Proceedings of IMECE2005
2005 ASME International Mechanical Engineering Congress and Exposition
November 5-11, 2005, Orlando, Florida USA

IMECE2005-82342

A SYSTEM FOR PERFORMING CONTENT-BASED SEARCHES ON A DATABASE OF MECHANICAL ASSEMBLIES

Abhijit S. Deshmukh
Department of Mechanical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742, USA
dabhijit@umd.edu

Satyandra K. Gupta[1]
Department of Mechanical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742, USA
skgupta@eng.umd.edu

Mukul V. Karnik
Department of Mechanical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742, USA
karnik@glue.umd.edu

Ram D. Sriram
Leader, Design and Process Group
Manufacturing Systems Integration Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
100 Bureau Drive, MS 8263
Gaithersburg, MD 20899, USA
sriram@nist.gov

ABSTRACT

The increased use of 3D CAD systems by product development organizations has resulted in large databases of assemblies; this explosion of assembly data will continue in the future. Currently, there are no effective content-based techniques to search these databases. Ability to perform content-based searches on these databases is expected to help the designers in the following two ways. First, it can facilitate reuse of existing assembly designs, and thereby reducing the design time. Second, a lot of useful Design for Manufacturing and Assembly (DFMA) knowledge is embedded in existing assemblies. Therefore a capability to locate existing assemblies and examine them can be used as a learning tool by the designers to learn from the existing assembly designs and hence transfer the best DFMA practices to new designers. This paper describes a system for performing content-based searches on assembly databases. We identify templates for comprehensive search definitions and describe algorithms to perform content-based searches for mechanical assemblies. We also illustrate capabilities of our system through several examples.

Keywords: Content based assembly search, graph compatibility, assembly mating conditions, and assembly characteristics.

[1] Corresponding Author

1 INTRODUCTION

Over the last ten years 3D CAD (computer-aided design) systems have become very popular in the industry. These CAD systems are being used to generate 3D models of parts and assemblies. These models are used as a basis for engineering analysis and generate manufacturing plans. 3D models also allow virtual prototyping and hence reduce the need for physical prototyping. Nowadays, organizations routinely set up databases of CAD models to enable all participants in the product development process to have access to 3D data to support their functions. Specially, manufacturing and service engineers are expected to greatly benefit from these databases. These databases are updated with the latest versions of parts and assemblies and hence significantly improve information dissemination. CAD databases for even moderate size companies are expected to be large in size. A product assembly can contain many subassemblies and each subassembly can contain many parts. Therefore, even a small organization that has multiple product lines may add hundreds of assemblies to their database every year.

In addition to supporting downstream manufacturing and service operations, the assembly databases can be very useful during the design phase as well. There are two main uses of assembly database during the design stage.

Copyright © 2005 by ASME

- The first possible usage is to locate existing assemblies that can be reused in a new product. Such reuse of existing designs is beneficial from many different perspectives. It reduces design time by eliminating the need for modeling and analysis for the assembly being reused. Furthermore, the existing assembly is already tested and has an established manufacturing plan. This further reduces the product development time and cost. Sharing assemblies across multiple product lines also allows a company to take advantage of the economy of scale.
- The second possible usage is to provide access to existing design knowledge. Designing assemblies requires considerable effort. Creating good assembly designs require thoughtful analysis and careful application of Design for Manufacturing and Assembly (DFMA) principles. New designers can adopt and copy successful design templates. Moreover, once designers manage to find an assembly with the desired characteristics, they can also access associated data such as cost, reliability, and failure reports.

Currently, designers have access to the several types of search tools. If the assemblies are stored in hard drives, they can use file name--based search tools. This strategy only works if a meaningful file naming convention based on assembly contents is adopted. However, developing and deploying a content-based naming convention is impractical in many large organizations. Another way is to attach text notations to assemblies and store them in a PDM database. This scheme only provides limited search capabilities and has a limited discrimination power. Moreover, assemblies need to be manually annotated. In the recent past, several geometry-based search tools have emerged. However, these tools, although useful for part searches, are not very effective for assemblies. They can only account for the overall shape of the assembly and cannot account for relationships and structure that exist in assemblies.

Currently, content-based search tools do not exist for searching assemblies based on the specified criteria. Therefore, designers locate assemblies by manually opening various files and browsing through them using a computer aided design system. This is a highly inefficient use of designer's time, and becomes a serious problem as the numbers of assemblies in the database grow.

We describe a system for performing content-based searches on an assembly database. It is followed by the description of the templates for comprehensive search definitions that we have identified after studying various assembly models used in modern CAD systems. We also describe the algorithms developed to perform content-based searches for mechanical assemblies based on these search definition templates. These algorithms have been implemented in a prototype system. We also illustrate the possible usages of the prototype system through several examples.

2 OVERVIEW

Our goal is to develop a content-based assembly search system for searching assemblies from a database of existing assemblies based on different characteristics. The characteristics used by the system to search the database need to be extensive and also include most of the characteristics of a typical assembly. Hence, the system will need to support a comprehensive list of characteristics of assemblies based on which the user can define a search. The characteristics included in the system are enumerated in subsequent sections. The system should be flexible and allow the user to search based on any combination of the characteristics. It should also handle cases that result in too few or too many search results. Thus, if the search system results are too few then the user should be able to lower the constraints (strictness) on the search criteria by increasing the cut-off values. Also, if the search results are too many then the user should be able to perform iterative refinement. This is achieved by constraining the search by including more assembly characteristics in the search, then performing search and again refining search definition. This iterative refinement is very effective in producing the right number of search results. At any time in the search, the user should be able to exclude any assembly from further search. Finally, the system should have an easy-to-use interface and should be efficient so as to locate assemblies from a database in few seconds.

To build a list of all the characteristics of an assembly on which search can be performed, an extensive review of existing CAD systems and literature in the assembly modeling field [Anan96, Boot94, Brun00, DeFa87, Home91, Khos89, Lee85, Lee93, Moll93, Shah93] was performed. We decided to interface our system with the Pro/Engineer CAD system. Therefore, we also studied the list of all characteristics available in Pro/Engineer. Based on the published assembly characteristics and information available in Pro/Engineer models, we developed a preliminary list of assembly characteristics to support content-based searches. To ensure that these characteristics are not specific to Pro-Engineer, we also studied the assembly characteristics available in another CAD system – Unigraphics. We ensured that our list is compatible with the information available in Unigraphics.

The identified characteristics were then categorized into four main categories. The assembly design process was used as the basis of categorization of characteristics. Then, a suitable format, independent of any CAD system, was developed to store all the characteristics in an assembly as its signature. The assembly format defined in [Gupt01] is used as basis to store the signature of an assembly. In [Gupt01], each part is described by a name, a pointer to the geometry, and a transformation, which places the part in its assembled position in the assembly. Every joint is described by a type, a name, and the names of the base and attached parts forming the joint. The joint and mating condition data stored in the signature can be read to construct a graph. Characteristics not covered by [Gupt01] are appended to the signature.

Copyright © 2005 by ASME

The following four major categories of assembly characteristics are used for defining search:

- **The assembly statistics**: The assembly statistics include assembly characteristics such size and number of parts in the assembly. Section 3 deals with the search based on these statistics.
- **The characteristics of the constituent parts**: Some parts in an assembly can have specific characteristics with regard to material, size and other characteristics. Use of an uncommon part in an assembly characterizes the assembly and can be used as search criteria. Section 4 deals with the searches that utilize characteristics of the constituent parts.
- **The relationship between mating parts**: The mating conditions in an assembly play an important role in its function. Therefore, we need to support searches based on the type of mating conditions that exist in assemblies. This search is discussed in Section 5.
- **The relationship between joints in articulated assemblies**: Joints in articulated assemblies define the possible motion between the parts. Various joints restrict different degrees of freedom and are used as identifiers of the assembly. Therefore, we need to support searches based on joints attached to rigid links. This search is discussed in Section 6.

We have developed a system to support content-based searches. This system has been implemented using C++ and Microsoft Foundation Classes (MFC) library on Windows platform. The results are displayed using the Visual Search software [Karn05]. Visual Search provides an OpenGL based representation of the assembly files. To implement OpenGL based representation, Visual Search uses OpenSceneGraph library which is a higher level library built on top of OpenGL. The interface to define search is shown in Figure 1. The options available to define search criteria in each tab are described in the relevant sections. Figure 2 shows an example of the result of a search. Pro-Engineer Wildfire educational edition was used create assemblies. Pro-Toolkit provided with Pro-Engineer was used to extract the data from assembly files. The algorithm requires user to specify a top-level directory. It then iteratively searches for the signature files in the directory and all its sub-directories. Various specific examples of the usage of the system will be described in Sections 3 through 6.

3 SEARCH BASED ON OVERALL ASSEMBLY STATISTICS

A possible way to search for existing assemblies is based on the overall assembly statistics. The following scenario illustrates why this type of search is useful in certain situations. Let us consider an organization that designs and builds prosthetic devices. Let us now consider the case when a customer approaches the organization with his own specific requirements. The designers in this organization would prefer to locate an existing assembly that is close to the given requirements and then adopt this existing assembly to the new requirements. The ability to effectively locate the most appropriate existing assembly will eliminate the need to design the assembly from scratch and hence reduce the design time significantly. A possible way to search for existing prosthetics will be to search based on the size of existing prosthetic assemblies. This scenario illustrates the benefits of being able to search based on overall assembly statistics.

3.1 Search Definitions

In our framework assembly search can be performed based on the following criteria related to shape statistics.

- **Size**: The user can search assemblies based on the *bounding box size* or *bounding sphere size* of the assembly. The bounding sphere is defined using the radius of the sphere and the bounding box is defined by the length, the width, and the height of the bounding box. The data required for performing searches based on these two sizes are obtained from the Pro-Engineer assembly model and OpenSceneGraph library.
- **Number of Parts**: The user can search assemblies based on the number of parts in an assembly. In addition, the user also has an option to either include or exclude the *standard fasteners* from the part count in the assembly. This option has been provided to overcome the situations where a user would remember the main parts in the assembly but not remember the total number of fasteners used in the assembly. To perform this search the number of parts is extracted from the Pro-Engineer assembly model. In addition, we also determine if a part being used in the assembly is a standard fastener (e.g., screw, bolt, nut, and washer).
- **Number and Types of Articulated Joints**: The user can also search assemblies based on the number and types of joints in the assembly. The types of joints that can be defined in Pro-Engineer are *pin, U-joint, gimbal, cylindrical, slider, planar, ball, and bearing*. Besides these joints, Pro-Engineer allows the user to define the connection as *cam-follower, slot-follower,* and *gear pairs.* This type of search is defined by indicating the number of joints in each selected joint type. If a joint type is not selected by the user, then the system excludes that joint type from the search. The type and number of each joint are extracted from the Pro-Engineer file. Even though the system currently uses Pro-Engineer joint types, we can easily extend it to work with joint types found in other CAD systems.
- **Number of Usages in Other Assemblies**: An assembly such as a motor may be a popular assembly and hence used in many other assemblies. So some users might remember the large number of usage associated with an assembly. Hence, users can specify the number of usages of an assembly in other assemblies as a possible definition of search. This might be an effective way of searching a frequently used assembly.

Copyright © 2005 by ASME

- **Overall Shape Characteristics**: Assemblies may have overall shape characteristics that a user might remember. For example an assembly may predominately consist of rotationally symmetric parts or sheet metal parts. Such characteristics can often be used as a possible way to search for an assembly. Currently, we support the following two ways to search for assemblies based on overall shape characteristics. First, the user can specify the *percentage of rotationally symmetric parts* in the assembly. Second, the user can specify the *percentage of sheet metal parts* in the assembly.
- **Names of Conformance Standards**: Often assemblies are designed to meet certain testing and/or performance standards. Names of these standards are often included as notes on an assembly drawing in Pro-Engineer. Therefore, a possible way to search for assemblies is to specify standards to which an assembly conforms. We allow users to specify names of conforming standards as strings. Pro-Engineer assembly drawing notes are used to extract names of possible standards to which an assembly conforms.
- **Designer Name**: Assembly file attribute also contains the name of the person who created the assembly. Therefore, assemblies can also be searched by specifying the designer's name as a string.

Many of the above criteria require the users to specify either an integer or a real number for constraining the search. Two types of search definitions are implemented for specifying such searches. The first type of definition is based on range. In this case the user can specify an upper and lower limit on the search attribute. For example, a user can indicate that the number of part needs to between 30 and 50. We also allow the user to leave either the upper limit or the lower limit as unspecified. For example, if the user specifies the lower limit as 30 and leaves the upper limit as unspecified, then the search attribute has to be greater than or equal to 30. If both the upper limit and lower limit have the same value, then the attribute in database assembly has to exactly match the specified value. The second type of definition is based on the target attribute value. In this case the user specifies only the target value. All relevant entry in the database are compared to this target value and ranked based on their closeness to the target value.

A user can also select the multiple different criteria from the above list to define a search. For example, a user can define a search in the following manner: $30 \leq$ number of parts ≤ 40 AND $5 \leq$ number of slider joints. We currently only support conjunctive (AND) operators to combine search based on multiple different criteria.

3.2 Search Methods

All search attributes are either defined using numbers or strings. For search attributes that are defined using numbers with a range option all attribute instances in the database that meet the search definition are considered as feasible match. We do not rank the results if this type of search definition is used.

For search attributes that are defined using numbers with a target option all relevant attribute entries in the database are compared to this target attribute value and the penalty function $|t-a|/n$ is used to rank order the matches, where t is the target value, a is the value of attribute in the database assembly, and n is the normalization value. The value of n needs to be selected carefully to suit an organization's needs. The value of n will determine how many assemblies are considered as matches for a given target value. The value can be selected based on the size of the database of assemblies in the organization and the expected deviation of the required assemblies from the chosen target value. In addition, one can also set a cut off value: if the attribute value in the database assembly is farther than the cut-off value then that assembly is excluded from the results reported to the user.

For search attributes that are defined using strings we use two methods for identifying matches in the database. The first method is based on the exact string matching. In this case all database entries that contain the search string are considered as matches. The second method uses approximate string matching and uses the closeness of the strings to rank order the matches.

3.3 An Example

In the first test, a criterion to find all assemblies that had bounding sphere size between 3 and 20 was specified. This criterion gave a list of 7 assemblies from the database. Then the search criterion was made stricter by specifying the same range for size and another criterion was added for the range for number of components to be between 15 and 20. A search over the entire database gave a list of 2 assemblies. Thus, more specific search results were found by increasing the strictness of the criteria, i.e., by imposing an additional constraint on the criteria. Figure 3 shows these 2 assemblies.

4 SEARCH BASED ON CONSTITUENT PARTS

Assemblies can be searched based on the constituent parts of the assembly. Consider a scenario where the designer wants to search for a rocket motor assembly that contains a Beryllium liner of a specific size. Rocket motor assemblies are custom made to satisfy specific requirements. The designer would search for an assembly by specifying the size and material for a part of the assembly. These criteria will allow the designer to search for an assembly containing a part with specified size and material. The DFMA rules embedded in the assembly can be reused for the design of a new assembly.

4.1 Search Criteria

The system supports search based on the geometry of the part and the characteristics of the part. A combination of the two criteria is also supported. The two criteria for search are:

Copyright © 2005 by ASME

- **Geometry**: The geometry-based assembly search has different inputs based on whether a part is a standard part or a custom part.
  - **Standard part**: Every organization has a library of standard parts. A single assembly can contain a set of many of these standard parts. This criterion is useful when designer knows that a certain set of standard parts were used in the assembly. In this method, the user can select any set of standard parts from the library and search for assemblies containing these parts.
  - **Custom part**: This is useful in a scenario when the designer knows that a part used in the assembly is exactly similar to a part in the database or approximately matches with a part in the database. The user can select any Pro-Engineer part from the database as input geometry. The system allows search based on exact and approximate geometry matching.
- **Part Characteristics**: The following criteria for part characteristics-based search are supported:
  - **Material of the part**: Some assemblies contain a part made of a specific material. This criterion is useful to search for assemblies that contain a part that is made of an uncommon material. Users can specify any material from the available list of materials in the database of the organization.
  - **Part attributes**: Attributes are the textual data stored in the CAD files. Organizations have a set of standard attributes that help classify the assemblies in the database. For example, a attribute called *part source* can have values 'bought out part' or 'in house part'. Attributes can have values in the form of a numbers or strings. The system supports search for both the types and values of attribute. The user can define the title and value for attributes. If the value of an attribute is a string, the designer can define either an exact or an approximate search. If the value of the attribute is a real number, the designer can define search based on a target value or a range.
  - **Name of the owner**: CAD files store the name of the creator and the modifier in part history. In most CAD systems, this refers to the login names of users on the Operating System (OS). This data is useful since the designers can search for assemblies by the name of a designer who worked on a specific project. The user is allowed to select the name of a designer from the database of designers' names in the organization.

We support only single criteria based search and conjunctive search, similar to those described in Section 3.2. The user can define any combination of the above criteria to define a single search. The range and target value definitions for numbers and exact and approximate match options for strings are also available.

4.2 Search Methods

Searches based on attributes defined using number and strings are handled exactly the same way as described in Section 3.2.

The problem of finding an approximate match is usually referred as "search for similar parts." This problem has been explored in many different design and manufacturing contexts [Card03, Li04]. There are broadly two different kinds of methods. The first method uses the overall object shape in identifying similar parts. Representative techniques in this area include [Hila01, Karn05, McWh01, Osad01, Sung02]. The second method uses shape features in identifying similar parts. Representative techniques in this area include [Card04, Cici01, Rame01]. Both of these approaches have useful attributes. Depending upon a particular application, one might prove to be better than the other.

Exact part matching is implemented in our system in the following manner. The principal axes are found for the query part and the part in the database. These data are available in the form of rotation matrix. The parts are then aligned based on their centroid and then their principal axes. A Boolean subtraction operation is performed on the query model and model in the database by interchanging the tool and target bodies. If the result of both Boolean operations is a body with a volume smaller than a very small user-defined value, then the parts are treated as exact matches. If a database part matches exactly to the query part geometry, then the assembly containing the part is added to set of matches. We use moments of inertia and volumes as pruning criteria. If the two parts do not have matching moments of inertia and volumes, then they need not be analyzed further.

4.3 An Example

Consider a scenario where the designer is searching for an assembly that uses a variant of Beryllium liner. The liner is a standard part in the organization and its geometry is available in the library of standard parts. The designer also knows who designed the assembly, the size of the assembly and the number of parts used in assembly. The search criteria used were the geometry of the Beryllium liner, the number of parts in the assembly, the target bounding box size and the name of the assembly owner. It was found that these options collectively are sufficient to identify a set of assemblies from the database. This set contains the desired rocket motor assembly which satisfies the designer's exact requirement and is shown in Figure 4.

5 SEARCH BASED ON PART MATING CONDITIONS

The mating conditions are the restraints (constraints) imposed on the location of a part with respect to other parts in the assembly. Different set of restraints imposed on same set of parts can constitute different assemblies. This search criterion uses mating conditions to search for an assembly. The designer specifies the mating condition between parts of a subassembly or an assembly by building a query mating graph. This query

Copyright © 2005 by ASME

mating graph is compared with mating graphs corresponding to the assemblies in the database. The results of the search are all assemblies whose mating graphs are compatible with the query mating graph.

5.1 Search Definition

The designer defines an input mating graph representing a subassembly/assembly. The graph consists of parts defined as nodes and the mating conditions between the parts as edges of the graph.

Each node has the following attributes:

- **Category**: This represents whether the part is a standard part or a custom part. The user can select between either of the two options, or leave this attribute unspecified.
- **Geometry**: This attribute is a pointer to the Pro-Engineer geometry of the part. The geometry for standard parts is referenced from the library of standard parts in the organization. This attribute can also be left unspecified.
- **Type**: This criterion is defined only for standard parts and specifies the subcategory of the standard part. The available subcategories are: *bolts, nuts, washer, bearings, resting pads in fixtures, mold base, ejector pins, springs, circlips, rivets, retaining rings, hydraulic and pneumatic cylinders, chains, belts, gears, brakes, couplings, engine, actuators, pumps, valves, oil seals, vacuum seals, collars joints, universal joints, solenoids, switches, heating elements* and *limit switches*. These options are available to the designer in a pull down menu. The designer can select a specific variant of the part from another pull down menu after selecting the category to be a standard part. This attribute can also be left unspecified.
- **Type of match to perform**: It represents the type of search to perform. The user can specify either an exact match or an approximate match. This option can be specified only for a custom part.
- **Degree**: It represents the number of mating conditions that are defined for the part represented by the node. This attribute can either be given a specific value or a range of values.

Every edge in the graph has the following attributes:

- **Type**: This represents the type of mating condition represented by an edge. The search tool supports all mating condition options available in Pro-Engineer. The options available to the designer are: *mate, align, insert, tangent, point on line, point on surface*, and *edge on surface*. This attribute can also be left unspecified.
- **Vertex-1**: This attribute stores the identifier of the node from where the edge originates. This attribute cannot be empty. The designer needs to specify the node from where the edge originates.
- **Vertex-2**: This attribute stores the identifier of the node where the edge terminates. The query graph specified by the designer can be a partial graph with unspecified terminating node for an edge. This attribute can be empty for an edge with unspecified terminating node or can store the identifier of the node.

Please note that the query mating graph need not be a fully specified graph. Many of the attributes in the query graph can be left unspecified (i.e., equivalent to wild cards in string search definitions). This means that the query graph is not a unique graph and many different database graphs might be compatible with the query graph.

We will now provide illustrations of compatible and incompatible mating graphs. Figure 5 shows an example of a mating graph. This graph will be used as a query graph. The graph has four nodes. Node 1 includes a custom-built part and includes a reference to a file describing part geometry. Nodes 2 and 3 include standard bolts and hence explicit reference to part geometry files are not needed. Node 4 again includes a custom built part. However geometry is not specified for this part and hence during the search process, parts with different geometries will be able to match this node. In the next step, edges are created between nodes. They represent the mating conditions between parts. Both custom parts are connected to the bolts through mating conditions that mate two faces on the parts. An edge is defined between Nodes 1 and 4 but an exact mating condition is not specified for this edge. Hence during the search process, this edge will be able to match with many different mating conditions.

Figure 6 shows mating graph for an assembly from the database. The graph defined in Figure 5 is compatible with the graph shown in Figure 6. Nodes 1 and 4 in Figure 5 match with Nodes A and F in Figure 6. Nodes 2 and 3 in Figure 5 match with Nodes B and D in Figure 6. The mating condition between standard part and custom parts is the same in graphs shown Figures 5 and 6.

Consider a graph shown in Figure 7 that represents another assembly in the database. In this graph, the node labeled A does not have an edge connecting it to a node representing the custom part. However the query graph shown in Figure 5 has an edge between the two nodes representing custom parts. Thus, the query graph defined in Figure 5 is incompatible with the graph shown in Figure 7.

5.2 Search Methods

The system builds a mating graph for every assembly in the database off-line. The parts are represented as nodes. For each node four attributes namely, category, geometry, type and degree, are determined and initialized. If two parts are mated together in an assembly, then an edge is created between the nodes representing the parts. The type of mating condition used and the identifier of two mated parts are the attributes of the edge. As all the information about part and the mating conditions can be extracted from the Pro-Engineer files, this graph is a completely specified graph and does not include any wild cards that are typically associated with query graphs.

The query mating graph needs to be compared with the mating graph of every assembly in the database. However, the

Copyright © 2005 by ASME

query graph may not be a fully specified graph due to presence of wildcards in the query graph. Thus, the available subgraph isomorphism techniques cannot be used in the present form to solve this problem. We call this problem as graph compatibility problem. By adapting graph isomorphism techniques, we have developed an algorithm to perform graph compatibility check.

The algorithm described in the next paragraph for graph compatibility needs to use combinations of possible node matching. This step can be computationally expensive. Thus, to ensure that results are obtained in real-time, a two stage pruning process is initially carried out before the graph compatibility check. The search criteria defined in Sections 3 and 4 are computationally less expensive. If they are applicable they are used first. All assemblies that do not satisfy the criteria are not considered for graph compatibility tests. In addition, the pruning step compares the sum of degree of the nodes in database assembly graph with the minimum possible sum of degree of the nodes in the query graph (degree of a node refers to the number of connections associated with that node). If the former is less than the latter, then two graphs cannot be compatible because query graph will have more nodes than the database assembly graph. Such graphs need not be considered further.

The graph compatibility problem is solved using a depth-first branch and bound strategy. For the two graphs to be compatible, each node from the query graph should have a corresponding node in the database assembly graph. The first step of the search identifies node with the least degree in the query graph. This node is matched the nodes with appropriate degrees in the database graph. This process goes on until all the candidate nodes from the database graph are systematically examined. If any of the steps to match nodes fail, then nodes cannot match and the search continues with the next node from the database graph. The steps to match nodes are as follows.

1) Compare if the category of the two nodes are compatible.
2) Compare if the geometry of the two nodes are compatible.
3) Compare edges. All edges originating from the two nodes are listed. The edges are compared to ensure that for every edge in the list for the query graph an edge exists in the list of database graph such that they represent the same mating condition. If a match can be found for every edge from the list of query graph, this check is passed.

For selecting the next nodes for comparison the following steps are used:

1) Find all the nodes from the query and database assembly graph that have a common edge with the nodes matched in the previous step. This step return two lists -- one from the query graph and one from the database graph.
2) Match the nodes from the two lists in the increasing order of cardinality of node degrees.

The above two steps are used to examine all possible node matches at different levels of the branch and bound search tree.

If an attribute is not defined in the query graph, then it need not be considered during the matching process. If no match is found along any branch of the search tree, then the graphs cannot be compatible. The search continues until a corresponding node is found for each node of the query graph or the graphs are found to be incompatible.

5.3 An Example

Consider a scenario where the designer wants to search for a subassembly commonly used in the organization. The subassembly consists of some bolts and a custom part for which the exact geometry is known. The designer looks for an assembly where another custom part was used along with the parts listed above. The designer knows the mating conditions between the parts and defines the query graph (shown in Figure 5). The criteria for the search are specified as follows:

- The mating graph as shown in Figure 5
- The number of parts as a target value of 10
- Exact geometry match for one of the custom parts
- Two standard parts in form of a bolt and nuts

The assembly shown in Figures 8 and 9 is an assembly retuned by the system that matches the above criteria. In this case, A is the custom part and the bolts shown as B, C, D and E are the standard parts.

6 SEARCH BASED ON JOINT RELATIONSHIPS

Consider a scenario where a robot designer is searching for an assembly consisting of two revolute joints at right angles to each other to mimic the motion of a human arm. The main characteristic of this assembly is the joints defined between these parts. A typical search for such an assembly would be a manual search. Another possible way to search for this assembly is to specify the type of joint and their orientations with respect to each other. The joints in articulated assembly are important to define the function of the assembly. We propose a set of criteria to search based on joints and their interrelationships.

6.1 Search Definition

In the search tool, the designer specifies the type of joint and the orientation between the joints. The designer can select joint of any type that is available in Pro-Engineer. The possible orientations between joints are as follows:

- Parallel
- Perpendicular
- Angle (This is specified as range value for angle between the two joints)
- No relationship

The angle relationship allows the designer to specify a range of values for the angle between the joints. The designer will provide a set of joints that are the defining characteristics for the assembly and the relationship among them. This list may include only the set of joints in the assembly that are important from the design perspective.

The user input is provided in the form of a query graph. Each node in the graph represents a joint in the assembly. The type of joint is an attribute of the node. The relationship

Copyright © 2005 by ASME

between joints is represented as an edge. The possible relationship between the joints is represented as an attribute of the edge. Only joints defined on a single rigid body could be defined in one graph. The user can define multiple query graphs to search for a single assembly. The system searches for an assembly that has all the query graphs as its subgraphs.

Consider the robot assembly shown in Figure 10. Figure 11 represents the joint graph for the assembly shown in this figure. The two joints define the range of workspace for the robot. They are oriented perpendicular to each other. The two joints are represented as nodes and the edge represents the relationship between these nodes. Figure 12 shows a query graph that can be used to search for this joint graph. This graph is compared to the joint subgraph of the assembly in the database.

6.2 Search Methods

The input is received as one or more query graphs. For an assembly from the database to satisfy the search criteria, it should contain the joints with the relationship between them as defined in each of the query graphs. A graph is constructed for each assembly in the database. This graph contains all the joints and the relationship among them. The problem here is of the category of attributed subgraph isomorphism. The algorithm proposed by Gold [Gold96] is adopted for this purpose. For the joint graph, the designer may specify more than one graph to search for a single assembly. Each query joint graph can have only one rigid link and two joints. For an assembly from the database to match the query, the database assembly graph should have all the query graphs as subgraphs.

6.3 An Example

Consider a scenario where the designer wants to search for a spatial mechanism that allows a base to have translational motion along three axes, 120 degrees apart from each other and lying in one plane. The second motion is translational and lies along an axis which is perpendicular to the earlier plane. This motion can be achieved by using three ball joints and one slider joint. The designer constructs a query graph as shown in Figure 13 to search for such a mechanism. The designer has a restriction on the total number of joints in the assembly, and the size and total number of parts in the assembly. The criteria for search are specified as follows:

- The joint graph
- The total number of joints in the assembly (should not exceed 20)
- The number of ball joints in the assembly (should not to exceed 8)
- The number of slider joints in the assembly (should not to exceed 6)
- The bounding sphere volume of the assembly (should be between 2 and 3 inches)
- The total number of parts in the assembly (a target value of 13 is specified)

With these criteria, the system is able to retrieve the Stewart platform shown in Figure 14. This mechanism comes closest to the designer's requirement. The platform uses 6 ball and slider joints. The motion given by slider joints is not exactly perpendicular to plane of motion defined by ball joints. However, the designer can get the required motion by editing the geometry of Part A. The joint graph of the Stewart platform is shown in Figure 15.

7 CONCLUSIONS

This paper describes a comprehensive framework for performing content-based search. The search definition templates support a wide variety of search queries that can be posed to the system. Search definition templates described in the paper spans nearly all aspects of the assembly model. Hence, it provides a designer with a very expressive search definition capability. A variety of search options are provided to allow users to define from very narrow to very broad searches based on their needs. Searches can be iteratively refined to better direct the search. Search results can be browsed through a convenient visual interface. Conservative pruning used by search algorithms ensures efficient search performance and at the same time does not exclude results that might be of interest to the user.

We expect that the system described in this paper will serve two purposes. First, it will allow designers to reuse existing assemblies by giving them a means to identify assemblies with the desired characteristics. Second, it will provide designers an access to the DFMA knowledge contained in the assembly database, and hence transfer best practices to new designs.

The current implementation works only with the assembly characteristics available in Pro-Engineer CAD system to build the signature of the assembly. We plan to extend the search system to search assemblies from other CAD systems also.

An assembly can be defined as a collection of parts to fulfill a function. The function of the assembly is thus the primary characteristic of an assembly. However, a function of an assembly is not always explicitly stored in CAD files. Often, the function of an assembly cannot be inferred from its geometric characteristics. Hence, the designer cannot search for a design fulfilling a particular function. We plan to extend our work to support queries based on functions.

The proposed search method works only on the basis of the geometric characteristics of the assembly. This search is thus not applicable to assemblies that have characteristics from other domains of engineering like electromechanical or electrochemical engineering. The search works on the assumption that the designer explicitly defines joints and mating conditions or such relation can be implicitly extracted to form a signature of the assembly. If this data is not available, the assembly cannot be included in the search and this may result in false negatives during the search process.

Copyright © 2005 by ASME

8 ACKNOWLEDGMENTS

This research is supported in part by the Center for Energetic Concepts Development at the University of Maryland and National Institute of Standards and Technology's (NIST) Manufacturing System Integration Division. Any commercial product or company name in this paper are given for informational purposes only. Their use does not imply recommendation or endorsement by the NIST or the University of Maryland.

9 REFERENCES

[Anan96] R. Anantha, G.A. Kramer, and R.H. Crawford. Assembly modelling by geometric constraint satisfaction. *Computer Aided Design* 28 (9): 707-722, 1996.

[Boot94] G. Boothroyd. Product design for manufacture and assembly. *Computer Aided Design*, 26(9):505--520, 1994.

[Brun00] G. Brunetti, and B. Golob. A feature-based approach towards an integrated product model including conceptual design information. *Computer Aided Design*, 32 (14): 877-887, 2000

[Card03] A. Cardone, S.K. Gupta, and M.V. Karnik. A survey of shape similarity assessment algorithms for product design and manufacturing applications. *Journal of Computing and Information Science in Engineering*, 3(2):109-118, 2003.

[Card04] A. Cardone, S.K. Gupta, and M.V. Karnik. Identifying similar parts for assisting cost estimation of prismatic machined parts. In Proceedings of the *ASME Design for Manufacturing Conference*, Salt Lake City, Utah, 2004.

[Cici01] V. Cicirello, and W.C. Regli. Machining feature-based comparisons of mechanical parts. In Proceedings of the *International Conference on Shape Modeling & Applications*. 2001.

[DeFa87] T De Fazio, and D Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Transactions on Robotics and Automation*, 3(6):640--658, 1987.

[Gold96] S. Gold, and A. Rangrajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 18, No. 4., 1996.

[Gupt01] S.K. Gupta, C.J. Paredis, R. Sinha, and P.F. Brown. Intelligent assembly modeling and simulation. *Assembly Automation*, 21(3):215--235, 2001

[Hila01] M. Hilaga, Y. Shinagawa, T. Kohmura, and T.L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH*, pages 203-212, 2001.

[Home91] L.S. Homem de Mello, and A.C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Transactions on Robotics and Automation*, 7(2):228--240, 1991.

[Karn05] M. V. Karnik, D. K. Anand, E. Eick, S. K. Gupta, and R. Kavetsky. Integrated visual and geometric search tools for locating desired parts in a part database. *Computer-Aided Design & Applications*, 2(6):727-736, 2005.

[Khos89] P Khosla, and R Mattikalli. Determining the assembly sequence from a 3D model. *Journal of Mechanical Working Technology*, pages 153--162, 1989.

[Lee85] K. Lee, and D C Gossard. An hierarchical data structure for representing assemblies: part 1. *Computer-Aided Design*, 17 (1):15--19, 1985.

[Lee93] S Lee, G. Kim, and G. Bekey. Combining assembly planning with redesign: An approach for more effective DFA. In Proceedings of the *IEEE International Conference on Robotics and Automation*, 1993.

[Li04] Z. Li, M. Liu, and K. Ramani. Review of product information retrieval: representation & indexing. In Proceedings of the *ASME Design Engineering Technical Conferences*, Salt Lake City, UT, 2004.

[McWh01] D. McWherter, M. Peabody, A. Shokoufandeh, and W.C. Regli. Transformation invariant similarity assessment of solid models. In Proceedings of the *ASME Design Engineering Technical Conferences*, Pittsburgh, PA, 2001.

[Moll93] E. Molloy, H. Yang, and J. Browne. Feature-based modelling in design for assembly. *International Journal of Computer Integrated Manufacturing*, 6 (1,2):119--125, 1993.

[Osad01] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In Proceedings of the *International Conference on Shape Modeling and Applications*, Genova, Italy, 2001.

[Rame01] M.M. Ramesh, D.Y. Hoi, and D. Dutta. Feature-based shape similarity measurement for retrieval of mechanical parts. *Journal of Computing and Information Science in Engineering*, 1(3):245-256, 2001.

[Shah93] J. J. Shah, and M. T. Rogers. Assembly modeling as an extension of feature-based design. *Research in Engineering Design*, 5(3,4):218--237, 1993.

[Sung02] R. Sung, H. J. Rea, J. R. Corney, D. E. R. Clark, J. Pritchard, M.L. Breaks, and R.A. MacLeod. Assessing the effectiveness of filters for shape matching. In Proceedings of the *ASME International Mechanical Engineering Congress & Exposition*, New Orleans, LA, 2002.

Copyright © 2005 by ASME

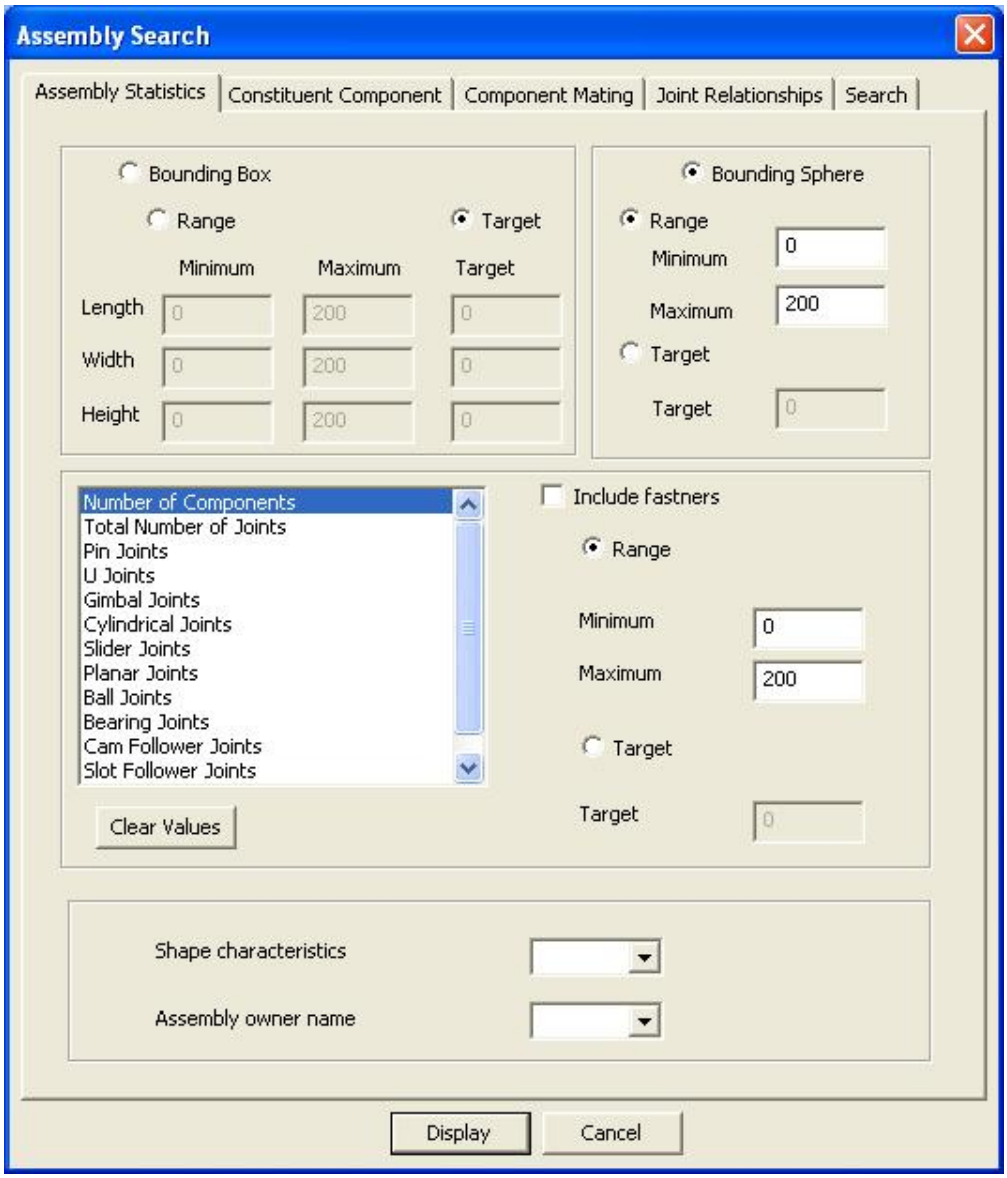


Figure 1: User interface to define search

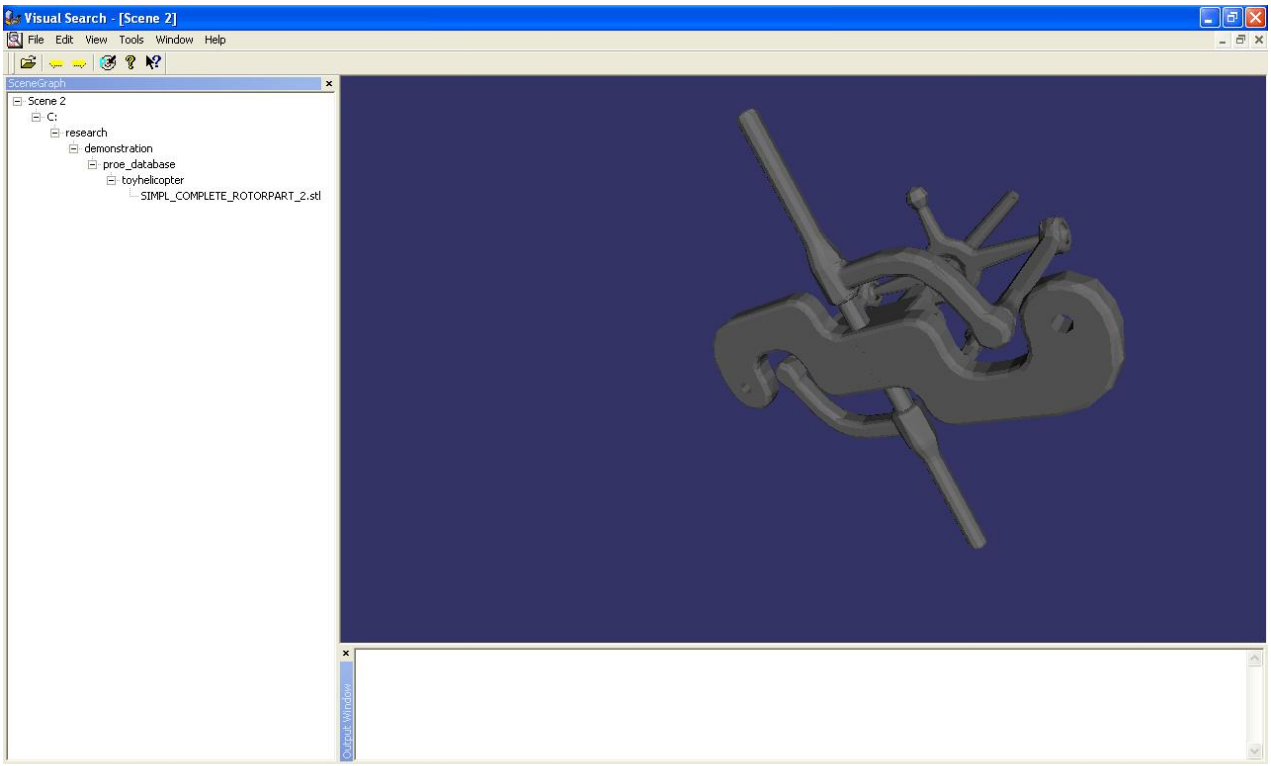
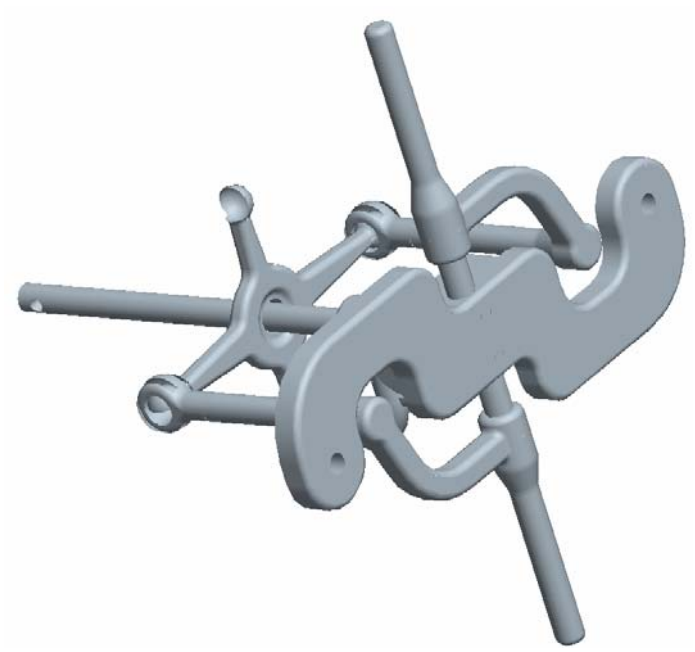
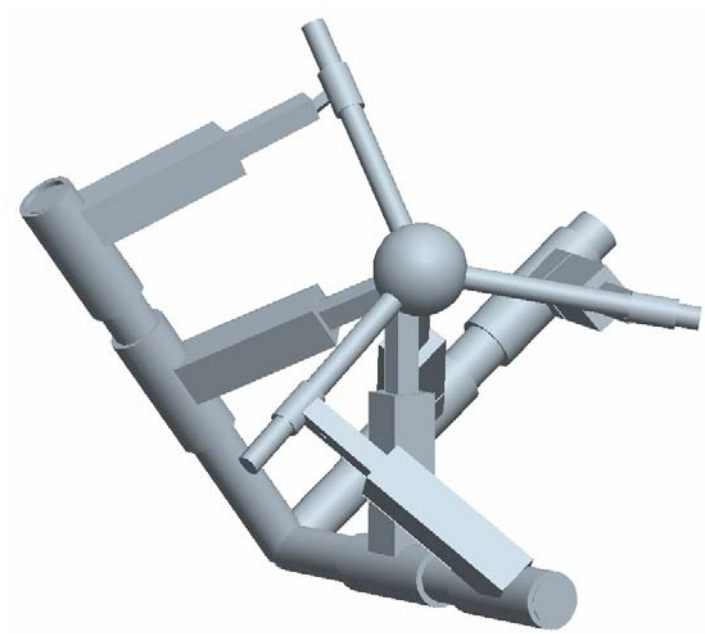


Figure 2: Results window



Rotor assembly



Stewart Platform

Figure 3: Two assemblies obtained by assembly statistics based search

Figure 4: A match for search by constituent part search –
Rocket Motor Assembly

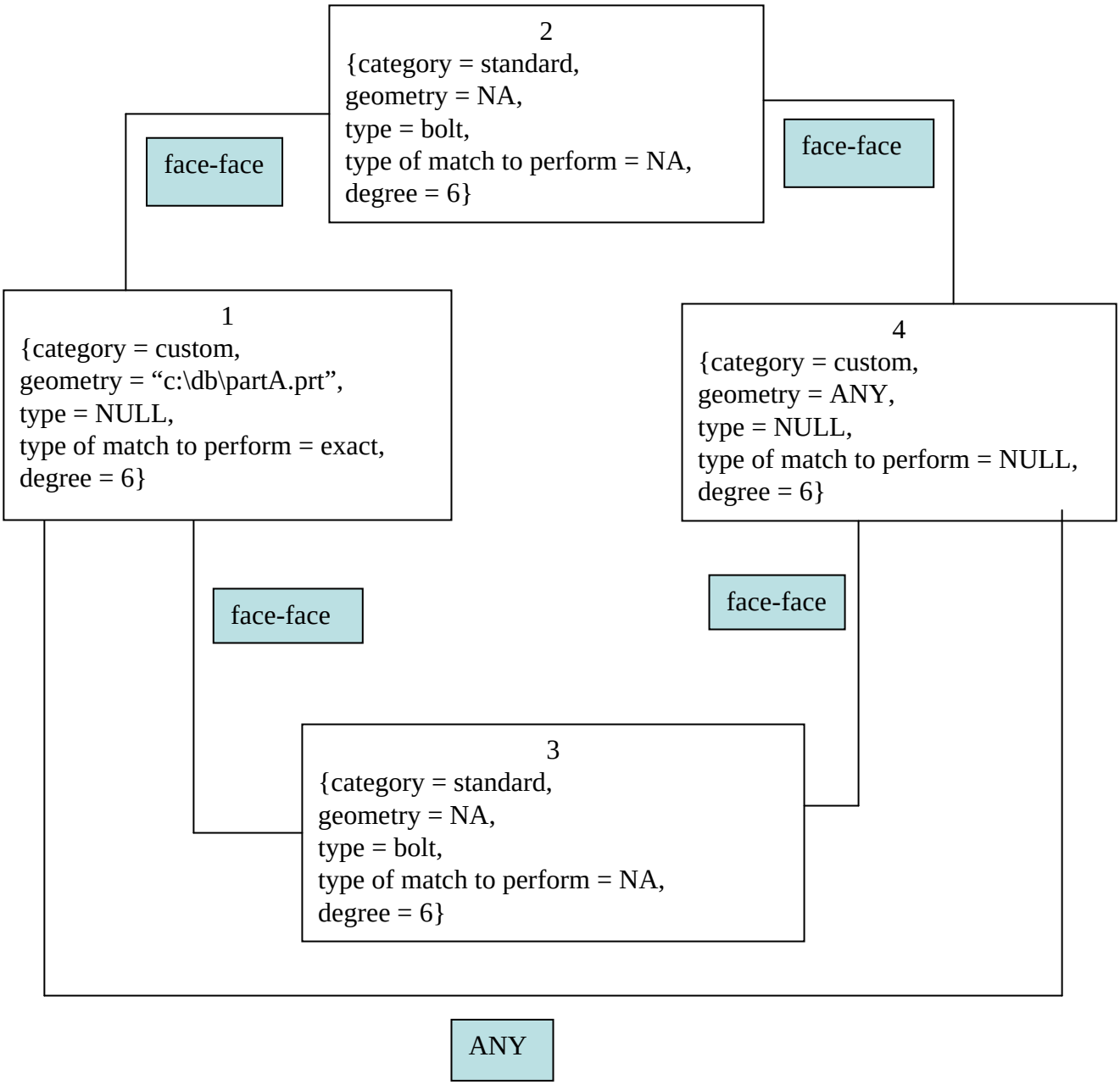
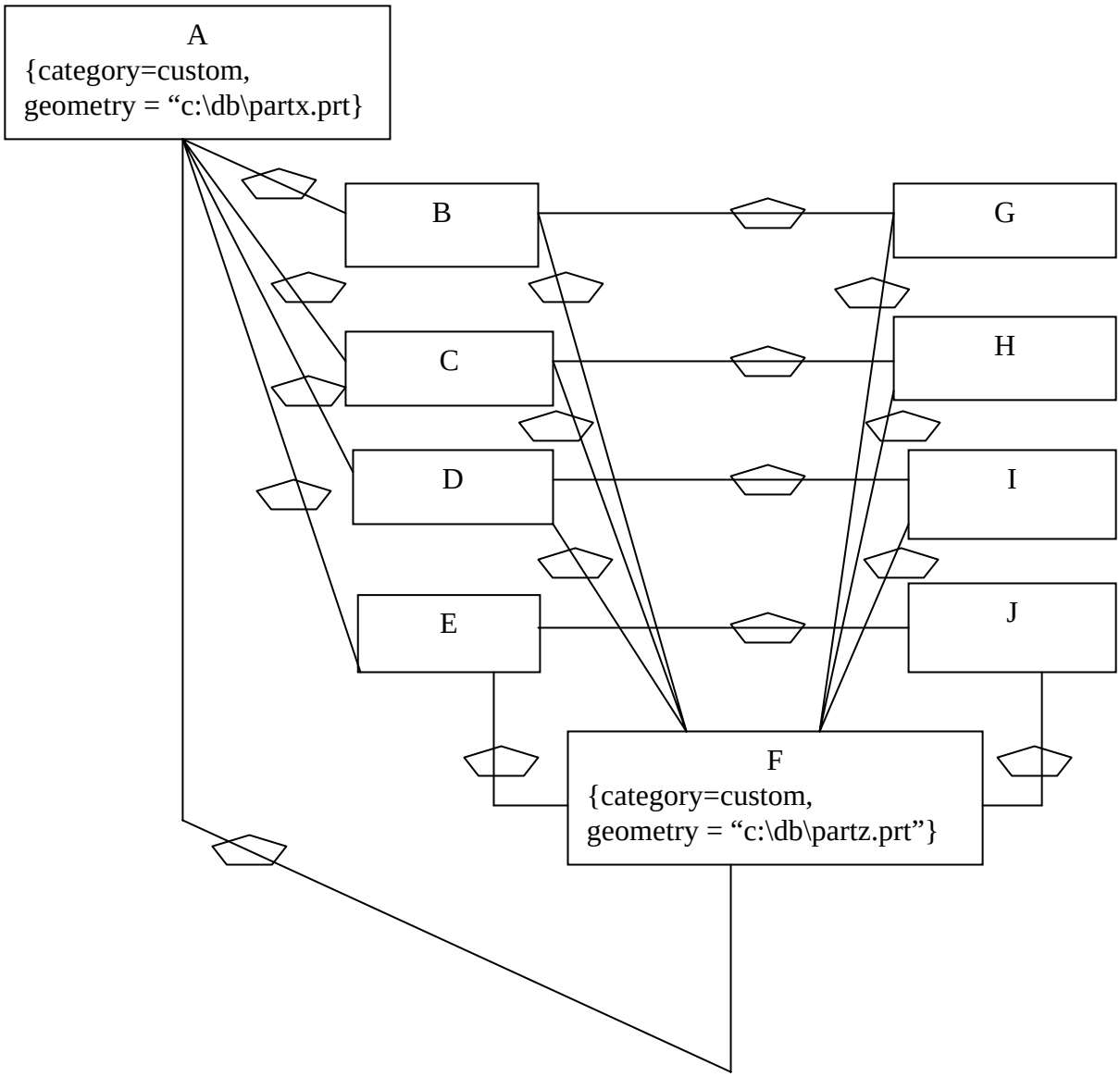


Figure 5: The query graph defined for part mating conditions

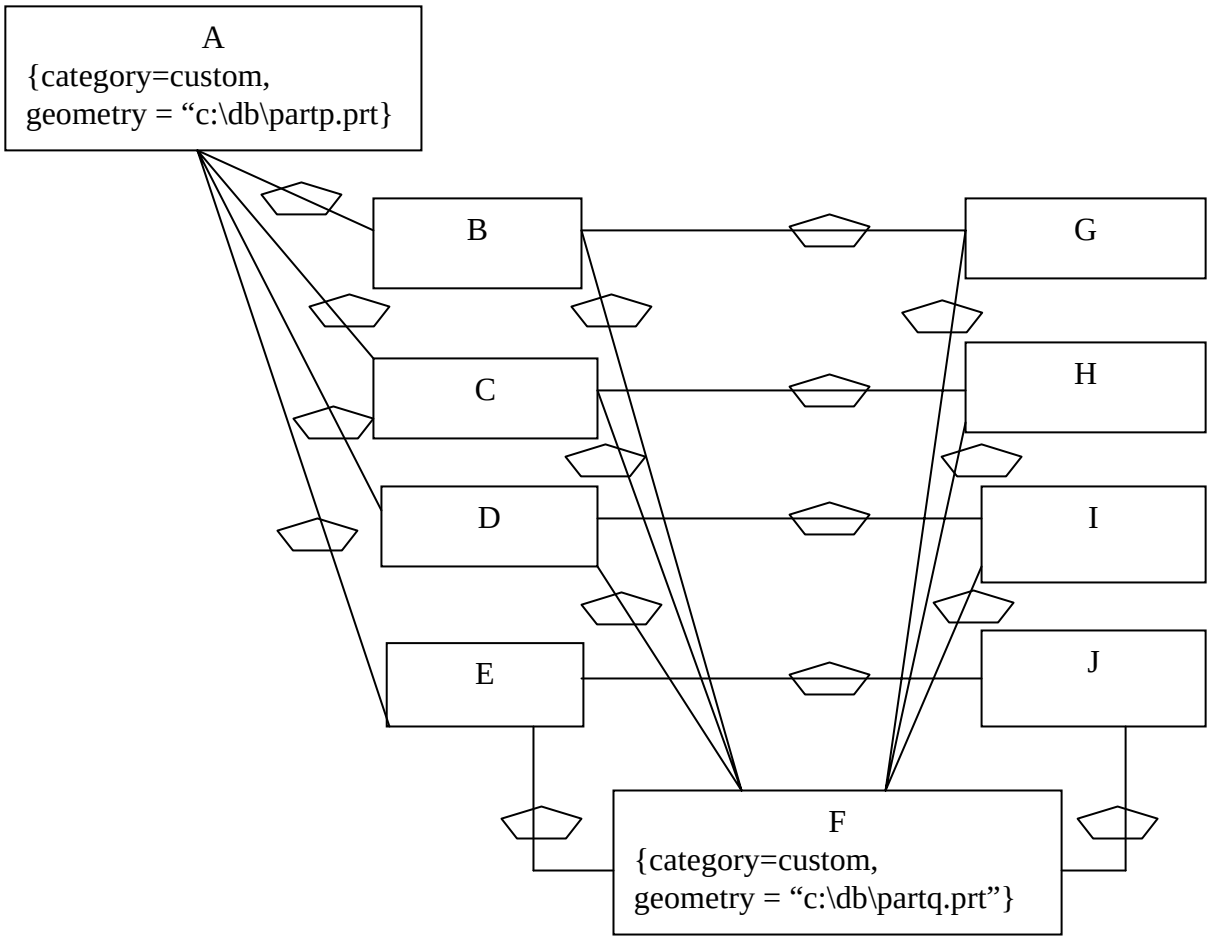


Represents face-face mating

B, C, D and E are {category = standard, type = bolt}

G, H, I and J are {category = standard, type = nut}

Figure 6: A database mating condition graph. The query graph is compatible with this graph



Represents face-face mating

B, C, D and E are {category = standard, type = bolt}

G, H, I and J are {category = standard, type = nut}

Figure 7: Another database mating condition graph. The query graph is incompatible with this graph

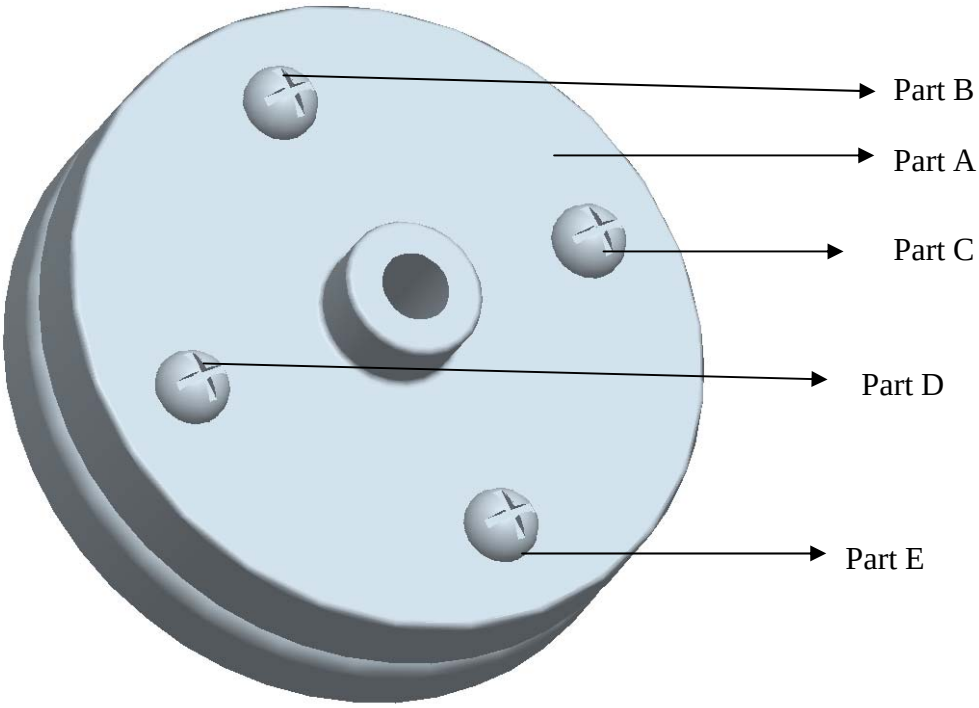


Figure 8: An assembly obtained by mating condition graph
- Flange Assembly

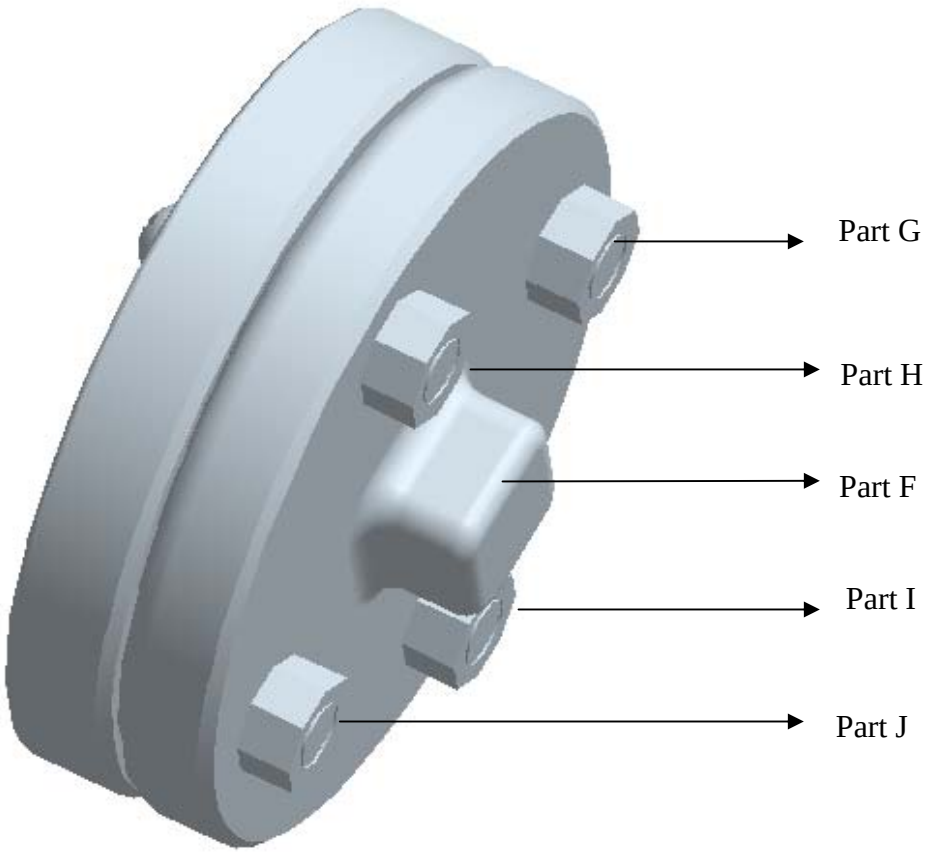


Figure 9: Another view of Flange Assembly

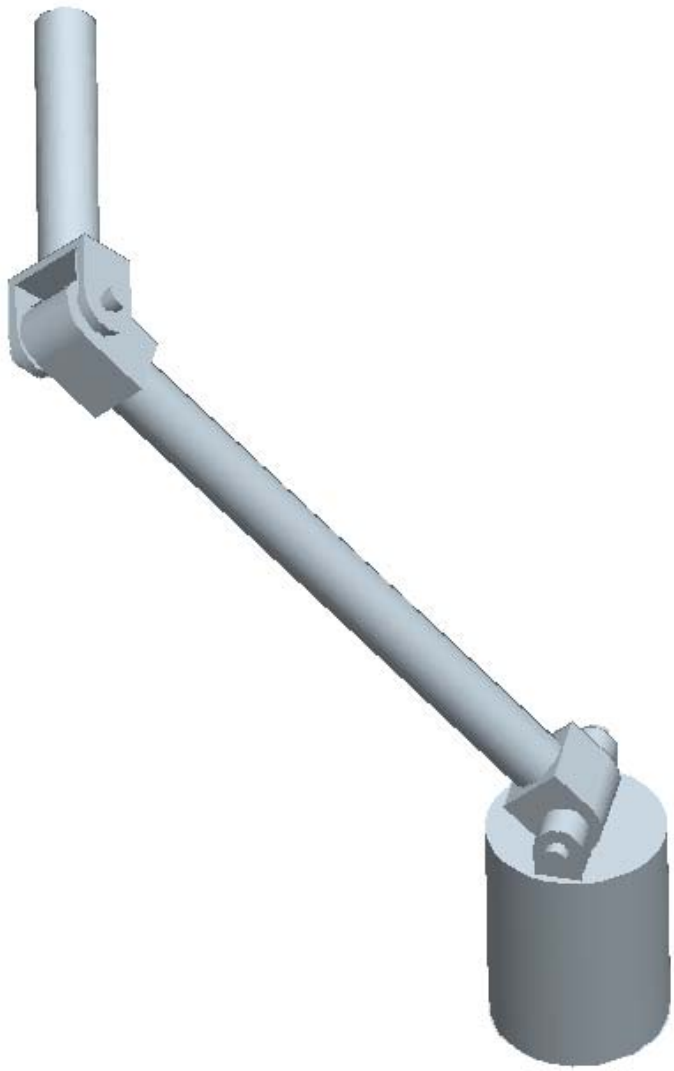


Figure 10: A robot arm with two revolute joint allowing perpendicular motion and attached on one link

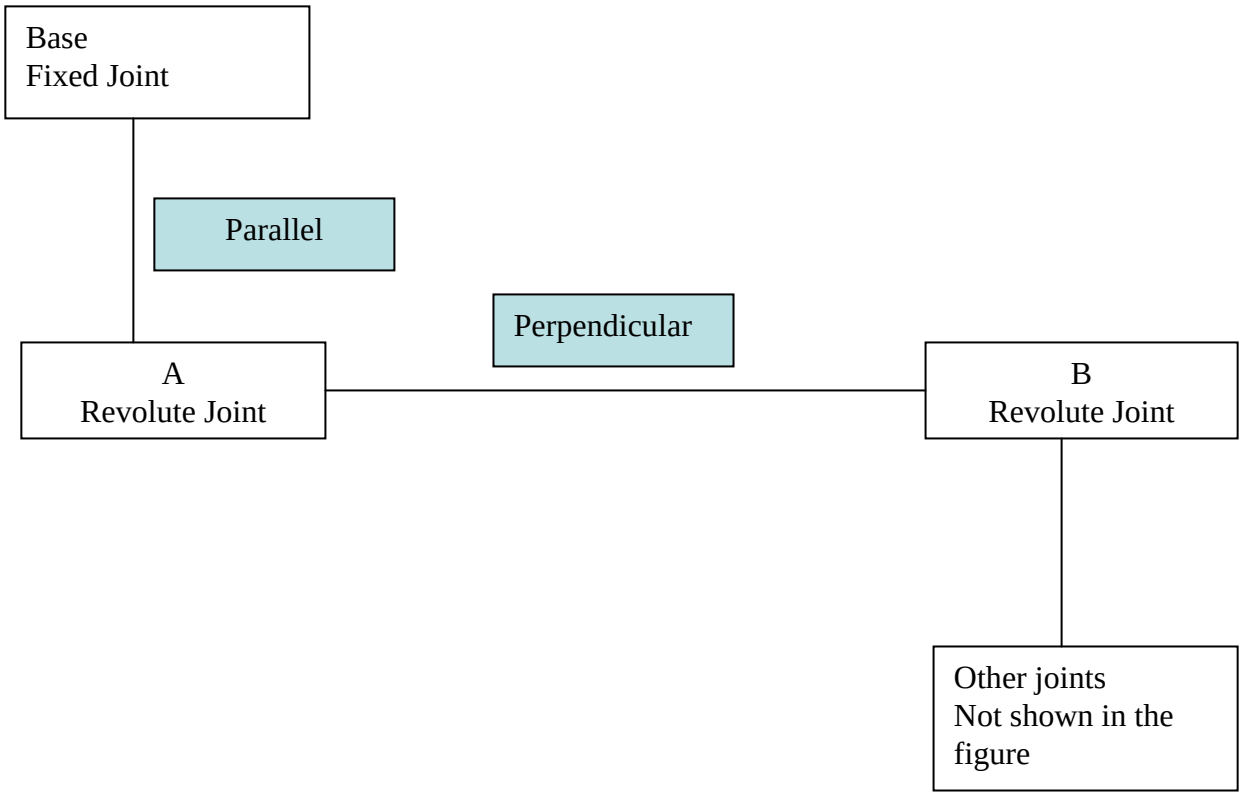


Figure 11: The graph stored in database and is constructed from the assembly in the database

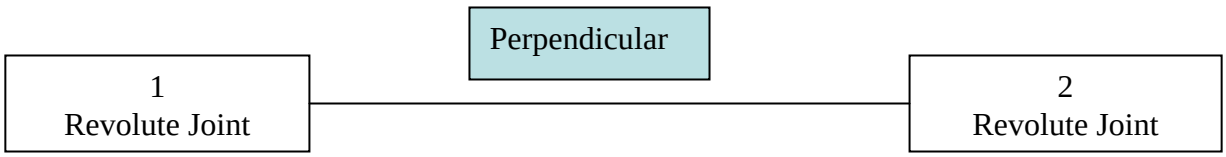


Figure 12: A query graph to search for assembly show in Figure 8

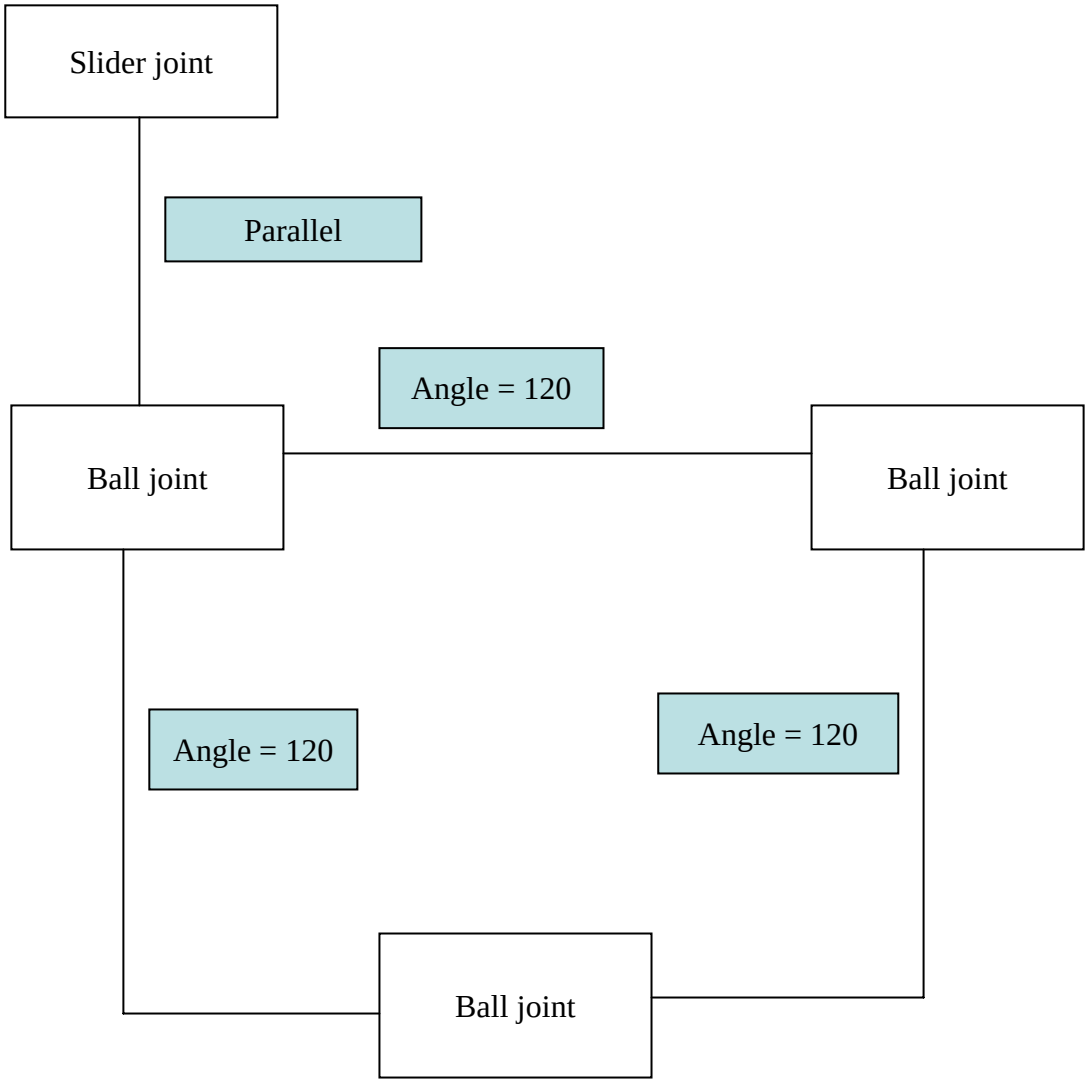


Figure 13: A query graph defined to joints based search

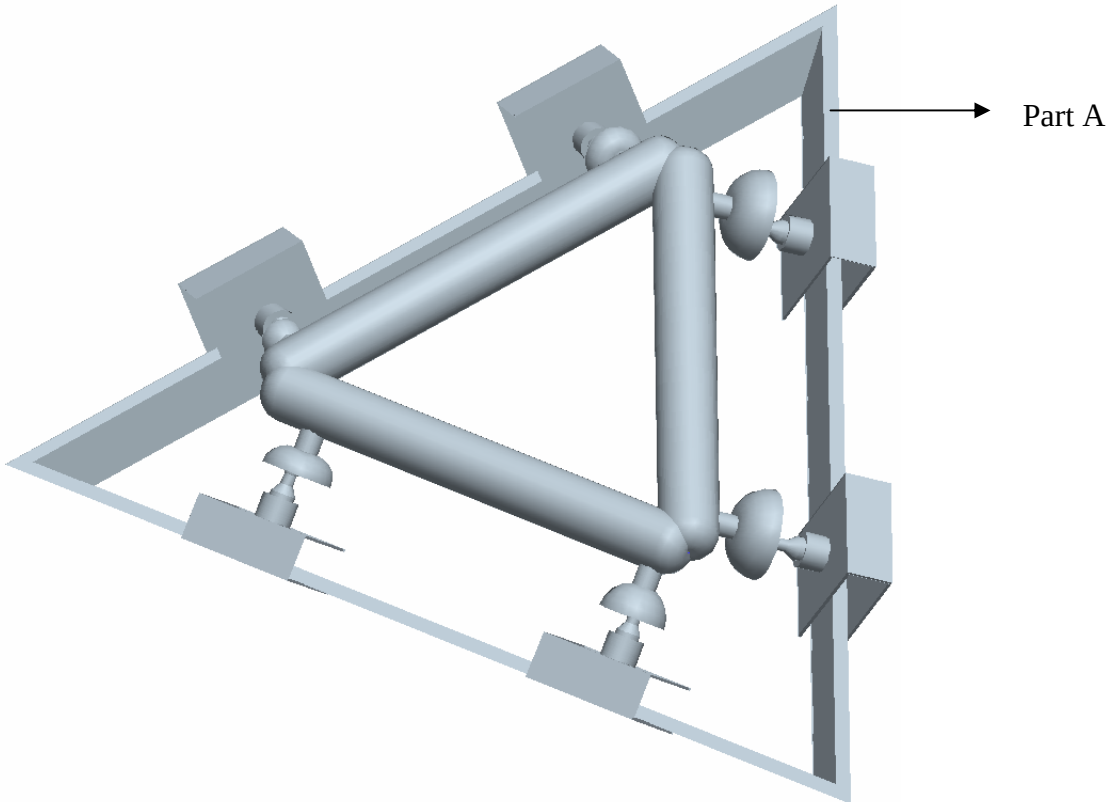


Figure 14: A match for search by joint relationship
– Stewart Platform

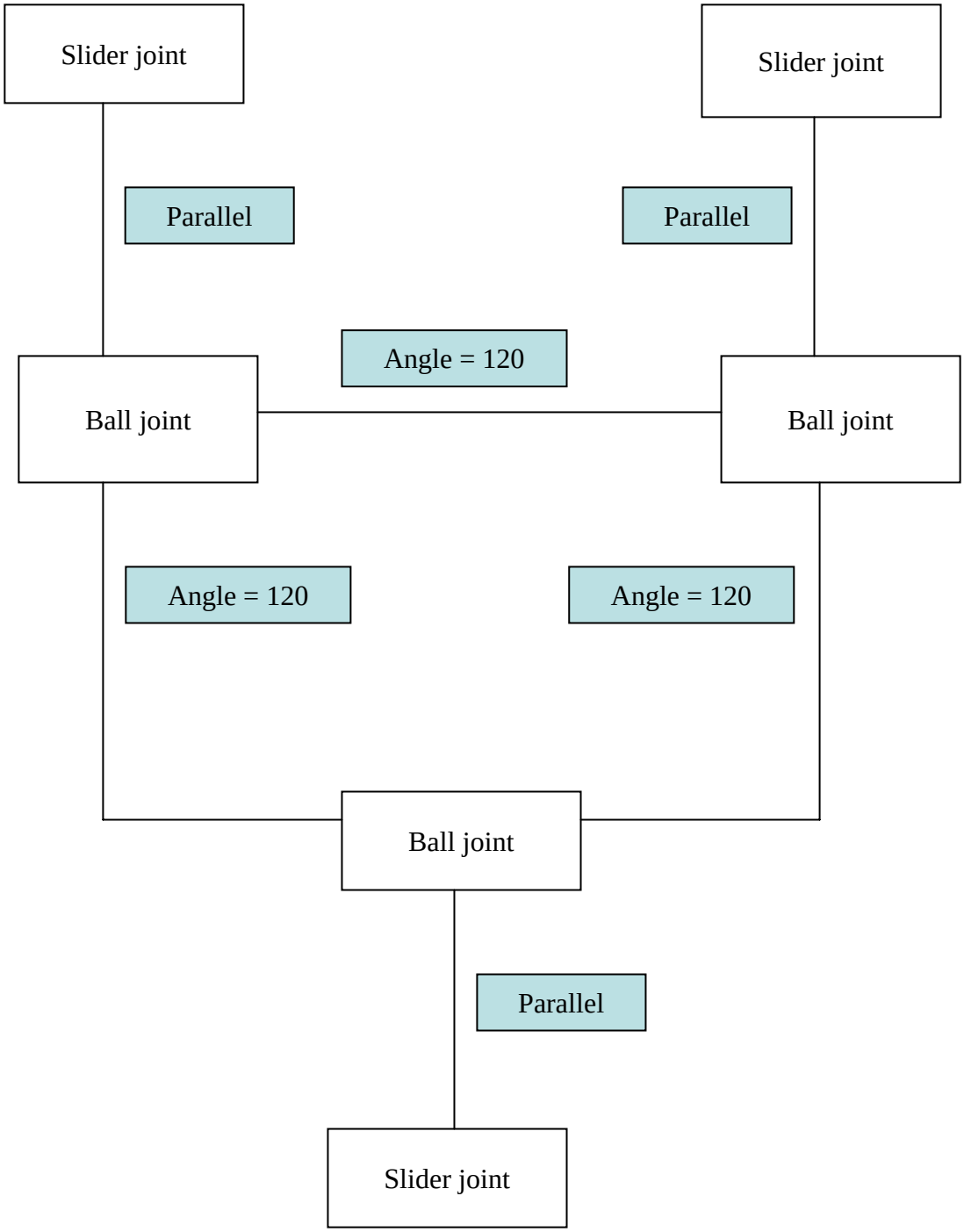


Figure 15: The joint graph for the assembly shown in Figure 12