manufacturing, Dr. Wang works closely with academia and industries with strong interests in CAD/CAM/CAE, product design, process planning, artificial intelligence, Java and web-based systems. He is currently a registered Professional Engineer, a Senior Member of SME, and an Adjunct Professor at University of Western Ontario.

Dr Weiming Shen is a research scientist at Integrated Manufacturing Technologies Institute of National Research Council Canada. His current research interests include applications of Intelligent Agents, Internet and Web technologies to Collaborative Design and Distributed Intelligent Manufacturing. He has been working in this area for more than 8 years, published one book and about 70 journal/conference papers and co-edited five conference/workshop proceedings in the related areas. He got his Bachelor and Master degrees in Mechanical Engineering from Northern Jiaotong University, China in 1983 and 1986, respectively, and his PhD degree in Computer Science from the University of Technology of Compiegne, France in 1996. He is a Senior Member of IEEE and a member of AAAI.

Helen Xie is a research officer at the Integrated Manufacturing Technologies Institute, National Research Council of Canada. She received her MSc in Mechanical Engineering from the University of Saskatchewan in Canada, MSc from Beijing University of Aeronautics and Astronautics, and BSc from Tianjin University in China. She has been actively involved in the area of product data representation, product configuration management, and production planning and scheduling. Her current research interests are in application of artificial intelligence and Web technologies on product configuration, conceptual design, and distributed and collaborative design.

Joseph Neelamkavil is a Senior Research Officer at the Integrated Manufacturing Technology Institute (IMTI) of the National Research Council of Canada (NRC). As a member of IMTI's concurrent engineering team, he is actively involved in the creation of tools that support engineering designers during the early stages of product development. His research interests encompass conducting research on techniques for design representations, knowledge capture and designs reuse to accelerate product development process. He is a registered Professional Engineer in the Province of Ontario, Canada. He received his BSc in Mechanical Engineering from India, MSc in Production Engineering from Trinity College, Dublin, Ireland, and MASc in Mechanical Engineering from the University of Toronto, Canada.

Ajit Pardasani is a research officer at the Integrated Manufacturing Technologies Institute, London, Ontario. He has a MSc in Information and Systems Science from Carleton University. His research interests are in the area of early design, modelling and simulation, operations research, and object-oriented software engineering.