Heterogeneous Collaborative Design in P2P-based Manufacturing Grid

Lei Chen, Wei Peng, Xiuzi Ye
College of Computer Science/State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, P.R. China

Abstract: Today, it has become more difficult for individual designer to fulfill the complicated design work due to the complexity of modern products. Moreover, an increasing trend toward product development in a collaborative environment has resulted in the involvement of multitudinous enterprises and the use of various software tools to enhance the product design. P2P-based manufacturing grid which combines the advantages of client-server model and pure P2P (Peer-to-Peer) model and avoids their shortcomings provides a supporting environment for heterogeneous collaborative design. And a semantics-based heterogeneous collaborative design methodology is proposed in the paper. Using semantics, product data exchange and modification for collaborative design of heterogeneous CAD systems is implemented. Based on above technologies, a prototype system named HCDGrid is developed. At last, a case study is further given to illustrate the effectiveness of the system.

Key-Words: P2P; manufacturing grid; heterogeneous collaborative design; product data exchange

1 Introduction
As the Internet becomes more reliable and widespread, it offers the possibility of enabling rapid collaborative product design between appropriate partners almost regardless of their geographical locations. Much research has been undertaken to achieve this aim using Web-based distributed collaborative environments and related technologies with some limited, success. Name and Eaglestein listed the tools for a distributed environment [1], Roy et al reported the development of a prototype Web-based collaborative product modeling system [2], Adapalli and Addepalli described different ways of integrating manufacturing process simulations by mean of the Web [3], Kim et al developed a system to store STEP data in an object-oriented database and covert STEP data into VRML data [4], Huang et al studied the Web techniques that can be used for developing collaborative systems [5], Chen et al studied on a network-supported collaborative design over the Internet/Intranet based on dynamic data exchange [6,7], and Lee et al presented a Web-enabled approach for feature-based modeling in a distributed design environment [8]. The authors' research team has been actively involved in this area, for example, Su et al conducted research in network support for integrated design [9-11], and developed a multi-user Internet environment for gear design optimization [12-14]; Hull et al developed a software tool for collaborative design and manufacture over the Internet [15].

We define peer-to-peer (P2P) as a class of applications that takes advantage of resources - storage, cycles, content, human presence - available at the edges of the Internet [16]. Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P design requirements commonly include independence from DNS and significant or total autonomy from central servers. Their implementations frequently involve the creation of overlay networks [17] with a structure independent of that of the underlying Internet, We prefer this definition to the alternative "decentralized, self-organizing distributed systems, in which all or most communication is symmetric," because it encompasses large-scale deployed (albeit centralized) "P2P" systems (such as Napster and SETI@home) where much experience has been gained. Current Grids provide many services to moderate-sized communities [18] and emphasize the nature of the strengths and weaknesses of the two approaches suggest that the interests of the two communities are likely to grow closer over time. Integration of substantial resources to deliver nontrivial qualities of service within an environment at least limited trust. Nevertheless, we shall argue that in fact the two environments are concerned with the same general problem, namely, resource sharing within VOs that may not overlap with any existing organization. Clearly, the two types of system have both conceptual and concrete distinctions, which we shall identify and illuminate, focusing as noted above on characteristics of deployed systems. We shall show that the distinctions seem to be the result of different target communities and thus different evolutionary paths. Grids have incrementally scaled the
deployment of relatively sophisticated services and application, connecting small numbers of sites into collaborations engaged in complex scientific applications. As system scale increases, Grid developers are now facing and addressing problems relating to autonomic configuration and management. P2P communities developed rapidly around unsophisticated, but popular, services such as file sharing and are now seeking to expand to more sophisticated applications as well as continuing to innovate in the area of large-scale autonomic system management. We expect the definition of persistent and multipurpose infrastructure to emerge as an important theme [19].

Sharing relationships are often not simply client-server, but peer to peer: providers can be consumers, and sharing relationships can exist among any subset of participants. Sharing relationships may be combined to coordinate use across many resources, each owned by different organizations. For example, a computation started on one pooled computational resource may subsequently access data or initiate sub-computations elsewhere. The ability to delegate authority in controlled ways becomes important in such situations, as do mechanisms for coordinating operations across multiple resources.

The organization of p2p is suitable to cooperative design in manufacture Grid. In the cooperative design, the designers trust between each other and can share the resources, for instance the chart. This paper describes a framework of cooperative design.

2 A Framework of the Manufacturing Grid
There are three main parts in this framework: Web portal, Design Service Component and Grid Component. Web portal is an interface for designers, in which designers make cooperative design. It will be provided to facilitate all designing application users. In the designing environment, the designer needn't concern about the other designers who are there. Through the portal, the designer feels in the same room to design [20].

Design Service Component and Grid Component are connected through service components such as Web service, Corba. Fig. 1 shows design service components with design data access and processing components.

Applications of manufacture grid require a framework built on top of service component-connection modules where new functionality can be simply inserted and used. Manufacture grid will employ some popular technologies and component technologies, such as XML and Web Services, to provide a service-oriented application framework of useable components that can be configured to produce and run custom grid-aware designing application systems. It supports the reuse, integration, management and execution of distributed software elements, and packaged applications, and allows movement of data and services. The Design Service Component mainly includes design processing, and data storage. The design processing provides the ability for processing the designing made by designers, for supporting cooperative designing, and for processing the charts made by designers. The catalogue access supports to access the different notes (peers) through searching the catalogue and gets some useful resources. The date storage supports to store the data that use in designing. The component is the type of data storage.

3 P2P Systems as Components
In our architectural framework, complex functions are performed by cooperating components, each of them being responsible for one specific aspect or task. A component is by itself a complete P2P system in the traditional sense, running on its own overlay network.

Figure 2 explains our terminology in more detail. It illustrates one component, or alternatively one P2P system.

In the following, we use the terms component and P2P system synonymously. The participating peers, represented by dark grey nodes, are interconnected by an overlay network.

Note that the links in the overlay network of one component may be different from those of other components, and they are usually also different from the underlying physical network topology. Components deliver their results to the outside world or to other components via their component interface. This interface, not shown in Fig. 2, is defined by an API. Note that components, whose interfaces comply
to the OGSA standard [21], for example, can be integrated into existing grid environments.

![Diagram of peer algorithms and component](image)

Figure 2 A component performs its tasks by executing identical instances of the peer algorithm on each peer. Peers communicate with a P2P protocol via peer interfaces. Component interfaces, not shown here, deliver results to the outside world.

Each peer of a component executes independently the same instance of the peer algorithm. Peers communicate with each other via a P2P protocol through peer interfaces.

Peers have their own view on the system, called the visibility horizon. The visibility horizon is in some cases limited to the direct neighbors in the overlay network, while in other cases it may span several communication hops.

In summary, a peer consists of the following three parts:
1. A peer interface for internal communication between peers.
2. A peer algorithm that represents the "intelligence" of the system, ensuring that all peers work together in a coordinated way.
3. A component interface that allows interacting with a component from the outside world.

We distinguish two kinds of components, functional components and self-management components. The former provide services to consumers (users or other components) and the latter keep the system in a stable state. The various components are typically independent from each other - especially those for self-management. Components can be arranged to higher functional units to perform more complex operations.

4 A semantics-based heterogeneous collaborative design methodology

Using semantics, collaborative design for heterogeneous CAD systems is supported in the methodology. The methodology is composed of the following two parts. One is product data exchange between heterogeneous CAD systems by means of feature semantics. And the other is modeling of assembly which comprised of different parts created by heterogeneous CAD systems by means of assembly semantics.

4.1 Product data exchange based on feature semantics

The major challenges of product data exchange based on STEP or IGES standards are as below.
1. High-level design intention is lost.
2. The validity of exchanged geometry data can't be guaranteed.
3. Neutral files are used to exchange in current STEP or IGES standards. Neutral data files are often huge in size and inappropriate in providing online data exchange support.

These challenges hinder current CAD DE methods in the new applications such as collaborative design. And modern most CAD systems features parameterized feature modeling. So Product data exchange based on feature semantics is proposed in the paper.

4.1.1 Neutral XML document approach for data exchange

Neutral XML document and neutral XML document interfaces are needed in order to exchange product data between CAD systems. Direct translators exist but the number of required translators becomes too great if there are many CAD systems involved in the data transfer. For each pair of CAD systems to be able to communicate two translators are required, one for each direction. For a new additional CAD system, several translators have to be added to each existing CAD system.

Figure 3 shows the situation using direct translators. When using a neutral XML document format only one pre- and post-processor is needed for each CAD system. When a new CAD system is added, only one pre- and post-processor has to be added.

Figure 4 shows the situation for data transfer using a neutral XML document format. The star architecture is used in our data exchange.

4.1.2 Feature semantics described by XML

Generally, commercial CAD system supports feature and parameterization technique and applies feature tree structure (Fig. 5). And XML (Extensible Markup Language) which is a flexible data description language is appropriate for describing the structured data [22]. So XML is applied to describing feature semantics in the paper. In the XML description method, each feature corresponds to special marks and structures. Detail definitions of structures and
types is stored in the DTD (Document Type Definition) or Schema.

![Diagram of feature tree structure and part model](image)

Figure 5 feature tree structure and part model

Each feature is defined by XML elements. The nesting structures of the XML elements comply with roles of feature tree expression. There are different data structures of the XML element to different feature types. The basic structured definition of features described by XML Schema is presented as below.

```xml
<ElementType name="Feature" content="eltOnly">
  <attribute type="ID"/>
  <attribute type="NAME"/>
  <group order="one">
    <element type="Extrude"/>
    <element type="Sweep"/>
  </group>
</ElementType>
```

<!-- Extruding feature is described as below. -->

```xml
<ElementType name="Extrude" content="eltOnly">
  <attribute type="ExtrudeStyle"/>
  <attribute type="ExtrudedDirection"/>
  <element type="Designer" minOccurs="1" maxOccurs="*"/>
  <element type="CreatedTime"/>
  <element type="RefPlane"/>
  <element type="Sketch"/>
  <element type="EndCondition" minOccurs="1" maxOccurs="*"/>
  <element type="RefObject" minOccurs="0" maxOccurs="*"/>
</ElementType>
```

4.1.4 Implementation of data change

Data transmission not based on 3D solid models but based on semantic models described by XML decreases the load of network of synchronous collaborative design. There is the capability of updating local knots because of tree structures of XML. And XML supports XQL (XML Query Language), so graphic elements and features can be fast searched. Using these natures, the system converts all kind of modeling operations into modifying XML documents automatically.

To sum up, the data change method based on feature semantics described by XML preserves design history, and greatly reduces transmitting quantity, and meets the needs of display and modeling. And it can be extended conveniently, and supports cross-platform, cross-network and cross-database.

4.2 Assembly modeling based on assembly semantics

Using assembly semantics, assembly modeling is implemented. Research on product assembly design based on semantics can prominently enhance the straightforwardness and efficiency of assembly design.

4.2.1 Assembly semantics

Assembly semantics is the abstract description of assembly relationships, which implies the constraint between parts, assembly rules, assembly knowledge and actions [23]. Assembly semantics, that is more suitable for the engineer to express design intention, has the abstract character in expression and is dependent of application domain.

![Diagram of key connection semantics](image)

Figure 6 "Key connection" semantics

We built a semantics dictionary that is composed of semantics in common use according to the joint relation, position relation and transmission relation of the assembly. For example, the joint semantics includes "bolt nut connection", "cylindrical pin connection", "taper pin connection", "key connection" (Figure 6), "multiple key connection", "shape connection" and "river connection" etc. The transmission semantics includes "straight gear mesh", "skew gear mesh", "belt transmission", "chain transmission" and "worm wheel gear mesh" etc. The position semantics for shaft includes "shaft shoulder - locked collar mate", "shaft shoulder -elastic collar mate", "shaft shoulder - sleeve mate", "shaft shoulder - round nut mate" and "shaft shoulder - end collar mate" etc.

The semantics model is represented as a graph, in which the node represents the mechanical part and
the arc between two nodes represents the constraint relationship of mechanical parts. A direct arc connects the parts, which co-related to the same semantics. The direction of the arc implies the assembly sequence of the parts. A direct semantic loop is formed by connecting the first part and the last part in the semantics. Thus, a semantics model can be expressed as SCG = (H, E, S), where H means the set of mechanical parts, E means the set of constraint relationship, S means the set of assembly semantics. Figure 6 shows an example of semantics model.

Node: mechanical part, Arc: assembly constraint Direct Loop: assembly semantics S1: "key connection" S2: "shaft shoulder - sleeve mate" S3: "straight gear mesh"

Figure 7 Assembly semantics model

4.2.2 Algorithm in semantics modeling
Algorithm: Conversion of assembly semantics
Let Si be the semantics to be converted, H={hi, i=1 to j} be the set of mechanical parts which are connected by semantics Si, H1(hi ∩H) be the set of parts which have added to semantics model. Now add hj+1 to the semantics model.
- step1: Convert semantics Si to constraint set C={ci, i=l to n}
- step2: if the element Ci of C describes the constraint relationship between hj+1 and H1, add Ci to φ={φi |i = k to r} 
- step3: Judge the conflicts between φ and the constraints that have existed in semantics model. If there is no conflict happen, go to step5
- step4: Report the conflict message to the designer, return 
- step5: If node does not exist in the semantics model, then add node 
- step6: Add constraint information to semantics model according to φ 
- step7: Connect hj to hj+1 in direct arc. If (j+1)=m, then connect hj+1 to h1 to close the semantics loop.

5 Implementation of HCDGrid and a case
A heterogeneous collaborative assembly system is implemented based on ACIS which is an excellent and component 3D CAD geometric kernel. It's encapsulated as an ActiveX component. So it can be downloaded from the grid and used by web browsers.

The system provides functionality of data exchange, model display, feature modification of part model, assembly modeling and etc. And we developed two Addins for SolidWorks2004 and Catia V5. Using APIs of corresponding CAD systems the Addins mainly provide functionality of product data exchange based on feature semantics. Using the Addins, we can exchange all kinds of special format files of CAD systems to neutral XML documents.

All product data is transmitted in XML format through SOAP and HTTP and all remote invoking is implemented by invoking the interface exposed by Web Services.

Figure 8 A case

Figure 8 shows a case. Using web browser a user input data of several models to a CA system. And assemble activity is implemented by using key connection.

6 Conclusion
At present, the research of collaborative design in manufacturing grid is strengthened. We have done some research works on it. P2P-based manufacturing grid which combines the advantages of client-server model and pure P2P model and avoids their shortcomings provides a supporting environment for heterogeneous collaborative design. Using semantics-based data exchange and assembly modeling technology, collaborative design for heterogeneous CAD systems is basically implemented.

It is no doubt that future manufacturing is global, virtual and network-based. Its object is to realize agile design, agile manufacturing, agile detection, agile response and agile recombination. And the research of Heterogeneous Collaborative Design in P2P-based Manufacturing Grid only takes first step.
There are some problems unsolved at this time. They include communication and control issues, knowledge representation and semantic issues etc.

References


