13. Assembly Modeling And Assembling Parts

Components can consist of existing parts and subassemblies, or components can be created directly within assembly mode. Placing existing components to form an assembly is referred to as bottom-up assembly design, while creating components within assembly mode is referred to as top-down assembly design.

Parts in assembly mode maintain their associativity with their separate part files. Within part mode, if a dimension value is modified, the part instance in assembly mode is modified. Correspondingly, if an instance of a part is modified in assembly mode, the component in part mode is modified. In addition, when a part is created within assembly mode using top-down assembly design, a new pan file is created that can be modified separately within part mode. When a component is placed into an assembly, the component's separate part or assembly file is placed into memory, and it remains there until the parent assembly is erased from memory.

13.1. CREATING ASSEMBLY AND ASSEMBLY MODELING

During assembly design, parts for an assembly are created using normal part modeling tools and techniques (Figure 13.1). As with any parametric model, the intent of the design needs to be considered. In addition, how components of an assembly are parametrically linked is also an important consideration.

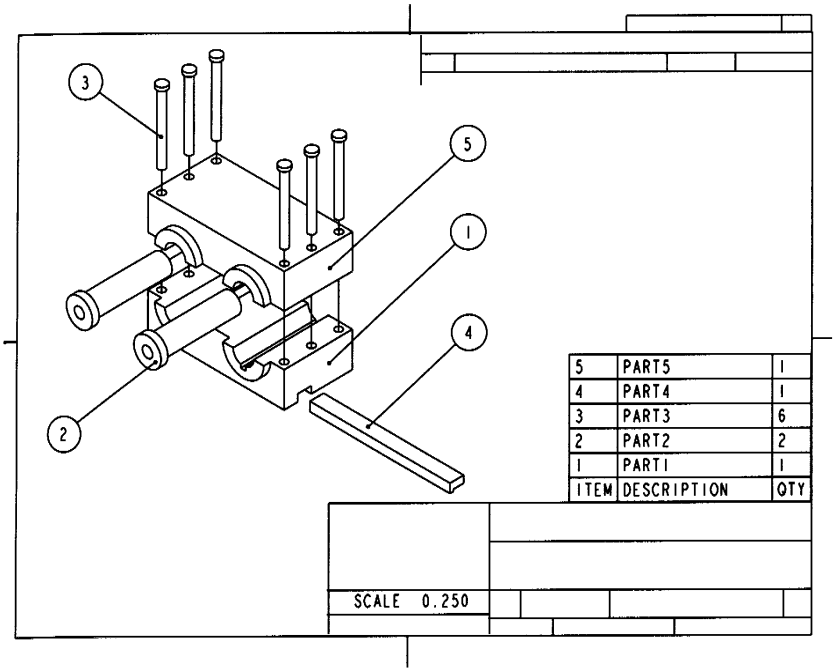

Figure 13.1. Assembly drawing of a system

You are familiar with most of the techniques used to create solid models with computer tools. Most engineered systems, however, do not consist of a single part, but comprise multiple parts that work together and form the system or assembly. A car engine is composed of several subsystems, each of which serves a distinct purpose in keeping the engine working properly. Some of these subsystems are the cooling system, the exhaust system, the fuel and combustion system, the drive train, and so on. Each subsystem is composed of other subsystems or pans. For example, the fuel and combustion system contains many single parts (the fuel line, the fuel filters, etc.), as well as other subsystems (e.g., the fuel pump).

13.2. ASSEMBLY HIERARCHY

It is easier to work with assembly models if they are organized in a logical manner. An assembly can usually be thought of as composed of several smaller subassemblies, each of which in turn may consist of other subassemblies or individual components. The organization or structure of a system is sometimes referred to as its hierarchy. The *hierarchy* can be thought of as similar to an inverted family tree. Another way of thinking of an assembly hierarchy is that it is analogous to a corporate structure. The president of the company is at the top of the hierarchy and has several vice presidents reporting to her or him at the next level. Each vice president, in turn, is responsible for several managers, and so on until you reach the lowest level in the company hierarchy, where the laborers who work for the firm are located.

A simple assembly is shown in Figure 13.2, along with an accompanying hierarchy. The assembly consists of two subassemblies, each of which in turn consists of two separate components—a gear and a shaft.

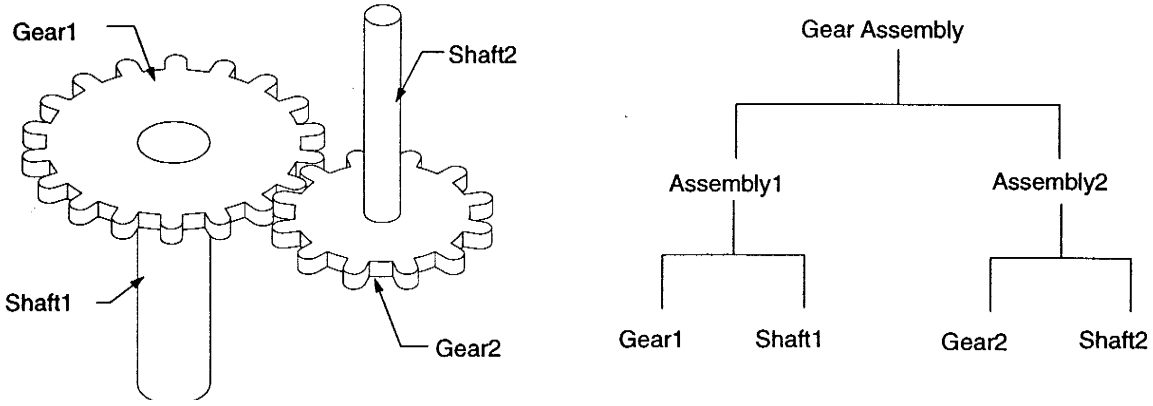

Figure 13.2. Simple Assembly and Hierarchy

Organizing an assembly in this way will enable you work more efficiently with it. The technique is similar to the way you organize your files on a computer, establishing folders to group files together in your workspace. One of the advantages in creating an assembly hierarchy is that subassemblies can be dealt with as a whole rather than as separate components. For example, one subassembly can be moved as a single unit within the system, rather than moving each of the individual components of the system separately, just like the way you can move around entire folders and all of the files in them on your computer workstation.

13.2.1. Nonhierarchical Organization

Most 2-D CAD systems support the concept of layers. Layering is a facility that allows the various graphics elements of a drawing to be grouped together in the database. This facility is used most often to control what is seen and/or editable on the screen and what is printed or plotted. Layering in most systems is *nonhierarchical;* that is, no one layer has precedence over another.

13.2.2. Hierarchical Organization

Often, groups of objects must be brought into assemblies. Such assemblies usually reflect the grouping of elements in the final product. For example, solids representing nuts and bolts might be grouped together in a hardware assembly. The most natural way to create these assemblies uses a hierarchical approach in

which the real relationships of parts in an assembly are reflected. In the example, the hardware assembly would be a subassembly of the final product. A hierarchical structure allows parts to be grouped together logically into subassemblies

13.2.3. Networked Hierarchical Organization

The use of standardized parts to improve manufacturing efficiency is a technique that goes back to the 19th century. To assist this manufacturing goal, 3-D modeling systems must share common parts, both within an assembly and with other assemblies. This sharing across assemblies constitutes a networked hierarchy in which parts exist in several hierarchical trees. The basic geometries of standard parts are stored in a central database and are never changed. When a variation on a standard part is needed, the geometry is copied, and the *copy* is modified. Standard components are stored in central locations where all members of the design team can use them.

13.4. ASSEMBLY CONSTRAINTS

The first thing you will likely want to do after setting up your system hierarchy is to orient the instances so that they are properly located in space relative to one another. When creating pans, you used constraints to establish geometric and dimensional relationships between 2-D entities. Thus, you constrained two lines to be parallel or perpendicular to one another, or you constrained the diameter of a circle to be a specific size and its center to be located given distances from lines on the drawing or from edges on an object. In assembly modeling, you can apply constraints between two 3-D parts so that the pans will maintain dimensional or geometric relationships with respect to one another within the assembly.

One of the more useful assembly constraints that can be applied is to define different lines on two different parts to be coincident with one another. This is especially useful in dealing with cylindrical shafts that fit within a cylindrical hole in another part. Figure 13.3 shows the coincident-lines constraint between the centerline of Shaft2 and the centerline of Gear2.

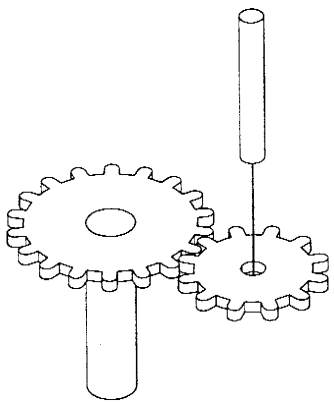

Figure 13.3. Coincident-Lines Constraint

When a component is placed into an assembly using the Assemble option, it can be fully constrained to existing components and features. This type of assembly is referred to as a parametric assembly. CAD software provides a variety of constraint types for the placement of components (Figure 13.4). The following is a description of each:

Mate . The Mate constraint type is used to place two surfaces coplanar. Any datum plane, part plane, or planar surface can be used. As shown in Figures 13.4, selected surface faces are placed along a common

plane but do not actually have to touch. When a datum plane is selected, since a datum plane has no defined thickness, CAD software provides the option of selecting either the positive side or negative side of the datum plane.

Mate Offset. The Mate Offset constraint type is similar to the Mate constraint type. Unlike the Mate type, the Mate Offset constraint places a user-specified offset distance between selected features.

Align. The align constraint type is used to place two surfaces coplanar and facing in the same direction. Like the mate constraint, the surfaces do not have to touch. In addition, the align constraint type is used to align axes, edges, curves and points.

Align Offset. The align offset constraint option is used to align two surfaces with a user-specified offset distance between each surface. The value can be either positive or negative.

Orient. Like the align constraint type, the orient constraint type orients two surfaces facing in the same direction. Unlike align, the two selected surfaces are not coplanar and no offset distance is specified.

Insert. The insert constraint type makes the axes of two revolved features coincident. The user is required to select the surface of each feature.

Tangent. The tangent constraint type makes a cylindrical surface tangent to another surface. The user is required to select the surface of each feature.

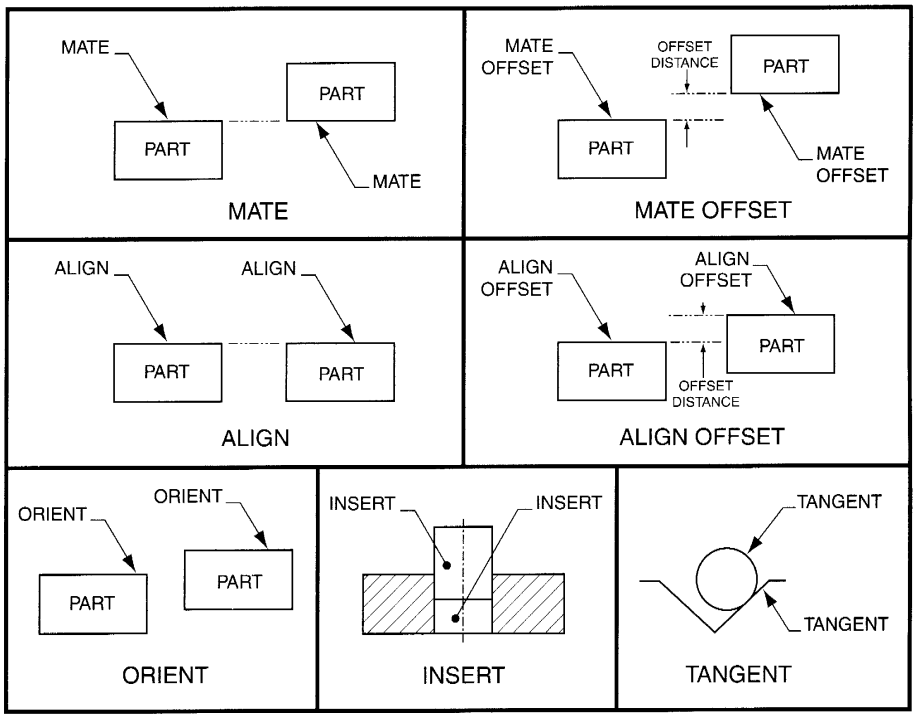

Figure 13.4 Assembly constraints

13.5. CONFIGURATIONS

Once you have established a hierarchy for your system, you can move the parts around within the system to create various configurations for the assembly. A configuration is merely a "snapshot" of your system with the parts located at specific orientations in space. From one configuration to the next, the parts and the hierarchy for the system have not changed; however, the *orientations* of the parts relative to one another have changed. For example, a car consists of many parts assembled into a system. If you open the car door, all of the same parts are still in the assembly, only its configuration has changed: The door is in an open position relative to the body of the car. If you open the hood and leave the door open, this is another configuration for the system. If you open all the remaining doors and roll down the windows, you will have yet another configuration for the system. In this way, you can see that numerous configurations are associated with any given assembly. Figure 13.5 shows three different configurations of the gear assembly of Figure 13.3.

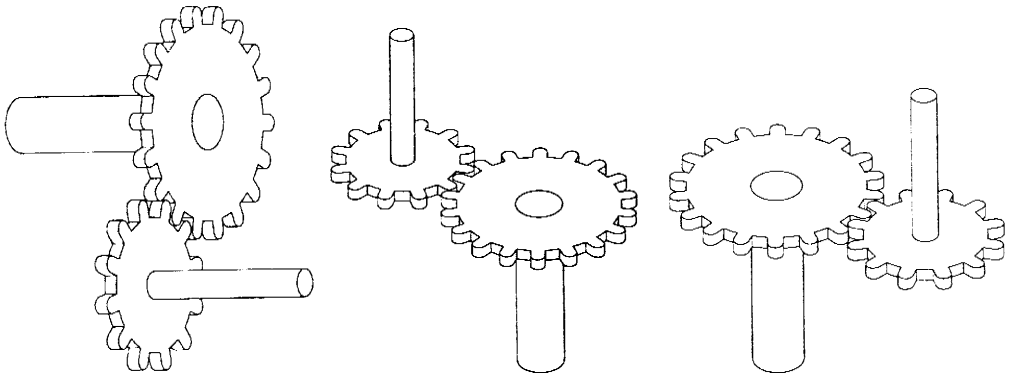

Figure 13.5. Three Configurations of Gear Assembly

13.5.1 Exploded Configurations

For most mechanical systems, an assembly drawing is necessary to put the system together. You have probably seen such a drawing if you have ever put together a furniture kit or a model. Once you have assembled your system, you can create an exploded configuration that will essentially be an assembly drawing for the system. Figure 13.6 shows an exploded configuration for the gear assembly of Figure 13.3. Note that this configuration shows how all the parts in the system will be put together or assembled.

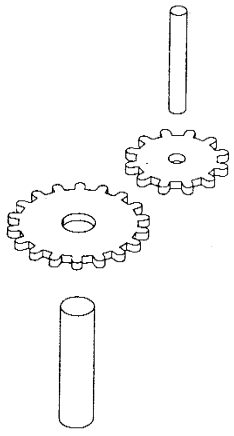

Figure 13.6. Exploded Configuration

13.5.2. Animations

Some software packages allow you to animate your system by displaying slightly different configurations in rapid succession. If you imagine rotating Gear 1 about Gear 2 in the gear assembly, the first configuration to be displayed could be the initial one. Subsequent configurations can show Gear I rotated about Gear 2 in increments of 10 degrees. In this way, there would be a total of 36 different configurations showing Gear 1 rotating in steps about Gear 2. If these configurations are displayed in rapid succession, the effect is the same as that seen with a "flip book" used to animate cartoon characters.

13.6. ASSEMBLY STRATEGY

Once again, the most important thing that you can do to successfully model an assembly is plan ahead. Before you sit down to create an assembly, you should plan the system hierarchy on paper. Objects that are logically grouped together (such as a bolt and the corresponding nut and washer) should be part of the same subassembly. Objects moved together as a unit should be grouped together in one subassembly. In short, in assembling a system, you should proceed as follows:

1. Create all required parts in the Modeler. If there is more than one copy of a given object in the assembly, you do not have to create more than one copy of the part in the Modeler.
2. Switch to the Assembly and establish your system hierarchy. Create subassemblies as necessary and logical for the assembly you are working with.
3. Orient the instances in the system relative to one another, applying assembly constraints as desired. If you use assembly constraints, remember to ground one instance so that it will remain stationary and the other instance will move relative to it.
4. Create the desired configurations for the assembly by orienting the instances and saving intermediate configurations as appropriate. Create sequences for animation from these configurations if that is desired.
5. Perform the necessary interference and clearance checks for the system. If parts are interfering that shouldn't be, reorient one or the other. You may have to go back into the Modeler to change the size of some features in order to remove unwanted interference from your system. Since CADD software has bi-directional associativity, if you change the part geometry, the instances associated with it in the assembly will also change.
6. Obtain a bill of materials for the assembly if that is desired.

References

1. D.S. Kelley, **Pro/Engineer Instructor**, McGraw-Hill, 2001
2. R.W.Lueptow, M.T.Snyder, J.Steger, **Graphics Concepts with Pro/Engineer**, Prentice Hall, 2001
3. G.R.Bertoline, et.al., **Technical Graphics Communication**, WCB McGraw-Hill, 1997
4. J.Rooney, P.Steadman, **Principles of Computer-aided Design**, UCL Press, 1997
5. F.E. Giesecke, et.al., **Engineering Graphics**, Prentice Hall, 2000.
6. F.E. Giesecke, et.al., **Modern Graphics Communication**, Prentice Hall, 2001.
7. N.N. **I-DEAS Master Series Student Guide**, SDRC, 1998.
8. R.Rizza, **Getting Started with Pro/Engineer**, Prentice Hall, 2002.
9. S.A.Sorby, **Solid Modeling With I-DEAS**, Prentice Hall, 2000.
10. S.J. Ethier, C.A.Ethier, **Instant AutoCAD Mechanical Desktop 5.0**, Prentice Hall, 2002.