A Framework for Distributed Collaborative Engineering on Grids

Liqing Fan, Huabing Zhu, Shung Hwee Bok and A. Senthil Kumar

[1]National University of Singapore, mpeflq@nus.edu.sg
National University of Singapore, engboksh@nus.edu.sg
National University of Singapore, mpeask@nus.edu.sg

ABSTRACT

Distributed collaborative design and manufacture enables manufacturing organizations in maintaining competitiveness in the fiercely competitive global industry. This requires the distributed system not only maintains data consistency across globally distributed location seamlessly, but also allows team members to access storage system and computing resources transparently and securely. In this paper, a framework for distributed collaborative engineering developed on the Grid platform is introduced. The system architecture is presented for this framework. This paper also describes data Grid, computational Grid and data compression to support collaboration efficiently.

Keywords: Collaborative Design, Product Design, Infrastructure, Grid Computing.

1. INTRODUCTION

The advent of dynamic markets, customer demands and product development competition point towards a need for lower cost, shorter product lead times in the fiercely competitive global industry. To stay competitive, manufacturers must be able to 1) manage increasing product complexity and product innovation driven by market demands, 2) have faster and flexible product development cycle, and 3) control globally distributed outsourcing operations. Distributed Collaborating Design and manufacture enables manufacturing organizations in maintaining competitiveness by leveraging the Internet and operating seamlessly with heterogeneous systems, tools and data. Therefore, extensive research and development works on distributed collaborative CAD systems have been carried out [1][2][3][4][5].

However, most of these works are based on current Internet infrastructure which is designed for information transmission and publication. A collaborative environment for engineering design & manufacturing requires the infrastructure much more than that. For example, designers and engineers in a team group at geographic distributed sites need to share and transfer data to accomplish a common product. This requires a managed system that can maintain data consistency and transport data among globally distributed sites to support collaboration seamlessly. Each site permits secure remote access to computers and storage systems. Moreover, in the iterative, multidisciplinary process of engineering design, simulation and analysis play more and more important roles. They are deployed to evaluate and verify the design suitability, to predict the design performance, to reduce trial-and-error cost, to decrease the design process cycle, and even to optimize the design. However, many simulation and analysis processes, e.g. finite-element method (FEM) and computer fluid dynamic (CFD), are data intensive, computationally expensive and consume huge computing resources. This requires the infrastructure should allow users in the team group to access remote computing resources to process simulation jobs transparently. Recent advancement of Grid computing technologies has provided the incentive for a new solution to integrate simulation and optimization in engineering design. The term “Grid" [6] was coined in the mid of 1990s and its primary target is to enable large-scale, dynamic collaboration among participants from highly heterogeneous and distributed computing environments. It is one such paradigm that proposes aggregating geographically distributed, heterogeneous computing, storage and network resources to for Grids that provide unified, secure and pervasive access to the combined capabilities of the aforementioned resources.

2. RELATED RESEARCH

2.1 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 Inter-disciplinary

Design) [8]. Various distributed collaborative applications have been developed for different engineering areas. The architectures of these systems can be generally grouped into two categories, i.e. Client/Server and Peer-to-Peer.

There are two approaches in the Client/Server systems. One puts a whole geometry kernel in each client. A server plays as an information agent and exchanger to broadcast CAD model and commands generated by one client to other clients [12][13][14][15]. Stand CAD systems can be conveniently distributed through this mechanism. However due to the heavy-weighted client mechanism, it is hard to migrated to web application. The models stored in clients are light-weighted visualization model only. The geometry kernel is in the server side [5][16]. The commands are passed from client to server side to carry out the modeling activities. In this way, data consistency is easily kept since the primary models are created and maintained in the server.

The Peer-to-Peer collaborative design systems [17] provide some services or modules which can be shared and manipulated by other collaborative engineering applications. Due to the heavy burden of networks, the manipulation efficiency of system is low. The approaches of the collaborative design systems fall into two domains: web-based approach and agent-based approach [8]. In the some research works, the Web was only be used to monitor the design process and to check the status of the working system [18]. Some other researchers use Web as a medium to share data, information and knowledge. 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 [15][19][20][21]. Some applications even use the advantage of Web for product data management and project management [22].

In agent-based collaborative design systems, agents have mostly been used for supporting co-operation among designers, providing semantic glue between traditional tools, or for allowing better simulations [8]. Unlike the Web-based design systems using the client/server architecture, most agent-based system architectures fall into the third architecture category. It is a loosely coupled network of problem solvers. Various agent-based projects have been carried out. Each focuses on some aspects of collaboration. PACK [23] might be one of the earliest successful projects in this area. IDIE [18] uses autonomous approach to study system openness, legacy problem integration, and geographically distributed collaboration. Fruchter et al. [24] put their efforts on building a shared graphical modeling environment for collaborative design. SiFAs [25] focuses on the issues of patterns of interaction, communication, and conflict resolution using single function agents.

From the previous review, it can be found that most collaborative systems are based on current Internet technology. Therefore, the collaboration is still on the level of information sharing and the systems have shortcomings in either

- not providing a problem-solving environment for product process development;
- lack of unified access to data/resource management mechanism;
- tightly coupled with network location/address; or
- lack of user/node discovery middleware.

In this paper, a collaborative engineering environment will be introduced which is build on Computational Grid [6] and Data Grid [7] as middleware to support distributed collaborative design and manufacturing. In this environment, the data Grid component is used to control the data access and flow for collaboration. The computational Grid component supports the CAE simulation and analysis. This environment provides services for application plug-in.

3. SYSTEM OVERVIEW

From information sharing to resource discovery and sharing, our research of distributed collaboration focuses on the infrastructure of the collaborative environment. Figure 1 shows the scenario of the environment.

From this figure, it can be found that this system has a hierarchical architecture. On the top level, the system is built on peer-to-peer style. There are multiple servers in the environment. They share the modeling information. The data stored in these servers should keep consistency. On the local level, the environment is built on modeling server + visualized manipulation client (strong server + thin client) style.

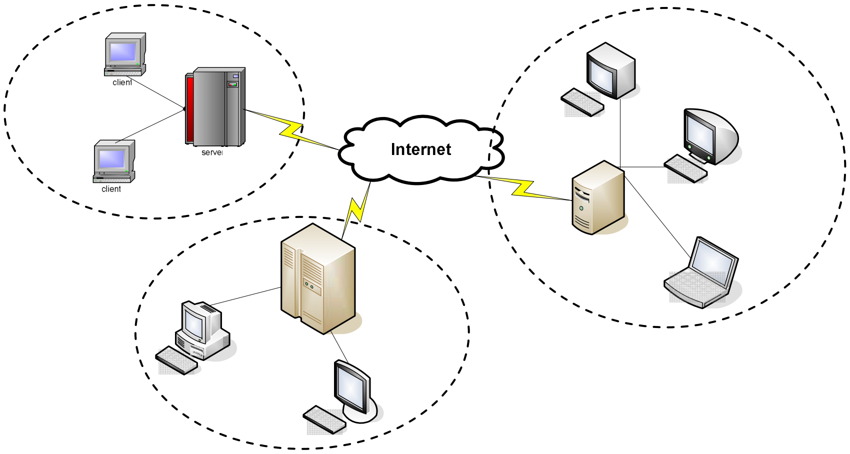

Fig. 1: Environment Scenario.

A system with peer-to-peer architecture style has some strong points which benefit the collaborative design environment [27]. Firstly, peer-to-peer data grid is flexible to extend to future users. When more users join the collaborative environment, it will not significantly impact the performance of the data grid systems. Secondly, peer-to-peer systems are highly fault tolerant with respect data routing. Thirdly, peer-to-peer systems typical integrate heterogeneous nodes.

In a collaborative manufacturing environment, various design changes occur and it is inefficient to manufacturing systems would need to obtain the entire CAD model each time a design change occurs to co-design or analyze the part for manufacturability. To overcome this inefficiency, standard CAD models are located at central databases. A visualization model is generated from the standard CAD model and sent to the client for manipulation.

Grid infrastructure is the base for the realization of collaborative design & manufacturing grid. There are some distinguished grid platform developed, but they are design for computing-intensive or data-intensive applications, and cannot used directly for collaborative design & manufacturing. Fig. 2 shows the software architecture of the environment which can be divided into four layers: resource layer, grid middleware layer, service layer and application layer. Resource layer includes all kinds of distributed resources such as, super computers, clusters, network resources, and distributed data storages.

Grid middleware layer overlaps the gap among different distributed resources to make the distributed system transparent. Computational Grid provides the infrastructure to support high performance computing over distributed resources. Data Grid provides the infrastructure to support remote data transparent access and meta-data index in a distributed environment. Grid security infrastructure provides single sign on (SSO) mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords. In grid environment, a virtual organization (VO) consists of multiple distributed and heterogeneous individuals/organizations providing resources/services, each of which must confront diversities of access technologies among its participants. VO management and grid security service are import facilities for distributed collaboration, because free and open communications need ability to know the participants' identifications. Moreover, they are providing the secure collaboration environment to support informal, spontaneous and formal collaborations.

Service layer provides a set of services to support varying collaborative applications in the environment, e.g. co-design modeling system, simulation based design systems and so on. The services include a CAE simulation service, a solid modeling kernel, model compression package, basic collaboration support tools and project management package. The simulation kernel is a finite element method (FEM) module which can be used to evaluate and verify design. The solid modeling kernel is a boundary representation (B-rep) modeling module which is developed on the CAD software development platform. A B-rep model can not be rendered and visualized directly by graphics hardware. Therefore, the solid modelling kernel can tessellate the B-rep solid model and convert it to triangle surface model. For efficient transmission of the triangle model, a model compression package is developed, which is a loss-less connectivity compression scheme.

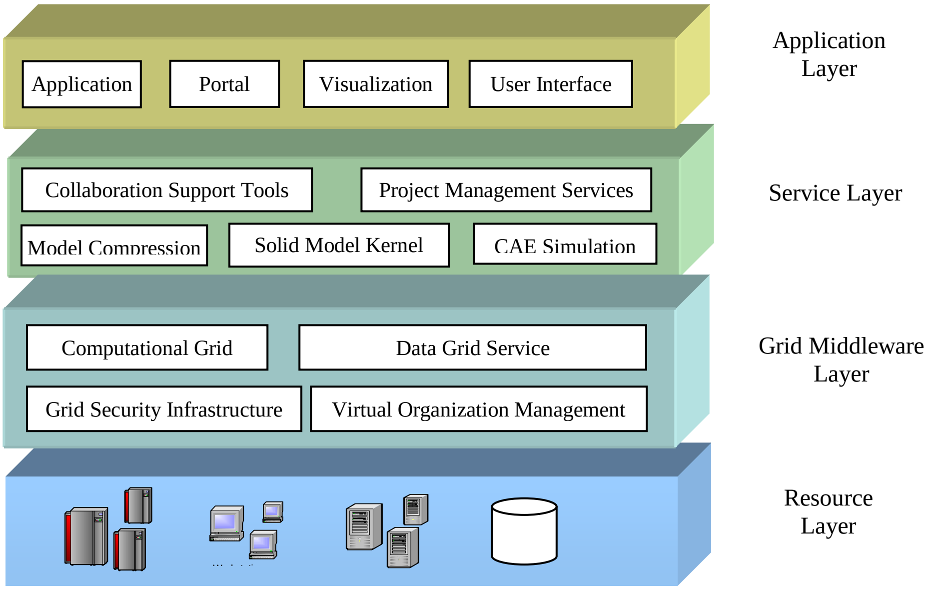

Fig. 2: Layered Software Architecture.

Collaboration support tools, including information communication means, real-time interactive tools, and threaded discussion services, etc., support the coordinated exchange of information within potentially large user communities synchronously or asynchronously. Project management consists of user management, file management, workflow management etc. It is implemented with grid VO management function as its core component, with which information about wide distributed organizations/individuals is organized into hierarchical structure of groups, and the structure can act as background index for large-scale collaboration.

Base on grid middleware layer and service layer, application layer provides good interactive interface and convenient usage to access service and grid resources for users, and allows users to display results in graphics.

4. SYSTEM DEVELOPEMENT

In this section, the system components developed for the distributed collaborative system are described. Due to limitation of pages, here only data Grid, computational Grid, and model compression package are discussed.

4.1 Data Grid

Data Grid component provides transparent access and synchronization services for the distributed data. Therefore, the user or developer does not have to take care of the physical distribution and replication of the data on the Grid. They can access and update the data through the logical file name. The environment will help user to update the replications over the data grid and keep the consistency. A peer-to-peer data grid framework is implemented with JXTA [29].

JXTA is a modular network-programming platform for building distributed services and applications. It is a three-layer architecture containing six XML-based protocols (discovery, membership, routing, etc.) and abstractions for several components (peer groups, pipes, advertisements, etc.). Since JXTA provides protocols instead of APIs, it can be implemented in any language on any operating system. JXTA defines low-level protocols that are more suitable for facilitating lightweight communication among devices rather than for creating a service based network with a large footprint and very large network traffic.

Figure 4 shows the design detail of the data grid component. A client daemon residing at client application maintains session with server and other resources in network, and takes care of discovery and transport. It periodically synchronizes with server, identifies the location of other resources in the local network and synchronized with peers.

Data Grid services in the central server at each site not only maintain user information, job sessions and data transactions, but also discover the resources available for the participants and rules for accessing and using the resources. The server at each site communicates with servers at other sites or organizations using JXTA technology and acts as a router through which clients inside the local network could access resources in other sites.
Based on the concept of VO[6], different organizations come together to share resources and collaborate in order to achieve a common goal. A VO defines the resources available for the participants and provides secure and transparent access to common resources.

In a VO, if there are any files to be shared with users of other participated sites, the index will update at all participating servers, which synchronize periodically with each other to keep consistency. The shared files are proactively sent to the server of other sites in a VO and keep there in cache for fast retrieval. A VO is created using Certificate Authorities (CAs) mechanism for security.

4.1.1 Data Indexing

Central server maintains the database of each data file added to the project. A data file can be shared with few or all members/resources. A new data file can be submitted for sharing either interactively or using a request template. This file can be shared with some or all members of the project. File once submitted can not be modified, but may be removed using the interface provided. The index of files will be maintained in a local database, and will be synchronized with central database.

Moreover, a list of users to be notified should be tagged with the file. The user notification (alert) can be processed actively for already logged-on users and passively for disconnected users. It might also be desirable to use *Look-ahead synchronization,* and push the files actively to users, who might need the file in near future. This will reduce the turn around times to distribute the files judiciously.

4.1.2 Data Categorization

Any new data file will be categorized into long-term file set or transient files set. A transient file set is a temporary file set which has a short life-span and can be created/ destroyed as required. Such file sets are typically the file transfers with compute clusters. Long-term file sets are generally files which members of projects share with each other.

4.1.3 Data Transporting

The data is transported to its required destination directly using either pushing or pulling mechanism depending on network reachability. During data transport, any intermediate resource might be utilized to complete the network path. The data can also be available on multiple resources, and intelligence has to be incorporated to choose more appropriate resource for data retrieval.

Once data is available on more than one resource, it can be categorized as data source for other members, thus providing better transport speed and redundancy. Whenever the user searches for a file based on some known criteria (filename, owner name, release number etc), the middleware should be able to retrieve indexes from all connected resources, and provide the URI of the file to the requesting user.

After the file or file set is completely downloaded, it should be possible to initiate an action, like “start the simulation process” or “display an alert message”. Every resource maintains a list of actions in local and central database, and only those actions can be acted upon for security reasons.

4.1.4 Data Security

Every transaction with the central server should be encrypted. The public key of central server is distributed along every installation, and every resources tells server about its public key on the first execution. Similarly, every file transfer should be encrypted with the public key of the requestor. The central database stores the public key of all resources. The data will be decrypted by the recipient upon download completion.

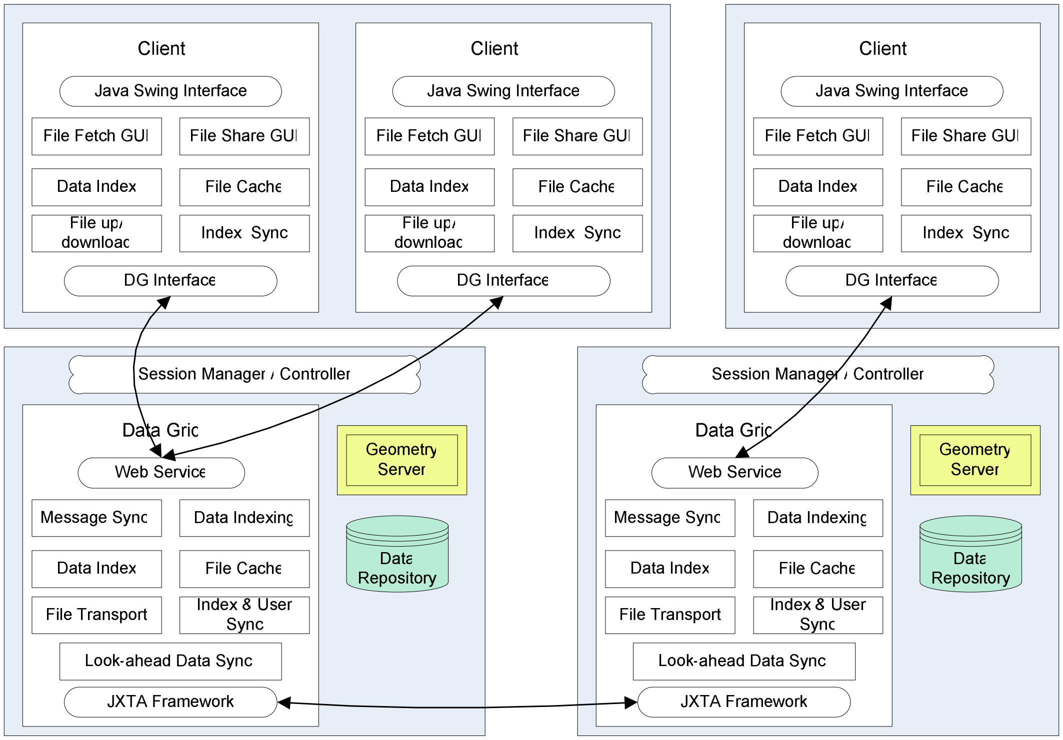

Fig. 4: Data Grid Component.

4.2 Computational Grid

The computational grid is primarily concerned with job management and analysis realization within large-scale pooling of computational resources. The job management provided in computational grid is related to resource discovery and allocation, resource monitoring, task scheduling, etc.

JXTA supports server-less decentralized discovery of resources, including clusters, data storages, and so on. It enables ad-hoc self-organization of computers. Ganglia is used to collect resource information inside each node of cluster and provides resource information to meta scheduler. Meta scheduler manages all locale schedulers, Sun Grid Engine (SGE) [30], running at clusters behind the firewall.

4.2.1 Analysis Realization

The design verification and evaluation job is computational intensive. It usually takes long time to complete it by single CPU. In order to improve performance, parallel processing has been adopted in this project. Design verification usually includes three steps: pre-processing, Finite Element Method (FEM) solving and post-processing.

In the pre-processing, the boundary condition is defined for the Finite Element Analysis (FEA). To archive load-balancing among different processors, Metis is used to evenly partition the job into small sub-jobs. Metis is a library for partitioning unstructured graphs and hypergraphs and computing fill-reducing orderings of sparse matrices. The algorithms implemented in Metis are based on the multilevel graph partitioning schemes described in [28].

After partitioning, the job is submitted to the computing cluster through the meta scheduler and SGE to do the simulation. SGE allocates the resources for the tasks. FEM solver is running on each computing nodes in the cluster. The sub-jobs are computed by the solvers. The simulation results will return to user to do the post-processing. In the post-processing, the data from the solvers will be rendered into a 3D animation to show FEA simulation result.

4.3 Model Compression

The primary objective of this research is to develop an engineering model compression scheme as a seamlessly integrated capability in a Grid-based distributed collaborative design environment. It has linear time complexity in the number of vertices used to represent the geometrical models. It can support incremental design changes.
Edgebreaker [31] algorithm is adopted in this model compression scheme. Model compression is usually achieved by reducing repeated references to vertices that are shared by many polygons/triangles. Generally, connectivity compression involves 2 phases: initialization process and compression process. Initialization process formats the geometry data and marches throughout the bounding edges and vertices to establish the connectivity between all the labeled triangular meshes.

Compression process is a recursive procedure that traverses the mesh along a spiraling triangle-spanning-tree and encodes the vertices and connectivity to generate compressed model.
Model decompression is roughly a reverse process of the model compression. It includes decompression process and post process. The decompression process reconstructs the mesh from the input streams. The post process recovers the holes and converts data formats. Figure 5 shows the compression and decompression procedure.

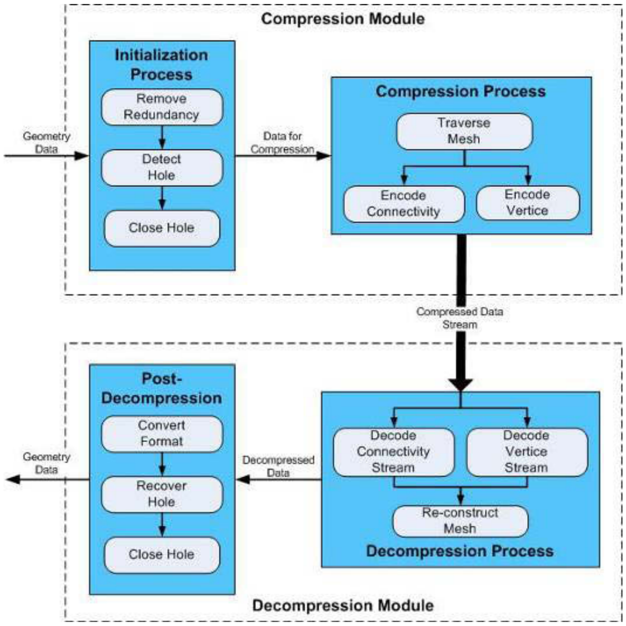

Fig. 5: Model Compression.

5. CASE STUDY

In this section, a mechanical fixture design case will be introduced to reveal how the environment infrastructure can be used to solve the collaborative engineering problem.

A key challenge is that of achieving distributed collaborative design and manufacturing capabilities that leverage the Internet and operate seamlessly with heterogeneous systems, tools and data. One key design and tooling i.e. manufacturing process is fixture design. Fixtures serve the purpose of holding a work-piece securely in position and orientation to ensure that successful manufacturing e.g. machining of tools necessary to the product's fabrication.

In fixture design, there are two necessary steps, i.e. Design Synthesis and Design Analysis. Fixture design synthesis determines the areas on a work-piece for locating, supporting and clamping; and their associated fixture elements; and thus the spatial location and configuration of these fixture elements onto a fixture body. Efforts in fixture design analysis aim to arrive at accurate mathematical models to verify and evaluate the suitability of the fixture design configuration during the entire machining process in terms of stability, total restraint, accessibility and deformation.

In this section, a case study involving three companies that focuses on product design, manufacturing and analysis is illustrated to present the benefit of the system architecture discussed above. In such scenario, four relevant roles and application views shown in Fig. 6 are involved: a). Part designer in company A who develops CAD solid models for products; b). Fixture designer in company B who carries out a tooling process to plan for a manufacturing process, i.e. machining process; c). Tool path planner in company B who generates cutting tool path for a machining process; d). Fixture analyzer in company C who analyzes the fixture design for manufacturability assessment. For the particular project, the four roles at different company collaborate with each other to form a VO.

In this system, the workflow is created first to indicate control flow and data flow. With the aid of workflow, the part designer in company A logs in the system and creates the CAD solid model named “Part1” (Fig. 7) and deposits it into Data Grid. The company B manufactures “Part1” with casting and machining features. In current setup, a milling process will be executed at the top of the part. Based on machining information, the fixture designer and the CAM programmer will load the model in their applications and cooperate with each other to work towards fixture design configuration (Fig. 8) as well as cutting tool path. The fixture design and cutting tool path file associated with “Part1” is handled by Data Grid and established relationships with Part1. The analyzer in company C firstly prepares for manufacturing assessments based on fixture design and cutting tool path provided by company B, then execute and monitor the fixture analysis job(s) on the Engineering Grid, and interpret and assess the results of the analysis job(s). The analysis result is shown in 3D in Fig 9 in which the small block at the bottom represents a support positioning on baseplate and holding the reaction force as well as friction force from the workpiece. From Fig. 9, a gap is found between the workpiece and the block representing support. That means that the workpiece is lifted off from the support with exerting the external machining force and clamping forces. This will result to machining error and failure.

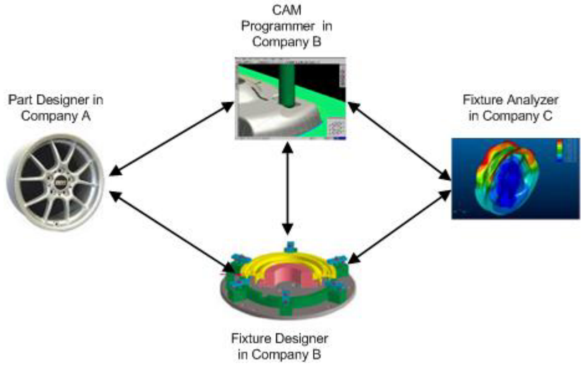

Fig. 6: Distributed collaborative application view.

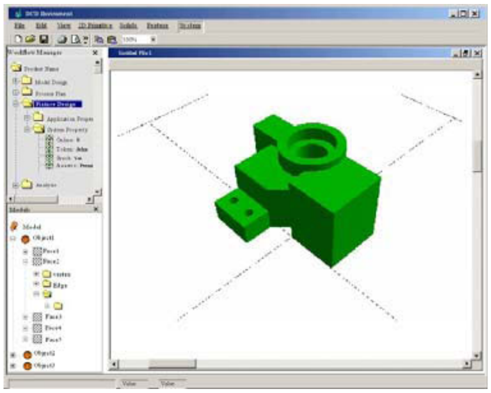

Fig. 7: The designed part to be deposited into data grid.

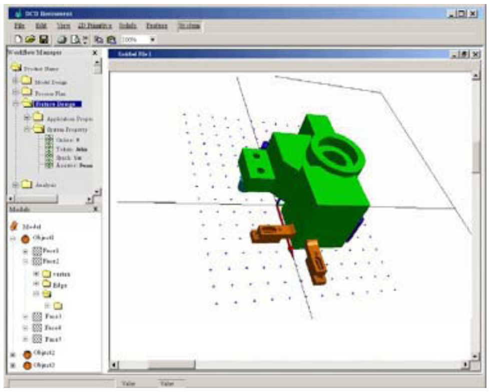

Fig. 8: The initial fixture design.

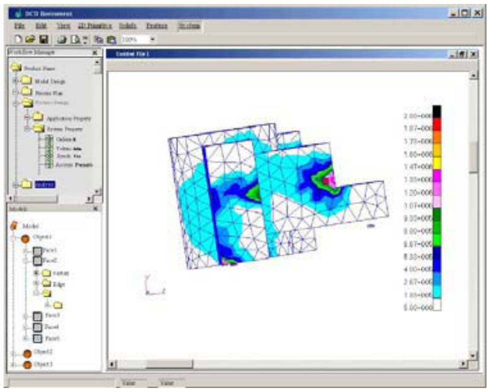

Fig. 9: Analysis result for initial fixture design.

Based on the fixture analysis results, the fixture designer at company B modifies the exiting fixture configuration by adding a side clamp on the workpiece (Fig. 10). The modified fixture design model is passed to company B via data Grid, and the analysis job for the fixture design is altered by change boundary condition based the modified model and re-submitted into the computational Grid for simulation. The new analysis result shown in Fig. 11 is interpreted and considered the modified model (Fig. 10) as a feasible fixture design configuration.

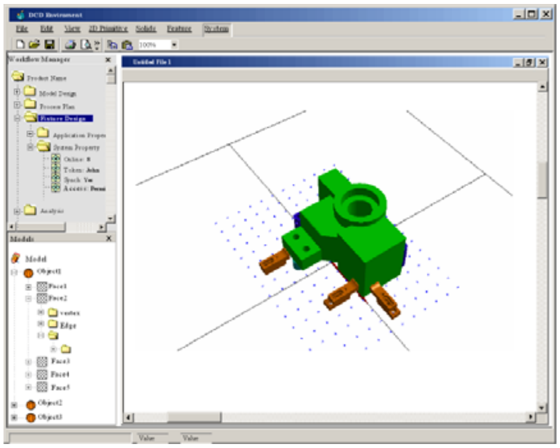

Fig. 10: The Modified Fixture Design.

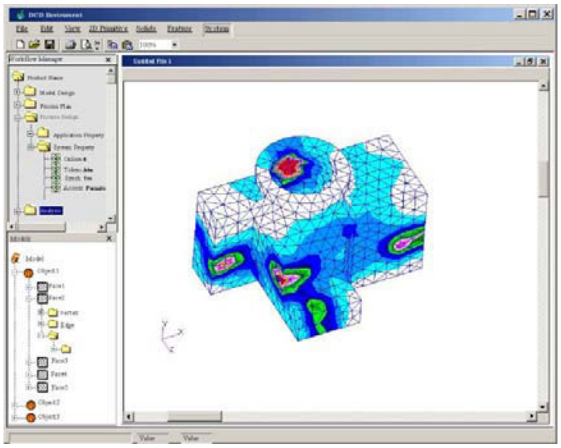

Fig. 11: Fixture Analysis Result for the Modified Fixture Design.

6. CONCLUSION

In this paper, a system framework is proposed for developing distributed collaborative design/manufacturing applications in the environment of the Grid in order to reduce product and management cost and to increase competency in the global markets.

The distributed data can be accessed transparently and be transferred efficiently via Data Grid components. Evaluation of alternative designs and multidisciplinary analysis integrated in the design process that is regarded as computational intensive engineering problem is solved in Computational Grid to improve performance. The job is submitted via global meta-scheduler and local scheduler, SGE, to computing resources. Efficient transfer of product data is also described using compressed model information for product data visualization at the application client. The developed environment is realized based on grid security infrastructure and virtual organization management. A case study in manufacturing design is demonstrated in this paper.

The framework of distributed collaborative engineering on Grid will further validated and the future research work will be focused on product and process integration, as well as design and analysis integration for distributed product-process modelling and Simulation-based design to enhance this framework.

7. REFERENCES

[1] Bidarra, R.; van den Berg, E.; Bronsvoort W. F.: Collaborative modeling with features, Proceedings of DETC'01 September 2001, Pittsburgh, Pennsylvania

[2] Martino,T. De; Falcidieno, B.; Habinger, S.: Design and engineering process integration through a multiple view intermediate modeler in a distributed object-oriented system environment, Computer-Aided Design, 30(6), 1998, 437-452.

[3] Hoffmann, C.; Joan-Arinyo, R.: Distributed maintenance of multiple product views, Computer-Aided Design, 32(7), 2000, 421-431.

[4] Gupta, S.; Vadhavkar, S.; Pena-Mora, F., Yeung, J.: An Integrated Framework to Support Distributed CAD over the Internet, Proceedings of 10th wetice, 2001

[5] Mervyn, F.; Senthil Kumar, A.; Bok, S. H.; Nee, A. Y. C.: Development of an Internet-enabled interactive fixture design system, Computer-Aided Design, 35(10), 2003, 945-957.

[6] Foster, I.; Kesselman, C.: The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2nd edition, 2003.

[7] Chervenak, F.; Kesselman, I.; Salisbury, C.; Tuecke, S.: The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets, Journal of Network and Computer Applications, 23, 2001, 187-200.

[8] Wang, L.; Shen, W.; Xie,H.; Neelamkavil, J.; Pardasani, A.: Collaborative conceptual design - state of the art and future trends, Computer-Aided Design, 34(13), 2002, 981-996.

[9] Adler, R. M.: Distributed Coordination Models for Client/Sever Computing, Computer 28(4), 1995, 14-22.

[10] Edelstein, H.: Unraveling Client/Server Architecture, DBMS 7, 5, 1994, 34(7).

[11] Lethin, R.: Technical and Social Components of Peer-To-Peer Computing, Communications of the ACM, 46(2), 2003.

[12] CollabCAD: National Informatics Centre, India; 2001. http://www.collabcad.com

[13] Nam, T.; Wright, D.: CollIDE: a shared 3D workspace for CAD, Proceedings of the 1998 Conference on Network Entities, Leeds, UK; 1998, 389-400.

[14] Qiang, L.; Zhang, Y.; Nee, A.: A distributed and collaborative concurrent product design system through the WWW/Internet, Int J Adv Manufact Technol 17(5), 2001.

[15] Pahng, G.; Bae, S.; and Wallace, D.: A web-based collaborative design modeling environment, In Proc. of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises 1998.

[16] van den Berg, E.; Bidarra, R.; Bronsvoort, W.: Web-based interaction on feature models, Proceedings of the Seventh IFIP WG 5.2 Workshop on Geometric Modeling: Fundamentals and Applications, Parma, Italy, 2000, 319-320.

[17] Inventore collaboration tools. Autodesk Inc.; 2001. www.autodesk.com

[18] Shen, W.; Barthes, J. P.: An Experimental Environment for Exchanging Engineering Design Knowledge by Cognitive Agents, Knowledge Intensive CAD-2, eds., Mantyla M., Finger S., and Tomiyama T., Chapman & Hall, 1996

[19] Sabbaghian, N.; Eppinger, S.; Murman, E.: Product Development Process Capture and Display Using Web-Based Technologies, In Proc. of the IEEE conference on system, Man and Cybernetics 1998

[20] Huang, G.; Mak, K.: Web-based collaborative conceptual design, Journal of Engineering Design, 10(2), 183-194.

[21] Roy, U.; Kodkani, S.: Product modeling within the framework of the World Wide Web, IIE Trans, 31, 1999.

[22] Numata: Knowledge amplification: an information system for engineering management, Sony Innovation in Management Series, Vol.17. Japan: Sony Corporation, 1996.

[23] Cutkosky, M.; Engelmore, R.; Fikes, R.; Genesereth, M.; T. Gruber, Mark, W.; Tenenbaum, J.; Weber, J.: PACT: An experiment in integrating concurrent engineering systems, IEEE Computer, 26(1), 1993, 28-37.

[24] Fruchter, R.; Reiner, K.; Toye, G.; Leifer, L.: Collaborative mechatronic system design, Concurrent Engineering: Research and Applications, 4(4), 1996.

[25] Brown, D.; Dunskus, B.; Grecu, D.; Berker, I.: SINE: support for single function agents, In Proc. of Applications of AI in Engineering, Udine, Italy, 1995.

[26] Open CASCADE Technology: http://www.opencascade.org/

[27] Finkelstein; G. C.; Lewis-Bowen, J.: Relating requirements and architecture: a study of Data-Grid, Journal of Grid Computing, 2, 2004, 207-222.

[28] Karypis, G.; Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM Journal on Scientific Computing, 20(1), 1998, 359-392.

[29] JXTA http://www.jxta.org/

[30] SUN Grid Engine http://www.sun.com/software/n1gridsystem/

[31] Rossignac, J.: Edgebreaker: Connectivity compression for triangle meshes, IEEE Transactions on Visualization and Computer Graphics, 5(1), 47-61, 1998.