Advances in Collaborative CAD: The-State-of-the-Art

J. Y. H. Fuh[1] and W. D. Li[2]

[1]National University of Singapore, mpefuhyh@nus.edu.sg
[2]Singapore Institute of Manufacturing Technology, wdli@simtech.a-star.edu.sg

ABSTRACT

In collaborative design and distributed manufacturing, the need to co-develop parts by designers at different geographical locations often arises. For designing a promising product, there is always a need for collaboration among the design, marketing, financial and procurement departments, and the top management. Global manufacturing makes it difficult to frequently gather all the departments in a meeting room to discuss because of geographical constraints. In order to address this issue, recently, a number of software tools and research works have arised to provide collaborative solutions. In this paper, some important works in Web-based visualization and 3D concise representations, 3D streaming technology and co-design systems and feature-/assembly-based representation are elaborated. Meanwhile, previous works done by a project led by the authors towards this direction are also highlighted.

Keywords: *Collaborative CAD, Web-based visualization, co-design systems*

1. INTRODUCTION

With the faster and complex demands of new and customized products, companies need to participate in global design chains and collaborate with each other and overseas partners to pursue competitive advantages. Consequently, designers are increasingly faced with the challenges of integrating distributed multi-disciplinary product design and development teams made up of increasingly diverse sets of skills (Fig. 1), varying design processes and different business measures. Product design, manufacturing and analysis have therefore the strong needs for various levels of collaboration in a distributed environment. According to a survey made by the CIMdata Inc. at Ann Arbor, US in 1999, the market of Collaborative Product Commerce (CPC) will reach \$2.2 billion for 2000 and increase at a compound annual growth rate of 20 percent through 2004, when market size is expected to exceed \$4.4 billion [1]. In another survey conducted by M2Research at San Diego, US in 2001, for the question "when do you foresee the need for your company to use collaborative design software ?", 70% of respondents in product design area said their company will have a need for collaborative design software [2]. It is both technically and commercially imperative to develop new collaborative design tools or renovate traditional stand-alone CAD system by making it collaboration-native.

Recently, extensive research and development works have been carried out to develop prototype systems and methodologies for collaborative CAD, and software vendors have launched a number of systems in markets to take advantage of the huge business opportunities in this area. Innovative design in infrastructure design, communication algorithms and geometric computing algorithms for developing a collaborative CAD system have been made to address the complexity of collaborative design activities and the specific characteristics/requirements of CAD systems, therefore, to facilitate the organization of a collaborative

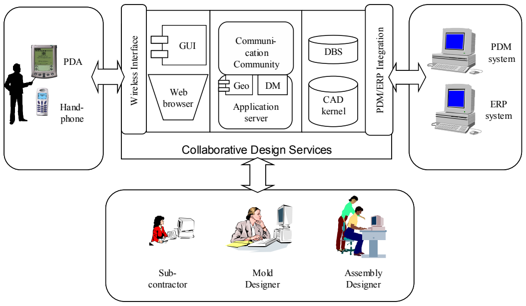


Fig. 1. The scenario of collaborative product develop-ment.

activity as either a vertical way to link upstream design and downstream manufacturing, or horizontal way to collocate a design team from the same or different disciplines to realize synchronous or asynchronous design collaboration for a complex design task. For the related works, a survey is made by authors [3]. In this paper, some significant problems are highlighted and deliberated, including Web-based visualization and 3D representations for Web-based application, 3D streaming over networks, and feature- and assembly representation for collaborative CAD. Meanwhile, some previous works towards this direction made by authors are briefly summarized here.

2. WEB-BASED VISUALIZATION AND 3D CONCISE REPRESENTATIONS

The Web is one of the most popularly used Internet tools to provide a light-weight and an operation system-independent platform for users to search, browse, retrieve and manipulate information disseminated and shared remotely. A visualization-based collaborative system contained in a Web browser can dynamically share and update 3D models through a standard communication protocol, i.e., HTTP (HyperText Transfer Protocol) in an Internet environment., to facilitate an on-line team to take on design discussion, product review, design remark and customer survey to enhance collaborative new products and conceptual design. However, designers at different locations find it hard to directly share their latest design for discussion because the large CAD files require an unbearable downloading time over the Internet. When parts being designed were modeled from different software vendors or even different versions, the visualization of the CAD model would not be possible if there were no general standards for 3D graphical representation and effective transmission strategies over networks. In order to suit the requirements of collaborative systems in the Internet and Web with limited bandwidth capability, research has been carried out to innovate light-weight 3D standards and 3D streaming communication (the 3D streaming communication will be discussed in next Section).

In order to deliver and manipulate interactive 3D objects effectively in the web, some concise formats, such as VRML, X3D (eXtensible 3D) (www.web3d. org), W3D (Web 3D) (www.macromedia.com) and MPEG-4, have been launched and the geometry of 3D CAD models can be represented as visualization-used triangular meshes and trimming lines. VRML is fundamental for these standards to represent geometric elements and scenes, while X3D and MPEG-4 are extended to support VRML and video/audio application in compressed binary formats, respectively. Some formats such as OpenHSF (www.hoops3d.com/openhsf) and ZGL (www.realitywave.com) are equivalent to the VRML standard in function while they define data for effective 3D streaming transmission through providing functions in data compression, mesh simplification and object prioritizing. The above formats are for generic usage and they are not suitable for representing complex CAD models since they lack feature and assembly structures to organize information. The trend in this area is to support and provide complex engineering data and the attributes, advanced streaming and compression formats, strong interoperability and cross-platform capabilities. Some concepts of VRML, MPEG-4, X3D and OpenHSF are chosen for brief illustration.

(1) VRML

Essentially, the VRML files describe the 3D scene in terms of objects, operations, and properties of the scene. It has the advantage of being written in text format, so that anyone with the desire to change or read the model file can do so with ease. The triangular mesh is the most popular choice of polygon for representing the mesh of a VRML model even though other types of polygons, such as quadrangle and hexagon, can also serve to represent it. In a VRML, primitive 3D, arbitrary and complex objects can be supported. Sensors are embedded to allow VRML objects and scenes to sense and respond to the passing of time and user activity. Objects are allowed to be defined and reused in programming such as Java, JavaScript and ECMAScript. The major disadvantages of VRML include files are large, no compression has been built in and there is no streaming technology, so that the download and display speeds are not satisfied.

(2) X3D

X3D is a major upgrade from VRML and retain backwards compatibility with a huge base of available 3D content. It is being developed under the Web3D Consortium's (www.web3d.org) standardization process that provides full and open access to the specifications and eventual submission to ISO for ratification to provide long-term stability for Web3D content and applications. The main motivation for moving to X3D is to have a more light-weight representation and hence eliminate the need for downloading of a heavy browser. The major features of X3D include: (i) X3D is XMLized, that is, nodes in X3D are represented in XML tags so as to take full advantages and potentials of XML on the Internet; (ii) X3D utilizes an open profile/components-based architecture enabling custom-crafted scalable implementations; and (iii) X3D incorporates numerous advanced 3D techniques including advanced rendering and multi-texturing, NURBS (Non-Uniform Rational B-Spline) surfaces, GeoSpatial referencing, Humanoid

Animation (H-Anim), and IEEE Distributed Interactive Simulation (DIS) networking.

(3) MPEG-4

The bulky sizes of VRML files hinder their effective transmission over the Internet. MPEG-4 has been proposed to define a binary compression format, i.e., BIFS (Binary Format For Scenes), to encode VRML in binary representation. Therefore, a BIFS file is often 10 to 20 times smaller in size than its VRML equivalent. As the major objective of adopting MPEG-4 is for multimedia applications, MPEG-4 supports media mixing and audio composition, and it can easily mix with rich forms of media, video and audio for multi-media collaborations in a Web-enabled environment.

(4) OpenHSF

VRML, X3D, MPEG-4 and other formats such as ZGL and W3D are designed to aim at more general and light-weighted uses of visual information for the Web and Internet applications. However, they are not very suitable for engineering data types such as complex assemblies and associated 2D drawings. OpenHSF has been proposed (Hoops3D Inc.) (www.hoops3d.com) to handle specific visualization requirements of mechanical CAD and architecture/construction software over the Internet. Some advantages of OpenHSF include: (i) it supports engineering geometry and engineering attributes such as vertices, arcs and circles, NURBS, multi-byte text, images, cutting planes, etc.; (ii) it supports 3D streaming and compression through multi-resolution objects (Level-of-Details), file ordering and compression algorithms; and (iii) it has an open format and interoperability to realize data exchange with some leading CAD and CAE vendors.

(5) Java3D

Java3D is sometimes misunderstood as a Web-based representation scheme equivalent to VRML or other equivalent representations in function. However, they have fundamental differences while there are close relationships between them. Java3D is a high-level programming API for 3D graphics rendering. The code must be compiled to move it to executable form. VRML is a text based modelling language that is interpreted dynamically from the source files. VRML is “static” and consists of a series of text represented in certain formats, and Java3D is “dynamic” as it is a Java programming language developed specially for visualizing and manipulating 3D models. Usually, VRML is used as the input information for a Java3D program. Some features of Java3D include: (i) it has rich set of 2D and 3D objects and behaviors; (ii) it encompasses 3D geometry compression and support of Level-of-Detail (LOD) objects. A binary geometry compression format (a generalized triangle strip format) is utilized in Java3D both as a run-time in-memory format for describing geometry, as well as a storage and network format; and (iii) it supports a wide variety of file formats to accommodate many vendor-specific CAD formats, interchange formats and VRML.

3. THREE-D STREAMING TECHNOLOGY

Faster visualization of CAD models during collaborative design has been needed for a long time. Recently, a new scheme for visualization has been presented, viz., the 3D streaming technology. Streaming is defined as listening or viewing media in real time as it comes across the Internet, such as conventional streaming of video and audio. It does not require the user to download the entire file before he can see the data, thus a portion of the data can be seen while downloading is still in process. Through the 3D streaming technology, users can view and manipulate the portion of the model they need. Streaming technology is especially vital to the distributed CAD system, enabling faster transmission and visualization of 3D models in real-time. Taking an automobile as an example, a full model CAD file can be a few hundred or thousand megabytes. It seems impossible to view the entire model over the Internet. Moreover, not all components or details of the automobile may be required for viewing at each instant. Different groups of people may view only certain portion of the model for their purpose. With streaming technology, there is no need for the client to download the entire model and only a portion of the model needs to be viewed at one time. Hence, the body of the automobile can be streamed first, gradually increasing the level of detail or visual quality, while hidden components need not be streamed over. When specific parts of the automobile such as the engine need to be viewed, additional information will then be streamed over. This greatly facilitates the navigation of the model over slow network connection such as a dial-up modem. Hence, 3D streaming is actually the incremental refinement process through progressive transmission over the internet, as will be illustrated in greater detail later.

Conceptually, it requires at least two steps to implement 3D streaming over the Internet: mesh simplification and mesh refinement. Simplification is the concept of removing as many polygons as possible from the mesh model in order to lower the storage requirements for rendering in the computer. Whereas, refinement provides functions to gradually retrieve the simplified model back to its original. The situation is shown in Fig. 2. To implement 3D streaming in a client/server architecture, the server is supposed to provide the functions of mesh simplification and refinement as well as communication with the clients. The clients should be able to send a streaming request, continuously receive the data and dynamically display the mesh through the viewer.

3.1 Mesh Simplification

The initial step for 3D streaming is to simplify a 3D file (e.g., a VRML file) to a smaller size while keeping the original overall shape to a certain acceptable degree. In other words, the goal of a simplification algorithm is to reduce the total number of triangular meshes preserving the original topology, and to achieve a good approximation to the original shape. Base on Garland and Herbert [13], the available simplification algorithms can be categorized into three classes: vertex decimation, iterative edge contraction and vertex clustering. In addition, geometric error functions have been used to determine the quality of simplification.

(1) Vertex decimation

Vertex decimation was initially introduced by Schroeder et al. [4] (illustrated in Fig. 3). This method can actually be divided into 3 steps: (i) vertex classification; (ii) vertex selection for removal; and (iii) re-triangulation of the resulting hole after removal. In the phase of vertex classification, it characterizes the local geometry of vertices as simple, complex, boundary, interior edge, and corner, among which only vertices of simple and interior edge types are possible candidates for removal. The step of vertex selection is performed by evaluating the decimation criteria to iteratively find the vertex with the smallest value of d, which is defined as the distance from the vertex to the best-fit average plane for the vertex's neighboring vertices. The last step is the triangulation of the resulting hole after the vertex removal. Schroeder et al. [4] suggested using a split-plane to test the feasibility of decimation to avoid overlapping of triangles. Many improved algorithms have been developed following Schroeder's decimation idea. Franc [5] discussed the advantages and disadvantages of vertex decimation algorithms compared with iterative edge contraction. Instead of determining the vertex importance by finding the distance of the vertex and its average plane, he presented a method to use a hash function to bucket the vertices with similar importance into clusters of the same reasonably small length, and the vertices are removed cluster by cluster from the least to the most important cluster after a sorting process. This approach improves the efficiency of the decimation algorithm. Moreover, it frees the restriction of Schroeder's algorithm and can be used to decimate non-manifold meshes. Soucy and Laurendeau [6] described a more sophisticated decimation algorithm similar to Schroeder's but provided reasonable improvement in terms of efficiency and quality. Garthwaite and Reposa [7] also developed a deliverable computer program that followed Schroeder's method closely. However, they did not deal with the color as well as the texture associated with the meshes when performing decimation and re-triangulation.

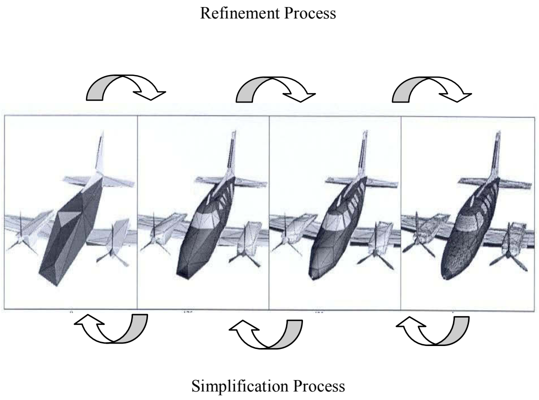
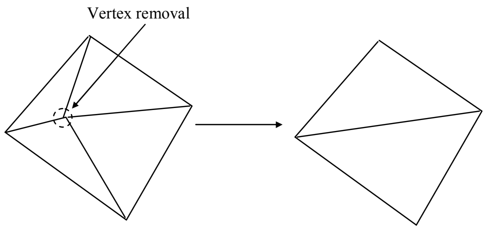


Fig. 2. Algorithms behinds a 3D streaming process.

(2) Iterative edge collapse (contraction)

Iterative edge collapse (contraction) is another promising simplifying approach that preserves volume and other geometric properties better than vertex decimation. Some of the common steps used by iterative edge collapse consist of selection of vertex pairs (either edge or non-edge type), determination of target point placement and reconstruction. First of all, a selection criterion has to be employed to choose the proper pair of vertices for collapse. Generally, it is ideal to iteratively collapse the vertex pair that least influences the overall shape after contraction. As a matter of fact, besides this approach, a vertex pair can be collapsed to the optimal location of the target vertex after contraction. Finally, all the neighboring vertices connecting the chosen vertex pair before collapse should be reconstructed to connect the target vertex. In Fig. 4, it



Number of triangles is reduced from 4 to 2 through vertex decimation

Fig. 3. A vertex decimation process.

is obvious to see that each collapse can reduce two triangles (shaded) and one vertex ($v_1, v_2 \to \bar{v}$).

There are some edge-collapse algorithms [8-12] that have been published. However, the essential difference between them lies only in the way they choose candidature of vertex pairs to contract or determine the target vertex placement after edge collapse. Garland and Herbert [9] introduced a new surface simplification algorithm based on the iterative contraction of vertex pairs. This algorithm provides efficiency (rapid simplification process), quality (high fidelity to the original model) and generality (contracting both connected and unconnected pair of vertices). It chooses qualified vertex pair and maintained surface error approximation based on similar quadric metrics presented by Ronfard and Rossignac [13]. This way can facilitate much better approximations, both visually and with respect to geometric error. Nevertheless, it is based on the assumption that topology is less important than the overall appearance. Garland and Herbert [10] further improved this algorithm to produce high quality approximations of complex polygonal surface models with material properties such as colors, textures, and surface normal. A natural extension of their original quadric error is presented to account for a wide range of vertex attributes.

(3) Vertex clustering

Vertex clustering is another simplification method significantly different from the former two. It was first introduced by Rossignac and Borrel [14] to process arbitrary polygonal input of mesh representation. Regardless of the original shape, it places a bounding box around the original model and divides it into a grid. All the vertices inside a certain cell will be treated as one vertex so that the original mesh model will be simplified cell by cell. This process can be implemented very fast but dramatically alters the model's topology. Moreover, the quality of simplification is hard to control since it depends on the size and number of grid cells, which cannot ensure a good approximation with topological loyalty.

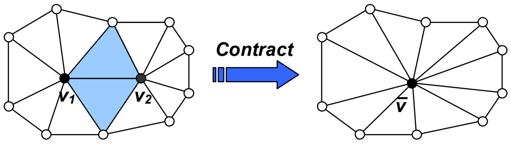


Fig. 4. An edge contraction process.

3.2 Mesh Refinement

In 3D streaming, the simplification process is followed immediately by mesh refinement. As far as we are concerned, there are two refinement methods available. One is represented by the *progressive forest split* (PFS) *scheme* to refine the mesh model from low-detail to high-resolution. However, it does not support the path from the simplified version to its original, which means that it is a smoothing process based on the current level of mesh regardless of whether this refinement can or can not restore the mesh to its original form. The other is represented by the *multi-resolution (or progressive) meshes* scheme, which has strict loyalty to the original mesh by simplifying it into a series of versions with continuous levels of detail from the coarsest simplification to its original.

(1) Progressive forest split (PFS)

PFS, presented by Taubin et al. [15], features an adaptive refinement operation, which arbitrarily adds more vertices and connections within the forest of edges to incrementally make the mesh smoother. Based on the current detail of mesh (generally a manifold triangular mesh with low resolution), the PFS scheme refines the mesh by cutting it through the forest, splitting the resulting boundaries apart, filling each of the resulting free boundary loops with one of the simple polygons, and finally displacing the new vertices. Since the PFS scheme offers a natural extension to the LOD representation, it does not lead to the original mesh but rather to a post-smoothing model based on the level of detail of the particular model.

(2) Multi-resolution meshes

Multi-resolution meshes are similar to the concept of level-of-detail (LOD), which generates a series of versions with different levels of detail of original model to facilitate 3D streaming. It is initially proposed by Funkhouser and Sequin [16] to improve rendering performance by providing an adaptive display in complex visual environment. For a certain mesh model $\hat{M}$, by simplifying it into n-level detail of $\{M_n = \hat{M}, M_{n-1}, \dots, M_1, M_0\}$ along the coarser direction, this multi-resolution structure can be used to render smooth visual transition along the sequence of $M_0 \to M_1 \to \dots \to M_{n-1} \to M_n = \hat{M}$.

Hoppe [8] introduced *Progressive Mesh* (PM) which transforms the original mesh models into simpler base meshes (then render them back to the full resolution mesh) and has the "ability to store compactly and incrementally transmit and reconstruct geometry." In PM form, any original mesh $\hat{M}$ is replaced with a much coarser mesh M_0 together with a sequence of *n* number of detail records that represents how to gradually restore

M_0 to the original model. This is especially an important development for network-based data transmission and rendering. Another attractive advantage of PM is its support of *selective refinement*, which depends on a Boolean function to decide whether the neighboring area around the mesh will be refined or not.

Gueziec et al. [17] proposed a framework for the progressive transmission of LOD geometry. It gives a flexible LOD storage scheme, which is termed *progressive multi-level mesh*. This scheme, which has benefited from a compressed data structure that receives and exploits LODs, requires low memory and provides an easy access to the various LODs (thus suitable for efficient rendering). Moreover, this representation is not tied to any particular polygon reduction algorithm. In fact, it uses the output from any polygon reduction algorithm based on *vertex clustering* (including the edge collapse operations used in several algorithms).

4. CO-DESIGN SYSTEMS AND FEATURE-/ASSEMBLY-BASED REPRESENTATIONS

Comparing to a visualization-based system, a co-design system can enable more active participations of a design team. The organization of a co-design activity can be enabled in two paradigms: synchronous co-modelling/co-modification design, and asynchronous assembly-based co-design. To satisfy the requirements, different system infrastructures are specified.

In a synchronous co-modelling/co-modification paradigm, each user is enabled to participate in design collaboration synchronously with modelling and modification capabilities. During iterative design sessions, changes imposed by a designer can be communicated with other project participants through sharing these changes to be merged with others' concurrent design models. Therefore, suitable coordination and synchronization mechanisms are crucial to schedule a design activity in parallel and ensure no conflict arises during this real-time and iterative design process. Meanwhile, as real-time data sharing, which is an essential requirement to ensure the collaboration, is almost impossible due to the contradiction of huge design models and limited bandwidth of the Internet, a new kind of feature representation are in active exploration.

In an asynchronous assembly-based paradigm, a co-design activity is centrally coordinated in an assembly level. Assembly constraints are encapsulated as interfaces to provide different designers platforms to cooperate, which can ensure sub-assemblies and components allocated to individual designers are compatible with each other. Although real-time sharing is not a must, an optimized representation strategy for assemblies to simplify data to avoid the sluggish transmission is still desired. Meanwhile, a propagation mechanism for changes happened in a sub-assembly or component to the entire assembly structure is imperative to maintain the assembly consistency.

As thus, the following two aspects will be investigated:

- *An effective system architecture based on the available IT infrastructures, such as client/server, peer-to-peer and Web service.*
- *New feature and assembly representations and schemes to optimize data sharing, transmission and management in the distributed environment.*

(1) System architectures

The architectures of the underlying systems to enable co-design can be classified into three types:

- *Communication server + modelling client (Thin server + strong client)*
- *Modelling server + visualized-based manipulation client (Strong server + thin client)*
- *Application or service sharing (Peer-to-peer)*

The major features, characteristics, implementation strategies and comparisons are summarized in Table 1.

Most of the available distributed CAD systems use the first two types of infrastructures Comparing them, the latter is getting more popular since it can bring a new kind of business model – application service provider (ASP). With such infrastructure, small and medium enterprises (SMEs) or even individual designers with specific domain knowledge can rent on-line high-end CAD systems, so they are able to participate and co-operate in the design process with large firms. Meanwhile, the scalability of system can be enhanced since it is convenient to add new seats in the distributed system. However, the implementation difficulty is increased (see Fig. 5).

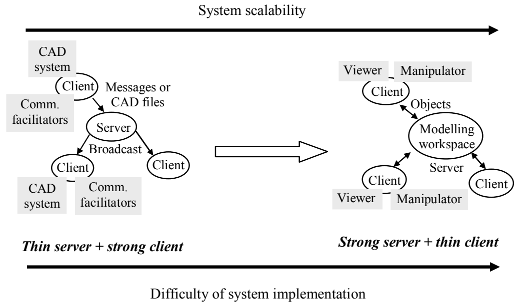
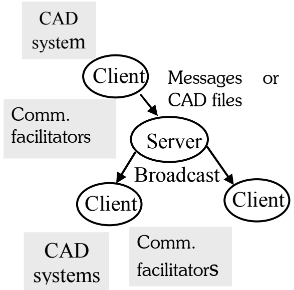
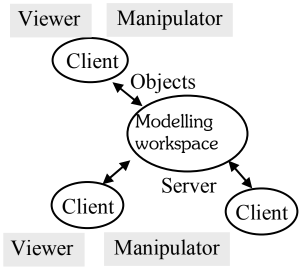
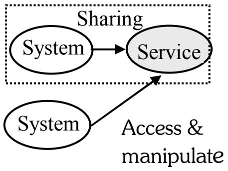


Fig. 5. Comparisons of two infrastructures.

Table 1. System infrastructures of distributed CAD systems

Types of infrastructures	**Functions**	**Characteristics**	**Diagrams**
Thin server + strong client	• Clients are equipped with whole CAD systems and some communication facilitators. • A server plays as an information agent and exchanger to broadcast CAD files and commands generated by a client to other clients.	• Standalone CAD systems can be conveniently distributed through this mechanism. • Due to the heavy-weighed client mechanism, it is hard to be migrated to web applications.	
Strong server + thin client	• The data structures in clients are light-weighed and they primarily support visualisation and manipulation functions. • The main modelling activities are carried out in a common workspace in the server side.	• Modules can be rent out as an Application Service Provider (ASP). • Data consistency is easily kept since the primary models are created and maintained in the server.	
Peer-to-peer	• The services or modules of a system can be shared and manipulated by other systems. • For the Inventor collaborative tool, an MS Netmeeting tool is embedded for application sharing.	• This mechanism enables a convenient manipulation on remote services or applications. • Due to the heavy burden of networks, the manipulation efficiency of systems is low.	

(2) Feature- and assembly-based representations

A significant problem for the above systems is that communication efficiencies are still quite far from satisfactory when large-size feature- and assembly-based models are designed collaboratively. In order to address this problem, some works have been appeared recently to optimize or simplify geometric entities of distributed design feature- or assembly feature-based models to accelerate the communication.

Wu and Sarma [18] developed an algorithm to incrementally update the B-Rep of a design model based on a cellular representation in a distributed environment. Based on the cells from the segmented B-Rep of a design model, the algorithm can identify and extract those regions that have been modified by a designer, and dynamically transmit and embed the modified regions into a B-Rep at another site. Lee [19] proposed a network-centric virtual prototyping system in a distributed computing architecture, in which a shape abstracting mechanism was developed to provide a light-weight Abstracted Attributed B-rep (AAB) in clients to represent a feature-based model stored and maintained in a server for concise and transparent communication between the server and the clients over the network. A naming consistency paradigm was established to maintain the interoperability and identification between geometric entities of the server and the clients during a concurrent design process. Li et al. [30] developed a distributed feature mechanism to filter the varied information of a working part during a co-design activity to avoid unnecessary re-transferring of the complete large-size CAD files each time when any interactive operation is imposed on the model by a client, so as to enhance the effectiveness of the information communication for co-design activities.

In order to support collaborative assembly design activities effectively, Shyamsundar and Gadh [20] developed a new geometric representation named as AREP and a collaborative prototyping system based on the representation to perform real-time geometric modification for components/sub-assemblies in an assembly model. In AREP, an envelope mechanism was designed to simplify the some internal geometric structures and entities, which are irrelevant to assembly constraints, of components designed separately and collaborated around the assembly constraints. Points are kept in envelopes to refer to corresponding detailed entities for further query and retrieve. Chen et al. [21] proposed an assembly representation for collaborative design. Their functional modules include a Master Assembly Model (MAM) and a Slave Assembly Model (SAM). The MAM is a complete representation stored in the server, and SAM is a simplified version of MAM used for visualization-based manipulation in the client. However, it does not address the real-time design modification in a collaborative design environment. The research direction is towards supporting optimized traffic and real-time feature and assembly design. Kim et al. [22] proposed a design formalism to capture the non-geometric aspects of designer's intent on assembly joining process in a co-assembly design environment, the joining relations can be used for inferring mathematical and physical implications, and an assembly design model are used for some assembly design activities, such as joining analysis, process planning and so on.

In order to support feature-based applications to cross application domains (e.g., from design to manufacturing) in a collaborative design environment, Gadh and Sonthi [23] developed a four-level representation scheme for features to address different applications effectively. The representation consists of boundary representation, aggregate geometric abstraction representation, domain independent geometric abstraction representation and domain dependent features. The motivation of this representation is to provide several layers of geometric abstractions and aggregations in a server to response to different manufacturing applications efficiently. Han and Requicha [24] and De Martino et al. [25] separately developed a distributed system consisting of a design-by-feature client and a downstream manufacturing feature recognition client connected by a geometric server. The functions of the geometric server are twofold: first, it is a repository to store features generated by these two clients; second, it transfers design features in the design-by-feature client to the feature recognition client. The distinction of these two works is in their feature recognition algorithms. The former used a hint-based reasoning method depending upon the augmented design features as hints, whilst the latter developed a graph-based reasoning method to work on the geometric models converted from the design feature models. In the above works, changes made in the design-by-feature client can be propagated to the feature recognition client automatically to achieve data completeness and consistency. However, this information flow is unidirectional. If a modification of a design part is required by the manufacturing feature recognition client, it should be made in the design-by-feature client, which process forces a user to think in a way that is not natural for him or her and blurs the functional differences among design and manufacturing. Hoffmann and Joan-Arinyo [26] proposed a master model scenario to store shared design information and a multi-way communication mechanism among design and manufacturing clients. However, the features supported in this work are still limited to some simple types and the work is still far from practical applications. This problem can be effectively solved through developing a generic and robust integration strategy of design-by-feature and feature recognition algorithms to support multiple views of a design model, which is actively investigated [27-29].

5. WORKS DONE IN THE "DCAD" PROJECT

From 2002, a collaborative project – development of key technologies for supporting collaborative CAD systems (in short, DCAD), has been carried out between Singapore Institute of Manufacturing Technology and National University of Singapore. Some prototype systems and methodologies for collaborative CAD have been developed. The related works developed in the DCAD project are briefly reported here.

(1) Feature-based co-design system [30]

A client/server framework has been developed to enable a dispersed team to accomplish a feature-based design task collaboratively. A manipulation client + modelling server infrastructure has been proposed to facilitate consistent primary information modelling for multiple users and adaptability of the system. Based on feature-to-feature relationships, a distributed feature manipulation mechanism has been proposed to filter the varied information of a working part during a co-design activity to avoid unnecessary re-transferring of the complete large-size CAD files each time when any interactive operation is imposed on the model by a client. In the distributed environment, a design task and the engaged clients are organized and connected through working sessions generated and maintained dynamically with a collaborative server.

For the manipulation client + modelling server strategy, two representations, “light” on the client side and “heavy” on the server side respectively, have been proposed to fulfil the functional requirements and enhance the performance of the system. A “light” face-based representation is established on the client side to support the interactive visualization and manipulation functions (selection, transformation and changing visualization properties of displayed parts). On the server side, a “heavy” representation with features and part information is set up and maintained to provide primary feature-based modelling functions. Such strategy can reduce the weight of clients to optimize the overall performance of the system.

(2) Web-based manufacturing optimization service [31]

A process planning module, which can optimize the selection of machining resources, determination of set-up plans and sequencing of machining operations to achieve optimized process plans, has been wrapped as services and deployed in the Internet to support distributed design and manufacturing analysis. The module includes four intelligent approaches – genetic algorithm (GA), simulated annealing (SA), tabu search and hybrid GA-SA. A Web-based prototype system has been setup for users to carry out visualization-based manipulations and process planning of design models by invoking the services remotely. The Web-based system has been integrated with a distributed feature-based design system, and the latter can generate design models and re-represent them in an XML representation based on VRML and attributes of features to provide the input of the former.

(3) Integration of co-design and process planning optimization service [32]

The process planning optimization service and co-design are complementary in functions since the former emphases a vertically seamless linkage between the upstream design and the downstream manufacturing processes through the creation of intelligent strategies for evaluating and optimizing the manufacturing process, while the latter focuses more on the horizontally interpersonal aspects of group work in the upstream design phases. With the trend for global competition and the rapid advances of the Internet technologies, both of them are moving towards supporting distributed applications, in which geographically dispersed users, systems and resources can be integrated in an Internet/Intranet environment beyond the traditional boundaries of physical and time zones. An Internet-enabled system has been developed to support collaborative and concurrent engineering design through seamlessly integrating three functional modules, i.e., co-design, Web-based visualization and manufacturing optimization, based on Java and Web technologies. In the co-design module, designers are equipped with co-modelling and co-modification facilities to carry out a design task collaboratively. The Web-based visualization module provides a portal for users, who are not involved in the co-modelling process directly, to view and analyze a design part conveniently. Services in the manufacturing analysis module can be invoked by users dynamically to evaluate and optimize the manufacturing costs and the manufacturability of a design part so as to implement the concurrent engineering methodology during a co-design process.

(4) Assembly model to support co-design [33]

In order to support real-time design modification in a collaborative assembly (co-assembly) environment, an assembly representation model, viz. feature-based hierarchical co-assembly representation, has been proposed and a new definition of the assembly feature has been given to resolve the co-assembly design issue. In order to realize real-time design modification propagation control, an XML schema has been developed to transfer the assembly design information by defining each feature using the XML format based on the co-assembly representation proposed.

6. REFERENCES

[1] CIMData Inc., http://www.cimdata.com/

[2] M2 Resaerch, http://www.m2research.com/

[3] Li WD, Fuh JYH and Wong YS, 2004, Collaborative computer-aided design – research and development trend. Proceedings of CAD’04 conference, May 24-29, Pattaya, Thailand.

[4] Schroeder WJ, Zarge JA and Lorensen E, 1992, Decimation of triangle meshes. Computer Graphics (Proceedings of SIGGRAPH '92), 26(2), pp 65-70.

[5] Franc M, 2002, Methods for polygonal mesh simplification. Technical Report No. DCSE/TR-2002-01, University of West Bohemia in Pilsen. [Online available]: http://herakles.zcu.cz/~marty/html/rig/mpms.html

[6] Soucy M and Laurendeau D, 1996, Multiresolution surface modeling based on hierarchical triangulation. Computer Vision and Image Understanding. 63(1), pp 1-14.

[7] Garthwaite T and Reposa J, 1999, Mesh decimation. Major Qualifying Project Report, MOW-2733, Worcester Polytechnic Institute.

[8] Hoppe H, 1996, Progressive meshes. Proceedings of SIGGRAPH’96, pp.99-108. [Online available]: http://www.research.mircosoft.com/research/graphics /hoppe

[9] Garland M and Heckbert PS, 1997, Surface simplification using quadric error metrics. Proceedings of SIGGRAPH'97, pp. 209-216. [Online available]: http://www.cs.cmu.edu/~garland/quad rics/

[10] Garland M and Heckbert PS, 1998, Simplifying surfaces with color and texture using quadric error metrics. Proceedings of IEEE Visualization'98 Proceedings, pp. 263-269.

[11] Hoppe H, DeRose T, Duchamp T, McDonald J and Stuetzle W, 1993, Mesh optimization. Proceedings of SIGGRAPH'93, pp. 19-26. [Online available]: http://www.research.mircosoft.com/research/graphics/hoppe

[12] Melax S, 1998, A simple, fast and effective polygon reduction algorithm. Game Developer Magazine, November 1998, pp. 44-49. [Online available]: http://www.melax.com/polychop /gdmag. pdf

[13] Ronfard R and Rossignac J, 1996, Full-range approximation of triangulated polyhedra. Computer Graphics Forum (Proceedings of Eurographics'96).

[14] Rossignac J and Borrel P, 1993, Multi-resolution 3D approximation for rendering complex scenes. Modeling in Computer Graphics: Methods and Applications, pp. 455-465.

[15] Taubin G, Gueziec A, Horn W and Lazarus F, 1998, Progressive forest split compression. Computer Graphics (Proceedings of SIGGRAPH'98), pp. 123-132.

[16] Funkhouser T and Sequin C, 1993, Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. Computer Graphics (Proceedings of SIGGRAPH'93), 27, pp. 247-254.

[17] Gueziec A, Taubin G, Horn B and Lazarus F, 1999, A framework for streaming geometry in VRML. IEEE Computer Graphics & Applications, Special issue on VRML, 19(2).

[18] Wu D and Sarm R, 2001, Dynamic segmentation and incremental editing of boundary representations in a collaborative design environment. Proceedings of 6th ACM Solid Modeling'01, Michigan, USA, pp.289-300.

[19] Gadh R and Sonthi R, 1998, Geometric shape abstractions for internet-based virtual prototyping. Computer-Aided Design, 30, pp.473-486.

[20] Shyamsundar N and Gadh R, 2002, Collaborative virtual prototyping of product assemblies over the Internet. Computer-Aided Design, 34, pp.755-768.

[21] Chen L, Song ZJ and Feng L, 2004, Internet-based real-time collaborative assembly modeling via an e-assembly system: status and promise. Computer-Aided Design, 36, pp. 835-847.

[22] Kim KY, et al., 2004, Design formalism for collaborative assembly design. Computer-aided Design (in press).

[23] Gadh R and Sonthi R, 1998, Geometric shape abstractions for internet-based virtual prototyping. Computer-Aided Design, 30, pp.473-486.

[24] Han J and Requicha AAG, 1998, Modeller independent feature recognition in a distributed environment. Computer-Aided Design, 30, pp.453-463.

[25] De Martino T, Falcidieno B and Habinger S, 1998, Design and engineering process integration through a multiple view intermediate modeller in a distributed object-oriented system environment. Computer-Aided Design, 30, pp.437-452.

[26] Hoffmann CM and Joan-Arinyo R, 2000, Distributed maintenance of multiple product views. Computer-Aided Design, 32, pp.421-431.

[27] De Kraker KJ, Dohmen M and Bronsvoort WF, 1997, Maintaining multiple views in feature modelling. Proc. Solid Modelling'95, New York, USA, pp.123-130.

[28] Jha K and Gurumoorthy B, 2000, Multiple feature interpretations across domains. Computers in Industry, 42, pp.421-431.

[29] Li WD, Ong SK and Nee AYC, 2002, Recognizing manufacturing features from a design-by-feature model. Computer-Aided Design, 34, pp. 849-868.

[30] Li WD, Ong SK, Fuh JYH, Wong YS, Lu YQ and Nee AYC, 2004, Feature-based design in a collaborative and distributed environment. Computer-Aided Design, 36(9), pp. 775-797.

[31] Li WD, Ong SK and Nee AYC, 2004, A Web-based process planning optimization system for distributed design. Proceedings of CAD'04 conference, May 24-29, Pattaya, Thailand.

[32] Li WD, Fuh JYH and Wong YS, 2004, An Internet-enabled integrated system for collaborative design and concurrent engineering. Computers in Industry, (in press).

[33] Lu C, Fuh JYH, Wong YS, Li WD and Lu YQ, 2004, Assembly design for in a collaborative CAD environment. Proceedings of CAD'04 conference, May 24-29, Pattaya, Thailand.