Complex Assembly Variant Design in Agile Manufacturing.
Part I: System Architecture and Assembly Modeling Methodology

Aihu Wang
Department of Management Science and Engineering, School of Business Administration
South China University of Technology
Guangzhou, Guangdong, P.R.China 510640

Bahattin Koc, and Rakesh Nagi[*]
Department of Industrial Engineering, 342 Bell Hall
University at Buffalo (SUNY)
Buffalo, NY 14260, USA

March 31, 2004

Abstract

In the distributed and horizontally integrated manufacturing environment in Agile Manufacturing, there is a great demand for new product development methods that are capable of generating new customized assembly designs based on mature component designs that might be dispersed at geographically distributed partner sites. To cater for this demand, this paper addresses the methodology for complex assembly variant design in agile manufacturing. It consists of fundamental research in two parts: *assembly modeling and assembly variant design methodology*. This paper, the first of a two-part series, presents the assembly variant design system architecture and the assembly modeling methodology. First, complementary assembly modeling concept is proposed with two kinds of assembly models, the Hierarchical Assembly Model and the Relational Assembly Model. The first explicitly captures the hierarchical and functional relationships between constituent components while the second explicitly captures the mating relationships at the form feature level. These models are complementary in the sense that each of them models only a specific aspect of assembly related information while together they include required assembly related information. They are further specialized to accommodate the features of assembly variant design. As a result, two kinds of assembly models, the Assembly Variants Model (AVM) and the Assembly Mating Graph (AMG), are generated. These assembly models serve as the basis for assembly variant design which is discussed in the second part.

Keywords: Variant Design, Agile Manufacturing, Assembly Modeling.

---

[*] To whom correspondence should be addressed, nagi@buffalo.edu, Tel: 716-645-2357x2103, Fax: 716-645-3302

1 Introduction

Having experienced Mass Production, Lean Manufacturing, and Follow the Leader, American manufacturing industry has come to a novel manufacturing paradigm: Agile Manufacturing. Agile manufacturing characterizes a different form of industrial competition where the traditional roles of competitor, supplier, and customer firms may frequently change to take advantage of opportunities in the marketplace [2]. To keep a competitive edge in the new manufacturing environment, the enterprises are highly desired to have the ability to capture the customer demands accurately, to design and manufacture swiftly and to delivery the final products to the market timely. Owing to the various constraints such as capacity and resource limitations, inadequate technological support and even the inherent nature of the organization and strategy of the enterprise, the worldwide business opportunity may not be exploited by a single enterprise. Fortunately, the high performance computer networks and advanced information technologies allow geographically distributed companies to collaborate electronically. Products that might not be feasible for companies to produce individually become feasible when the same companies share data and technology. This results in the new organizational structure: the virtual enterprise.

This manufacturing paradigm shift brings fundamental changes to the way enterprises do business. In turn, the new way of doing business demands new product design and manufacturing methodologies. Under agile manufacturing, how to develop new or customized products effectively and efficiently based on available resources (mature designs, process plans and related information) that might be dispersed on different partner sites becomes more important. Variant design is a common practice introduced to relieve designers from iterating similar design processes, shorten product development, reduce cost, and finally enable manufacturing companies to develop individualized products based on existing mature designs. However, literature review shows that most current variant design methods are focused on the redesign of a single component or a set of components (see Part II). There lacks significant effort in developing complex assembly variant design methodologies though most successful manufacturing products turn out to be highly complex assemblies. This motivates us to develop a systematic assembly variant design methodology that serves the virtual enterprises by establishing new and customized assembly designs, based on mature component designs.

The methodology developed is composed of two parts: *assembly modeling and assembly variant design methodology*. Assembly modeling is essential to unambiguously capture the structure and relationship of components in complex assemblies. However, literature review shows that there are no universally accepted assembly modeling methods. Researchers either generate higher level assembly models from the CAD models or lower level assembly models, or establish assembly models that supposedly contain all assembly related information. The first approach is a nontrivial and error-prone procedure while the latter approach is challenging in designing the structure of the assembly model. Instead of generating assembly models from CAD models or squeezing all assembly related information into a single model, two kinds of complementary assembly models, the Hierarchical Assembly Model and the Relational Assembly Model, are proposed to be used. The first explicitly captures the hierarchical and functional relationships between constituent components while the second explicitly captures the mating relationships at the form feature level. These models are complementary in the sense that each of them models only a specific aspect of assembly related information while together they include required assembly related information. They are further specialized to accommodate the features of assembly variant design. As a result, two kinds of assembly models, the Assembly Variants Model (AVM) and the Assembly Mating Graph (AMG), are generated. Assembly variant design methodology will be covered by the second paper of this two-part series [29].

The paper is organized as follows. Relevant literature is reviewed in Section 2. Section 3 describes the architecture of the assembly variant design system. The complementary assembly modeling method is proposed and instantiated in Section 4. Finally, the concluding remarks are provided in Section 5.

2 Literature Review of Product Assembly Information Modeling

Product information modeling is the logical accumulation of all product information during its life cycle, which includes data, structures, access mechanisms and manipulation algorithms. Owing to the complexity of the assembly product and heterogeneous activities involved in the product life cycle, assembly product development is subdivided into the development of components and finally the assembly of the components. Correspondingly, two types of product data models: component model and assembly model, are resulted.

Component modeling takes place in the parametric design stage. The major function is to represent the geometry and tolerance of Components, perhaps with the help of a CAD system, where it is called geometric modeling. Geometric modeling primarily focuses on the modeling (capture the data and structure) of a single component and the ultimate purpose is to represent the design specifications. Constructive Solid Geometry (CSG) and Boundary Representation (B-rep) are the two representative component modeling methods. Such models are good at efficient and unambiguous geometric modeling single piece parts, however complex and irregular their shape may be. They are weak at supporting assembly design information. Even though commercial CAD packages are beginning to support simple assembly modeling issues, only limited assembly modeling capability is available at present.

Contrasted to component modeling, assembly modeling concentrates on the representation of the relationships among the components of an assembly product, and the ultimate purpose is to guide the assembly tasks (assembly sequence generation and evaluation, tolerance chain analysis, etc.) in order to assemble the final assembly product. It is less mature and no commonly accepted data structures and modeling methods exist. Practical assembly modeling methods fall into the following two categories: general assembly models and task specific assembly models.

2.1 General Assembly Models

Similar to CSG and B-Rep, these kinds of models propose a data structure that captures not only the topological and geometrical information of components but also the mating conditions between components. Lee and Gossard [12] propose a general hierarchical data structure for representing assemblies. The relationships between the components in an assembly are represented using the “virtual links”. A virtual link is the complete set of information required to describe the relationship (e.g., rigid attachment, conditional attachment, translational constraint, and rotational constraint) and the mating features between the mating pair. The proposed data structure is directly used by Lee and Andrews [11] to infer the positions of components in an assembly. A modified version of the data structure is used by Kim and Lee [9] to perform dynamic and kinematic analyses.

Gui and Mäntylä [5] develop a general top-down assembly model that can provide both a function-oriented view and a module-oriented view of the assembly structure. The former relates to functional and behavioral knowledge and the latter to manufacturing knowledge. The objective is to support both conceptual design at high levels of abstraction and feature modeling at low levels.

2.2 Task Specific Assembly Models

This category of assembly models target at solving various kinds of domain specific problems. The feature of the assembly model is that only task related information is captured and information is organized in such a way that the tasks can be solved efficiently by manipulating the assembly model. These kinds of models can be further divided into the following types:

1. **Assembly Modeling for Automatic Tolerance Chain Generation:** Wang and Ozsoy [30] develop a scheme of representing assemblies so that tolerance chains of assemblies can be generated automatically in order to facilitate tolerance analysis at the assembly level. It is a hierarchical data structure where the connectivity information is carried by the instances of components and subassemblies, and the mating relations between each pair of mating entities are described by mating links, mating paths, mating conditions, and mating features.

2. **Assembly modeling for assembly and task planning:** Thomas *et al.* [25] classify the commonly used assembly modeling methods for task planning into two categories.

    (a) **Implicit Representation Scheme:** There is no direct mapping from the assembly tasks into the elements of the representation. Instead, these types of representations consist of conditions (precedence relations, etc.) that must be satisfied by the assembly sequences.

    (b) **Explicit Representation Scheme:** There is a direct mapping from assembly tasks into elements of the representation such as tree hierarchy, AND/OR tree, precedence graph and Petri nets.

Thomas *et al.* [26] present a boundary graph representation to handle geometric information for assembly. Each assembly part is represented by a boundary graph. The assembly operations joining two parts together can be modeled simply as the splicing of two corresponding boundary graphs. The authors claim that the boundary graph representation allows fast feasibility evaluation. The main issue to be solved is the mapping of current boundary representations of complex objects on to the data structure proposed.

Tseng and Liou [28] develop a new graph-based representation model to integrate assembly and machining planning. First, an assembly-machining operation graph is developed to represent the spatial relationships between the components as well as to express the operational precedence of the machining and assembly operations. Next, the integrated assembly and machining sequences are generated using a tree structure called the assembly-machining sequence tree. Using a graph-based methodology, all the feasible integrated assembly and machining sequences can be generated and evaluated. The main objective is to provide a complete model for integrating assembly and machining sequences.

3. **Assembly modeling for articulated mechanisms:** Rocheleau [19, 20] proposes assembly modeling methods for articulated mechanisms that facilitate the kinematic and dynamic analyses.

The above survey shows that most assembly modeling methods use boundary representation (B-Rep) to represent the components of the assembly. But as to the methods to capture the relationships between components, they are different according to the nature of the tasks and the characteristic of the assembly products. No unique method is universally accepted. Some such efforts on unified assembly modeling can be found in the works of Whitney [31], [32], and Srikanth and Turner [24].

2.3 Critique of Existing Assembly Modeling Methods

The literature review shows that the existing assembly modeling principles have the following limitations.

Bottom-up Approach: Many assembly modeling methods adopt a bottom-up approach [24]. Based on the component CAD models, the mating relations are input by the designers manually so that the relational assembly model is established. Then the hierarchical model is created by manipulating the relational model whenever the hierarchical model is necessary. The modeling procedure is a bottom-up approach. The authors believe that the bottom-up assembly modeling approach makes the modeling process unnecessarily complex since the hierarchical assembly model should have been specified at the end of conceptual design stage. In addition, it is also time consuming and error-prone to derive the hierarchical model from the relational model.

Intermingling mating relationships with the functional relationships: Assembly relationships among components can be classified into two categories: mating relationship and hierarchical relationship. The literature review shows that a relational graph is appropriate to model the mating relationships while a hierarchical tree structure is good at modeling the hierarchical relationships. In reality, the two kinds of assembly relationships are seldom distinguished and most assembly modeling methods try to cover the two different kinds of relationships into one single model. As a result, only one of the two kinds of relationships can be modeled explicitly. If needed, the other kind of relation has to be derived by manipulating the single assembly model. For example, Lee and Gossard [12] model the mating relations explicitly using virtual links in their assembly model and the hierarchical relations are not captured. Later Ko and Lee [10] develop a method to order a set of components related by virtual links into a hierarchical tree which they use to generate an assembly procedure. In contrast, Wang and Ozsoy [30] model the hierarchical relation explicitly and use mating links to store the mating relations. The explicit mating relations between components have to be obtained by manipulating the mating links.

Unified Assembly Modeling: It is a trend to establish unified assembly models. Basically, there are two approaches. The first approach can be named as collective approach which squeezes all assembly related information into a single model so that different kinds of users can glean whatever they need out of the single model. The work introduced in [31, 32] belongs to this approach. The second approach can be named as generative approach which starts from component CAD models and derives the relational model from them and then further derives the hierarchical model from the relational model. Srikanth and Turner's [24] approach belongs to this category. Both approaches are in their early stages. For the collective approach, it is challenging in how to integrate the heterogeneous assembly information into the unified model systematically. For the generative approach, how to maintain the efficiency of the assembly model is a challenge.

The authors believe that it is favorable to model an assembly at different levels and from different perspectives. As a result, multiple models are generated. Here, two key issues need to be addressed. First, the perspectives should be carefully selected so that the assembly models can be complementary to each other. Though there may be overlaps between models, different assembly tasks should be able to be carried out by focusing mainly on one or a few of the several models. Second, it is important to keep the structure of the assembly models open so that designers can accommodate new assembly features that are not covered by existing models. The basic idea is to model those most prominent assembly information explicitly using various models so that different assembly tasks can be expedited by manipulating corresponding models. Since assembly related information is stored in different models, it is important to keep the consistency among them. This issue is addressed in Sections 4.2.3 and 4.3.3.

3 Assembly Variant Design System Architecture

In the agile manufacturing paradigm the product development activities may be dispersed at different partner sites. A comprehensive view of the assembly product development process is maintained in this paper because the approach to capture and model assembly product information and the establishment of assembly variant design method are emphasized. However, the decentralization of product development process and the distributed nature of the manufacturing environment can always be assumed.

Fig. 1 describes the overall architecture of the assembly variant design system. The first column shows the heterogeneous databases that contain the available resources for the variant design. The second column outlines the assembly product data modeling procedure which aims at capturing the compositional/functional and mating relationships among constituent components. The third column displays the complementary assembly models which are the output of the second column. Finally, column four sketches the assembly variant design procedure. The heterogeneous databases are supposed to be known and might be dispersed at different partner sites. Since our attention is focused on how to represent known information so that the assembly variant design can be expedited, the procedure or method to generate the assembly models is beyond the paper's scope. There are different approaches to establish the assembly variants model from the heterogeneous databases. The procedure described in the top part of column in Fig. 1 is proposed by Ramabhatta [16, 17]. A new data mining approach to the same problem has recently been developed by Romanowski [21, 22]. First, Pareto analysis is used to identify major assembly variants which are classified into different categories based on the key assembly design characteristic values. The hierarchical structure of each assembly variant is then identified and represented as a generic bill of material (GBOM) using object-orientation. Finally, the attributes are collected and the ultimate assembly variants model is resulted. The procedure of obtaining the mating graphs is left for further research. It is not the intention of this paper to develop procedures to generate the assembly variants model and assembly mating graphs from heterogeneous databases. The contribution is to propose the complementary assembly modeling concept which models mating relations explicitly at the form feature level and to establish the variant design method which is targeted at complex assemblies. These two parts are outlined in the third and fourth column.

3.1 Assembly Modeling

Noticing that current assembly modeling techniques emphasize either on the relational aspect or the hierarchical aspect of the assembly, the complementary assembly modeling concept is proposed. Two kinds of complementary assembly models, hierarchical assembly model and relational assembly model, are employed to explicitly model hierarchical and relational relationships respectively. The first is designed to capture the hierarchical and functional relationships among components while the second for the mating relationships. The two kinds of models are further specialized to suit the need for assembly variant design

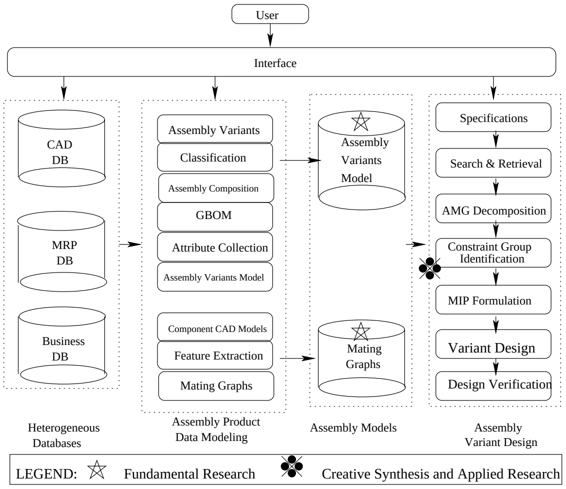

Figure 1: The System Architecture

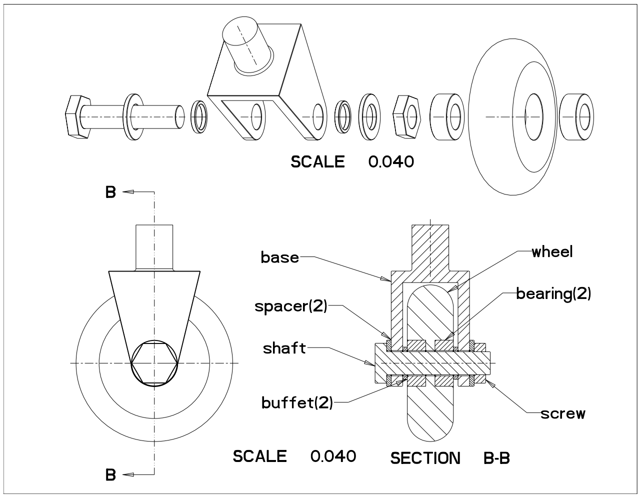

Figure 2: Wheel Support Assembly (Low Speed)

and this results in the assembly variants model (AVM) and assembly mating graphs (AMG). The AVM and AMG form the foundation of the assembly variant design. The second column in Fig. 1 briefly outlines the assembly modeling procedure. The AVM is obtained by the following procedure. First, the representative assembly variants are identified and then classified by the Key Assembly Design Characteristics which are parameters that specify the fundamental assembly features such as capacity, load and speed. Then the composition of each assembly variant is identified and the Generic Bill of Material (GBOM) is abstracted. Finally the AVM is created by collecting related attributes. This procedure can be partially automated. For the AMG, the form features of each component are either available or extracted from the CAD models. Then the mating graphs are created by the help of the designer. Since form feature extraction is a very complex procedure and many problems need to be solved, the AMG is developed manually at present. In addition, the paper's focus in assembly modeling is not on how to generate the AVM and AMG automatically from the component CAD model. Instead, it is concerned with proposing the representational schemes to capture the compositional/functional and mating relationships so that the assembly variant design can be facilitated.

3.2 Assembly Variant Design

The assembly variant design is performed by manipulating the assembly variants model and assembly mating graphs. The matching components are searched and retrieved from the AVM and then the Constraint Groups (CG) are identified by manipulating the AMGs. Finally, the assembly variant design process is formulated as a Mixed Integer Programming (MIP) problem which is solved using a standard solver.

It is challenging to model assembly related information and create new assembly designs based on mature component designs. The assembly variant design process is a complicated decision making procedure. Brown and Chandrasekaran's statement best describes the nature of assembly variant design.

> “The choice may be simple at each point in the variant design, but overall the task is still too complex to be done by merely looking in a design database because there are too many possible combinations of initial requirements. Simple choices do not imply simple designs or a simple design process [1].”

4 Complementary Assembly Modeling Methodology

As stated in Section 2, current assembly modeling methods are application oriented and unified assembly modeling techniques are still in a conceptual stage. Though the research is targeted at the assembly variant design, it is not favorable to develop assembly models that are only adequate for the variant design since successive research may involve assembly planning and disassembly analysis. This motivates us to establish open assembly models that capture the fundamentals of assembly products and that can be further specialized to suit different kinds of assembly related applications. This section discusses the open data structures that capture the fundamentals of assembly products.

4.1 Three-level modeling

There is no doubt that assembly modeling should be based on individual component modeling. In addition, the relationships among components can be further classified into two distinguishable categories. The first category captures the detailed mating relationships among components while the second captures the compositional or hierarchical relationships among components. Therefore, the information captured by the assembly models can be divided into three layers: individual component model, mating relationships among individual components, and the hierarchical relationships among individual components. The three-layer information results in the three categories of product information models: CAD models, Relational Models and Hierarchical Models. From an assembly modeling perspective, CAD models are in the lowest level while the hierarchical models are in the highest level. The following sections further elaborate each of these layers.

4.1.1 CAD Models

CAD models represent individual components as if they were not related at all [24]. Constructive Solid Geometry (CSG) and Boundary Representation (B-rep) are the two representative component modeling methods. Both methods are good at representing the geometry and specifications and even tolerance [4] of the components. But as far as assembly modeling is concerned, B-Rep is more preferred. This is because the higher level assembly modeling, relational modeling, is focused on the relationships of the boundary features of components. Hence, it is more advantageous to represent components by B-Rep rather than CSG. Though data structures other than B-Rep and CSG, *e.g.*, winged edge data structure [12], are also used to represent individual components, B-Rep is the most widely used [26, 30]. It is also used in STEP ISO 10303 standard for product data exchange. This work assumes that CAD models are available in some neutral B-Rep format, and therefore is not restricted to a particular CAD system.

While CAD models contain the representation of design information as geometric and topological entities, these entities must be typically reinterpreted for the purpose of different down stream activities (manufacturing, assembly, etc.). Reinterpretation of manufacturing and assembly information as *form features* in a neutral format is currently an active research area.

Without getting into the complex notation for a formal mathematical definition, a feature can be defined as characteristic of the part which carries significance or higher semantic meaning to a particular application. These various applications could be manufacturing, engineering, design, assembly etc. The meaning of the term feature as it may apply to these disciplines is provided below.

In works of Shah and Rogers [23], the term *feature* was defined as a set of information related to an object's description. This description could be for design, for manufacturing or even for administrative purposes. The authors have classified features into sets related to product engineering applications as follows:

- form features: which identify the combination of geometric and topological entities in such a way that it makes practical sense during the various stages of the products life cycle; for instance, shoulder and boss are examples of form features which are important during design and manufacturing.
- assembly features: which assist in the easy location/mating of parts for assembly, e.g. holes, slots, etc.;
- material features: which specify material composition and condition information such as properties/specification or treatment applied to materials and surfaces;
- tolerance features: such as geometric tolerances or surface finish; and
- functional features: such as performance parameters, operating variables or design constraints, e.g. the aerodynamic shape of the wing of an airplane.

In summary, *form features* are configurations on the object that may be used for engineering-analysis during design, for process planning during manufacturing and during assembly. The same configurations could carry different connotation to different applications such as design and manufacturing. For example holes, pockets and steps are types of form features that are represented as a set of surfaces during design and can be associated with manufacturing activities like drilling, end milling and slab milling. The advantage of working with form features in this research is that a number of objectives can be simultaneously met. (1) Specific geometric entities/surfaces of two form features belonging to different component parts that make contact in an assembly can be gleaned. (2) Key Component Design Characteristics (KCDCs, see Part II [29]) can be extracted. (3) design and manufacturing cost and time changes for dimensional change of a form feature KCDC can determined because it is associated with a design and manufacturing interpretation (in Part II [29]). If form features are not available, these objectives can still be met but with more human effort.

4.1.2 Relational Models

The relational models are based on the CAD models and are focused on the representation of the mating relationships among components. They are usually represented as graphs. In the graphs, the nodes represent the components while the arcs represent the mating relations. There may be only one arc or more than one arcs for each pair of components. The arcs store the homogeneous coordinate transformation that specifies the relative position of one component with respect to the other. As a result,

the transformation matrices for relative positioning can be derived from the mating relationships. Usually, the relational assembly model is used together with boundary representation of individual components to model the assembly product in order to perform assembly related tasks such as assembly sequence generation. One prominent feature of the relational models is that the user must manually specify the mating relationships most of the times. The remaining part of this section will focus on the types of mating relationships that have been studied.

Three aspects about mating relations should be emphasized. They are: (i) the relationship between the type of mating relations, (ii) the concept of Degree of Freedom (DOF) and (iii) the taxonomy used to describe the types of mating conditions, and types of mating conditions considered.

Since each mating pair between a pair of components usually takes away at least one DOF, it is natural to associate the type of mating relationships with the DOF. Morris and Haynes [14] describe 20 different mating feature relationships in terms of the DOF constrained by each pair of mating features. But in reality, only a few types of mating relations (much less than 20) are distinguished. For example, in their initial attempt, Lee and Gossard [12] only deal with two kinds of mating relations: *Against* and *Fit*. This proves that the DOF is not the only factor that distinguishes mating relations. Literature review shows that researchers are more concerned about the way mating features contact with each other rather than DOF.

The second and third aspects of mating relations are the taxonomy and scope of mating conditions considered. The taxonomy and scope vary from case to case. Lee and Gossard [12] only deal with two kinds of mating relations and name them as *Against* and *Fit*. Lieberman and Wesley [13] describe relations using *part-of*, *attachment*, *constraint* and *assembly*. Obviously, no standard taxonomy is available in describing the types of mating relations. In despite, the mating relations from three perspectives: contact, DOF and

Attachment	Non-Attached		Attached
Contact	Planar Contact (Against)		Rigid Against
	Coaxial Contact	Coaxial Fit	Rigid Fit
		Spherical Fit	
		Screw Fit	
	Gear Contact		
	Rack-and-pinion Contact		
DOF	Non-Rigid (DOF>0)		Rigid (DOF=0)

Table 1: Types of Mating Relations

attachment are summarized in Table 1.

Essentially, mating is concerned with the way components contact with each other. Therefore, it is straightforward to distinguish mating relations in terms of the way components interact or contact. As shown in Table 1, the mating relations can be classified into planar contact (commonly known as against), coaxial contact, gear contact, and rack-and-pinion contact. The coaxial contact can be further divided into coaxial fit, spherical fit and screw fit. Both the classification and the meaning of each kind of mating relation are self-explanatory. Though this classification is not exhaustive, it covers most of the mating relations found in common practice or mentioned in the literature.

The second aspect of mating relations to understand is the DOF. If a pair of mating features completely constrain the two components (leaving no relative motion and DOF=0), this kind of mating relation is called rigid mating relation. Otherwise, it is called non-rigid mating relation. As indicated by Table 1, the basic types of mating relations are non-rigid while those with attachment operations are rigid. One important issue of assembly analysis is about the tolerance analysis. The non-rigid mating relations can be further divided according to the ASME Y14.5M-1994 tolerancing standards [3].

Each of the classes of mating relations defined above takes away at least one DOF. The relative positions of two components are still not completely constrained. In certain cases, the two components need to be fully constrained with no relative motion by operations such as riveting, soldering, tight fitting, no matter how the components actually contact with each other. This kind of operation is defined as attachment. The attachment operation can be used together with the major types of mating relations defined above. For example, against plus attachment results in rigid against while coaxial fit plus attachment results in rigid fit as shown in Table 1.

Though the classification is not exhaustive, it proposes a way to understand common mating relations. All the mating relations can be interpreted from the three aspects: contact, DOF and attachment.

4.1.3 Hierarchical Models

Except the mating relationships among components, there are functional and compositional relationships among components. The entire assembly product can be separated into different functional units that are usually taken as subassemblies. The subassemblies in turn may consist of other subassemblies or individual components. The functional relationships among components are usually not captured by the relational models though many researchers try to capture them in their relational models by using “part-of” and “assembly” mating relationships classification [13]. The functional relationships among components are manifested by the hierarchy of the assembly product and is usually represented by a tree structure which is called the hierarchical model of the assembly. AND/OR graphs and Petri nets [25] are widely used to represent the hierarchical model.

The assembly hierarchy implies a definite assembly sequence. Hence the hierarchical assembly model can be used to guide the assembly planning. In addition, it is also useful for enterprise resource planning and production planning [15].

4.2 Complementary Assembly Models

Based on the component CAD models, another two kinds of assembly models are necessary to model assembly related information. They are the hierarchical assembly model and the relational assembly model.

4.2.1 The Hierarchical Assembly Model

The hierarchical assembly model is defined as the tree structure in which the root node of the tree represents the assembly product, the intermediate nodes represent the subassemblies and the leaf nodes represent the single components. In addition to the hierarchical relationships, it also captures the functional relationships among different parts of the assembly. The leaf nodes in the hierarchical assembly model represent the single components and hence are connected with corresponding CAD models. In this way, the hierarchical level assembly information is captured and is linked with component CAD models. The concept of hierarchical models is not new since Bills-Of-Materials can be considered as a special structure of hierarchical assembly model. In this research, the hierarchical concept has been extended to a richer modeling construct that might be helpful in many other design/manufacturing activities, e.g., configuration design, classification and coding, and cellular manufacturing.

Theoretically, the hierarchical assembly model is the final result of the assembly conceptual design stage in the top-down assembly design approach. Since the conceptual design stage is the least computer supported and documented design stage, this model is usually established in a bottom-up fashion. For example, Ramabhatta *et al.* [16, 17] propose an approach to abstract the hierarchical assembly model from representative assembly products and model it using object-oriented method. This model captures the compositional assembly relations and hence is good for the production planning, resource planning and even assembly planning. But it is not enough for tolerance analysis, disassembly analysis and other assembly tasks that require the knowledge of detailed mating relationships among components. This is the reason for developing the relational assembly model.

4.2.2 The Relational Assembly Model

One limitation of traditional assembly relational models is that the mating conditions are modeled at a very rough level. They show in which way components mate (component A is fit or against with component B) but they do not show how the components mate (which surface of component A is fit or against with which surface of component B). The intention for establishing the relational assembly model is to explicitly capture the assembly mating information at the form feature level which is usually contained in various assembly relational models. The following two factors explain why mating conditions need to be modeled at the form feature level in this research.

- **Design and Form Features are Different:** The features used by CAD packages such as Pro/Engineer to model three dimensional objects are different from manufacturing/form features and there is no simple one-to-one mapping between them. In Pro/Engineer [27], everything is represented as a feature. For example, the coordinate system is a feature and so does a datum plane. It is likely that the geometry of a complex component be modeled using only one feature, e.g., the buffet in Fig. 3 can be modeled as plane and curved surfaces. The features in Pro/Engineer have to be re-interpreted from manufacturing/assembly perspective if they are to be used in downstream manufacturing and assembly activities. This is exactly the function of feature extraction. As stated in Sections 2 and 4.1.1, feature extraction itself is a broad research area.
- **To Facilitate the Evaluation:** The purpose for assembly modeling is to facilitate assembly analyses such as tolerance chain analysis, assembly planning, variant design and disassembly analysis. Often times, different options (tolerance assignment, assembly plans, etc.) need to be evaluated and compared. The evaluation is usually form feature based or related. For example, tolerance analysis needs to consider the impact of different tolerance assignments toward the manufacturing cost which is manufacturing feature based. Variant design needs to consider the impact of different design schemes toward the incurred manufacturing and assembly costs which are also manufacturing feature based.

In summary, the assembly applications usually take form feature-based information as input which is very difficult to be derived from the component CAD models directly. To promote the performance of assembly applications, it is beneficial to model this kind of information explicitly. The observations motivate this research to model the mating relationships explicitly at the form feature level.

Though it is beyond the scope of this paper, the basic procedure of obtaining the relational assembly model can be outlined as follows which is shown in Fig. 1:

- **Feature extraction:** Taking the component CAD geometric models as input, re-interpret the designs from manufacturing/assembly perspective and then obtain the list of form features of each component.
- **Construct the relational assembly model interactively:** Taking the geometric features (e.g., surfaces, edges and vertices) of each component as input, the designer constructs the relational assembly model by establishing the links between components based on actual mating conditions.

A relational assembly model is defined as a 4-tuple $g =< V, E, \alpha, \beta >$ where

- V is a finite set of nodes (or components);
- $E : V \times V$ is a finite set of edges (or component relationships);
- A is the set of node attributes;
- B is the set of edge attributes.
- $\alpha : V \rightarrow A$ is a function which associates a set of node attributes with each node;
- $\beta : E \rightarrow B$ is a function which associates a set of edge attributes with each edge;

The openness and flexibility of the relational assembly model are guaranteed by the following two strategies:

(1) Flexible definition of the attribute sets: Set A contains the node or component attributes. It can be the number of mating features of each component, the material of the component, the functional description of the component and so on. No matter how may kinds of component attributes in which the designers are interested, they can always be captured by further expanding the set A as an n-tuple, *i.e.* $A = \{A_1, A_2, ..., A_n\}$. n corresponds to the types of component attributes of interest. Similarly, set B contains the edge attributes and it can be the weight of the edge, the type of mating conditions and even the tolerance of corresponding mating pair. Therefore, the same approach can be used to expand the scope of the edge attributes.

(2) Flexible definition of the functional mappings: In this relational model definition, only two kinds of functional mappings are defined. If needed, more functional mapping can be defined to capture the relations between the subsets of A, B, V and E.

For the relational assembly model, an incidence matrix can be used as the mathematical representation of the corresponding graph. The relational assembly model can be used to expedite the disassembly analysis, tolerance analysis, assembly constraint analysis and assembly variant design.

4.2.3 The consistency of the hierarchical and relational assembly models

It is always a challenge to keep the consistency of the product data models. Different models are inter-related and hence the information changes in one model may incur the changes of other models. The following items outline the guidelines as how to keep the consistency of the hierarchical and relational assembly models.

- **The nodes of the relational assembly model are the leaf nodes of the hierarchical assembly model:** Though the nodes of the relational model may also be standard subassemblies, they correspond to the leaf nodes in the hierarchical model. The intermediate nodes in the hierarchical assembly model correspond to the subassemblies and represent the compositional and functional relationships between the atom components.
- **Hierarchical assembly model is not unique:** The same assembly product may be represented using different hierarchical models by different designers. Therefore, the changes of the hierarchical assembly model might not incur changes in the corresponding relational assembly model.
- **Relational assembly model is unique:** For each representative assembly variant, the relational assembly model is unique. It represents the actual mating relationships among atom components. Therefore, it is likely that the changes of relational assembly model incur the appropriate changes of corresponding hierarchical assembly model.

In summary, the three level assembly information, component information, mating relationships and the compositional relationships, is captured by the two kinds of assembly models which are based on the component CAD models. In Section 4.3, it will be shown how the two kinds of complementary assembly models can be specialized to facilitate the variant design of complex assemblies.

4.3 Assembly Models for Variant Design

This section demonstrates how the assembly models proposed above can be specialized so that the assembly variant design can be facilitated.

4.3.1 Assembly Variants Model (AVM)

Assembly variant design is based on existing representative assembly products. These assembly products can be classified into different categories (assembly variants) according their key assembly design characteristics. These characteristics can be defined as design, functional, or performance attributes associated with the assembly or product as a whole. They usually distinguish product families through classification. For example, Figs. 2, 3, and 4 show three kinds of wheel supportive assemblies classified based on the key assembly design characteristics "speed" and "load".

Notably, different assembly variants might have different compositional structure. Therefore, the major assembly variants need to be identified and their compositional structures are to be modeled. This can be done by further specifying the hierarchical assembly model to cover the major assembly variants. As a result, the Assembly Variants Model (AVM) shown in Fig. 5 is obtained. The assembly variants model serves as the basis for the component search and retrieval in the assembly variant design.

This AVM is represented using the object oriented methodology, where the classification operator (triangle) precedes the aggregation operator (diamond), because the assembly structure and components used can change by classification. A lot of the subassemblies and components might actually be shared between the variant classification types.

As mentioned in Section 3 the concept of AVM is not entirely new, Ramabhatta et al. [17, 18] proposed an object-oriented product model combining Group Technology with generic bill of materials (GBOM) concepts to help virtual enterprises identify similar designs. The GBOM, introduced by Hegge and Wortmann [6], is a single entity that encompasses all design options and alternate parts for a particular family of end products. Jiao and Tseng [7] and Jiao et al. [8] extend the basic GBOM by adding routing information and alternate operations. Recent efforts using data mining techniques to automate AVM development can be found in [21, 22].

The AVM is an important component of our variant design methodology and hence belongs to our set of complementary assembly models.

4.3.2 Assembly Mating Graph (AMG)

Since different assembly variants are different in structure, there should be a relational assembly model for each assembly variant. These relational assembly models designed for assembly variant design are named assembly mating graphs (AMG). This section presents how to specify the relational assembly model to generate the assembly mating graphs.

First, the node attribute set A needs to be expanded to cover the number of mating features, the dimensional key component design characteristics associated with each mating feature and the "weight" of each component (see below for definitions). Let the component set A be divided into two subsets, *i.e.* $A = \{A_1, A_2\}$.

- **Number of Mating Features and Dimensional Key Component Design Characteristics Associ-**

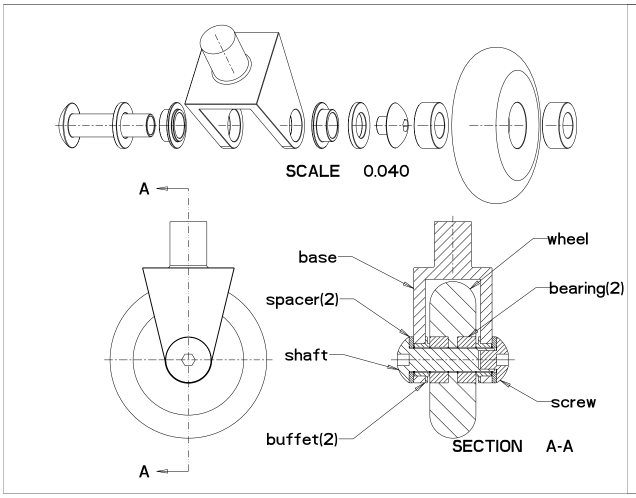

Figure 3: Wheel Support Assembly (Medium)

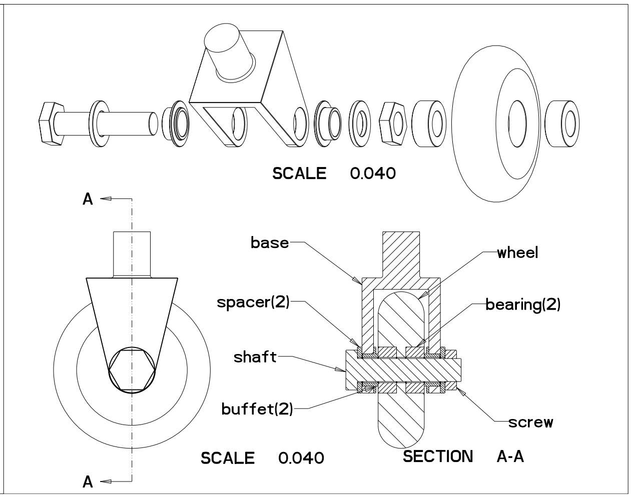

Figure 4: Wheel Support Assembly (High)

ated with each Form Feature: Each component may have one or more mating relationships with other components and each mating feature may have several dimensional key component design characteristics (KCDCs) associated with it which are parameters that specify the fundamental component features such as geometry, materials, and dimensions and which will be discussed in the second paper of this series [29]. Therefore, we define A_1 as a set of two-tuples $A_1 = \{(A_{i1}, A_{i2})\}$ where A_{i1} corresponds to the form feature index and A_{i2} corresponds to the set of KCDCs associated with that form feature, *i.e.* , $A_1 = \{(1, \{D_{11}, D_{12} \ldots\}), (2, \{D_{21}, D_{22} \ldots\}), (3, \{D_{31}, D_{32} \ldots\}), \ldots\}$ where D_{ij} is the jth dimensional KCDC associated with form feature i. Fig. 6 shows the mating features indices for the medium speed and load wheel supportive assembly. One issue that needs to be addressed here is the definition of the mating features in assembly variant design. Mating features can be manufacturing features but not necessarily. For example, a through hole can be both a mating and a manufacturing feature, but a single face of a manufacturing feature, square pocket, might be a mating feature. The mating features are usually single surfaces of the mating components.

- **Weight of the Node:** In graph theory, the weight of the node is a basic definition and is used in many cases. In application, the node weight has different meanings in different circumstances. However, the node weight is an indispensable component attribute that should be involved in the mating graph. $A_2 \in R$ is defined as the node weight. The weight could be a measure of the relative importance of a component in the assembly and might be closely related to the cost, contribution to customer satisfaction, or physical weight. In this research, these weights are assumed to be user specified and provided. Further research can be undertaken to formally derive these.

Second, the edge attribute set B needs to be expanded to cover the types of mating relationships and the weight of each edge. These are detailed in the following.

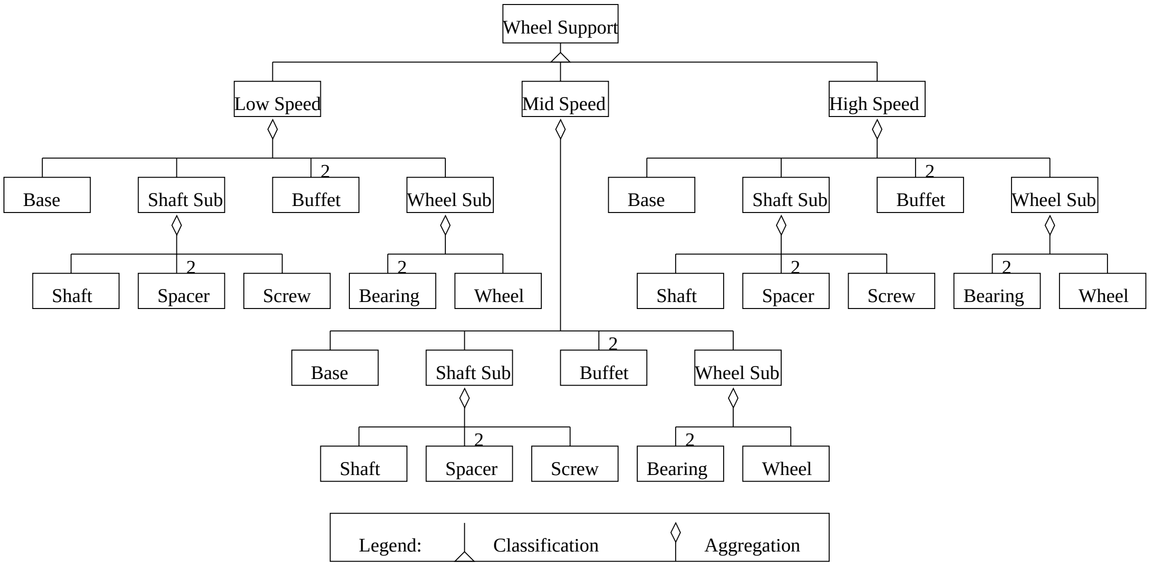

Figure 5: Assembly Variants Model

Types of Mating Relationships: As summarized in Table 1, the types of mating relationships can be represented as a triplet which covers the types of mating, DOF, and operator. In assembly variant design, the DOF and the operator are not of major concern. Instead, the primary interest is how the components are mated and how to make existing components fit as new variant designs are formed. And attention is given to the type of mating. For the purpose of demonstration, “against” and “coaxial fit” are enough. Therefore, the mating relationships are classified into “against” and “fit” which means “coaxial fit”. This set is extendible to the other mating relations outlined in Section 4.1.2 and Table 1 using the open modeling structure of the edge attribute set B (or more specifically, B_1 described later in this section).

In addition to the types of mating, another kind of edge attribute is important to assembly variant design. In assembly variant design, the dimensions of the components need to be changed in order to make them fit and function properly. If two components have mating relationship, it is obvious to check whether the dimensional change of one component will cause interference with the other one. The difficulty lies in the situation when two components do not actually mate with each other but a dimensional change of one may cause interference between them. This kind of situation to our knowledge has never been modeled in any assembly modeling methods. For assembly variant design, it is significant to model them. It is not only capable but also very easy to do so using the relational assembly model concept proposed in Section 4.2.2. Two kinds of mating conditions are distinguished. The first type of mating condition indicates actual mating characteristics and is called direct mating relationship. The direct mating relationships compose the essential part of almost all assembly models. The second type of mating condition is called indirect mating relationship. Since indirect mating conditions may be even more than direct mating conditions, it may be cumbersome to model all of them. As to which indirect mating relationship needs to be modeled, it depends

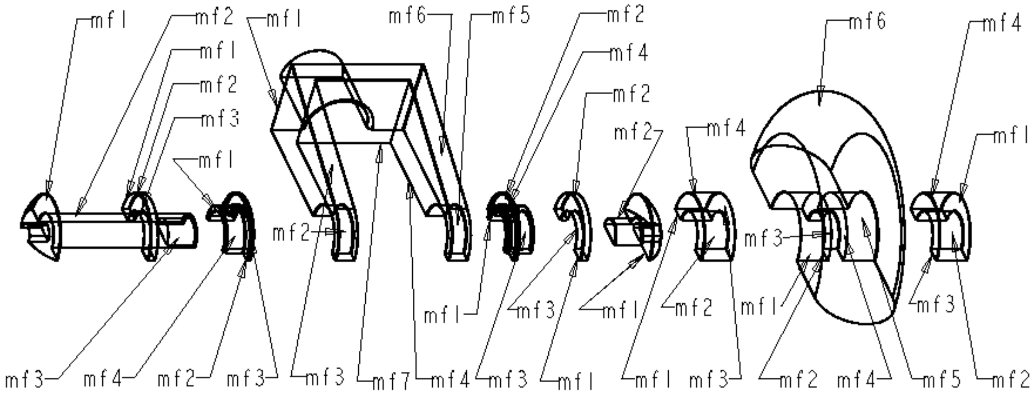

Figure 6: Assembly Product Mating Features

on the task and hence is left for the judgment of the designer. The direct and indirect mating relationships are complementary to each other.

Weight of the Edge: Similarly, the edge weight is an indispensable attribute that should be involved in the mating graph. It might be used to represent the coordination of each mating pair which can be used in tolerance chain analysis. Again, the weight could be a measure of the relative importance of the mating relationship in the mechanism, functionality, cost, etc. In this research, it is assumed that these weights are user specified and provided.

In summary, the edge attribute set B is defined as a triplet, *i.e.* $B = \{B_1, B_2, B_3\}$. B_1 and B_2 are two kinds of edge attributes. $B_1 = \{F, A\}$ where "F" represents "Fit" and "A" represents "Against". $B_2 = \{D, I\}$ where "D" represents "Direct Mating" and "I" represents "Indirect Mating". $B_3 \in R$ as the edge weight.

As a result, the AMG is defined as a 5-tuple $g =< V, E, \alpha, \beta, \gamma >$ where:

- V is a finite set of nodes;
- $E : V \times V$ is a finite set of edges;
- A is the set of node attributes and is defined as a tuple $A = \{A_1, A_2\}$. A_1 is further defined as a set of two-tuples $A_1 = \{(A_{i1}, A_{i2})\}$ where A_{i1} corresponds to the form feature index and A_{i2} corresponds to the associated key component design characteristics (KCDCs), *i.e.*, $A_1 = \{(1, \{D_{11}, D_{12} \ldots\}), (2, \{D_{21}, D_{22} \ldots\}),$ where D_{ij} is the jth dimensional KCDC associated with form feature i. $A_2 \in R$ represents the node weight;
- B is the set of edge attributes. Set B is a triplet, *i.e.* $B = \{B_1, B_2, B_3\}$. B_1 and B_2 are two kinds of edge attributes. $B_1 = \{F, A\}$ where "F" represents "Fit" and "A" represents "Against". $B_2 = \{D, I\}$

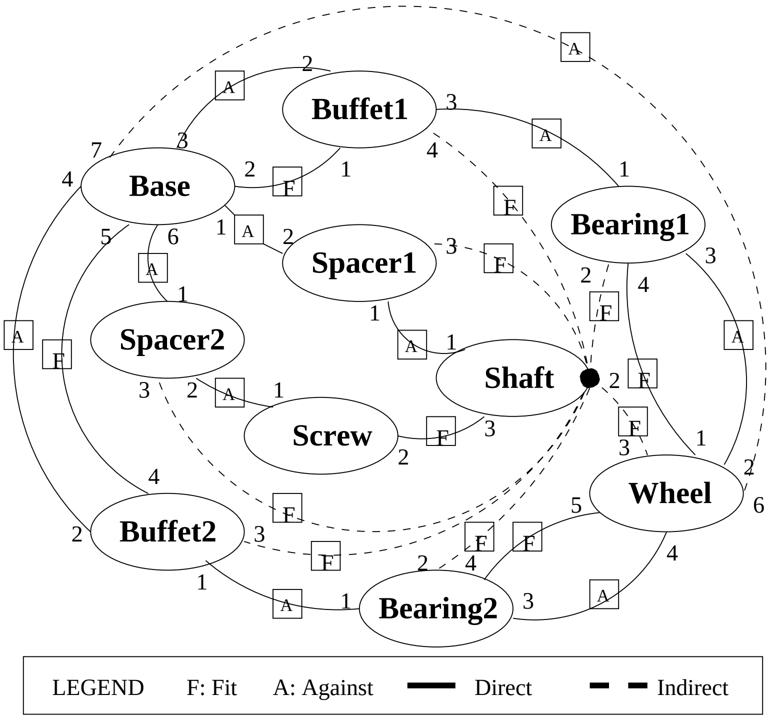

Figure 7: Assembly Product Mating Graph

where “D” represents “Direct Mating” and “I” represents “Indirect Mating”. $B_3 \in R$ corresponds to the edge weight;

- $\alpha : V \rightarrow A$ is a function which associates a set of node attributes with each node, e.g., the number of alternatives of each component;
- $\beta : E \rightarrow B$ is a function which associates a set of edge attributes with each edge, e.g., the tolerance information may be associated with each mating pair;
- $\gamma : A_1 \rightarrow B$ is a function which associates the set of node attributes A with each edge, e.g., associate the form features of each component with the edges.

The Assembly Mating Graph (AMG) can be represented either graphically or mathematically. Fig. 7 graphically shows the mating graph of the medium speed wheel supportive assembly shown in Fig. 3. In Fig. 7, each node represents a component or subassembly and each arc connects two nodes and represents a pair of mating features. The black dot around each node represent the mating features of the corresponding component. The mating features of each component of the medium speed wheel supportive assembly are identified and numbered in Fig. 6. Corresponding to the two kinds of mating conditions, there are two kinds of arcs. A solid arc represents a direct mating condition while a dashed arc represents an indirect mating condition. Each pair of mating features falls into either fit or against and is marked by either “A” or “F” where “A” represents “Against” and “F” represents “Fit”.

		Shaft			Spacer1			Base							Buffet1				Bearing1				Wheel						Bearing2				Buffet2				Spacer2			Screw	
		1	2	3	1	2	3	1	2	3	4	5	6	7	1	2	3	4	1	2	3	4	1	2	3	4	5	6	1	2	3	4	1	2	3	4	1	2	3	1	2
Shaft	1				AD																																				
	2						FI											FI		FI					FI					FI					FI				FI		
	3																																								FD
Spacer1	1	AD																																							
	2							AD																																	
	3		FI																																						
Base	1					AD																																			
	2														FD																										
	3															AD																									
	4																																	AD							
	5																																			FD					
	6																																				AD				
	7																											AI													
Buffet1	1								FD																																
	2									AD																															
	3																		AD																						
	4		FI																																						
Bearing1	1																AD																								
	2		FI																																						
	3																							AD																	
	4																						FD																		
Wheel	1																					FD																			
	2																				AD																				
	3		FI																																						
	4																														AD										
	5																															FD									
	6													AI																											
Bearing2	1																																AD								
	2		FI																																						
	3																									AD															
	4																										FD														
Buffet2	1																												AD												
	2										AD																														
	3		FI																																						
	4											FD																													
Spacer2	1												AD																												
	2																																							AD	
	3		FI																																						
Screw	1																																					AD			
	2			FD																																					

LEGEND: F: Fit A: Against D: Direct Mating I: Indirect Mating

Figure 8: The Incidence Matrix

In addition to the graphical representation, the AMG can be mathematically represented using the incidence matrix. Fig. 8 shows the incidence matrix of the medium speed wheel supportive assembly. In this figure, the first row and column are the names of the components. The second row and column are the mating features of each component. These features are shown in Fig. 6. The remaining rows and columns represent the interactions between the mating features in which "F" and "A" stand for "Fit" and "Against" while "D" and "I" stand for "Direct Mating" and "Indirect Mating".

4.3.3 Relationship between the AVM and AMGs

- In the assembly variants model, assembly variants are distinguished by the values of the key assembly design characteristics. And the component retrieval in the variant design stage is based on the desired values of these key characteristics. In the examples shown in Fig. 2, 3 and 4, the assembly variants are classified by their speed and load.
- Though the assembly variants belong to the same assembly product family, the structures of different

assembly variants in the AVM may be quite different. The reason is simple. It is very likely that functionally similar assembly products have different design schemes.

- The relationship between the AVM and AMGs impact the efficiency of the assembly modeling method. In this research, the assembly variants are classified in such a way that the assembly design schemes within each variant have the same AMG structure.

5 Conclusions

This paper has presented the assembly variant design system architecture and the complementary assembly modeling methodology. The system is composed of two parts: assembly modeling methodology which is discussed in this paper and assembly variant design methodology which will be described in the second paper [29].

Owing to the lack of universally accepted assembly modeling methodologies, the complementary assembly modeling concept is proposed in this paper. Instead of generating higher level assembly models from lower level assembly models, which is usually an nontrivial and error prone procedure or squeezing all assembly related information into a single model, two complementary assembly models, Hierarchical Assembly Model and Relational Assembly Model, are employed. The first models the hierarchical and function relationships between constituent assembly components explicitly while the second models the mating relationships between constituent components explicitly at the form feature level. They are complementary in the sense that each of them focuses on a particular aspect of the assembly related information while they together unify them. The hierarchical assembly model and relational assembly model are further specialized to establish the assembly variants model and assembly mating graphs which are the assembly models designed especially for the complex assembly variant design.

In conclusion, the theoretical contribution of this paper can be outlined as follows:

- A framework that facilitates the generation of new assembly designs based on mature component designs which may be distributed at different partner sites is developed.
- Complementary assembly modeling concept is proposed which is finally instantiated as the hierarchical assembly model (Assembly Variants Model) and relational assembly model (Assembly Mating Graph). These models can be used to expedite the application in assembly variant design, assembly planning and disassembly planning.

Acknowledgment

This work was supported in part by National Science Foundation, via grant no. DMI-9624309. This support is gratefully acknowledged. The authors also wish to acknowledge the help of three anonymous referees whose insightful comments have enhanced this paper.

References

[1] D.C. Brown and B. Chandrasekaran. Knowledge and control for a mechanical design expert system. *IEEE Computer*, 19(7):92–100, 1986.

[2] Richard DeVor, Robert Graves, and John J. Mills. Agile manufacturing research: accomplishments and opportunities. (special issue of design and manufacturing on agile manufacturing). *IIE Transaction*, 29(10):813–824, 1997.

[3] Gary P. Gooldy. *Dimensioning, Tolerancing and Gaging Applied*. Prentice Hall, New Jersey, 1999.

[4] D.C. Gossard, R.P. Zuffante, and H. Sakurai. Representing dimensions, tolerances, and features in mcae systems. *IEEE Comput. Graph. Appl.*, 8:51–59, 1988.

[5] Jin-Kang Gui and Martti Mäntylä. Functional understanding of assembly modeling. *Computer Aided Design*, 26(6):435–451, 1994.

[6] H.M.H. Hegge and J.C. Wortmann. Generic bill-of-material: a new product model. *International Journal of Production Economics*, 23:117–128, 1991.

[7] J. Jiao and M.M. Tseng. Methodology of developing product family architecture for mass customization. *Journal of Intelligent Manufacturing*, 10(1):3–20, 1999.

[8] J. Jiao, M.M. Tseng, Q. Ma, and Y. Zou. Generic bill-of-materials-and-operations for high-variety production management. *Concurrent Engineering-Research & Applications*, 8(4):297–321, 2000.

[9] S.H. Kim and K. Lee. An assembly modeling system for dynamic and kinematic analysis. *Computer-Aided Design*, 21(1):2–12, 1989.

[10] H. Ko and K. Lee. Automatic assembling procedure generation from mating conditions. *Computer Aided Design*, 19:3–10, 1987.

[11] Kunwoo Lee and Guy Andrews. Inference of the positions of components in an assembly: part 2. *Computer Aided Design*, 17(1):20–24, 1985.

[12] Kunwoo Lee and David C. Gossard. A hierarchical data structure for representing assemblies: part 1. *Computer Aided Design*, 17(1):15–19, 1985.

[13] L.I. Lieberman and M.A. Wesley. Autopass: An automatic programming system for computer controlled mechanical assembly. *IBM J. Res. Develop.*, 21:321–333, 1977.

[14] G.H. Morris and L.S. Haynes. Robotic assembly by constraints. *IEEE 1987 International Conference on Robotics and Automation*, pages 1507–1515, 1987.

[15] Steven Nahmias. *Production and Operations Analysis*. McGraw Hill/Irwin, New York, NY, 2001.

[16] V. Ramabhatta. An object oriented product model to represent complex assemblies for aiding information retrieval and work process database integration. M.S. thesis, SUNY at Buffalo, Buffalo, NY, USA, 1997.

[17] V. Ramabhatta, L. Lin, and R. Nagi. Object hierarchies to aid representation and variant design of complex assemblies in an agile environment. *International Journal of Agile Manufacturing*, 1(1):77–90, 1997.

[18] V. Ramabhatta, L. Lin, and R. Nagi. An object hierarchy scheme for complex assemblies to aid information retrieval and work process database integration. In *6th Industrial Engineering Research Conference*, pages 650–655, Miami, Florida, 1997.

[19] David Neville Rocheleau. An assembly representation for articulated mechanisms. *PD-Vol. 80, ASME 1996 Engineering Systems Design and Analysis Conference*, 8:23–28, 1996.

[20] David Neville Rocheleau. Interactive assembly modeling of articulated mechanisms using geometric constraints. *PD-Vol. 80, ASME 1996 Engineering Systems Design and Analysis Conference*, 8:29–36, 1996.

[21] C.J. Romanowski and R. Nagi. A data mining and graph theoretic approach to building generic bills of materials. In *11th Industrial Engineering Research Conference*, 2002.

[22] C.J. Romanowski and R. Nagi. A similarity measure for bills of material, a class of unordered trees. *submitted IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2003.

[23] J. Shah and M. Rogers. Functional requirements and conceptual design of the feature-based modeling system. *Computer-Aided Engineering Journal*, 5(1):9–15, 1988.

[24] S. Srikanth and J.U. Turner. Toward a unified representation of mechanical assemblies. *Engineering with Computers*, 6:103–112, 1990.

[25] J. Thomas, N. Nissanke, and K.D. Baker. A hierarchical petri net framework for the representation and analysis of assembly. *IEEE Transactions on Robotics and Automation (Special Issue on Assembly and Task Planning for Manufacturing)*, 12(2):268–280, 1996.

[26] J.P. Thomas, N. Nissanke, and K.D. Baker. Boundary models for assembly knowledge representation. *IEEE Transactions on Robotics and Automation*, 12(2):302–312, 1996.

[27] Roger Toogood and Jack Zecher. *Pro/ENGINEER Tutorial and MultiMedia CD*. SDC publications, 1999.

[28] Yuan-Jye Tseng and Li-Cheng Liou. Integrating assembly and machining planning using graph-based representation models. *Int. J. Prod. Res.*, 38(12):2619–2641, 2000.

[29] Aihu Wang, Bahattin Koc, and Rakesh Nagi. Complex assembly variant design in agile manufacturing. Part II: Assembly variant design methodology. *IIE Transactions on Design and Manufacturing*, 2004.

[30] N. Wang and T.M. Ozsoy. Representation of assemblies for automatic tolerance chain generation. *Engineering with Computers*, 6:121–126, 1990.

[31] D.E. Whitney. From geometric modeling to product data models: Collaboration between engineering, computer science, and industry at leeds university. World Wide Web, *http://web.mit.edu/ctpid/www/Whitney/Europe/LEEDS.pdf*, 1995.

[32] D.E. Whitney. The potential for assembly modeling in product development and manufacturing. World Wide Web, *http://web.mit.edu/ctpid/www/Whitney/papers.html*, 1996.