Available online at www.sciencedirect.com

SCIENCE DIRECT®

Advanced Engineering Informatics 19 (2005) 155–168

ADVANCED ENGINEERING
INFORMATICS

www.elsevier.com/locate/aei

A collaborative approach to assembly sequence planning

Tianyang Dong[a], Ruofeng Tong[b,*], Ling Zhang[c], Jinxiang Dong[b]

[a]*College of Software Engineering, Zhejiang University of Technology, Hangzhou 310032, China*
[b]*Institute of Artificial Intelligence, Zhejiang University, Hangzhou 310027, China*
[c]*Department of Economics and Management, Zhejiang University of Science and Technology, Hangzhou 310023, China*

Abstract

Distributed product development requires collaborative work among team members. For the sake of supporting assembly planning activities involving geographically dispersed designers, this paper presents an approach of collaborative assembly sequence planning to validate the assemblability of parts and subassemblies rapidly. In order to increase the planning efficiency and support the collaborative planning, role-based model is exploited to compress or simplify the product. In role-based model, the B-rep models are simplified according to the permissions associated with the role, so the surfaces invisible from outside of the model are removed. In collaborative planning, the planning tasks are assigned to different designers that carry out the collaborative planning, respectively. In this paper, a knowledge-based approach is proposed to the assembly sequence planning problem. This research shows that the typical or standard CSBAT (Connection Semantics Based Assembly Tree) can be applied to a given assembly problem. This paper presents the structure of the Co-ASP (Collaborative Assembly Sequence Planning System) and provides an example to illustrate the collaborative planning approach.

Keywords: Assembly planning; Collaborative design; Connection semantics; Knowledge base; Assembly model; Sequence planning

1. Introduction

Product design is typically a highly iterative activity involving a group of designers. Previously, all the collaborating designers were at the same geographical location within the enterprise. In addition, companies are often out-sourcing engineering activities to rapidly design and prototype the product [1]. The Internet enables many new tools for global collaboration and data sharing in a global marketplace. Successful companies will get the right information and tools to the right person at the right time, regardless of where the person is located [2]. This improved communication technology has lessened the impact of physical distances on design tasks and has resulted in the reconsideration of design activities where design tasks are geographically dispersed. One of these activities is virtual prototyping, which analyses a product without actually making a physical prototype of the product. The term *virtual* refers to the fact that the design is not yet created in its final form but that only a geometric representation of the object is presented to the user for observation, analysis and manipulation. This prototype does not necessarily have all the features of the final product but has enough of the key features to allow testing of the product design against the product requirements [3].

Assessment of the assemblability of a product is frequently neglected, but it is an important issue that should influence the design process. This is because assembly issues affect the partitioning of a product into functional parts as well as generating the sequence of assembly tasks. In the past, the definition and analysis of these issues has been completed towards the end of the product design phase, but this means that it is often too late to correct any deficiencies [4]. What's more, assembly planning plays a major role in the manufacturing industry. This automation constitutes one of the most important conditions to guarantee the future competitiveness of the industrial companies. Indeed, an optimal assembly plan can increase production efficiency and reduce the cost of a product.

This paper addresses the problem of Design for Assembly to achieve the product pre-assembly across the various CAD platforms and geographical locations by taking advantage of web-resources, to validate the

---

* Corresponding author. Tel.: +86 571 87951414; fax: +86 571 87982571.

E-mail address: trf@zju.edu.cn (R. Tong).

1474-0346/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.aei.2005.05.008

assemblability of parts and subassemblies rapidly, and to carry out collaborative planning for complicated product. In order to increase the planning efficiency and support the collaborative planning, role-based model is exploited to compress or simplify the product. In role-based model, the B-rep models are simplified according to the permissions associated with the role, so the surfaces invisible from outside of the model are removed. In collaborative planning, the planning tasks are assigned to several designers that carry out the collaborative planning, respectively. In this paper, a knowledge-based approach is proposed to the assembly sequence planning problem. This approach of assembly sequence planning is to generate the feasible plans for assembly directly and avoid the sophisticated merging of plans for substructures. The CSBAT hierarchy proposed in this paper provides an appropriate way to consider both geometric information and non-geometric knowledge.

The rest of this paper is organized as follows: Section 2 reviews the previous work on collaborative design as well as assembly planning. Section 3 presents a new elaborate scheme for role-based model simplification. Section 4 considers the representation problem of assembly model. The strategy of assembly sequence planning is described in Section 5 and the algorithm of generating CSBAT is described in Section 6. Section 7 gives the structure of Co-ASP and provides an example to illustrate the collaborative planning approach. Conclusions and areas for future research are finally discussed in Section 8.

2. Related works

2.1. Collaborative design

This section reviews related work and summarizes some of the highlights of collaborative design development. Bidarra et al. [5] presented a collaborative framework that does support integrated design of parts and assemblies. It enables members of a product development team to have synchronous collaborative modeling sessions via Internet. The collaborative framework not only offers facilities to simultaneously work on independent tasks in a product development process, but also synchronous facilities to really collaborate on the design of a same component. Mori and Cutkosky [6] proposed an architecture in which engineering design agents interact with each other, exchange design information and keep track of state information to assist with collaborative design. They presented an example involving CAD agents, for which each state corresponds to a particular design model. If a designer publishes a new design, the operation is recorded as a state transition that triggers action. Saar [7] presented VIRTUS, a multi-user platform based on VRML2.0, Java and TCP/IP, which eases the development and authoring of distributed environments with a special focus on collaborative work. Huang and Mak [8] focused on providing design for manufacture and assembly techniques on the Internet. An experiment is conducted to show how a well-known design for assembly technique can be converted into a web-based version, which is functionally equivalent to its version on a standalone workstation. Kim et al. [9] introduced an assembly design formalism to specify the assembly and joining relations symbolically to support collaborative assembly design. By using this formalism, assembly and joining relations are extracted from the assembly and the relation models have mathematically solvable implications.

Research has been performed to demonstrate the feasibility of collaborative design and deduce the requirements for collaborative design in the Madefast program [10]. Madefast differed from a conventional industrial project in that it was a community effort with no formal top-down management structure and no central authority. Cutkosky et al. [11] developed the Palo Alto Collaborative Tested (PACT), a concurrent engineering infrastructure that encompasses multiple sites, subsystems and disciplines. The authors concluded that: (a) in distributed design several groups communicate utilizing a predefined protocol and (b) during concurrent engineering integration of multitude of models required for the complex design process must be considered. Regli [2] has described some of the technology trends influencing a network based computer aided design framework, with a particular focus on how companies are assimilating new Internet and object-oriented concepts.

2.2. Assembly sequence planning

The assembly planning problem has received much attention in manufacturing industries over the past 20 years or so. Assembly planning aims to identify and evaluate the different ways to construct a mechanical object from its components. The problem can be formulated as follows: given a geometrical and technological description of a product, find an assembly sequence that satisfies the precedence relations between operations and meets certain optimization criteria. In the last decade, several approaches have been proposed to generate assembly sequences automatically. In summary, the existing approaches to generating assembly plans can be roughly classified into three main approaches: human-interaction, geometry-based reasoning and knowledge-based reasoning.

The method of human-interaction mainly focuses on each user's query either on the connection between a pair of parts or the feasibility of a single assembly operation [12,13]. Clearly, this method is far from automation. Thereafter a number of geometry-based reasoning approaches have been proposed. One general approach is the cut-set method by many researchers [14,15]. The cut-set method follows the compute-and-test scheme, where all possible ways to partition an assembly into two connected subassemblies are generated. The other approach of geometry-based reasoning is compute-and-generate strategy [16–18]. In order to generate good assembly plans, non-geometric assembly data, besides geometric assembly data, should also be used in assembly planning. There have been several approaches that generate assembly sequences by using high-level expert knowledge or experience. Swaminathan and Barber [19] developed an experience-based assembly sequence planner for mechanical assemblies. This approach utilizes the case-based planning to store, retrieve, and modify existing cases or experience to develop assembly sequences. Chakabarty [20] described a planner that uses the structure both as a framework for structure-dependent definitions of good plans, and as a tool for finding good plans more rapidly by reusing sub-plans for repeated substructures. Yin et al. [21] proposed a connector-based hierarchy approach that also seeks a plan reuse oriented solution to assembly planning based on the hierarchy description. There are other approaches using assembly knowledge or artificial intelligence [22–24].

3. Role-based model simplification

In collaborative planning, the planning tasks are assigned to many designers and they carry out the collaborative planning, respectively. An extremely large assembly is composed of millions of complex parts, a trend of many mechanical models. When a single computer or network deals with large or complex assembly, a special method to compress or simplify the assembly is needed. In CAD application, expensive and high performance computer systems are employed to deal with the accompanying large information. And a high network bandwidth is necessary to the collaboration from different sites [25].

Up to now, various ways has been proposed to simplify the models. Hoppe [26] and Hussain et al. [27] proposed some approaches of triangular mesh compression for storage and transmittance of triangular meshes, which are inapplicable to our research. Other researchers have considered features and topologies in the simplification of the models. Zhu [28] proposed an approach to simplifying B-Rep models by automatic fillet/round suppressing, which utilizes an incremental knitting process to handle various topological structures of fillets and rounds. Koo et al. [25] adopted the wrap-around operation to make multi-resolution model of part and assembly. In this method, the wrapping of products using the plastic wrap in the kitchen is imitated.

In this research, we propose a new elaborate scheme for multi-resolution hierarchies and access control mechanism, in which the surfaces invisible from outside of the model are removed and a tailored 3D model is customized for a specific user based on the roles defining the user's access permissions on the parts or subassemblies. There are two models for each subassembly, one is the simplified model, the other is the original model that has all detailed information for its parts or subassemblies.

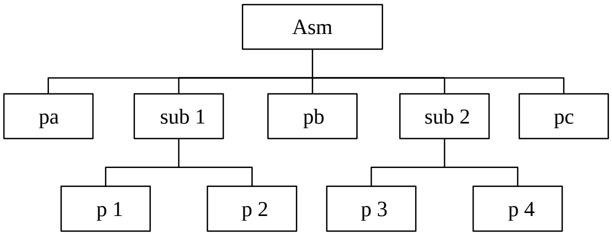

Fig. 1. Hierarchical structure model.

Up to now, several access control policies commonly have been developed. In this research, the Role-based Access Control (RBAC) model [29,30] is adopted. In RBAC, system administrators create roles according to the job functions in an organization, grant permissions (access authorizations) to the roles, and then assign users to the roles. The permissions associated with a role tend to change much less frequently than the users who fill the job function that role represents. Users can also be easily reassigned to different roles as needs change. Roles, $R=\{r0,r1,\ldots,rm\}$, are abstract objects that define both the specific users allowed to access resources and the extent to which the resources are accessed. The designers correspond to a set of *actors* $A=\{a0,a1,\ldots,an\}$, each of which will be assigned to a set of roles [30]. The entire product is represented as an assembly model *Asm*. We define a set of *security parts*, $SP=\{p0,p1,\ldots,pk\}$, where each pi is a part or subassembly of the product. Assembly models are often hierarchically represented, and Fig. 1 shows the hierarchical structure model of a certain assembly.

Table 1 is the access matrix for the subassemblies and parts of the product shown in Fig. 1. In this example, only three roles, $r0$, $r1$ and $r2$, are created. Suppose that actors *designer*0, *designer*1, and *designer*2 are assigned to roles $r0$, $r1$ and $r2$, respectively. Each cell of the access matrix is assigned *read*, *write*, or *pass* (only for subassembly) permissions. It is reasonable to assume that write permission of a part is exclusively given to single role. However, read permissions of a part or subassembly can be given to multiple roles. The pass permission of a subassembly allows an actor to access its children, and the permissions associated with the children determine their readability/writability.

For the role $r0$, it has *read* permission to *sub*1. Because *sub*1 is subassembly, $r0$ cannot directly access *sub*1's children, $p1$ and $p2$, and a simplified model of *sub*1 will be presented to $r0$. Then, $r0$ has *pass* permission to *sub*2, and the permissions associated with $p3$ and $p4$ should be

Table 1
Access matrix for the hierarchy of the assembly

Roles	*sub*1	*sub*2	pa	pb	pc	*p*1	*p*2	*p*3	*p*4
*r*0	*r*	*p*	*w*	*w*	*w*	–	–	*r*	*r*
*r*1	*p*	–	*r*	*r*	*r*	*w*	*w*	–	–
*r*2	–	*p*	*r*	*r*	*r*	–	–	*w*	*w*

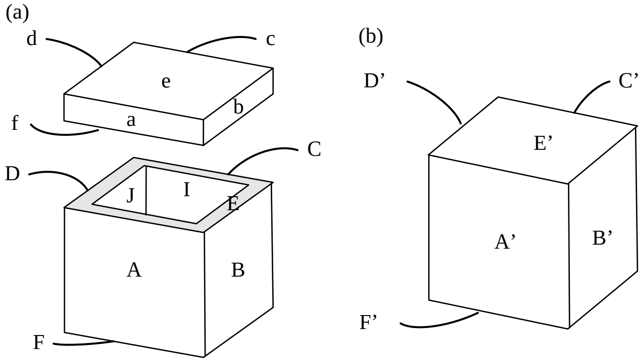

Fig. 2. An example of simplification of subassembly: (a) original model; (b) simplified model.

checked: *r*0 has *read* permission to *p*3 and *p*4. Finally, *r*0 has *write* permission to *pa*, *pb* and *pc*.

In collaborative planning, if the role only has *read* permission to the subassembly, the designer is provided with only the information of the simplified model. In this research, the simplified model of the subassembly is called a virtual part. The simplification of the subassembly will change the geometry and topology of the original B-Rep model. In order to prevent information missing and obtain a complete correct assembly model after the simplification of subassembly, we need to pay attention to the issues such as geometric and topological consistency and reversibility. In addition, the consistency and reversibility of assembly constraints are important [28].

In order to obtain a complete simplified B-Rep model by the simplification of subassembly, a systematic approach is developed to guide the information processing. We adopt the wrap-around operation to make multi-resolution model of the subassembly [25]. This method is composed of two steps. The first step is the part level wrap-around operation for the parts that compose the simplified subassembly. In this step, a convex inner loop is used as a clue to find concave space and fill this space by removing the convex inner loop. After filling the concave space, the surfaces that cannot be seen from outside of the model are removed. The second step is assembly level wrap-around operation. As a result of first step, an overlap between parts exists and surfaces that cannot be seen from outside of the model exist. These surfaces are deleted in the second step.

For the subassembly shown in Fig. 2, there are seventeen surfaces in the original model, but only six surfaces exist after removing the invisible surfaces. Only the simplified representations of the details for the subassembly are used in assembly planning, therefore, the algorithm of assembly planning will be more efficient. What's more, this simplification supports the collaborative planning for complicated product effectively.

The hierarchical representation of product is shown in Fig. 3. The product attribute and behavior information in collaborative planning is stored in the hierarchy of product, feature, geometry and display levels. It is noted that the product level may include several layers. These levels are linked together through the mapping of correlated data

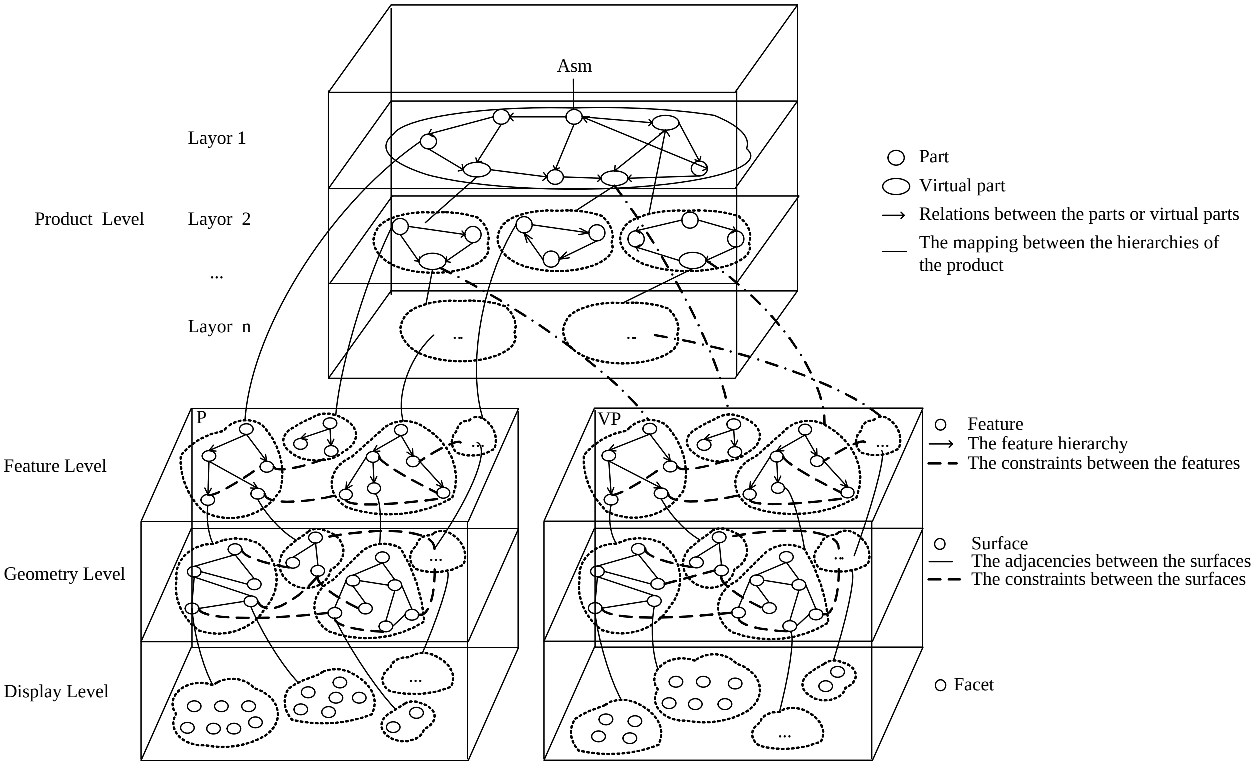

Fig. 3. The hierarchical representation of product.

and constraints. Product information organized in such a way cannot only support the collaborative planning, but also increase the planning efficiency.

4. Representation of assembly model

4.1. Connection semantics based assembly tree

A connector provides constraints on its jointed components to ensure that these components perform the required functions. Therefore, the connection is not only a thinking module to construct product design, but also the core block to provide the components restriction. Connection provides some significant relationships among two or more assembled parts and can act as a foundation of assembly clustering.

In this research, connection semantics is used to represent each connection. Connection semantics is denoted as *Connect-Type* $(C)[A]$, where C is the set of connectors, A is the set of all parts or virtual parts constrained by connector C, and *Connect-Type* represents the connection type including: *Bolt-Nut*, *Screw*, *Pin*, *Key*, *Roll-Fit*, *Mate*, etc. Some connection types are explained as follows:

Bolt-Nut (c_1, c_2): bolt-nut type connection, where c_1 is a set of bolts and c_2 is a set of nuts;

Screw(c): screw type connection, where c is a set of screws;

Key(c): key type connection, where c is a key;

Pin(c): pin type connection, where c is a set of pins; ……

Mate (a, b), mate type connection, where a and b are the mating parts, means two faces of a pair of parts are kept in contact with one another without the function of connectors.

Insert (a, b), Insert type connection, means that the inserted part a is inserted into the bounding part b. It is obvious that a and b can be extended to subassemblies.

It is noted that the connector parameter in some connection types is a set of connectors, namely a group of several connectors. Two connections *Connect-Type*1 $(C_1)[A_1]$ and *Connect-Type*2 $(C_2)[A_2]$ satisfy the grouping relation if: (1) $A_1=A_2$; and (2) C_1 and C_2 can be assembled and removed only in the same direction, but in any order with respect to each other.

4.2. Connection semantics based assembly relational model

To enable automated assembly sequence planning, all the related information should be organized and represented as assembly model. The effectiveness of an assembly planner relies heavily on the input of the assembly representation. The assembly sequence planning problem is essentially a geometrical one, the assembly representation applied in this research will emphasize on the geometric information such as the shapes of the parts, their positions and the contacts between the parts. A purely geometric description of the assembly cannot always generate a good assembly sequence. Some sequences may be feasible from a geometric point of view, but are impractical due to the special properties of some connections. Furthermore, the inclusion of non-geometric information helps to reduce the explosion of possible solutions. The assembly representation used in this research is the CSBARM (Connection Semantics Based Assembly Relational Model) that integrates both geometric and non-geometric assembly data.

The CSBARM of an assembly is an undirected graph that includes two types of nodes: parts and connectors. Each node has its own attributes. The relationship between the nodes describes the connection of the parts. Essentially, the CSBARM is similar to the CBRM used in the connector-based hierarchy approach [21]. But some important changes have been made to support the planning strategy proposed in this research. The CSBARM for assembly can be denoted as $\langle P, C, M\rangle$ where

- P is a set of nodes, each of which corresponds to a part or virtual part not belonging to the connectors in the assembly. The attributes associated with P include (1) the part geometry (feature list or surface list), (2) the contact surfaces as features, (3) the assembly/disassembly tools, such as screwdriver, spanner and gripper, etc. and (4) the physical properties of the part or virtual part such as weight.
- C is a set of nodes, each of which corresponds to a connector in the assembly. The attributes of a connector include (1) the connector geometry, (2) the mating volumes and the contact surfaces, (3) the connector type, (4) the assembly/disassembly tools, and (5) the physical properties of a part such as weight.
- M is a set of liaisons between two nodes, each of which corresponds to the connection between pairs of elements of $P\cup C$. The attributes of a liaison are described by (1) the connection type (2) the mating type (such as Against, Fit, Screw-Fit, etc.), (3) the male–female pairs of mating entities, (4) the contact surfaces, (5) the mating directions corresponding to the pairs of contact surfaces, and (6) the mating matrix.

Specifically, the most important information obtained from all mating features is the degrees of freedom of the mating entities. To accomplish this, mating feature is represented by a simple 3×4 matrix [18]. The elements represent the degree of freedom on the three major axes in 3D space. The configuration space for an assembly model typically has 3DOF in translation and 3DOF in rotation. It is usually subdivided into positive and negative directions and represented by individual elements in the matrix. This leads to the following mating matrix

$$\begin{bmatrix} x & -x & wx & -wx \\ y & -y & wy & -wy \\ z & -z & wz & -wz \end{bmatrix},$$

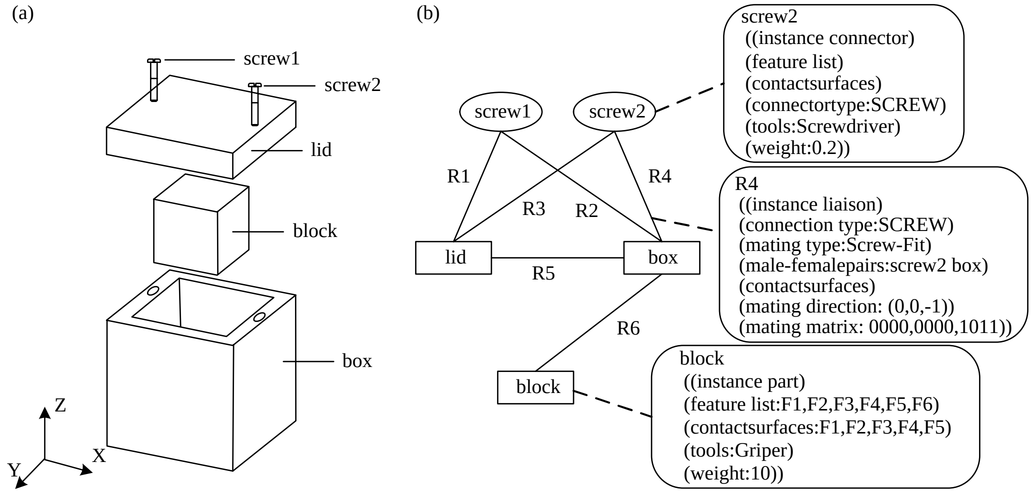

Fig. 4. An example of CSBARM: (a) the container assembly; (b) its CSBARM.

where $\pm x$, $\pm y$, $\pm z$ are linear translations and $\pm wx$, $\pm wy$, $\pm wz$ are the rotations about X, Y and Z axes, respectively. The values of the elements in the mating matrices are either 0 or 1. Integer 1 indicates freedom of motion in the direction along the corresponding principal axis. Integer 0 indicates the motion is disallowed in the axial direction.

The CSBARM of an assembly can be represented by an attributed liaison graph. A liaison exists between a pair of parts if one part constraints the freedom of motion of the other either by a direct contact or a near contact. As an example, Fig. 4(a) shows a container assembly and Fig. 4(b) illustrates the corresponding CSBARM. In order to simplify the graph, not all labels with the edges or nodes are described. In Fig. 4(b), nodes corresponding to parts are rectangles, and nodes corresponding to connectors are ellipses. All nodes contain labels indicating their corresponding entities. The attributed liaisons connecting two nodes in Fig. 4(b) correspond to the connections.

5. Planning strategy

The overview of collaborative planning strategy is shown in Fig. 5. This approach can generate assembly plans for complicated products in a distributed environment that is global, network-centric, and spatially distributed. It also enables product designers to communicate more effectively, obtain, and exchange a wide range of planning resources during product development. The utilization of knowledge stored in knowledge base is fundamental to the approach. Compared with other systems based on the reuse philosophy, this approach can generate the feasible plans for assembly directly and do not need to merge plans for CSBATs. The procedure of collaborative assembly planning is mainly composed of the following modules.

5.1. Decomposition and assignment of planning tasks

This module sets up a project for collaboration for the specific designers according to the access resources allowed and the extent to which the resources are accessed. This module plans and executes the decomposition of high-level goals into low-level actions and assigns the tasks to different designers. The planning tasks can be decomposed into sequential actions along the time line for a single designer, or into concurrent actions for multiple designers. After the decomposition and assignment of planning tasks, the system generates the role-based solid model and CSBARM, where the surfaces invisible from outside of the model are removed and a tailored 3D model is customized for a specific user based on the roles defining the user's access permissions on the parts or subassemblies. In collaborative planning, if the role only has *read* permission to the subassembly, the designer is provided with only the information of the simplified model.

5.2. Generating CSBAT hierarchy

To reuse of stored plans in knowledge base is the fundament of the proposed approach to assembly planning. If an assembly is represented as a CSBAT, the assembly plans can be generated by retrieving the knowledge base. There may be more than one CSBAT for an assembly. Therefore, it is necessary to select a preferred CSBAT out of multiple CSBATs that can be used to reuse of stored plans in knowledge base. In CO-ASP, a CSBAT hierarchy for an assembly is automatically derived from its CSBARM by geometric reasoning and knowledge-based reasoning according to some heuristic rules. The algorithms of generating CSBAT hierarchy will be discussed in Section 6 in detail.

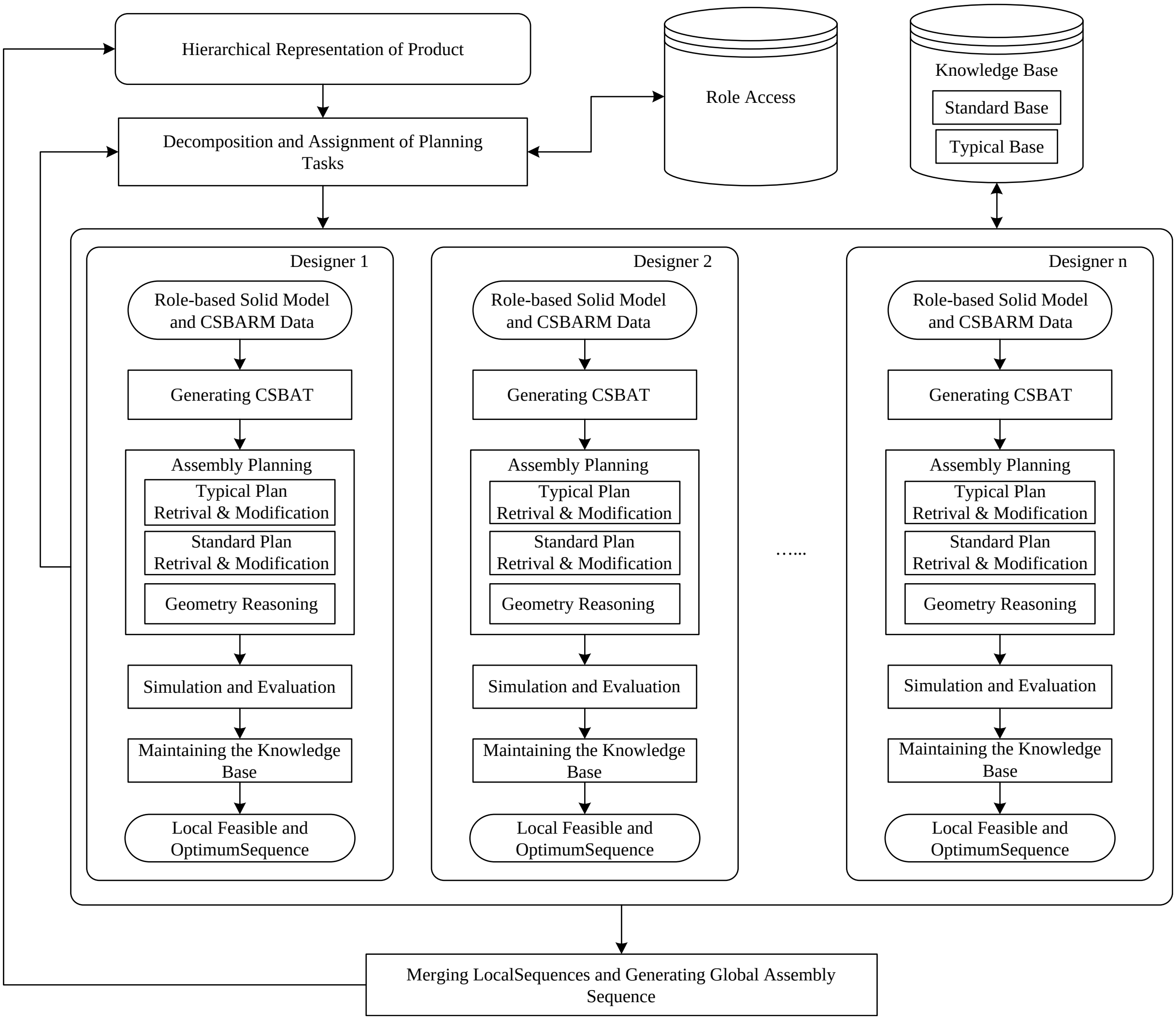

Fig. 5. The overview of collaborative planning strategy.

5.3. Assembly sequence planning

This phase involves the searching of the knowledge base to find a match for the CSBAT provided to the *assembly sequence planning* module. If the similar or same CSBATs in knowledge base do not give rise to useful plans for the provided CSBAT, the system will generate plans by geometric reasoning. In CO-ASP, there are three ways to find plans for a CSBAT and they are attempted in the following order: (1) by retrieving the typical base, (2) by retrieving standard base, and (3) by geometric reasoning. The plans obtained from the plan base are expressed in terms of part names that act as placeholders for actual parts. To make the plans useful, the dummy part names are converted to reflect the part names from the problem.

5.3.1. Typical plan retrieval and modification

It is generally desirable that similar or same structure in all assemblies should be built in the same way during assembly process, since this generally requires the smaller variety of operations. Moreover, retrieving from the typical plan base can speed up the planning process with better plans produced. Therefore, the plans of the typical assemblies are stored in typical plan base for reuse. Typical plan base consists of the knowledge of CSBATs that are typical assemblies or subassemblies in the enterprise. For example, the motorcycle engine is a typical assembly in the motorcycle enterprise. Therefore, the assembly plans for the CSBAT of motorcycle engine can be stored in typical plan base for reuse. To support the reuse of plans for typical CSBATs, all kinds of knowledge is stored for each CSBAT in typical plan base. There may be one or several assembly sequences corresponding to each typical CSBAT.

The problem of typical plan retrieval from the typical plan base turns to be the problem of matching the CSBAT and CSBARM. Because of the complexity of assembly relational model, the graph matching in assembly planning is very difficult. In our research, a novel approach to graph matching is proposed. We use a so-called partial assembly constrain satisfying strategy to dynamically prune improper typical CSBATs in which two assemblies or subassemblies are determined unmatched without necessity to check their details. The partial assembly constrain satisfying strategy is implemented by candidate CSBAT and target CSBAT.

5.3.2. Standard plan retrieval and modification

If the similar or same CSBATs in typical base do not give rise to useful plans for the provided CSBAT, the system next retrieves the stored plans from standard plan base. For most primitive CSBATs, there are common and preferred procedures to assembly them. What's more, these assembly processes for primitive CSBATs are invariable. For instance, there are standardized procedures to install a ball bearing type of CSBAT. During assembly planning, plans for a primitive CSBAT are obtained by retrieving suggested plans from the standard plan base, instead of by reasoning about the low-level interactions among the parts of the CSBAT. By doing so, not only a great deal of computation can be avoided, but also better plans are obtained by integrating manufacturing experience with building CSBATs. Standard plan base consists of primitive CSBATs that are indexed by the types of their connections.

The problem of standard plan retrieval and modification is also a problem of matching the CSBAT and CSBARM. The CSBAT to retrieve plans from standard base is transformed into a primitive CSBAT firstly. Each child CSBAT is looked as a whole, and the assembly constrains between parts in different child CSBATs are transformed into assembly constrains between child CSBATs.

5.3.3. Geometric reasoning

If the similar or same CSBATs in typical and standard base do not give rise to useful plans for the provided CSBAT, the system next generates feasible plans for the CSBAT by geometric reasoning. As the parts of a CSBAT are assumed to contact with each other, the geometrically feasible plans are generated mainly by reasoning about mating directions of each part in the CSBAT. In addition, directed-connector knowledge is exploited for the generation of assembly precedence constraint graph. By integrating geometry-based reasoning with knowledge-based reasoning, the computation complexity is reduced drastically and the assembly sequences obtained are more feasible and practical.

5.4. Simulation and evaluation

After the *assembly sequence planning* stage, the feasible and practical solutions are found. At this stage, the user can simulate the assembly or disassembly process in the virtual environment. The main benefits of *simulation and evaluation* stage are discussed as following. (1) It can provide a tested and valuable information that might otherwise have required time-consuming and expensive physical experimentation. (2) It also assists in training assembly operators using virtual machines and virtual workpieces. (3) It actually controls and runs the operation of the real assembly processes through manipulation of the virtual objects in the virtual environment. In addition, all feasible solutions are compared with each other at this stage.

5.5. Maintaining knowledge base

This unit decides whether the newly generated plans should be stored to the knowledge base for reuse. The user can tell the system to store the typical CSBAT in the typical plan base if it is a typical subassembly. If the plans of a primitive connection type do not exist in standard plan base, this module also stores the knowledge in standard plan base.

6. Algorithm of generating CSBAT

Each CSBAT is classified by the connection types. As stated before, there may be more than one CSBAT for an assembly. Therefore, it is necessary to select a preferred CSBAT out of multiple CSBATs that can be used to reuse of stored plans in knowledge base. The preferred CSBAT can be selected by evaluating tentative CSBATs based on selection indices (*SI*). The SI evaluates a cluster of parts in the CSBAT based on the following criteria [31]:

(1) *Stability Index* (*STI*): The stability index, STI (T), of a subassembly T represents how stable all child CSBATs of T remain during the disassembly operation of the connectors from T. If the child CSBATs are not stably fastened, the parts may deviate from their correct position, and the operator has to take more time to reposition them, and the index STI is defined as follows:

$$\mathrm{STI}(T) = \sum_{P_m \subset \mathrm{subtree}(T)} \sum_{P_j \in P_m} F_{\mathrm{st}}(\mathrm{stb}(P_j)) \tag{1}$$

where, subtree(T) is a set of all child CSBATs of T. In this research, F_{st} () maps the stability flag of a part into the time of basic motions: *reach, grasp, move, position, release and reach* as proposed by Kanai [32].

(2) *Operation Preference Index* (*PRI*): The operation preference index indicates the priority of the connection type. Usually, an assembly may have several CSBATs of different connection types. However, the operation complexity of each connection type is different. For instance, the CSBAT of *Screw* type connection can be

Table 2
Operation preference index (PRI)

Connection type	PRI
Mate, insert	0.3
Bolt, bolt-nut, screw, pin	0.5
Key, roll-fit, gear, roll-fit, belt-mesh, bearing	0.6
Rivet, welding	0.8

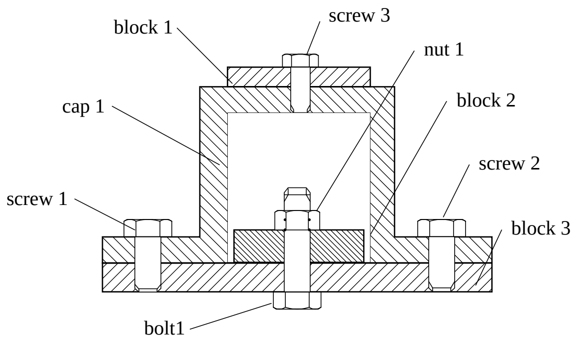

Fig. 6. A test assembly.

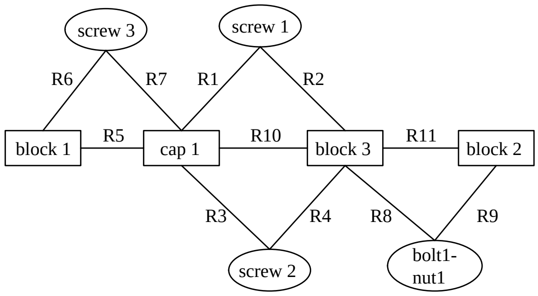

Fig. 7. The CSBARM of the test assembly.

disassembled or assembled more easily than the CSBAT of *Rivet* type. The index PRI is determined by the connection types. The index PRI is determined by the connection types as shown in Table 2.

(3) *Operation Continuity Index* (*CNI*): The operation continuity index indicates how much the operator has to make the extra motion by exchanging the connection types, mating directions and tools. The index CNI is defined by Eq. (2)

$$\mathrm{CNI} = k_1 \times \mathrm{TPI} + k_2 \times \mathrm{DRI} + k_3 \times \mathrm{TLI} \tag{2}$$

where k_1, k_2, k_3 are the coefficients and $k_1+k_2+k_3=1$. TPI indicates the change of the connection type; if the CSBAT has different root connection type from its parent CSBAT, the TPI=1, else TPI=0. DRI indicates the change of the mating directions. If the mating direction of the CSBAT is different from that of its parent CSBAT connection, the value of DRI=(*the angle of direction change*)/90. That's to say, if the mating direction of the CSBAT is same to that of its parent CSBAT, the value of DRI is 0. TLI depends on whether its own tools are same to that of its parent CSBAT. If the tools are different, the value of TLI is 1, else the value is 0.

(4) *Parallelism Index* (*PI*): Parallelism of a CSBAT can be measured approximately by the number of the connectors which construct the connection and the number of parts in each child CSBAT. The index PI is defined by Eq. (3)

$$\mathrm{PI} = k_1 \times \mathrm{CI} - k_2 \times \mathrm{SPI} = k_1 \times \frac{N_c}{N_{\mathrm{all}}} - k_2 \times \frac{1}{N_{\mathrm{all}}} \times \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} |N_i - N_j| \tag{3}$$

where k_1, k_2 are the coefficients and $k_1+k_2=1$, N_{all} is the number of all parts in the CSBAT, N_c is the number of connectors making the connection, and N_i is the number of parts in child CSBAT i. Higher value of CI implies the more the number of the connectors, which construct the connection. Lower value of SPI implies the more operations can be done in parallel for different child CSBATs.

(5) *Selection Index* (*SI*): The selection index is given as follows: $\mathrm{SI} = e^{-k_1 \cdot \mathrm{STI} - k_2 \cdot \mathrm{PRI} - k_3 \cdot \mathrm{CNI} + k_4 \cdot \mathrm{PI}}$, where k_1, k_2, k_3, k_4 are the assembly coefficients and $k_1+k_2+k_3+k_4=1$. The CSBAT that has the highest SI value is selected as the candidate CSBAT. The system prefers to select the CSBAT with smaller STI, PRI, CNI values, but larger PI. The coefficients can be assigned by the designer based on the relative significance of each selection indices on the overall assembly cost.

The proposed algorithm to generate the CSBAT of an assembly proceeds as follows:

Step 1. Push the provided assembly into stack *AsmSet*;

Step 2. Check if *AsmSet* is null or not: if yes, the CSBAT of the assembly has been constructed successfully; else pop a subassembly as *CurAsm*; If *CntrSet* is not null, then delete all elements from *CntrSet*.

Step 3. Group connectors that can be disassembled in *CurAsm* in all possible ways and store the sets of grouped connectors in the set *CntrSet* by a decreasing order according to the number of the connectors included; Check if *CntrSet* is null, if yes, go to step6;

Table 3
The process of generating the CSBAT of the test assembly

No	*AsmSet*	*CurAsm*	*CntrSet* and the value of SI	Selected connection
1	{*screw*1, *screw*2, *screw*3, *bolt*1, *nut*1, *block*1, *block*2, *block*3, *cap*1}	{*screw*1, *screw*2, *screw*3, *bolt*1, *nut*1, *block*1, *block*2, *block*3, *cap*1}	{*screw*1, *screw*2} 1.01398 {*screw*3} 0.93291	{*screw*1, *screw*2}
2	{*screw*3, *block*1, *cap*1} {*bolt*1, *nut*1, *block*2, *cap*3}	{*screw*3, *block*1, *cap*1}	{*screw*3} 1.18133	{*screw*3}
3	{*bolt*1, *nut*1, *block*2, *cap*3}	{*bolt*1, *nut*1, *block*2, *cap*3}	{*bolt*1, *nut*1} 0.92005	{*bolt*1, *nut*1}

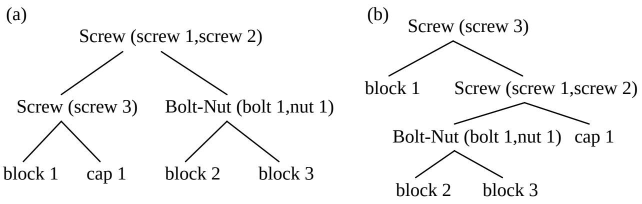

Fig. 8. Two CSBATs of the test assembly.

Step 4. Compute the selection indices SI for all grouped connectors, and select the group connector that has the highest SI value as the root of the current child CSBAT;
Step 5. Disconnect the nodes of *CurAsm* constrained by selected connectors from each other in CSBARM, and delete the nodes of these connectors and their edges, the disjoined sub-graphs are the children of the current child CSBAT; if the sub-graph has only one node, then the part is the leaf node of the child current CSBAT, else push the subassemblies into *AsmSet* and go to step2;
Step 6. Generate the *Insert* or *Mate* type child CSBAT by geometric reasoning and go to step2.

To illustrate the efficiency of the proposed algorithm, we describe an example of assembly planning for a test assembly shown in Fig. 6. The test assembly has been used by several researchers [21]. The product consists of nine parts, five of which are connectors: *screw*1, *screw*2, *screw*3, *bolt*1 and *nut*1. Fig. 7 shows its CSBARM, where there are four connector nodes: *screw*1, *screw*2, *screw*3 and *bolt*1-*nut*1. The process of generating the CSBAT is shown in Table 3, where the coefficient k_i of index has the average value. Certainly, the *screw*1 and *screw*2 can form a screw group {*screw*1, *screw*2}. By using the algorithm to generate the CSBAT, the connectors that can be disassembled are firstly grouped as two sets: {*screw*1, *screw*2} and {*screw*3}. Then the selection index of each connector group is computed and compared. Clearly, the subassemblies with respect to {*screw*1, *screw*2} and {*screw*3} are stable, therefore, the *STI* values of the CSBATs shown in Fig. 8(a) and (b) are zero. The CNI and PRI of the two CSBATs have the same value. However, the PI values of the CSBATs shown in Fig. 8(a) and (b) are 0.05556 and −0.27778, respectively. The SI of CSBAT in Fig. 8(a) is 1.01398, larger than that of the other. Therefore, the subassemblies with respect to {*screw*1, *screw*2} are selected to be subdivided in further. So {*screw*3, *bock*1, *cap*1} and {*bolt*1, *nut*1, *block*2, *cap*3} are pushed into stack *AsmSet* and subdivided with respect to {*screw*3} and {*bolt*1-*nut*1}, respectively.

The plans of the CSBAT in Fig. 8(a) can be retrieved from the *Standard Plan Base*, and the assembly sequences for the assembly are shown in Fig. 9.

7. Implementation

First of all, an architecture and database are designed to develop the concurrent and collaborative assembly sequence planning system. The database should include all information on products and resources for the product assembly. When the planner wants to prepare an assembly sequence, the planning tasks are assigned to different designers. In this research, MS IIS is used as a Web server, MS SQL server2000 as a DBMS, and PHP and Java script for creating interactive pages. The designers in separated environments can thus connect to this system through the Internet and carry out the planning tasks, respectively. During collaborative product design, both the design knowledge and expertise, which are distributed at geographically different locations, can be brought together into a common collaborative design space. The architecture of a prototype system, Co-ASP in short for collaborative assembly sequence planning system, that supports creations of a common collaborative planning space is proposed in this paper (Fig. 10) and includes the following modules.

Visualization Module. This module creates facet representation from the B-rep model. The facet representation is generated by approximating the faces of the B-Rep model by polygons, while maintaining edge consistency between adjacent faces. In addition, the module is in charge of role-based model simplification, and each client may request different data from the server according to the permissions associated with the role.

ASP Merger. In collaborative planning, the planning tasks are assigned to different designers and each designer obtains a local assembly sequence. ASP merger will merge all local assembly sequences for the product and generate a global assembly sequence.

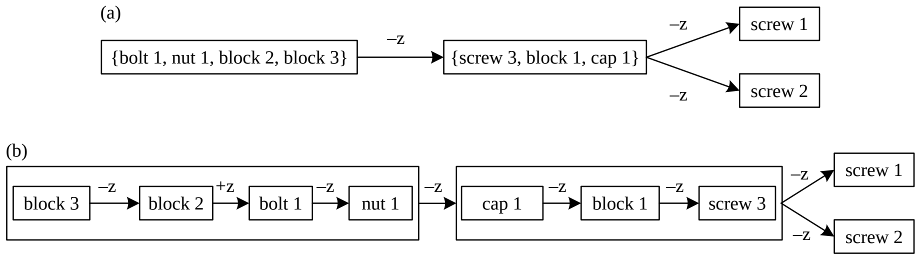

Fig. 9. Plan retrieval: (a) retrieve connection SCREW(*screw*1, *screw*2); (b) retrieve connection SCREW(*screw*3) and BOLT-NUT(*bolt*1, *nut*1).

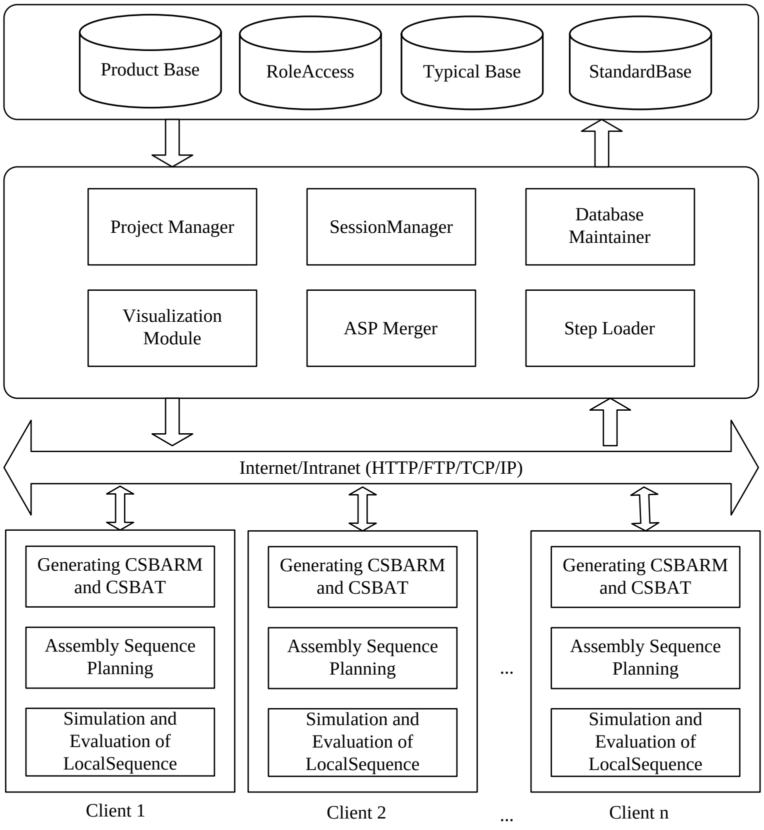

Fig. 10. System architecture of Co-ASP.

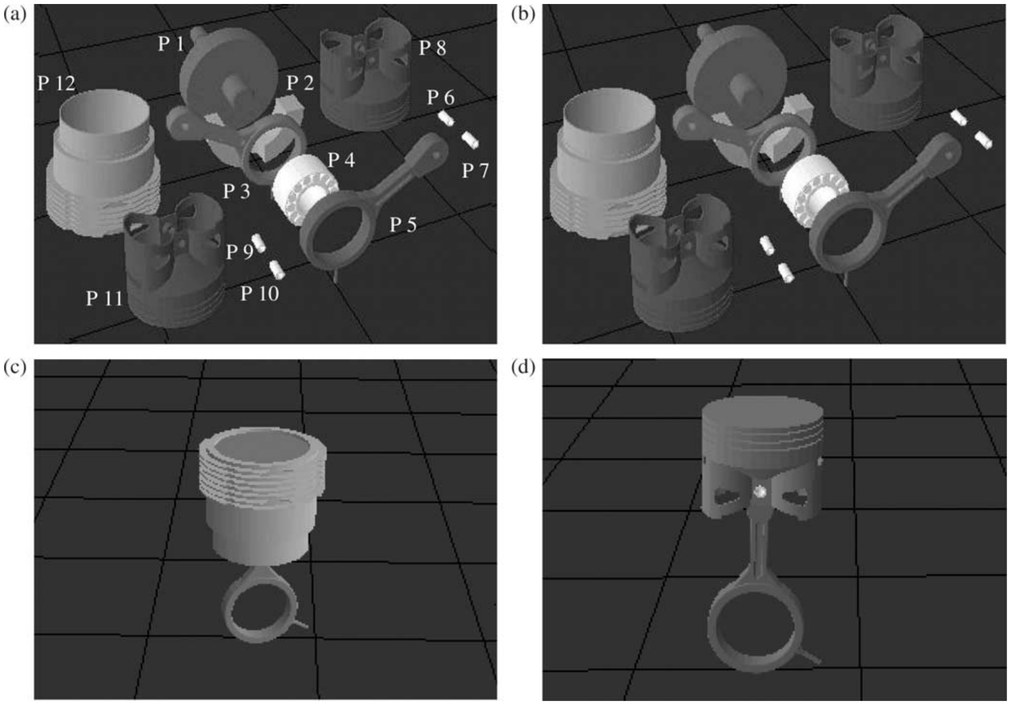

Fig. 11. Wave-hand: (a) its exploded view; (b) solid model for $r0$; (c) solid model for $r1$; (d) solid model for $r2$.

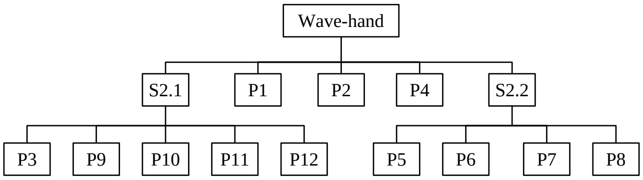

Fig. 12. Hierarchical structure model of wave-hand.

Database Module. This module is the layer between the server and the database that provides an interface to save and retrieve data from the database.

Project Management. This module sets up a project for collaboration. This module provides functions to support role creation and modification activities.

Step Loader. The STEP Loader is purported to upload STEP physical files from CAD designers' computers to the central server and then to parse these files to obtain the corresponding assembly models. These models are in turn mapped to the system database.

Session Management. This module manages collaboration, multiple session creation, transfer of editing control and maintaining a master copy of the model information.

In this paper, we present an example of collaborative assembly sequence planning for *Wave-hand* (Fig. 11(a)), which consists of 12 parts. According to its hierarchy (Fig. 12), *Wave-hand* is composed of three parts ($P1$, $P2$ and $P4$) and two subassemblies ($S2.1$ and $S2.2$) at the first layer. And the subassemblies $S2.1$ and $S2.2$ are composed of five and four parts, respectively. In this case, the system administrator creates three roles according to the job functions, grant permissions to these roles, and then assign three users to the roles. Table 4 is the access matrix for the assembly.

The role $r0$ is in charge of the planning of *Wave-hand*. *Write* permission to $P1$, $P2$ and $P4$ is assigned to $r0$, and therefore $r0$ can edit $P1$, $P2$ and $P4$. Of course, $r0$ sees the full resolution models of $P1$, $P2$ and $P4$. *Read* permission to $S2.1$ and $S2.2$ is assigned to $r0$, so $r0$ only can see the simplified model of these subassemblies. The role-based model for $r0$ is illustrated in Fig. 11(b). The role-based models for $r1$ and $r2$ are depicted in Fig. 11(c) and (d), respectively. For the original models of $S2.1$ and $S2.2$, there are 12,560 and 6759 facets, respectively. After role-based model simplification, there are only 6426 and 4281 facets remained. Finally, three local sequences are generated. The local assembly sequence gained by $r0$ is shown Fig. 13. Those of $r1$ and $r2$ are depicted in Fig. 13(b) and (c). Then the global assembly plan can be generated.

After generating assembly sequences, the user will simulate the assembly process in the virtual environment. In our research, the sequence chart is created as a specific file for assembly simulation. The sequence chart contains a 'start op' by default. The start op can reset the product assembly to its initial state before the first operation has occurred. The sequence chart is a fully interactive dialog box. This dialog box has a highlighted 'current' line. The current line determines the positions of the subassemblies and parts in the virtual environment. They will move to the positions defined through the currently highlighted sequence operation.

In order to analyze the efficiency of the approach proposed in this paper, a set of experiments was contrived using several industrial examples such as *toy Motor Grab* (Fig. 14) and *vacuum cleaner* (Fig. 15). These experiments compared the run-times with and without the use of simplified models. In these cases, we see exponential growth in the run-times without the use of simplified models. These simplified models can be considered as new components in other groups or assemblies. This type of representation, in addition to being intuitive, affords us a connection among operations or tasks to be done for the real assembly, by automatically generating the sequence of operations to achieve the assembly of a component or of a subassembly. For the complicated assemblies, the approach proposed in this paper is quite effective. It should be noted that the time spent in the simplification of models must be considered and the speed-up gained by the use of simplified models may be less dramatic than in these cases. But the simplified models can be used not only in assembly sequence planning but also in the collision detection. Most important of all, the simplified model can support the collaborative assembly planning. Thus, the influence of model simplification for complicated assemblies is great.

8. Conclusion

This paper has presented a new approach for collaborative assembly planning in a distributed environment. The environment is global, network-centric, and spatially distributed, which enables product designers to communicate more effectively, obtain, and exchange a wide range of planning resources during product development. It describes how the system architecture should be arranged

Table 4
Access matrix for the hierarchy of wave-hand

Roles	$S2.1$	$S2.2$	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$	$P8$	$P9$	$P10$	$P11$	$P12$
$r0$	*r*	*r*	*w*	*w*	–	*w*	–	–	–	–	–	–	–	–
$r1$	*p*	–	–	–	*w*	–	–	–	–	–	*w*	*w*	*w*	*w*
$r2$	–	*p*	–	–	–	–	*w*	*w*	*w*	*w*	–	–	–	–

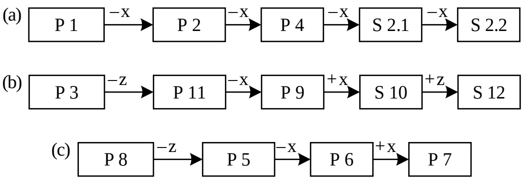

Fig. 13. Local sequences: (a) local assembly sequence for *r*0; (b) local assembly sequence for *r*1; (c) local assembly sequence for *r*2.

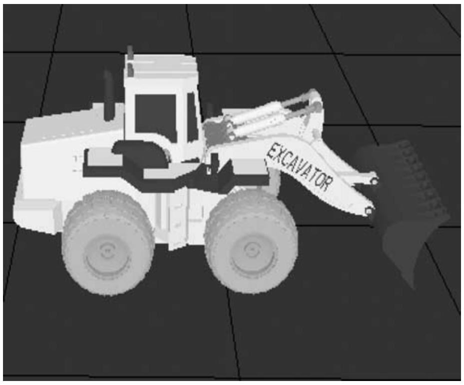

Fig. 14. Toy motor grab.

for cost-effective, flexible, and portable distributed modeling. In role-based model, the B-rep models are simplified according to the permissions associated with the role, so the surfaces invisible from outside of the model are removed. In collaborative planning, the planning tasks are assigned to several designers that carry out the collaborative planning, respectively. And a knowledge-based approach is proposed to the assembly sequence planning problem. The knowledge-based approach proposed in this paper solves the problem of assembly sequence planning by integrating geometry-based reasoning with knowledge-based reasoning. This realization assists in significantly reducing the complexity and amount of planning to determine the more feasible and practical sequences for the assembly. To verify the validity and efficiency of the approach, a variety of assemblies including some complicated products from industry are tested in our Co-ASP. Although the Co-ASP ingenerates a general purpose geometric reasoning with the knowledge about how to build specific structure, there remains much to do.

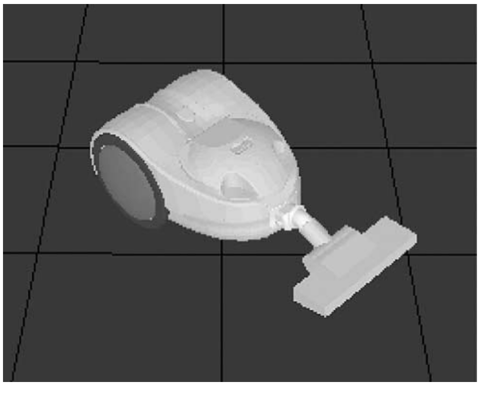

Fig. 15. Vaccum cleaner.

Acknowledgements

We gratefully acknowledge the support of the Special Funds for Major State Basic Research Program (973) of China under grant 2002CB312106. We would also like to thank the editors and the anonymous referees for their insightful comments.

References

[1] Shyamsundar N, Gadh R. Collaborative virtual prototyping of product assemblies over the Internet. Comput-Aided Des 2002;34(10): 755–68.

[2] Regli W. Internet-enabled computer-aided design. IEEE Internet Comput 1997;1(1):39–50.

[3] Gadh R, Sonthi R. Geometric shape abstractions for internet-based virtual prototyping. Comput-Aided Des 1998;30(6):473–86.

[4] Barnes CJ, Jared GEM, Swift KG. Decision support for sequence generation in an assembly oriented design environment. Robot Comput-Int Manuf 2004;20(4):289–300.

[5] Bidarra R, Kranendonk N, Noort A, Bronsvoort WF. A collaborative framework for integrated part and assembly. Trans ASME 2002;2: 256–64.

[6] Mori, T Cutkosky M. Agent-based collaborative design of parts in assembly. Proceedings of 1998 ASME design engineering technical conferences, Atlanta, Georgia, USA; September 13–16; 1998.

[7] Saar K. VIRTUS: a collaborative multi-user platform. Proceeding of the VRML'99 symposium. Germany: Paderborn; 1999 p. 141–52.

[8] Huang GQ, Mak KL. Design for manufacture and assembly on the internet. Comput Ind 1999;38(1):17–30.

[9] Kim KY, Wang Y, Muogboh OS, Nnaji BO. Design formalism for collaborative assembly design. Comput-Aided Des 2004;36(8): 849–71.

[10] Cutkosky MR, Tanenbaum JM, Glicksman J. Madefast: collaborative engineering over the Internet. Commun ACM 1996;39(9):78–87.

[11] Cutkosky MR, Genesereth MR, Mark WS, Tenenbaum JM, Weber JC. PACT: an experiment in integrating concurrent engineering systems. Comput IEEE 1993;26(1):28–37.

[12] De Fazio TL, Whitney DE. Simplified generation of all mechanical assembly sequences. IEEE Trans Robot Automat 1987;3(6):640–58.

[13] Wilson RH. Minimizing user queries in interactive assembly planning. IEEE Trans Robot Automat 1995;11(2):308–12.

[14] Homem de Mello LS, Sanderson AC. A correct and complete algorithm for the generation mechanical assembly sequences. IEEE Trans Robot Automat 1991;7(2):228–40.

[15] Baldwin DF, Abell TE, Lui MCM, Fazio TD, Whitney D. An integrated computer aid for generating and evaluating assembly planning. IEEE Trans Robot Automat 1991;7(1):78–94.

[16] Zhang B, Zhang L. A new algorithm for planning mechanical assembly sequences. Chin J Comput 1991;14(8):561–9 (in Chinese).

[17] Lee S. Subassembly identification and evaluation for assembly planning. IEEE Trans System, Man, Cybern 1994;24(3):493–503.

[18] Eng T, Ling Z, Walter O, Mclean C. Feature-based assembly modeling and sequence generation. Comput Ind Eng 1999;36(1): 17–33.

[19] Swaminathan A, Barber S. An experience-based assembly sequence planner for mechanical assemblies. IEEE Trans Robot Automat 1996; 12(2):252–66.

[20] Chakabarty S, Wolter J. A structure-oriented approach to assembly sequence planning. IEEE Trans Robot Automat 1997;13(1):14–29.

[21] Yin Z, Ding H, Li H, Xiong Y. A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. Comput-Aided Des 2003;35(1):37–56.

[22] Romeo M, Lee LHS, Abhary K. Assembly sequence planning and optimization using genetic algorithms: part I. Automatic generation of feasible assembly sequences. Appl Soft Comput J 2003;2(3): 223–53.

[23] Chen CLP, Pao YH. An integration of neural network and rule-based systems for design and planning of mechanical assemblies. IEEE Trans System, Man, Cybern 1993;23(5):1359–71.

[24] Zha XF, Du H. A PDES/STEP-based model and system for concurrent integrated design and assembly planning. Comput-Aided Des 2002; 34(14):1087–110.

[25] Koo S, Lee K. Wrap-around operation to make multi-resolution model of part and assembly. Comput Graph 2002;26(5):687–700.

[26] Hoppe H. Progressive meshes. Proceedings of SIGGRAPH, New York 1996 p. 99–108.

[27] Hussain M, Okada Y, Niijima K. Efficient simplification of polygonal surface models. Proceedings of the fifth international conference on information visualisation, London, England 2001 p. 464–9.

[28] Zhu H, Menq CH. B-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. Comput-Aided Des 2002;34(2):109–23.

[29] Sandhu RS, Coyne EJ, Feinstein HL, Youman CE. Role-based access control models. IEEE Comput 1996;29(2):38–47.

[30] Han J, Kim T, Christopher DC, William CR. Multi-resolution modeling in collaborative design. Lecture Notes Comput Sci 2003;2869:397–404.

[31] Dong T, Tong R, Dong J, Zhang L. Knowledge-based assembly sequence planning system. In: Proceedings of the eighth international conference on computer supported cooperative work in design, Xiamen, China, vol. 2; 2004, p. 516–21.

[32] Kanai S, Takahashi H. ASPEN: computer-aided assembly sequence planning and evaluation system based on predetermined time standard. Ann CIRP 1996;45(1):35–9.