This article was downloaded by: [University of Connecticut]
On: 12 October 2014, At: 13:30
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

A simultaneous collaborative framework for mould and die design

Chang Li [a] & Jian Jun Li [a]
[a] State Key Laboratory of Plastic Forming Simulation and Die & Mould Technology , Huazhong University of Science and Technology , Wuhan 430074, People's Republic of China
Published online: 04 Apr 2008.

To cite this article: Chang Li & Jian Jun Li (2008) A simultaneous collaborative framework for mould and die design, International Journal of Production Research, 46:11, 2989-3006, DOI: 10.1080/00207540600658054

To link to this article: http://dx.doi.org/10.1080/00207540600658054

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

International Journal of Production Research,
Vol. 46, No. 11, 1 June 2008, 2989–3006

Taylor & Francis
Taylor & Francis Group

A simultaneous collaborative framework for mould and die design

CHANG LI and JIAN JUN LI*

State Key Laboratory of Plastic Forming Simulation and Die & Mould Technology, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China

(*Revision received February 2006*)

The fiercely competitive global market is obligating mould and die manufacturers to embrace Computer-Supported Collaborative Design (CSCD) as a way to shorten delivery time and improve product quality. A number of collaborative tools have been developed to keep up with demand. Unfortunately, these existing tools support either collaborative review or collaborative design, but hardly consider the seamless integration of the two types of tools and the characteristics of mould and die. This paper presents a novel framework that combines co-review and co-design in one system so as to cover different aspects of mould and die design. The framework consists of four tiers. In the core, five databases serving different purposes are settled for geographically distributed engineers to inherit various disciplines and propagate their intents. Then, an access-control mechanism is employed to synchronize and secure data during the process of mould and die design in the next tier. For the purpose of interchanging information in real time, an event-driven strategy is applied in the third tier. Afterwards, the outmost tier provides uniform interfaces for invoking application tools and manipulating design information in the collaborative system. Finally, the framework proves effective to support simultaneous collaborative design of mould and die through an example scenario.

Keywords: CSCD; Product development; C4P; Mould and die

Downloaded by [University of Connecticut] at 13:30 12 October 2014

1. Introduction

In the past decade, moulds and dies have been increasingly widely used in the industries of mechanics, communication, electrics, automobile, aviation, and so on. To survive in a fiercely competitive market, mould and die manufacturers have to keep employing advanced technologies like C4P (CAD/CAM/CAPP/CAE/PDM) to shorten delivery time and improve the design quality of the mould and die. However, manufacturers are still confronted with great challenges such as the following:

- Nowadays, the structures of moulds and dies are become increasingly complicated, e.g. one set of progressive die assemblies normally has 500–2000 components with hundreds, even thousands, of pockets and detailing. It is such a complex and huge job that an integrated product team of individuals with different disciplines, whose members may come from different enterprises, even different countries, needs to be combined temporally to

*Corresponding author. Email: ljj@263.net.cn

International Journal of Production Research
ISSN 0020–7543 print/ISSN 1366–588X online © 2008 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540600658054

cope and to meet the increasingly reduced delivery times. Additionally, customers always like to make changes to requirements or trace the design status during the process of the mould and die design.
- In different stages of the mould and die design, each engineer concentrates only on their own disciplinary aspect and can benefit greatly from different professional software tools on the market, such as Rhino and Alias for industrial design, UG, CATIA and Pro/E for product design, Ansys and Deform for analysing, and Adams for motion simulation. However, each tool has its own concept and data representations, which prevent the capturing and interchanging of product information seamlessly between upstream design and downstream manufacture.
- Outsourcing is a common way to provide qualified products in a minimal cycle, since most mould and die manufacturers are not large and do not have enough engineers and machines to finish all the components of a set of mould and die. According to Rezayat (2000), almost 50–80% of components in a product are outsourced to external suppliers. On the other hand, it is important for customers to be involved in the process of mould and die design to avoid any misunderstanding from engineers. So, a light-weight and easily deployed tool is desirable for geographically distributed customers, suppliers, and managers to review survey design information, i.e. a description of team members, schedule of product design, and product models, across a network.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

With advancing technologies, especially information technology, many philosophies have come into existence to facilitate product design and realization processes (Li *et al.* 2004) Computer-supported collaborative design is an ideal solution to address such challenges and is having an increasing impact on modern industries. CSCD has three essential features, a group, a computer-supported environment, and a common object. This means that a geographically distributed integrated product team with multi-disciplines can carry out complicated design activities by processing the same product data and sharing the design information in a distributed computer environment.

In this paper, a simultaneous collaborative system based on a commercial CAD (UG) system is proposed to promulgate multi-disciplines among geographically distributed engineers in nature, beyond the restrictions of space, time, and platforms. After the related research work is discussed in section 2, the new collaborative framework and its implementation are detailed in section 3. Section 4 uses an example scenario to elaborate on the collaborative prototype. Finally, conclusions are presented in section 5.

2. Related work

Since its advent, CSCD has been the focus of attention from researchers of design methodologies all over the world, and a number of frameworks and tools have been developed. Applications to date have been aimed at sharing design data and interchanging information among participants.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

Sharing design data has received much attention in research work. The DICE project (Krishnamurthy and Law 1997) employs a centralized database to store all the project design data, with each agent in a distributed environment having their own local database. KADBASE (Howard and Rehak 1989) focuses on the issue of semantic and syntactic translations for data retrieval. However, neither of them can support the evolution of realistic design practices from multi-disciplines over time. Krishnamurthy and Law (1997) propose a three-layered model of versions, assemblies, and configurations. In the model, versions maintain evolving descriptions of a primitive entity, assemblies describe component instances in a composite entity, and configurations integrate assemblies in terms of versions in an entire project. Operators saving and managing data at each of the three levels provide designers with a more comprehensive data management model.

Due to the diversities of platforms and software, interchanging information in a collaborative environment becomes another important and complicated aspect of CSCD. So far, middle-ware technologies, e.g. CORBA, MS COM/DCOM, and J2EE, are widely used to encapsulate functions of different CAD/CAM kernels, so that design information from distributed participants can be transferred freely across networks. Liu (2000) proposes 'the common core interface' to wrap and expose API functions of different CAD kernels to provide a generic and neutral application layer through a MS COM/DCOM interface-based framework. Zhou *et al*. (2002) encapsulate the ACIS modelling kernel with CORBA and develop a Web-based real-time collaborative modelling system named WebCOSMOS. Nevertheless, the complexity and differences in different kernels result in a heavy system with less flexibility. Another popular encapsulating methodology is based on the agent, which is introduced from artificial intelligence. Cutkosky *et al*. (1993) first developed an agent-based distributed integrating concurrent engineering system called the Palo Alto Collaborative Testbed (PACT). In this system, tools for mechanical design, control system, digital circuit design, etc. are encapsulated into a series of agents. These distributed agents communicate with each other by Knowledge Query and Manipulation Language (KQML) through the Internet. As a result, autonomy and intelligence are brought into CSCD systems besides convenient information interchanging.

One significant area that has not been fully explored in CSCD is the simultaneous collaborative support for 3D CAD/CAM. According to the strategies and functions, some developed tools can be generally categorized into co-review and co-design.

Co-review assists customers, suppliers, managers, and engineers to participate in design activities by providing visualized scenarios across a network. This is necessary because parametric geometric models are always too large in volume to transmit in a network among geographically distributed participants efficiently. On the other hand, not all the participants need precise geometry models during product development, such as in conceptual product design, or in communication between customers and project managers. Web technologies, including http, Java, and some concise formats for Web applications, e.g. VRML, JPEG (Kan *et al*. 2001, Bidarra *et al*. 2002, Wang and Zhang 2002), can accommodate a light-weight and platform-independent infrastructure, so that participants can share their views, examine design validity, or make online annotations on corresponding views. Because the concise formats are not real geometric models but some graphic information, participants cannot create and modify objects described by these formats persistently. Autovue^{TM} (2005) is one such tool that enables geographically distributed participants to access, view, and markup a product, whatever the format involved. As a result, workflow efficiency and product quality are improved to a certain extent. Other co-review tools include eDrawing^{TM}, Streamline^{TM}, ConceptWorks^{TM}, ConceptStatio^{TM}, CyberReview^{TM}, etc. (Huang 2002, Cheng *et al*. 2004, Li *et al*. 2004).

Co-design is for geographically distributed engineers to implement product design by operating parametric geometric models in a collaborative environment. This is the most important and complicated part of CSCD. To describe product data precisely, capture design intentions exactly, and support design processes seamlessly, parametric geometric models always contain plenty of information besides geometric information, such as material attributes, tolerance attributes, etc. Historically, software tool vendors have developed all kinds of worldwide used C4P tools with their own data representations, as part of their competitive advantages (Szykman *et al*. 2001). A serious consequence is that a gap of transmitting information exists in product development. Co-design tools play the role to shield geographically distributed engineers with multi-disciplines from the gap, ensure design consistency during different stages of product design, engineering simulation and analysis, NC manufacturing, and so on. OneSpace^{TM} (2005) is a commercial 3D CAD software with collaborative PLM application, able to handle engineering models and animation across network. CollabCAD^{TM} (2005) is also a 3D CAD/CAM software based on Open Cascade geometry kernel and Java technologies, which facilitates basic feature modelling, assembly management and allows multiple designers viewing and modifying the same design through network. Weak support for conventional CAD/CAM capabilities and excessively strict check in/check out control over shared 3D models put them at a disadvantage in competition. Conversely, Syco3D (Nam and Wright 2001) focuses on addressing usability and usefulness in modelling, editing, and reviewing CAD models in a virtual collaborative environment through a designer-centred approach achieves good result.

Considering the merits and drawbacks of co-review and co-design, a new trend is to aggregate both along with additional audio, even video-conferencing applications into one environment, so that the whole phases of mould and die design and different demands from geographically distributed multi-disciplinary professionals can be covered.

3. System framework and implementation

Conventionally, design activities of mould and die are mainly carried out in a sequent mode. Interdependencies among different design phases are entirely ignored. Due to the weak communication between upstream design and downstream design, unreasonable decisions or invalid designs made in the early stage always emerge in the late stage. These can be corrected later, but this will increase the cost and delay the delivery time. On the contrary, CSCD allows geographically distributed engineers with multi-disciplines to work together in a concurrent manner throughout the process of product development. This method overcomes the drawbacks of more conventional methods and reduces the delivery time dramatically.

CSCD has great advantages, but it is not easy to support, especially for the design of mould and die. Mould and die is an order-oriented tooling industry. Due to an

Downloaded by [University of Connecticut] at 13:30 12 October 2014

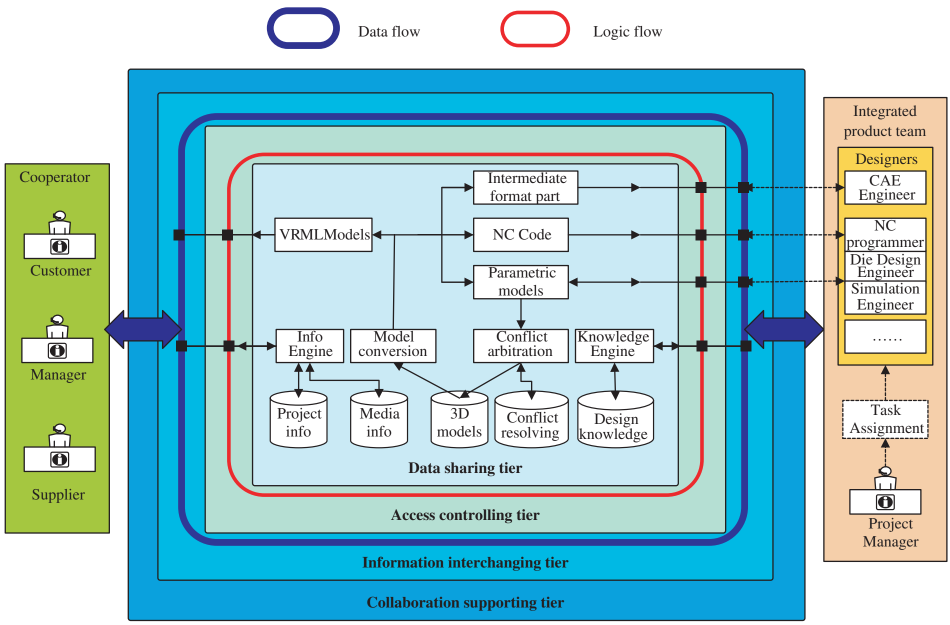

Figure 1. CSCD framework for a mould and die design.

ever-changing market, customers keep updating their wish lists during the process of mould and die design. This iterative process may bring a risk of failure or delayed delivery. In addition, the interdependence among components in a mould and die set means that changes in one phase potentially impact other phases. This prevents a mould and die set from being divided into relatively independent functional modules, which could be combined together after each module is designed, as is done in the automobile industry. This makes simultaneous design difficult. Moreover, because of indirect communications among geographically distributed participants, misunderstandings always occur and result in conflict design during the process of mould and die design; consequently, the efficiency of design decreases considerably. Making full use of all resources, including machines, human resources, application tools, design data, design knowledge, etc., is a vital part of CSCD for a mould and die design.

In order to address the aforementioned issues, a four-tier framework to support simultaneous CSCD for mould and die is proposed, as shown in figure 1. At the core of the framework, the Data-Sharing Tier functions as the data repository to store information related to product development, such as 3D models, media information, and design knowledge. The second tier is the Access-Controlling Tier. All the data from Data-Sharing Tier are appended by a list of attributes here so as to maintain information consistency. The third tier is responsible for transmitting information and providing a communication channel for participants. The outmost tier is the Collaboration-Supporting Tier, which offers participants an integrated application tool environment and common interfaces to capture and manipulate the desired design information.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

3.1 *Data-Sharing Tier*

Mould and die design is a complicated job, involving many different disciplines, each with its own area of concern and expertise according to functional concerns (Roseman and Gero 1999), such as material, mechanics, forming process, and mould and die structure. Even at the same stage of mould and die development, different engineers may have different designs. So, sharing all kinds of information across the whole product-development phase is an important aspect in a distributed collaborative environment (Frank and Mitschang 2002).

To provide an efficient information-sharing repository, the Data-Sharing Tier classifies the related information into five databases according to their serving purposes, as described below:

- 3D models database: used to store parametric 3D models of all the components in a product assembly. All other models (e.g. IGES models, VRML models) will be derived from the parametric models with the model conversion module. This is the kernel database to support co-design of a product.
- Project information database: used to store project-related information, including description of various projects, roles of members in each team, privileges of a role, task assignment, task evolution, and so on. It is important to assist access control and human resource coordination in the collaborative framework.
- Media information database: used to backup communication information, such as instant messages, annotated images, and meeting records.
- Conflict-resolving database: used to store the criteria determining whether design conflicts exist, thus guaranteeing the uniqueness of 3D models of a product in a collaborative environment.
- Design knowledge database: used to store multi-disciplinary design knowledge for participants to refer to, such as design rules, design experiences, best practice, and other standard documents, which provides all participants with a uniform knowledge background.

The architectures to implement data sharing can be categorized into centralized architecture and replicated architecture (Zhou and Lin 1999). In a centralized architecture (shown in figure 2(a)), the server site is equipped with full application tools, e.g. C4P, so all the operations are performed here. A single view of applications will be transmitted to all client sites as requests. Accordingly, a client site can send commands to application tools to modify the models in the server. The main advantage of this architecture is that the consistency of a database can be easily guaranteed. On the contrary, in a replicated architecture (shown in figure 2(b)), all the client sites run their own application tools, initially loading a copy of collaborative data, and the server site serves only as a shared data repository. Obviously, the replicated architecture accords more with reality, with a higher efficiency and lower network load, so the latter is chosen as the architecture of the current framework to sharing data. However, how to maintain the consistency of design data is a challenge that will be discussed in section 3.2.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

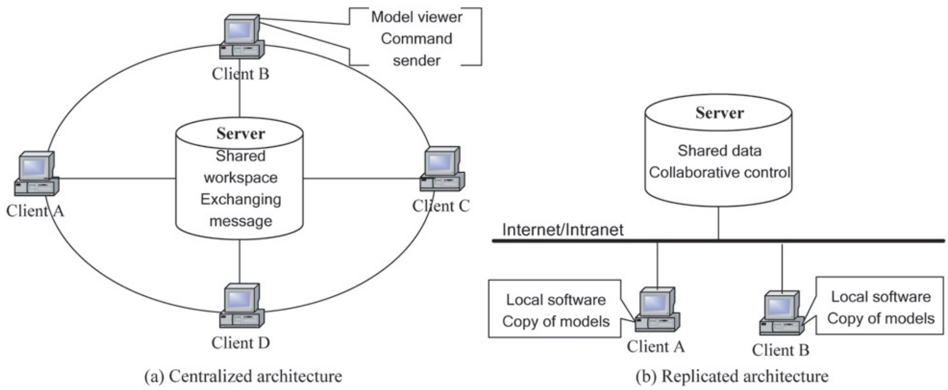

Figure 2. Two data-sharing architectures.

3.2 *Access-Controlling Tier*

Collaboration can have an asynchronous mode and synchronous mode based on time constraints. In the former, such as E-mail and BBS, time does not need to be considered. However, the latter emphasizes real-time consistency among geographically distributed participants. It is easy to imagine that the whole design work will fall into confusion soon if no restriction is applied to the shared product data. To ensure that what one person sees is what another sees all the time, a collaborative mechanism is proposed in the Access-Controlling Tier.

In a collaborative environment, a product team works together for a common goal. During this process, each member takes their own role according to their responsibilities. This means that a common goal (a product) can be divided into a series of tasks, and the tasks can be subdivided further until there is less coupling between the subdivided tasks. The task with less coupling is referred to here as the Atom Task, which is easy for an engineer to accomplish independently. Figure 3 illustrates an activity tree model, which is a hybrid tree of tasks and components (assemblies and parts). In this hierarchy, the tree is divided into an Executing Layer and Supervising Layer. The executing layer consists of all the components, in which the models of a product are defined. The supervising layer includes all the tasks, in which the dependencies of components are defined. The two layers are bridged by atom tasks, which are supervised by their parent task nodes in Supervising Layer and own child components in Executing Layer.

In the Executing Layer, a group of components owned by an atom task are the direct carriers of design intents. Once their parent, an atom task, is assigned to an engineer, all its children will be entirely under the control of the engineer, who will be involved in designing the actual components of a mould and die set. This control is exclusive. In other words, each atom task can be owned by only one engineer. While we examine a product composed of components, one aspect that cannot be ignored is that the components owned by each atom task are not isolated from each other, e.g. in figure 4, the shape of a piercing punch in Task Punch/die Design is dependent on that of a scrap in strip part of the Task Strip Layout, while its height and pockets are associated with the top diebase in the Task Diebase Design. Considering the

Downloaded by [University of Connecticut] at 13:30 12 October 2014

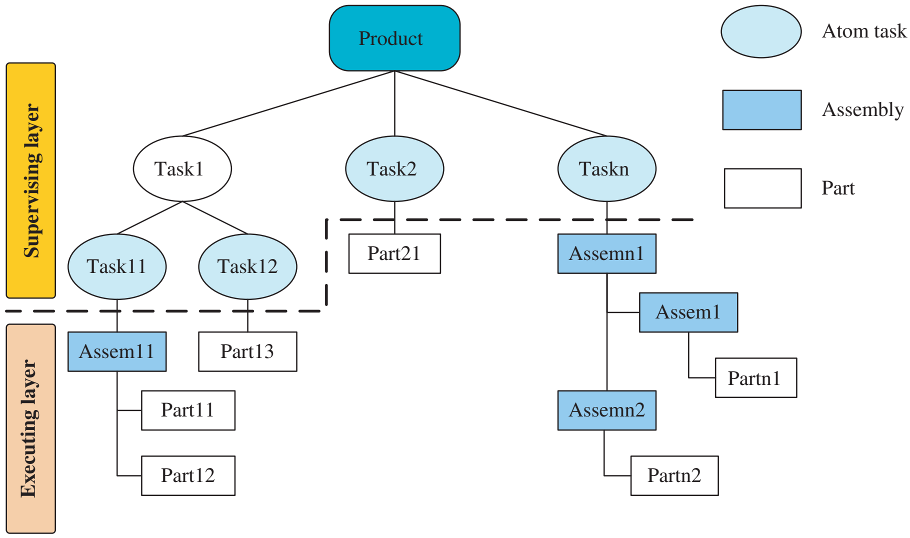

Figure 3. Activity tree model.

dependencies among components, three privileges, namely read, write, and coordinate are defined:

- read: allow loading components;
- write: allow modify components and save modification;
- coordinate: allow external participants having write privilege temporarily.

Each atom task holder has all three privileges for their task. External participants of an atom task can usually hold read and coordinate privileges. When an atom task holder requests for coordination to external participants, they can release the write privilege over their atom task temporarily to external participants, to modify the components and save the modification. At that time, the rule of 'first come, first serve' is observed. While atom tasks are executed respectively, the validity of a product composed of components is guaranteed by a conflict-resolving module, which utilizes the knowledge in a conflict-resolving database to resolve any design conflicts.

In the Supervising Layer, a parent node takes charge of supervising the evolution of its child nodes. If necessary, the activity tree may be refined by creating, deleting, merging, or subdividing some tasks. In an integrated product team, the project manager holds the root supervising privilege, who has the power to configure and optimize all the resource of a team through task division and task assignment, but never works on any component of a product directly. Only assigning atom tasks to an individual engineer is not enough to ensure that the collaborative mechanism works well because of the dependency between tasks. Therefore, prior relation and simultaneous relation are proposed to describe the relations between two child tasks with a common parent task:

- prior relation: a subsequent task can be carried out on the design result of its previous tasks;
- simultaneous relation: two tasks can be carried out simultaneously without any dependencies.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

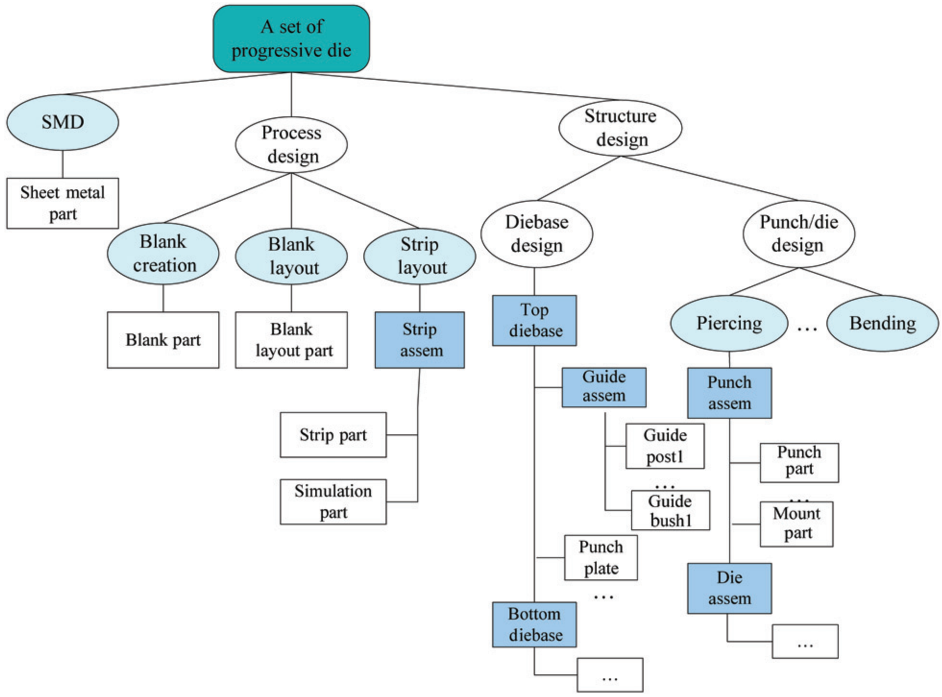

Figure 4. Example of an activity tree for a progressive die set.

As in figure 4, the Task Process Design cannot start until the Task SMD finishes, whereas Task Piercing and Task Bending can be carried out concurrently. The former relation is called a prior relation (SMD prior to Process Design), and the latter is called a simultaneous relation. Only after all its prior tasks are finished can a subsequent task start. For example, in figure 4, the Task Structure Design cannot begin until both the Task SMD and Task Process Design are finished. So, any child task must notify its parent task as soon as it is finished. This will be detailed in section 3.3. The relations among tasks are determined once task division is finished. Through these strategies, a collaborative mechanism is built and can keep participants working on shared data in regular sequence.

3.3 *Information-Interchanging Tier*

The Access-Controlling Tier provides a mechanism to keep data flow in order. However, all the information, such as a discrete media continuous stream (Garcia *et al.* 2004), generated and modified by geographically distributed participants, is actually interchanged in the Information Interchanging Tier. To deal with various format data and provide an efficient communication channel, three services are embedded into the information-exchanging centre:

- Communication service: before CSCD tools are developed, team members have to collaborate through a face-to-face meeting, email, fax, or telephone

Downloaded by [University of Connecticut] at 13:30 12 October 2014

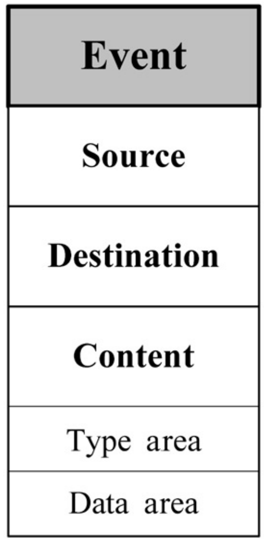

Figure 5. Event structure.

in a product team (Zhuang *et al*. 2000). Such methods are time-consuming and error-prone. This service aims to enable geographically distributed engineers to exchange ideas with each other in a direct and real-time way, to identify customers' demands and discuss design conflicts existing in product development.

- Transmission service: in a collaborative environment, different format data flows among geographically distributed engineers. Parametric 3D models play a central role, since a geometric model is a more general and effective way for most practising engineers to understand the design intents and find solutions (Zha 2002). This service is built to transmit 3D models among engineers to keep all views of participants up to date.
- Awareness service: as described in section 3.2, close dependencies exist in a collaborative system. It will be impossible to inform related and distributed engineers in time after a task is finished without any automated notification. This service aims to provide the capability of group awareness, such as instant messages, automated feedback, etc.

The three services are driven by an event-driven mechanism. The structure of an event is illustrated in figure 5, which indicates source, destination, and content. The source area records the IP address to identify the sender of an event, the destination area specifies the recipient of the event by recording the recipient's IP address, and the content area, including type area and data area, specifies the type of event and stores information (e.g. product models, textual description).

To demonstrate how the interchanging module works, the process of updating a 3D model is chosen as an example. As shown in figure 6, Engineer A and Engineer B are in the same integrated product team. Engineer A is designated as the sender,

Downloaded by [University of Connecticut] at 13:30 12 October 2014

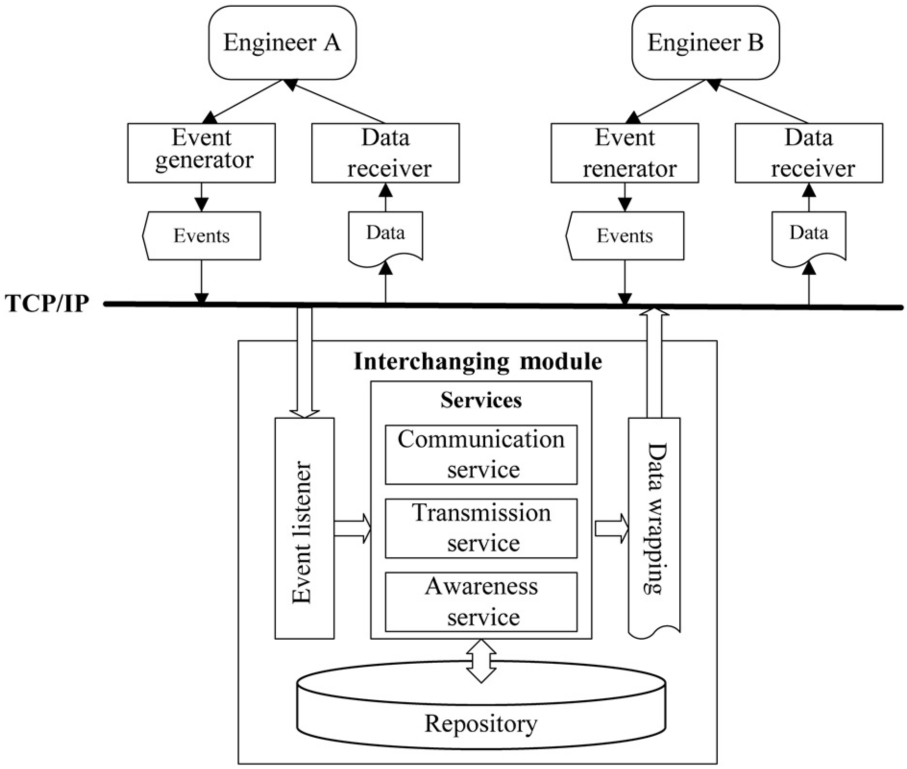

Figure 6. Mechanism of the information-interchange module.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

and Engineer B as the receiver. The whole interchanging process includes three phases, as follows:

- Update the server models: when Engineer A modifies their models and saves the modification, a modification request event is generated automatically (shown in figure 7(a)). In the interchanging module, an event listener keeps listening for any event occurring on the network. As soon as the modification request event is captured, the listener invokes the transmission service to update the models in a repository. Irrespective of success or failure, a cue message is attached to the event, as shown in figure 7(b). Engineer A will then receive the acknowledgement message.
- Send an awareness message: once Engineer A obtains a successful acknowledgement from the network, an update awareness event is triggered (shown in figure 8(a)). Once the event is captured by the listener, the awareness service is invoked to match a proper message from the repository. Then, the message is attached to the awareness event (shown in figure 8(b)). Since the destination is contained in the event, no other engineers other than the destination engineer/engineers receive the data. In figure 6, Engineer B will receive the data, an awareness message from Engineer A.
- Update the client models: if Engineer B ignores the received awareness, no further response will be given. Otherwise, an update request event is

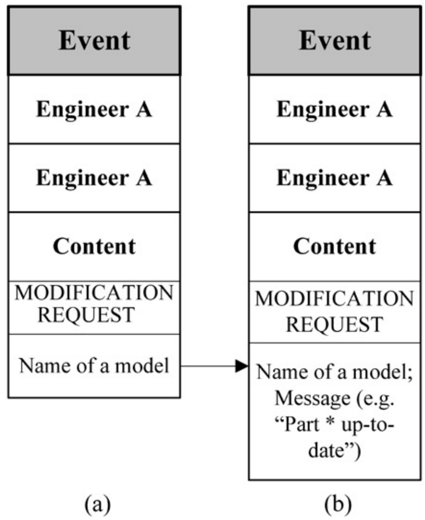

Figure 7. Structure of the modification request event.

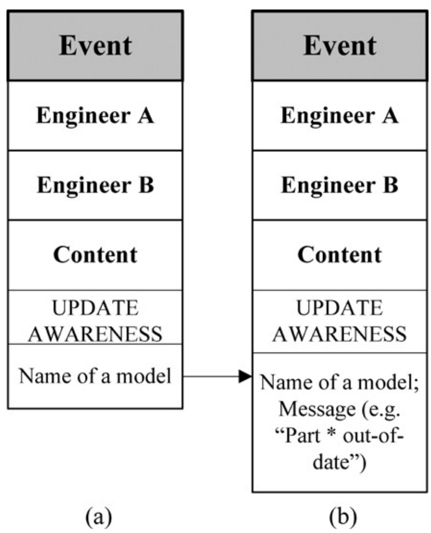

Figure 8. Structure of the update awareness event.

generated and sent back to the network (shown in figure 9(a)). Then, the transmission service will run to fetch modified models from the repository, which is attached to the request event (shown in figure 9(b)). Then, Engineer B can receive modified models and update their own models. If Engineer B wants to discuss the modification with Engineer A, they can send a negotiation event. Then, the communication service is activated to provide a virtual channel for them to discuss by instant messages and concise images. At the same time, all discussion will be stored in a repository as backup.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

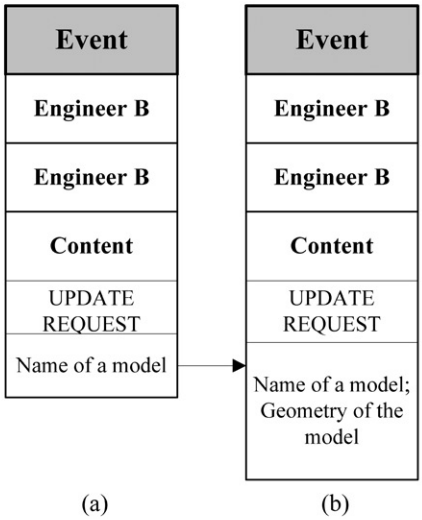

Figure 9. Structure of the update request event.

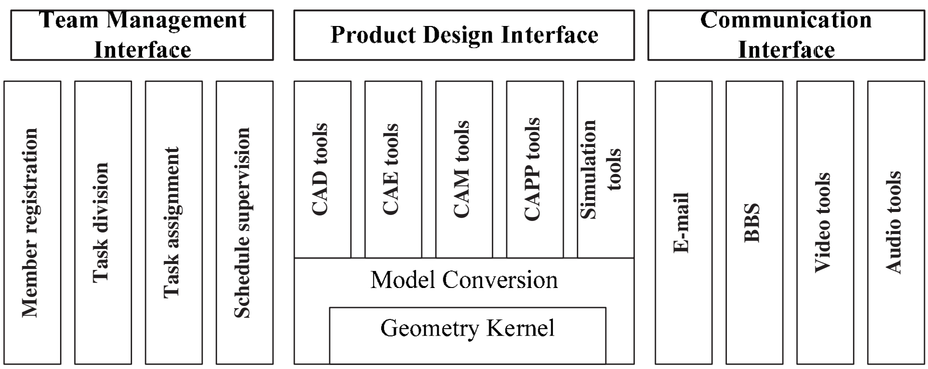

Figure 10. Collaborative supporting environment.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

3.4 *Collaboration-Supporting Tier*

During the process of product development, various kinds of application tools are used in different special domains, such as industrial design, product design, analysis, and simulation. Due to the complexity of product development, a large amount of human–machine interaction is needed. A uniform and friendly interface is therefore very important for a successful collaborative system.

In the Collaboration-Supporting Tier, three types of interfaces are partitioned off: the Team Management Interface, Product Design Interface, and Communication Interface, as shown in figure 10. The Team Management Interface integrates the applications to assign and optimize all kinds of resources. All design tools are put into the Product Design Interface, implementing product modelling, analysis, and simulation until manufacture. Communication tools are embedded in the Communication Interface to support peer-to-peer discussions or group net meetings. Using middle-ware technology, such as CORBA, a variety of software packages used in different design phases can be integrated into different interfaces in a plug-and-play manner. Moreover, through the protocols of TCP/IP, FTP, HTTP, etc., the three interfaces allow integrated packages activating each other, exchanging information, and manipulating shared data over network to provide a seamless, flexible, complete, and distributed collaborative environment.

4. Application case

An application case is illustrated here to demonstrate a simultaneous collaborative process. In the scenario, there are three participants working together over a distributed network to design the inserts of a progressive die collaboratively.

- Participant A: the project manager, responsible for coordinating the whole team work. First of all, they log in to the collaborative environment and create a project, as shown in figure 11. Then, they divide the project into atom tasks and assign them to different engineers. To avoid design conflicts, it is better for different engineers to take charge of different components. So, different components of a mould and die can be divided into different atom tasks according to their functions or positions, e.g. a piercing or bending insert can be considered as an atom task to assign to an engineer. As shown in figure 12, components in one or more stations in a progressive die set are defined as an atom task. Here, die inserts from Station 4 to Station 8 are

Downloaded by [University of Connecticut] at 13:30 12 October 2014

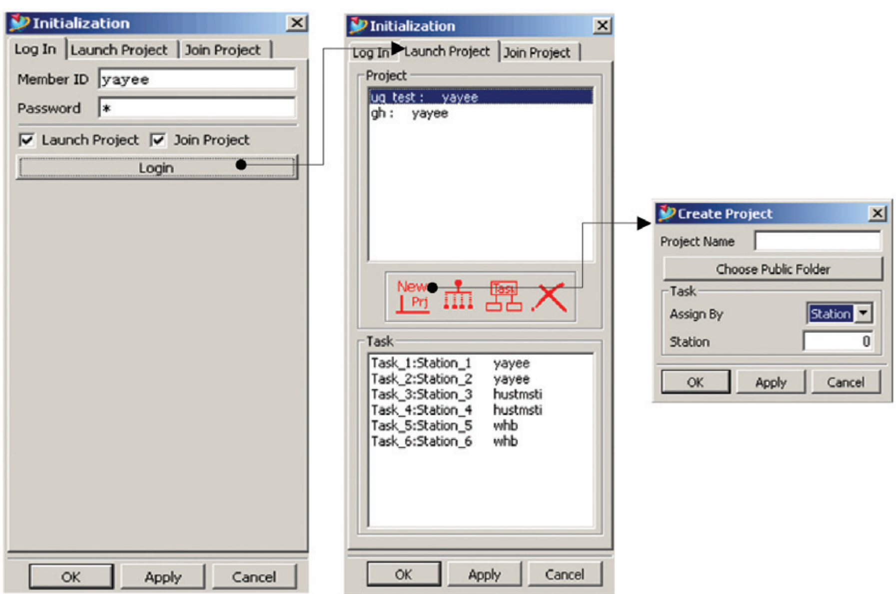

Figure 11. Process of logging in and creating a new project.

assigned to Participant B as an atom task, while the rest are assigned to Participant C. Then, the information of task division and task assignment is recorded in the project information database in the Data-Sharing Tier, and at the same time, an instant message is sent to both B and C to inform them of the tasks.

- Participant B: one of the two die designers who undertakes the task to design all the die inserts from Station 4 to Station 8. Once they log in to the collaborative environment, they will receive the instant message from Participant A, and their task is then retrieved from the project information database and displayed in the Initialization dialog, as shown in figure 13. After access validation from the Access-Controlling Tier, the Information Interchanging Tier loads the finished assemblies from the 3D models database to the client site of Participant B, including the results of the strip layout and diebase design; then, they concentrate on designing their die inserts, creating, editing, and saving models in their domain, as shown in figure 14 (Participant B). To obtain the whole view of die inserts and check whether there is any interference between the two designers' work, they can load assemblies of die inserts from Participant C at any time. It is worth noting that the assemblies from Participant C are read-only to them without Participant C's authorization.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

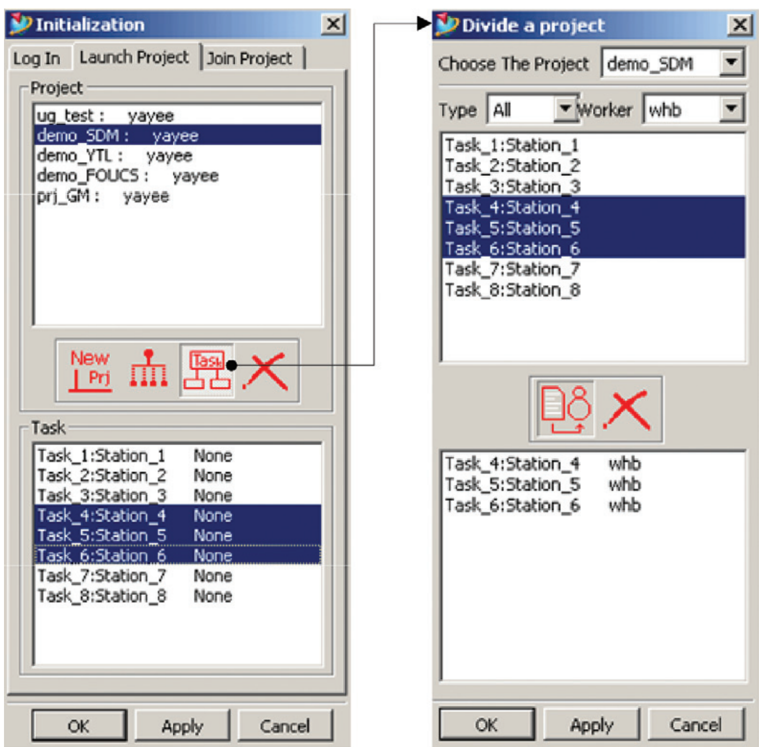

Figure 12. Task definition and assignment.

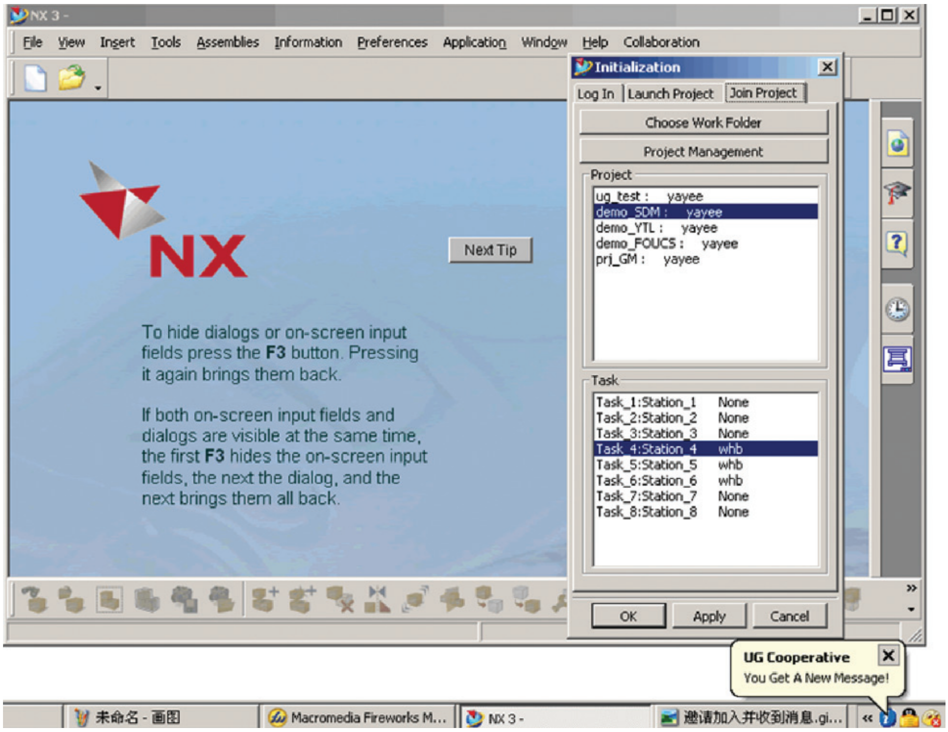

Figure 13. Awareness message from a project manager.

- Participant C: the other die designer who undertakes the design task of all the die inserts from Station 1 to Station 3. After logging in, they will also get the instant message. Their task is listed in the dialogue. They begin their task after loading previous assemblies. During their design, they can synchronize the die inserts from Station 4 to Station 8 from Participant B. Figure 14 illustrates the process for Participant C to synchronize one pair of piercing inserts from Participant B.

The example scenario, though simple, is sufficient to illustrate how the prototype system presented here serves to achieve 3D simultaneous collaborative design in a geographically distributed team.

5. Conclusions

Cost-effective and time-efficient mould and die design is achieved by reasonable resource distribution, simultaneous execution, expedited communication, and streamlined sharing. A framework has been proposed in this paper to cover these elements. A prototype system based on the framework has been developed to support a team of geographically distributed engineers to design moulds and dies in real time with fewer wrong decisions.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

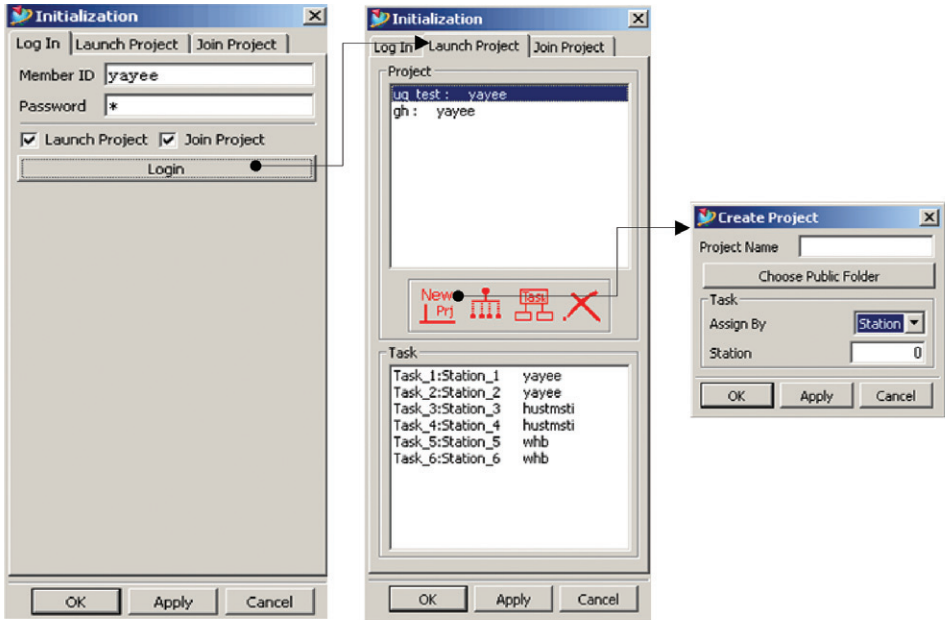

Figure 14. Synchronization of die inserts between two engineers.

In the framework, the information including 3D models, design knowledge, intermediate information, etc., is classified and organized into different databases in terms of its serving purposes. Furthermore, a role-based access control mechanism based on a hybrid tree of tasks and components is used so that the shared data can be utilized efficiently and securely. Finally, an event-driven information-interchange strategy and application-integrated interfaces accommodate an integrated product team with an efficient and friendly workspace.

There are several drawbacks with the current prototype system that need to be addressed in future work. The management toward the process of product development depends entirely on the project leader in the current system. This is a very heavy burden for an individual. A workflow subsystem is being investigated to enable process-management automation. Conflict is a popular issue in collaboration, and detecting various conflicts and resolving them automatically are challenges. Different methodologies need to be explored to replace the completely interactive method used in the current system. In addition, Web-based collaboration among managers, customers, and suppliers will also need to be enhanced in future endeavours.

References

Autovue^{TM}, Cimmetry Systems Inc. Available online at: http://www.cimmetry.com (accessed 18 September 2005).

Bidarra, R., Kranendonk, N., Noort, A. and Bronsvoort, W.F., A collaborative framework for integrated part and assembly modeling. *J. Comput. Inform. Sci. Eng.*, 2002, **2**, 256–264.

Downloaded by [University of Connecticut] at 13:30 12 October 2014

Cheng, L., Song, Z.J. and Feng, L., Internet-enabled real-time collaborative assembly modeling via e-Assembly system: status and promise. *Comput. Aided Des.*, 2004, **36**, 835–847.

CollabCAD^{TM}, National Informatics Center, India. Available online at: http://www.collabcad.com (accessed 20 September 2005).

Cutkosky, M.R., Engelmore, R.S., Fikes, R., Genesereth, M.R., Gruber, T.R., Mask, W.S., Tenenbaum, J.M. and Weber, J.C., PACT: an experiment in integrating concurrent engineering systems. *IEEE Comput.*, 1993, **26**, 28–37.

Frank, A. and Mitschang, B., A customizable shared information space to support concurrent design. *Comput. Ind.*, 2002, **48**, 45–57.

Garcia, E., Guyennet, H., Lapayre, J.C. and Zerhouni, N., A new industrial cooperative tele-maintenance platform. *Comput. Ind. Eng.*, 2004, **46**, 851–864.

Howard, H.C. and Rehak, D.R., KADBASE: interfacing expert systems with databases. *IEEE Expert*, 1989, **4**, 65–76.

Huang, G.Q., Web-based support for collaborative product design review. *Comput. Ind.*, 2002, **48**, 71–88.

Kan, H.Y., Duffy, V.G. and Su, C.J., An internet virtual reality collaborative environment for effective product design. *Comput. Ind.*, 2001, **45**, 197–213.

Krishnamurthy, K. and Law, K.H., A data management model for collaborative design in a cad environment. *Eng. Comput.*, 1997, **13**, 65–86.

Li, W.D., Fuh, J.Y.H. and Wong, Y.S., An internet-enabled integrated system for co-design and concurrent engineering. *Comput. Ind.*, 2004, **55**, 87–103.

Liu, X.D., CFACA: component framework for feature-based design and process planning. *Comput. Aided Des.*, 2000, **32**, 397–408.

Nam, T.J. and Wright, D., The development and evaluation of Syco3D: a real-time collaborative 3D CAD system. *Des. Stud.*, 2001, **22**, 557–582.

OneSpace^{TM}, CoCreate Inc. Available online at: http://www.cocreate.com (accessed 20 September 2005).

Rezayat, M., The enterprise-web portal for life-cycle support. *Comput. Aided Des.*, 2000, **32**, 85–96.

Roseman, M.A. and Gero, J.S., Purpose and function in a collaborative CAD environment. *Reliab. Eng. Syst. Saf.*, 1999, **64**, 167–179.

Szykman, S., Fenves, S., Keirouz, W. and Shooter, S.B., A foundation for interoperability in next-generation product development systems. *Comput. Aided Des.*, 2001, **33**, 545–559.

Wang, H.F. and Zhang, Y.L., CAD/CAM integrated system in collaborative development environment. *Robot. Comput. Integr. Manuf.*, 2002, **18**, 135–145.

Zha, X.F., A knowledge intensive multi-agent framework for cooperative/collaborative design modeling and decision support of assemblies. *Knowl. Based Syst.*, 2002, **15**, 493–506.

Zhou, J.W. and Lin, G., Implementation of collaborative design environment based on single user CAD systems, in *Proceedings of Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, Adelaide, Australia, 1999, pp. 78–83.

Zhou, X., Li, J., He, F.Z., and Gao, S.M., Collaborative solid modeling system over internet, in *Proceedings of China National Conference on CAD & CG*, Guiyang, 2002, pp. 720–725 (in Chinese).

Zhuang, Y., Chen, L. and Venter, R., CyberEye: an internet-enabled environment to support collaborative design. *Concurr. Eng. Res. Applic.*, 2000, **8**, 213–229.

Downloaded by [University of Connecticut] at 13:30 12 October 2014