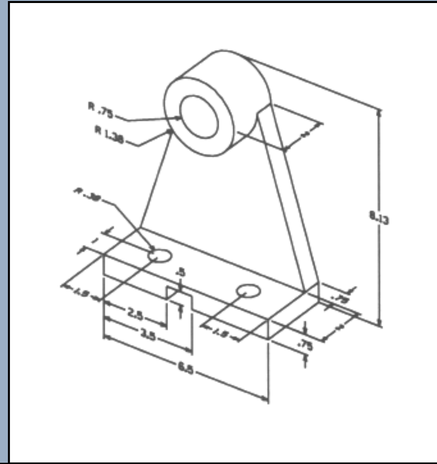
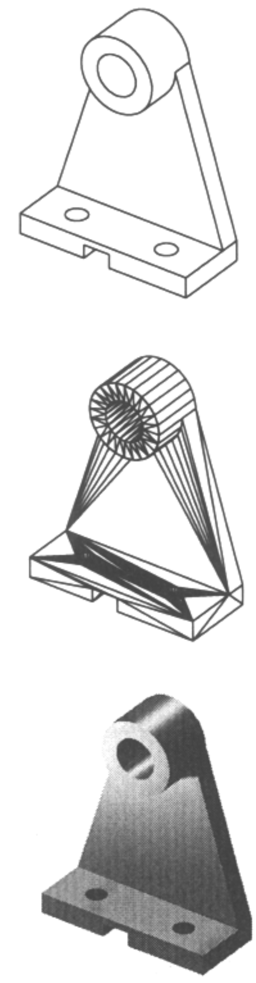
Introduction to 3D Modeling Using MicroStation V8

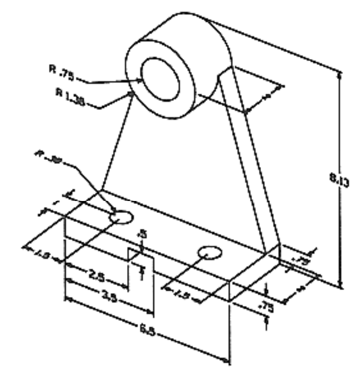


Andrew Anderson
University of Southern Maine



SDC PUBLICATIONS
Schroff Development Corporation
www.schroff.com

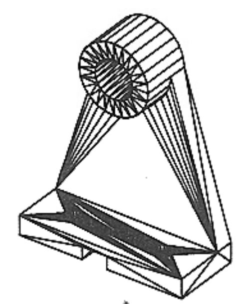
Better Textbooks. Lower Prices.

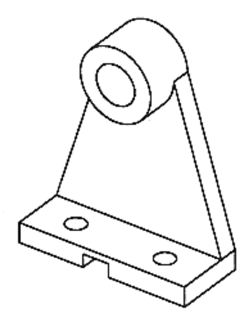


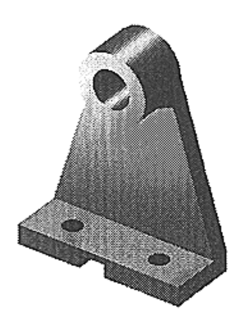
Chapter 2

3D Modeling

Introduction
Two-Dimensional
Three-Dimensional
Three-Dimensional Models
Solids and Surfaces
Wireframe Models
Surface Models
Solid Models
Feature Modeling
Parametric Modeling







Introduction

The continued development of computer-aided design software and the improved performance of computer hardware have combined to provide a variety of options for design and visualization. There is an increasing number of software packages that allow you to design three-dimensional models from which traditional 2d drawings can be extracted. This approach allows a design process more reminiscent of creating a prototype than the process of visually flattening three-dimensional objects in order to represent them with two-dimensional drawings.

Software packages not only offer different approaches to similar functions such as drawing a circle, but can also offer a much different approach to how one uses the tool as part of a design process. A basic comparison can be made between systems that primarily focus on drawing graphic representations of objects and those based on graphics used in conjunction with mathematically accurate models of objects. Two-dimensional drawing software is the computer equivalent of working on a drafting table, with the added functionality that can be derived with the assistance of a computer. It primarily focuses on representing the edges of objects on a two-dimensional plane. Three-dimensional modeling software creates something that is closer to a three-dimensional prototype. Its focus extends beyond representing the edges of objects to one that maintains information about the surfaces and spaces that result from the enclosure.

Regardless of the approach, what appears on the computer display is determined by the particular approach used for the modeling process. Martti Mantyla, in An Introduction to Solid Modeling (1987) describes the modeling process in terms of three levels.

Physical object:	An actual three-dimensional real world object.
Mathematical object:	The underlying mathematical description of an object's physical space.
Representation:	A representation of an object sent to a graphical display or other output device.

The goal of 3d modeling is to create a computer equivalent that is a reasonably complete and accurate mathematical description of an actual object. Naturally, the same physical model can be mathematically described and represented differently. What is ultimately displayed depends, to an extent, on the completeness to which the mathematical model describes an object. Also to some extent, the degree of completeness and accuracy that constitutes a model determines what can be done to the model and the information that can be extracted.

There is some difficulty in ascertaining a clear distinction between the different approaches to 3d computer-aided design software, and often it is not a significant distinction. It is a much easier distinction when comparing 2d and 3d software which has much different functionality. Software that assists in drawing, or the process of creating two-dimensional representations of an object, provides little assistance in depicting a design beyond that available with traditional board drawing. Software that assist in modeling or three-dimensional design typically provides tools to aid in the accurate visualization of objects as well as tools to assist in the basic development of a design.

One method of contrasting approaches is to assess how a system constructs, manipulates, and stores the drawing geometry. Each of the geometric shapes that is created within a design file occupies a relative position in space. Generally, that position is specified as a set of numeric values representing positions relative to coordinate axes.

Two-Dimensional

If a system maintains only two-dimensional information, the various points used to define the geometry are represented as a position on a flat plane (i.e. xy plane). Points on drawing geometry are indicated as a certain position on a horizontal (x) and vertical (y) axes. Such a system is the computer equivalent of mathematically describing a surface in terms of points on a sheet of grid paper.

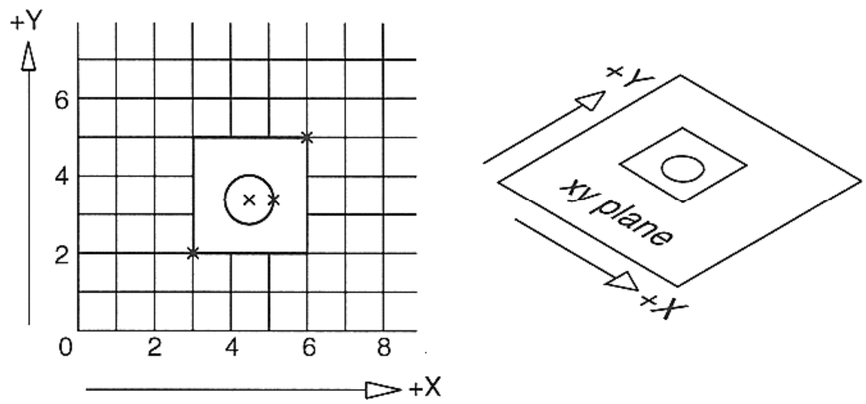


Figure 2 - 1. A 2d design plane.

The boundaries of the object placed on the 2d drawing plane are represented using line, polygon, and curve primitives. Each 2d representation of a particular view (top, front, right, etc.) must be manually constructed since the lack of depth information does not allow the user to view an object from different angles. Each view of an object must be created as an independent subset of the entire drawing. Although many systems provide assistance in creating different views, each view is essentially a separate drawing. When creating a traditional four-view drawing, the drawing is comprised of four separate drawings of the same object.

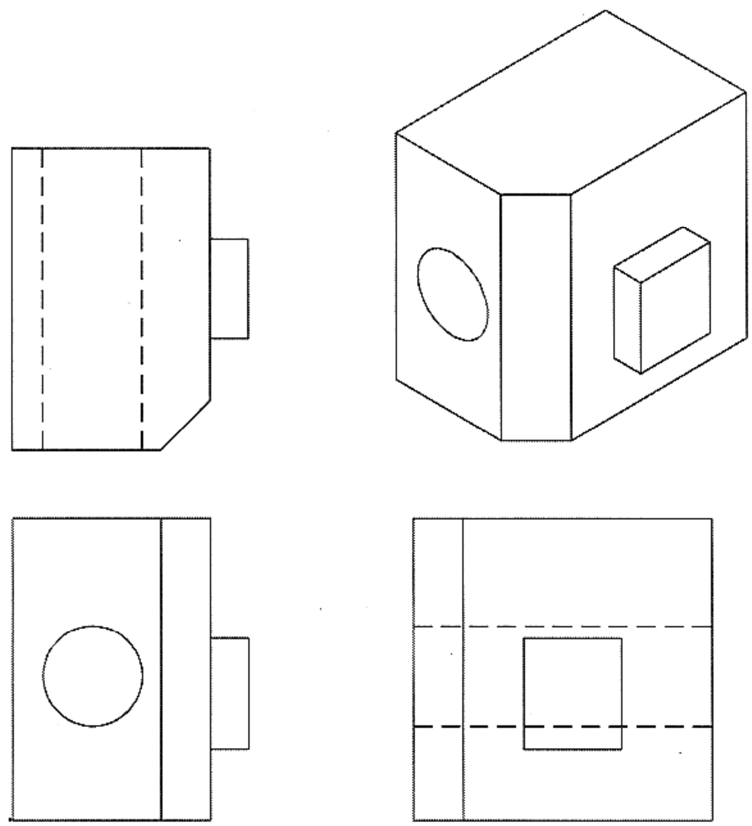


Figure 2 - 2. The four views depicted in this drawing were each drawn separately.

A major limitation of 2d drawing remains the inherent problem of visualizing a 3d object depicted as a series of 2d representations. Except for the trained drafter, mentally assembling flat projections into a 3d object often leaves the viewer with an ambiguous image of the design. Often, an equally difficult task is projecting 3d designs onto a series of 2d planes. Anyone enrolled in a basic drafting course can relate to assignments requiring the completion of missing views or the identification of inclined and oblique surfaces. Many of the limitations of 2d systems are the same as those experienced by a drafter working on a drawing board. Visualizing a design focuses heavily on the abilities of the user.

This is not to say that 2d systems are inadequate or fundamentally flawed. It is to say that they are capable of solving drawing and design needs only with 2d geometry. Items that can be adequately represented on a flat plane can often benefit from the less costly and easier 2d systems available. Even 3d systems have the capability of reducing a three-dimensional model to 2d production drawings. The most sophisticated 3d modeling software retains many of the 2d drawing tools that serve as the foundation for constructing 3d features.

Three-Dimensional

Three-dimensional systems offer the opportunity to move from the approach of representing objects by projecting onto a plane to one of creating objects more closely resembling their real world counterparts. If a system maintains three-dimensional information, the various points used to define the geometry are represented as a position within 3d space. Points on drawing geometry are often indicated as a certain position on horizontal (x), vertical (y), and depth (z) axes. The three axes define three mutually-perpendicular planes that define the space.

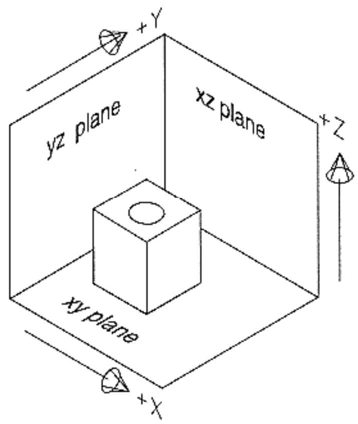


Figure 2 - 3. A 3d design volume.

The orientation of the xy-plane is relative to a particular view, typically the top or front of the object. MicroStation's x and y axes are horizontal and vertical relative to the top view. Therefore, when looking at the top of the object, you are looking at the xy-plane.

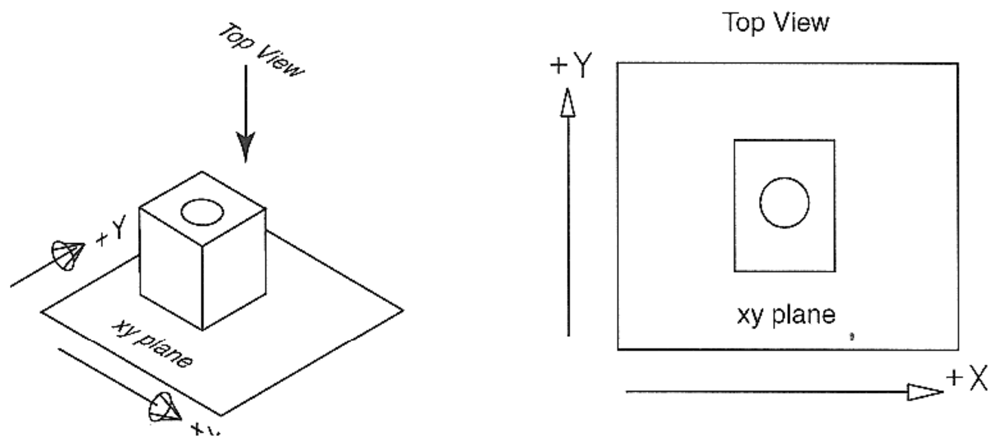


Figure 2 - 4. Top view of 3d design volume.

Since objects drawn in 3d possess height, width, and depth, they can be viewed from different angles allowing for a single representation to be used to produce multiple views. In most cases, the software allows you to view the object from fixed views (top, front, right, etc.) or an arbitrary view. When working with 3d models in MicroStation, the default views are typically top, front, right, and isometric. There are actually 8 standard views that are represented in the View Group tool box. Each view is shown in a separate view window.

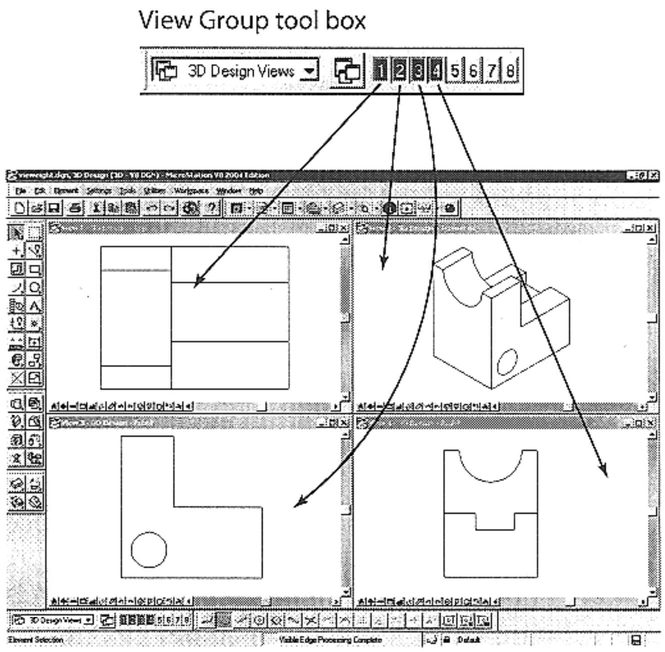
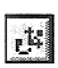


Figure 2 - 5. Display of the View Group tool box with four views enabled.

Initially, the settings in the seed file establishes the content of each view window - View 1, *top* view; View 2, *isometric* view; View 3, *front* view; and View 4, *right* view. Although each view window initially contains one of the standard views, a window can contain any standard or arbitrary view.



Selecting the Rotate View tool from the 3d View Control border tool box allows you to change the view in the window. Once you select the tool, the Rotate View settings box allows you to pick the method for setting the view.

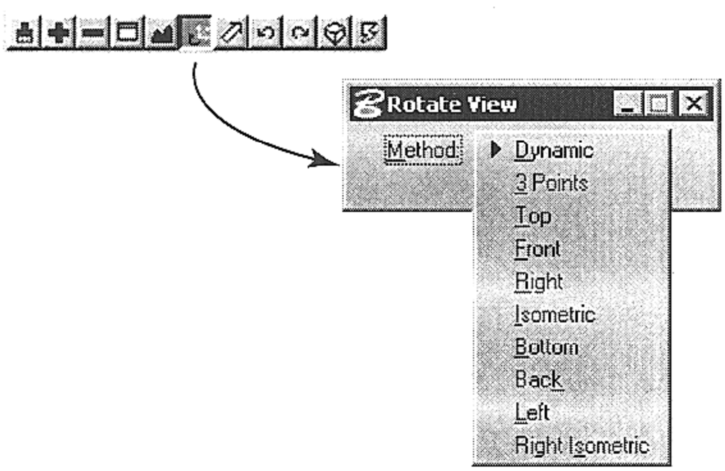


Figure 2 - 6. Rotate View tool settings.

The *Dynamic* option allows you to dynamically position the view. The *3 Points* option allows you to define a view plane by clicking three points that define a plane. The entire view is then rotated until the defined plane is perpendicular to the line of sight (LOS).

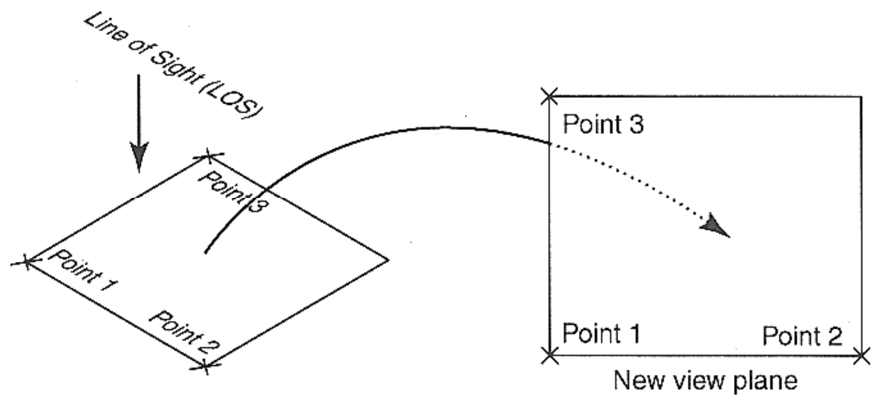


Figure 2 - 7. Rotate view using 3 points.

After selecting the view option, you are prompted to select the view (1, 2, 3, etc.) you want to apply the rotate method. The rotation is applied to only the view(s) you select. Like the other view border tools, you must cancel the command before continuing.

Three-Dimensional Models

Three-dimensional modeling comes in several different forms, with considerable variability in functionality and the extent of information regarding the geometry. Of the modeling techniques available, an approach called solid modeling provides one of the more comprehensive object descriptions and more closely parallels the actual process of creating real world objects.

Solid modeling is based on the generation of a more complete description of an object. Solid modeling software uses drawing geometry to describe an object's space rather than merely representing its edges or other features. Graphically and mathematically representing a "complete" item can provide an alternative to actually creating a mock-up or prototype of an object. In many cases, a model can be more readily used to study certain characteristics than actually using its physical counterpart. This is especially true when using some of the view "clipping" tools that allow you to look inside of an object. The more complete object information allows additional functions, including: automatically creating new surfaces when randomly cross-sectioning an object; calculating properties, such as volume, area, and moments of inertia; and checking interferences that exist between objects. It can also be used in conjunction with other software to do analysis of the design's structural performance.

While creating 3d models represents a change in approach from 2d drawing, there also exists several different ways in which to create 3d geometry. From the different approaches a distinction emerges between drawing tools and modeling tools. In a drawing context, real world objects are represented in a system as lines, arcs, and curves. For example, a system may depict a hole passing entirely through a cube as a circle drawn on opposite faces and then connected with one or more lines. In essence, an extension of the 2d drawing process with the addition of depth. In modeling, the approach is to create an object while maintaining its real world characteristics, such as edges, surfaces, and holes. Using the previous example, a model of the cube may be comprised of six sides with the hole beginning on one surface of the cube and ending on the opposite side surface.

As can be seen in the next figure, the hole passing through the object is formed by surfaces starting at one end of the shape and ending at the other.

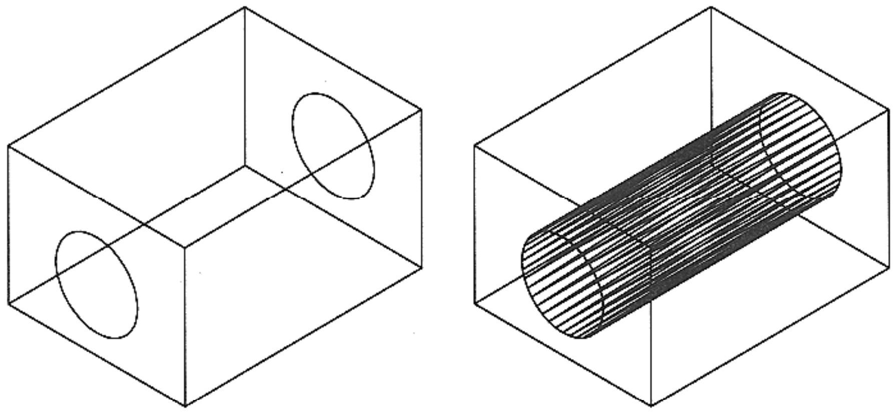


Figure 2 - 8. Two different ways to represent a hole in an object.

It is not always clear at which point one crosses the line between drawing an object and modeling an object, since creating 3d objects often starts by drawing 2d shapes or profiles. The profiles are then extruded to form 3d objects. It does become somewhat clearer when you begin to use some of the tools designed primarily for creating or modifying 3d objects. The nature of the tools used to create and manipulate shapes is based on the approach the software takes in creating three-dimensional models.

In MicroStation Version 8, the term model is used to describe both 2d and 3d designs stored in a single file. A MicroStation design file serves as a container that holds one or more models. Models are comprised of 2d (lines, arcs, circles, etc.) or 3d (slabs, cylinders, cones, spheres, etc.) design elements. One model can be referenced into another model.

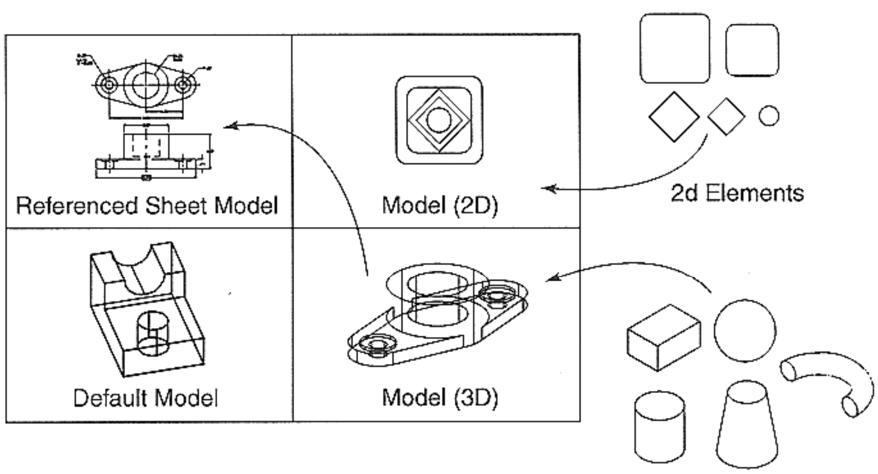


Figure 2 - 9. An illustration of the design file "container."

The design file can contain several models. Each model can be totally different in terms of working units and other design settings.

There are actually four different types of models used by MicroStation: 2d and 3d *design* models, and 2d and 3d *sheet* models. Models are accessed by selecting File > Models or by selecting the Models tool located on the Primary tool box.

When selecting the Models tool, a dialog box is displayed that lists current models in the design file. All design files have at least one default model.

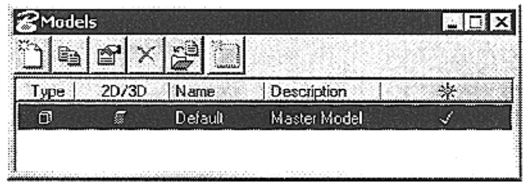


Figure 2 - 10. Models dialog box with default model.

The Models dialog shows the type of model (design or sheet), whether it is 2d or 3d, the model name, description, and if the model can be placed as a cell.

The dialog also provides tools for creating new models and modifying existing models. It also provides a tool for importing a model from another design file.

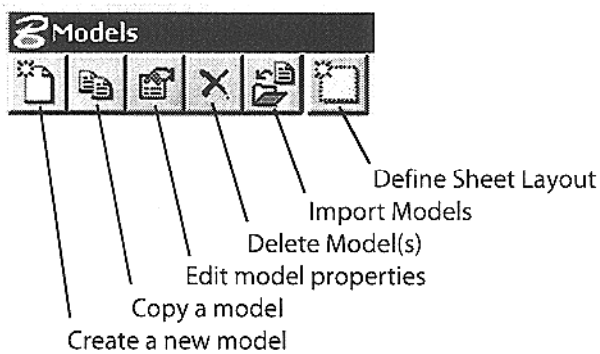


Figure 2 - 11. Tools available in the Models dialog.

The following describes the function of each tool:

Tool	Description
Create a new model	This tool is used to create a new model in the design file.
Copy a model	In addition to creating a new model, you can use this option to copy an existing model.
Edit model properties	This option displays and allows you to edit the properties of an existing model.
Delete Model(s)	An existing model can be deleted using this tool. Note that you cannot delete the default model.
Import Models	In addition to copying other models in the current design file, you can use this option to import a model from another design file. The import option requires you to select a design file and a model in the file for import.
Define Sheet Layout	If the model is a sheet, you can specify the sheet size, the location of the sheets origin, and the rotation of the sheet.

When you create a new model, you need to determine if the model is a *design* or *sheet* model. You must also determine if the model is *2d* or *3d*.

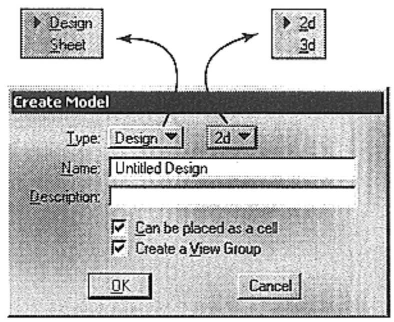


Figure 2 - 12. The create model dialog box.

A design model is used to develop a 2d or 3d design. In a 2d model, all points on drawing elements reside on a 2d plane. With a 3d model, the points reside within a 3d space. When you are working on a 2d model, AccuDraw provides only x- and y-axis input. All three axes are provided when working with a 3d model.

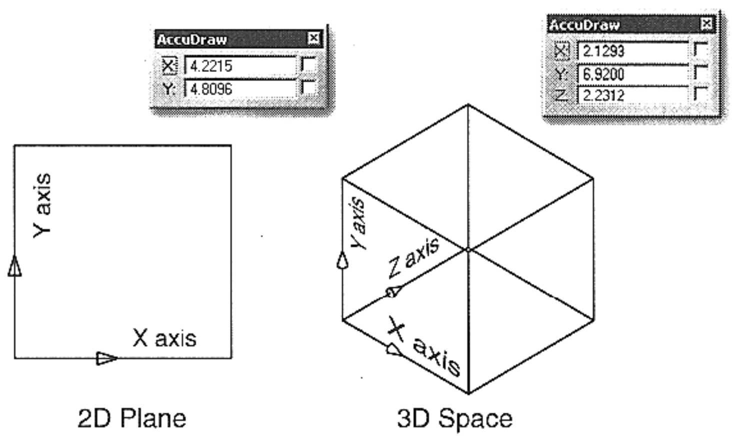


Figure 2 - 13. Coordinate axes define 2d plane and 3d space and AccuDraw.

A sheet model is typically used to reference several views (front, top, right, etc.) of a 3d model into a single view for the purpose of outputting the design to a printer, thus the name "sheet."

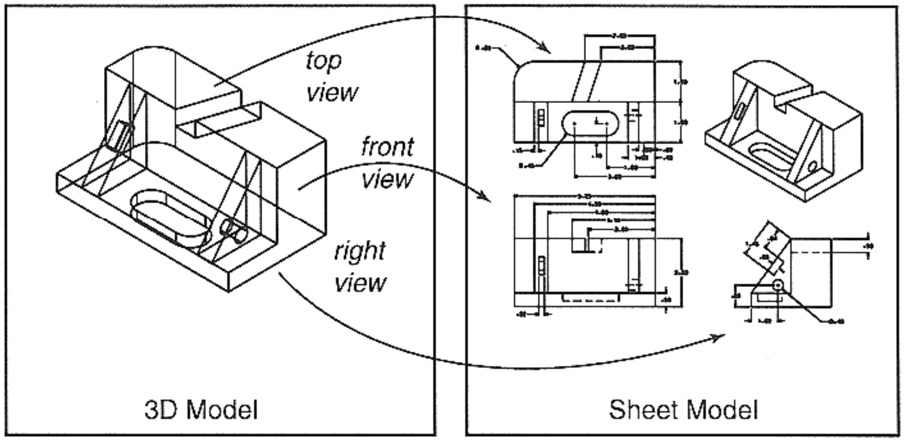


Figure 2 - 14. Views of a 3d model referenced in a sheet model.

When you create a new MicroStation design, the file used as a start-up (the seed file) for the new design determines if the *default model* for the file is 2d or 3d. Even though you may begin with either a 2d or 3d default model, other models created in the same design file can be either 2d and 3d. Similar to creating a new design file based on a seed file, you have the option to create a new model based on one of the models contained in a seed file. If you elect to base a new model on a seed, the settings for the new model are taken from one of the models in the selected seed file. Remember, seed files are merely other MicroStation design files from which settings are "borrowed." The design file is a container that may have several different models. You not only select the design file to serve as the seed, you also select which model in the design file is going to used. If the seed file contains only a single default model, that is the model used to create the new design file.

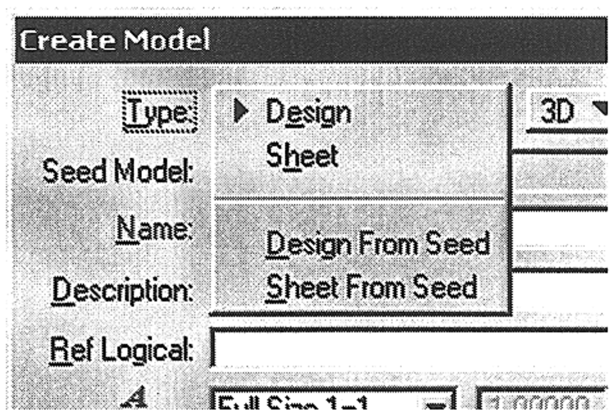
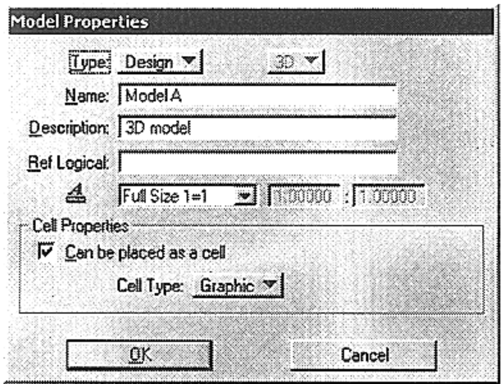
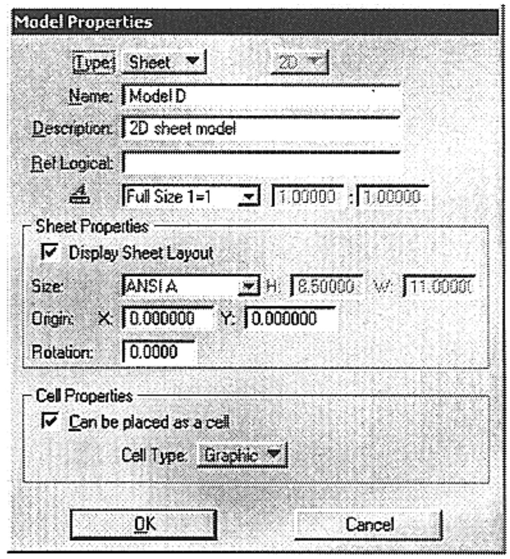


Figure 2 - 15. Options available when creating a new model.

Each model has several settings, based on the type of model.



Design model properties



Sheet model properties

Figure 2 - 16. Model properties are based on the type of model.

All models are provided a *name* and *description*. Models also can be given a *ref logical* name. This name is used to identify the model when it is referenced to another model. The XM Edition adds an option to specify the *Sheet Number* for sheet models only. Each model also contains an *annotation scale*. The annotation scale sets the scale to properly size text and dimensions found in the model. This allows you to maintain a consistent text and dimension size when printing output of the model at different sizes. The scale defaults to 1:1, but can be set to match the scale of the output. You can select from a list of common scales or select the *custom* option and set your own scale.

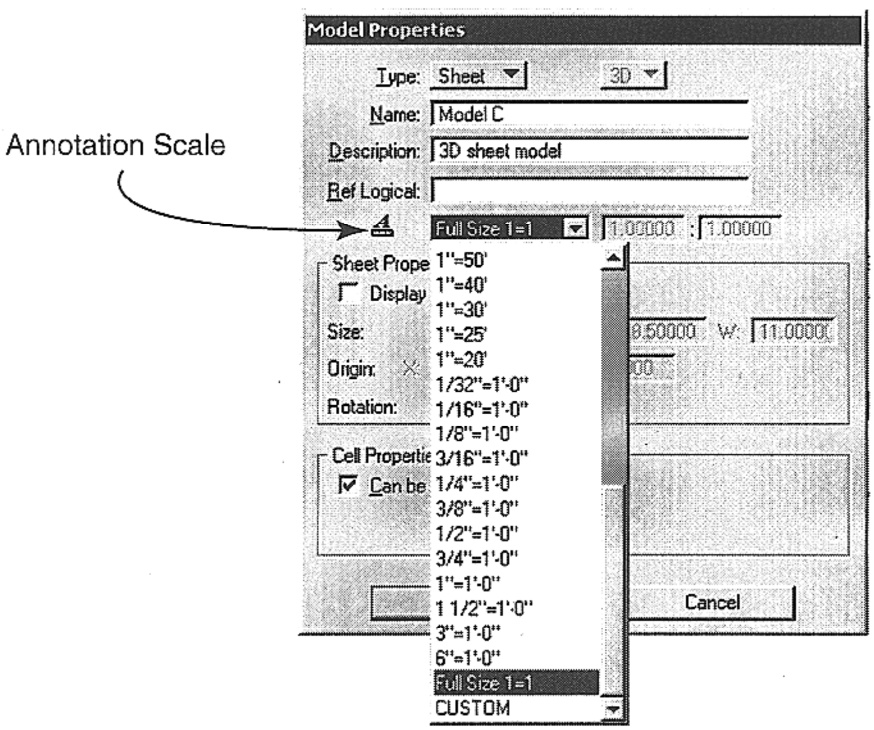


Figure 2 - 17. Options for setting a model's annotation scale.

The annotation scale is used to maintain a consistent output size when plotting and printing at different sizes (plot scales). It automatically adjusts the printed height and width, and the dimensions to maintain them at the size they were inserted into the file. It does not change the value of the text or dimension, only the text height and width attribute. For example, assuming you want the final text height size to be .2 inches, you could use the annotation scale to assure that the final output is maintained at .2 inches even if the plot scale is changed. If the annotation scale is set to 1:1 and the plot scale is set to 1:1, the text and dimensions that are inserted into the design file at a height of .2 inches would plot at .2 inches. If the annotation scale is adjusted to .5:1, the text and dimensions that are inserted into the design file at a height of .2 inches are automatically adjusted (scaled by the ratio of .5 to 1) to a height of .1 inches, thus when plotted at the larger scale of 2X, the final size would remain .2 inches. If the annotation scale is adjusted to 2:1, the text and dimensions would automatically be adjusted (scaled to a ratio of 2 to 1) to a height of .4 inches. The annotation scale is applied to only the text and dimensions that had annotation scale enabled when they were placed in a model.

The annotation scale parameter in the model's properties determines the size of text and dimensions that have been inserted with the annotation scale enabled. You have the option to enable the annotation scale when inserting text or dimensions. The options can be seen when using the Place Text and the Dimension Element tools.

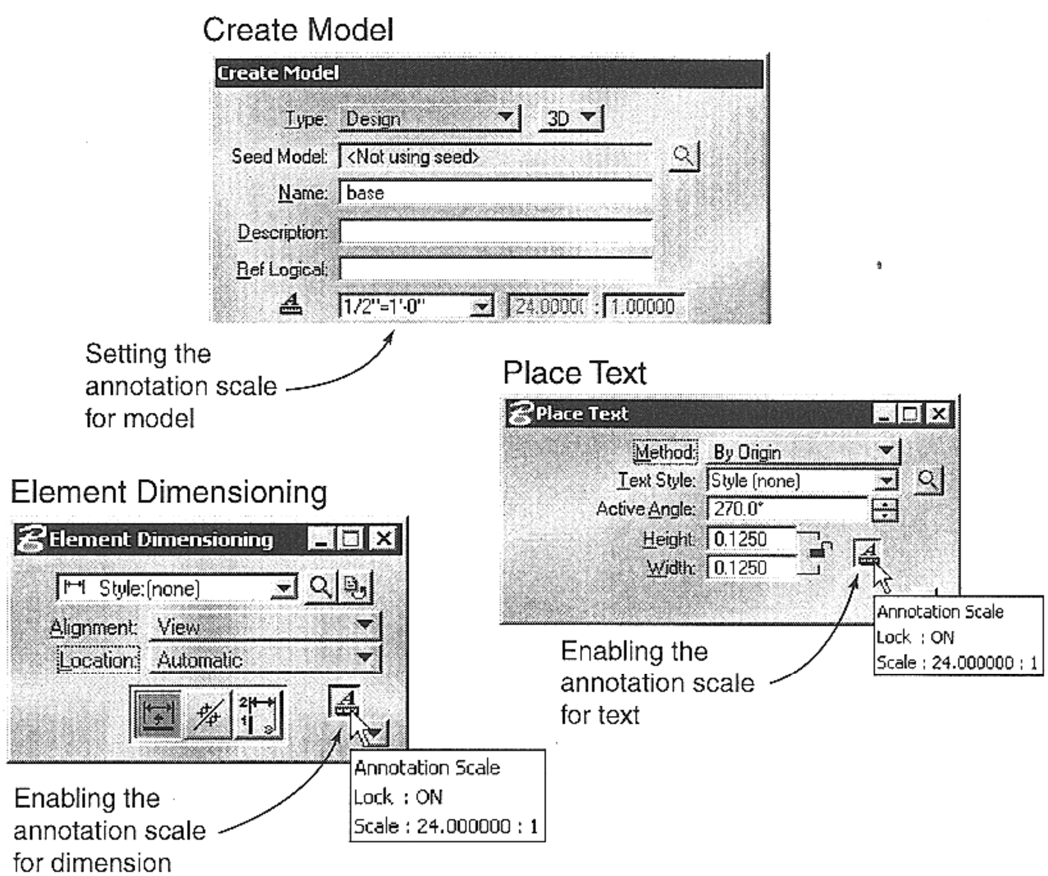


Figure 2 - 18. Annotation scale set for model and enabled for text and dimensions.

The XM Edition adds the option to set the *Line Style Scale* to the annotation scale or to a specified scale.

Each model can also be placed into a drawing as a cell. A cell is a drawing component created in MicroStation that is normally stored in a file referred to as a cell library. Typically used to store frequently used design components, a cell library can be attached to any number of design files. Any DGN file can serve as a cell library and any cell library can be opened as a DGN file. If you open a cell library using File > Open, each cell in the library is represented as a separate model. If you select Element > Cells and then use Files > Attach to attach a design file that contains several models, each model that has *Can be placed as a cell* enabled in the original file, is listed as a cell and can be inserted into the drawing as a cell.

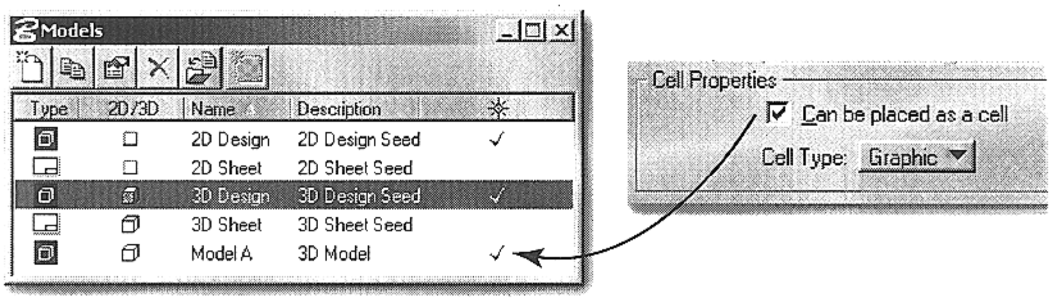


Figure 2 - 19. Models can be enabled to allow them to be inserted as a cell.

Sheet models have an additional set of *sheet properties*. These properties control the sheet size, the location of the sheet's origin, and the rotation of the sheet. It also determines if the boundary of the sheet layout is displayed. If the sheet layout is displayed, it is also adjusted to reflect changes in the annotation scale.

The type of model is shown in the dialog box in the *Type* column. Design models and sheet models each display a distinct icon. There is also distinct icons for 2D and 3D models displayed in the *2D/3D* column of the dialog box.

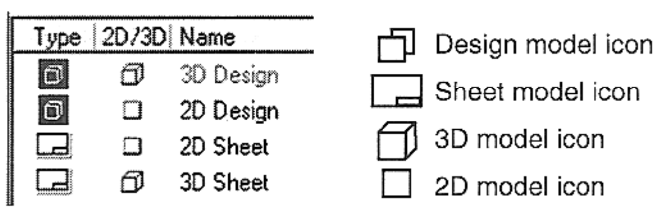


Figure 2 - 20. Icons used with 2D, 3D, design and sheet models.

The XM Edition allows you to associate a *Border Attachment* to a sheet model. Changes to the annotation scale of the sheet are automatically reflected in the border attachment.

The steps in creating a border attachment begin by creating a sheet model that contains the desired border and/or title block. Typically, this is located in a separate file (i.e. BorderB.dgn). You then can reference the border using the Attach Reference tool found in Tools > Reference. Make sure you reference the model in that file that contains the border. Enable *live nesting* if you referenced other files in creating the border or title block. The last step is to tell MicroStation you have attached a reference you want to treat as a border attachment. This is done by selecting Utilities > Keyin and entering *sheet set borderattachment* followed by the file name. For example:

Sheet Set Borderattachment BorderB.dgn

Once you have referenced the border attachment to a sheet model, any changes to the annotation scale are reflected in the attachment.

Solids and Surfaces

MicroStation provides tools to create both surfaces and solids. Surfaces are open shapes that are defined by a minimum of three points. A solid is a shape that completely encloses a volume of space. The next illustration shows some of the distinction between surfaces and solids. That distinction is made somewhat clearer by rendering the shape. In many cases, you will not know if it is a surface or solid without obtaining further information or attempting an operation that may require a particular type of geometry.

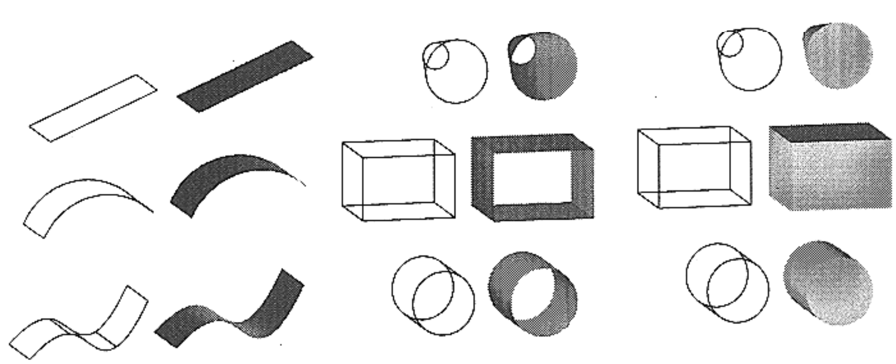


Figure 2 - 21. Surfaces and solids.

MicroStation prepares solids and surfaces in several different ways. There is not always a concise distinction between the different approaches to constructing surfaces and solids. It actually provides two different approaches to solid modeling. MicroStation's SmartSolids determines what is the best way to represent 3d design elements.

1. Start MicroStation and load the *3dmodels.dgn* file that is provided on the accompanying CD. The file needs to be copied into the default design file folder. If it is not located in that folder, you will need to identify the drive and folder that contains the file. If necessary, use Fit View to see the entire design file.

2. Select the **Utilities > Render > Phong** command. When prompted to select the view to render, click on View 2.

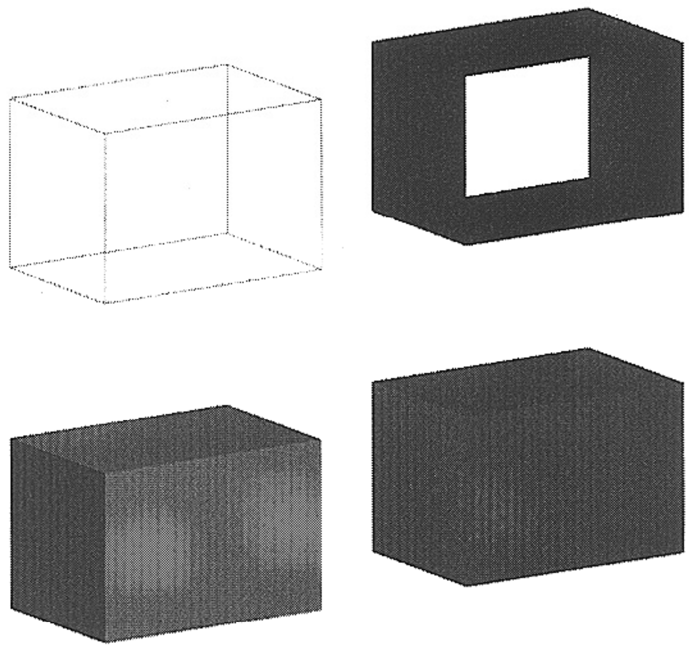


Figure 2 - 22. File depicts different approaches to modeling in 3d.

Three-dimensional models can be classified in terms of how the models are created and stored. The different approaches include wireframe, surface, and two different type of solid models. The *3dmodels.dgn* design file illustrates all four.

Wireframe Models

Wireframe models add a depth axis (z) point to the two-dimensional axes (x,y) points of a 2d system. Each line or "wire" represents an edge of the actual object. Only the boundaries of surfaces are represented and not the surfaces themselves. Note that the object at the upper left in the *3dmodels.dgn* file represents a wireframe model. The example provided is for illustration purposes since MicroStation does not typically create wireframe models.

3. Select the Update View tool in the view border and click View 2. Reset the command (click right mouse button) once the view is refreshed.

4. Select the **Element > Information** command from the menu or the **Element Information** tool on the Primary tool box.

Note that if a tool box is not displayed, select **Tools > Tool Boxes** and enable the item. If a particular tool is not displayed on a tool box, place the cursor over the tool box and click the right mouse button to list available tools.

Element information > Identify Element

Pick any edge on the object at the upper left corner (green object). Select the *Details* tab in the Element Information dialog.

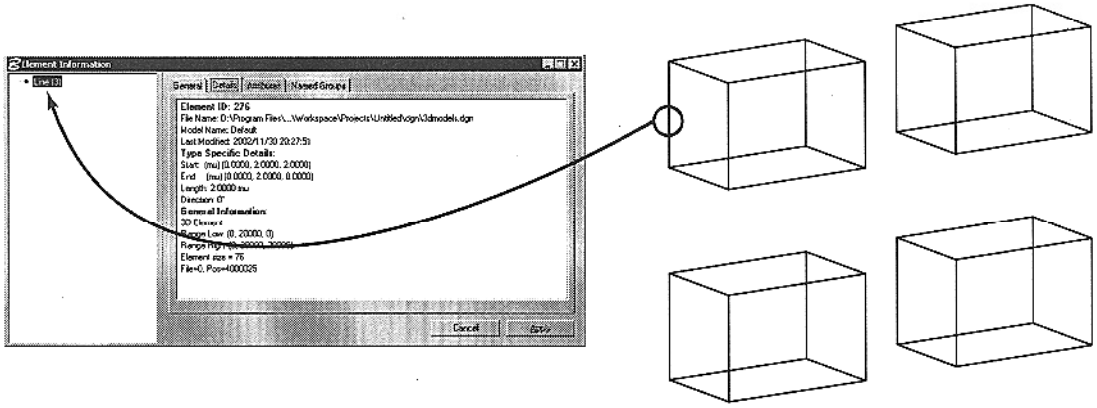


Figure 2 - 23. Element information for wireframe object.

The selected object is a wireframe comprised of a series of 3d elements (lines). You can pick any of the lines on the object and the *Detail* or *Geometry* (XM) section shows related information. Note that each of the lines that comprise the upper left-hand shape is a 3d element with start and end points residing at specific x, y, and z locations.

Wireframe modelers generally allow you to quickly edit a model since it does not need to maintain surfaces or the interrelationship that must be maintained between surfaces when working with solid objects. The major drawback of wireframe systems is the ambiguity that results when modifying and viewing models. Since it lacks surfaces and information about the relationship between surfaces, they can be modified in often undesirable ways and viewing both simple and complex drawings can lead to several visual interpretations of how the real object may appear.

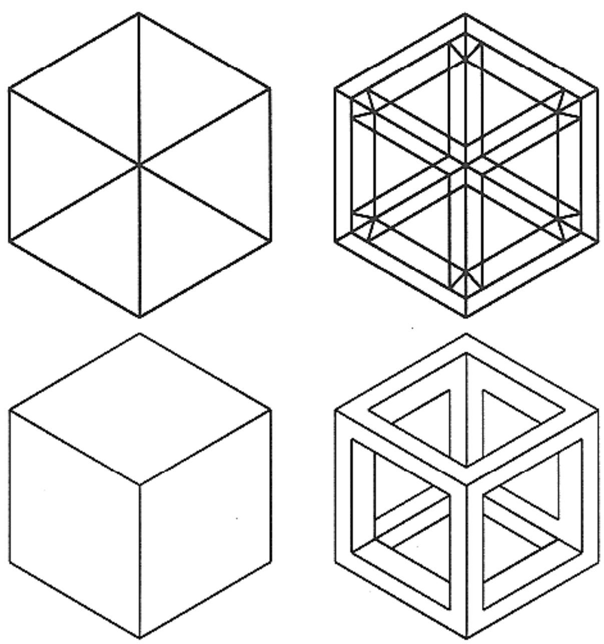


Figure 2 - 24. The illustrations at the top represent wireframe models and those on the bottom are the same models rendered as surface models.

Although the lines normally not visible in a particular view can be manually removed, the wireframe database lacks the surface information which can aid in determining the lines that are visible and those hidden. Although algorithms exist to perform some hidden-line removal on wireframe data, the lack of surface information does result in inconsistent interpretation of the lines. Some wireframe systems depict the connection between surfaces with attachment lines. Although such lines do indicate which primitive shapes are connected, they may not provide lines that properly depict a features edges when viewing the drawing from different angles. The following illustrates how the lines may be used to depict a hole and a cylinder in a wireframe representation of an actual hole and cylinder.

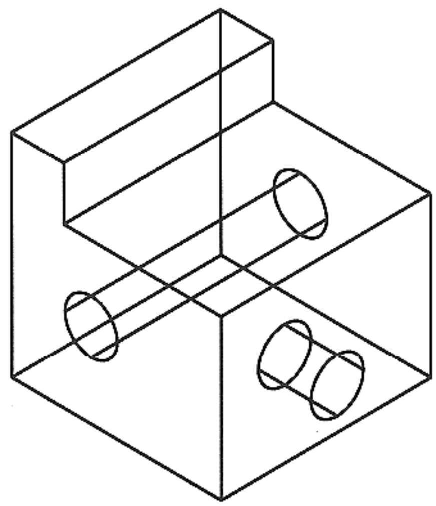


Figure 2 - 25. An illustration of a wireframe model's representation of a hole and a cylinder.

The same lack of surface and relational information does not allow cross-sectioning an object with an arbitrary cutting plane. The following illustrations represent the surfaces that would result from sectioning a solid model and the lack of surfaces that result from cutting a wireframe model.

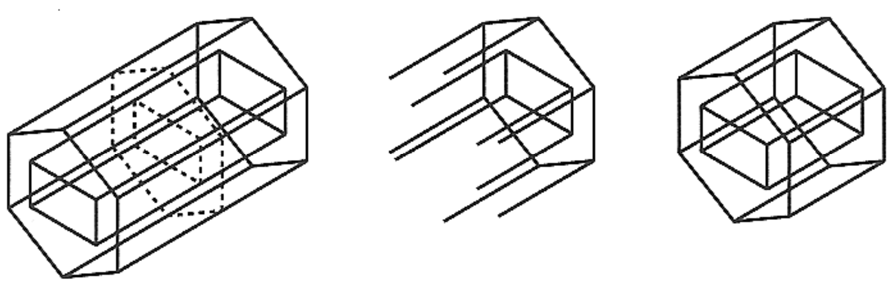


Figure 2 - 26. A cross-section of a wireframe and solid model shown without rendering.

Surface Models

As the name implies, a surface model is based on geometry that forms the surfaces of objects. Surface modeling is analogous to stretching a thin fabric over a wireframe. Since the boundary of an object consists of surfaces, the system has more information to determine the lines normally not seen in a particular view.

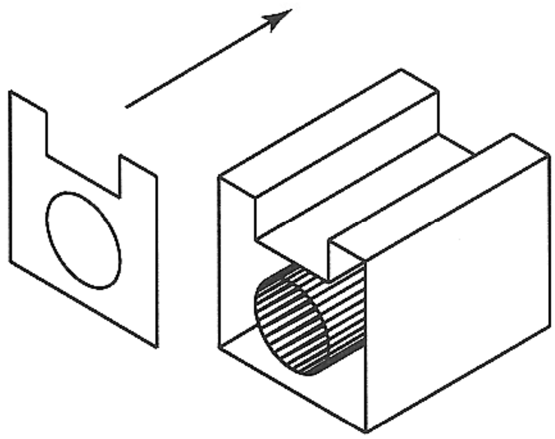


Figure 2 - 27. Two-dimensional profile extruded as a surface model.

Surface modeling adds a degree of complexity to the creation of an object. Although it results in a more complete database when compared to two-dimensional and wireframe systems, the inclusion of surfaces in itself does not result in a complete description of a 3d model. A surface modeler may lack sufficient information to determine what regions of 3d space are "inside" or "outside" an object.

5. Select the Element Information tool on the Primary tool box.

   Element information > Identify Element

   Pick any edge on the object at the upper right corner (red object) in *3dmodels.dgn*. Clicking the data (left mouse) button selects an edge of the object.

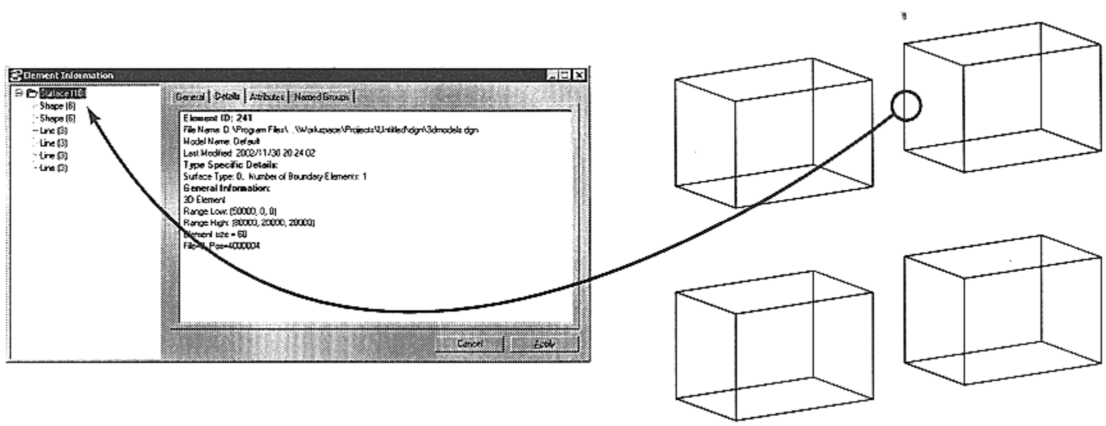


Figure 2 - 28. Element information for surface object.

The object is a surface model comprised of a series of 3d elements (shapes and lines). This model is similar to one created with MicroStation's **Place Slab** tool found on the 3d Primitives tool box. This tool provides options to create either a *surface* or a *solid* type model. In this case, the type was set to *surface* when the slab was created. Note the header (in element information) indicates the object is a surface model.

Solid Models

Solid models describe part geometry in terms of valid objects rather than a collection of surfaces. To represent a valid solid, all geometry comprising the part must be maintained in a manner that is consistent with the real world existence of the part. For example, in the other approaches described earlier, a drilled hole is often represented as a circle. It is relatively easy to draw one circle within another on the same plane. In other words, you could depict a hole within a hole. A solid modeling system attempts to mimic real world geometry and would not consider a hole inside a hole to be a valid solid model.

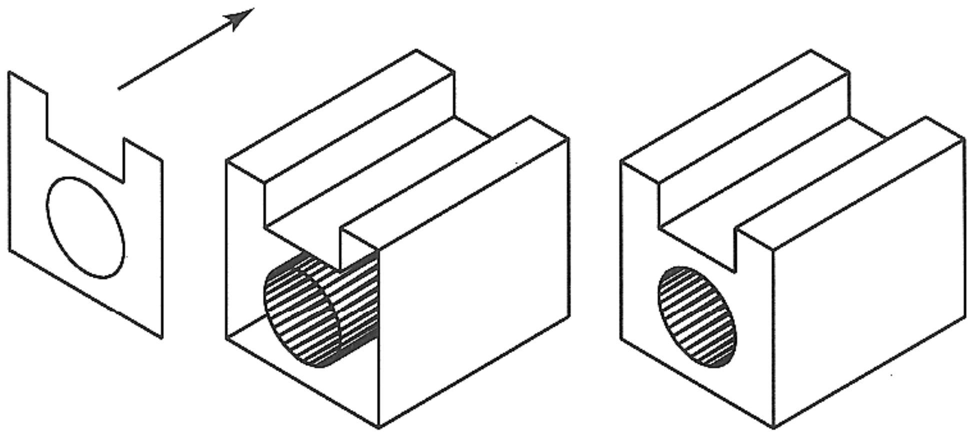


Figure 2 - 29. Two-dimensional profile extruded as a surface model and a solid model.

Another example can be used to illustrate the difference in surface and solid modeling. A surface modeler depicts a cube with six surfaces or two surfaces connected by lines. A solid modeler depicts the same cube shape with surfaces, but those surfaces enclose a volume and are integral to forming a valid cube-shaped solid. Trying to remove a surface from a solid cube violates the rules of a valid solid. The surfaces have no thickness of their own and merely represent the boundary of an enclosed space. The goal of a solid modeler is to maintain valid representations of solid objects. A valid representation of a solid must be able to describe a completely enclosed space, avoiding "dangling" lines and surfaces without shared edges.

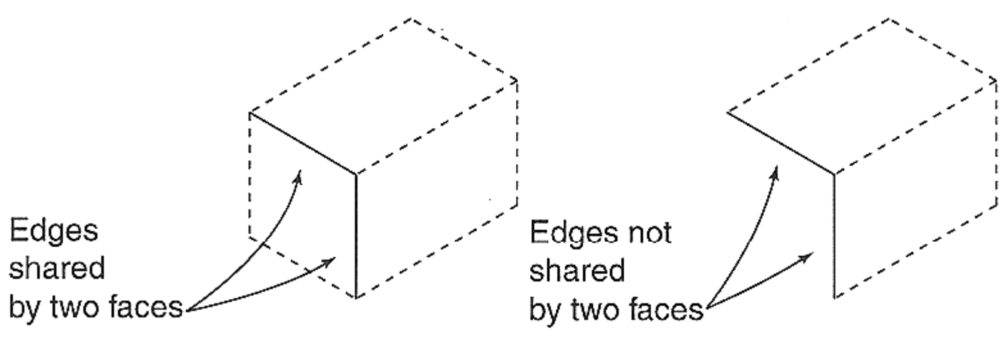


Figure 2 - 30. Depicting of an edge not shared with any other surface.

6. Select the Element Information tool on the Primary tool box.

Element information > Identify Element

Pick and accept any edge on the object at the lower left corner (yellow object) in *3dmodels.dgn*.

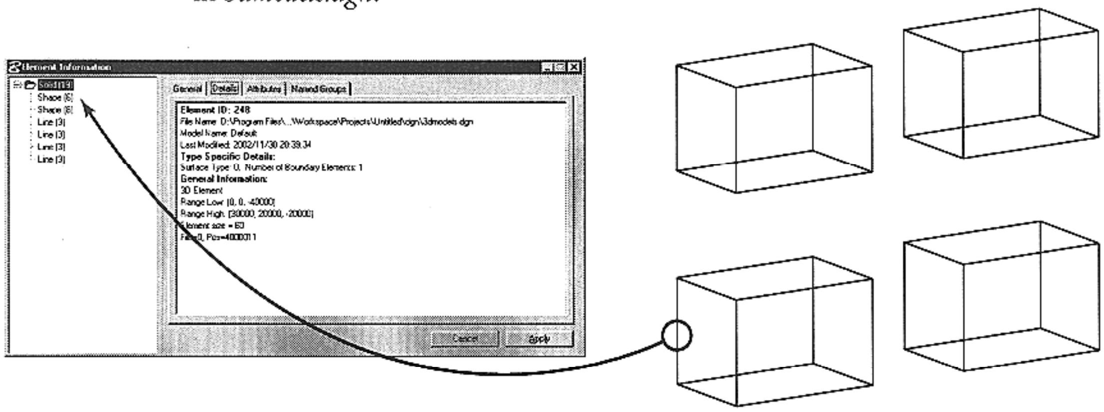


Figure 2 - 31. Element information for solid object created with MicroStation.

This object is a solid model comprised of a series of 3d elements (shapes and lines) similar to the surface model. This model is similar to one created with MicroStation's **Place Slab** tool found on the 3d Primitives tool box.

While this object is very similar to the previous surface model, the type parameter was set to *solid* rather than *surface* when the model was created. Note that the object is comprised of 3d elements similar to those for the surface model, but the header indicates it is a solid. In essence, a solid created with this tool in the 3d Primitives tool box is a "capped" surface model.

7. Select the Element Information tool on the Primary tool box.

   Pick any edge on the object at the lower right corner (light blue) in *3dmodels.dgn*.

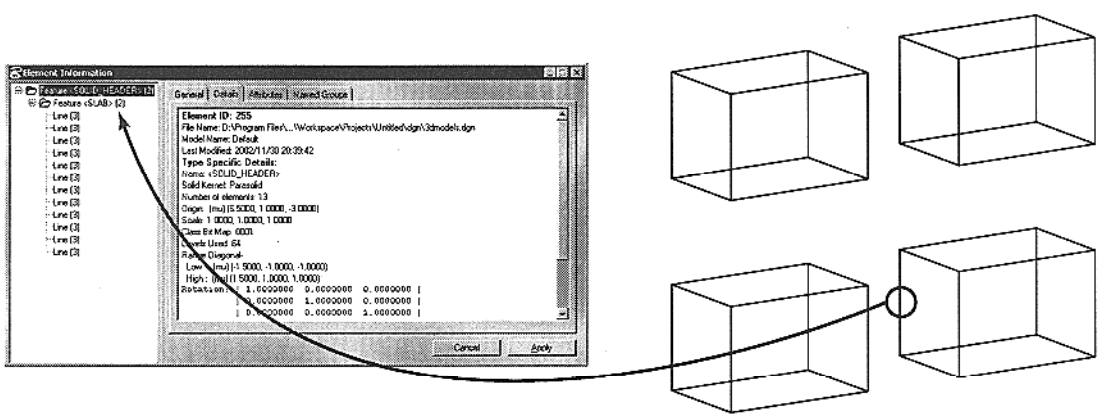


Figure 2 - 32. Information for solid feature object created with feature modeling tools.

This object is a solid model created with MicroStation's **Slab Feature** tool found on the Primitive Feature Solids tool box. This tool, available with feature modeling, creates a parametric solid using the Parasolids kernel.

Many solid modeling software packages use solid modeling kernels provided by companies specializing in solid modeling tools. Two of the most prominent packages are the Parasolid® solid modeling kernel from EDS and the 3d ACIS® Modeler kernel from Spatial Technologies Inc. MicroStation has incorporated both products at various times in their solid modeling applications. Which one of the solid modeling engines is used depends on the particular version and file format. In Version 8, MicroStation uses the Parasolids kernel when solid models are incorporated into a design file. When a design containing solids is saved as a DWG file or when working in the DWG mode, 3d elements are placed as ACIS bodies. Since AutoCad uses the ACIS 3d Modeler kernel, this assures the model is accurate and compatible when saved in the DWG file format.

After completing this activity, you can save the file as an AutoCad DWG file. You can then use the Element Information tool to explore the four shapes to see how they are different when saved in this format.

Feature Modeling

As was described in the earlier examples, MicroStation offers different options for doing 3d modeling. The MicroStation foundation product comes with the ability to work in 3d and create 3d models. The tools available with feature modeling offers true solid modeling and provide enhanced capabilities for creating, manipulating, and managing 3d objects. Although the various options available with MicroStation offer similar 3d tools, they create and manipulate solids based on different approaches or using different modeling kernels. Various functions when applied to 3d elements actually convert models from one form to another.

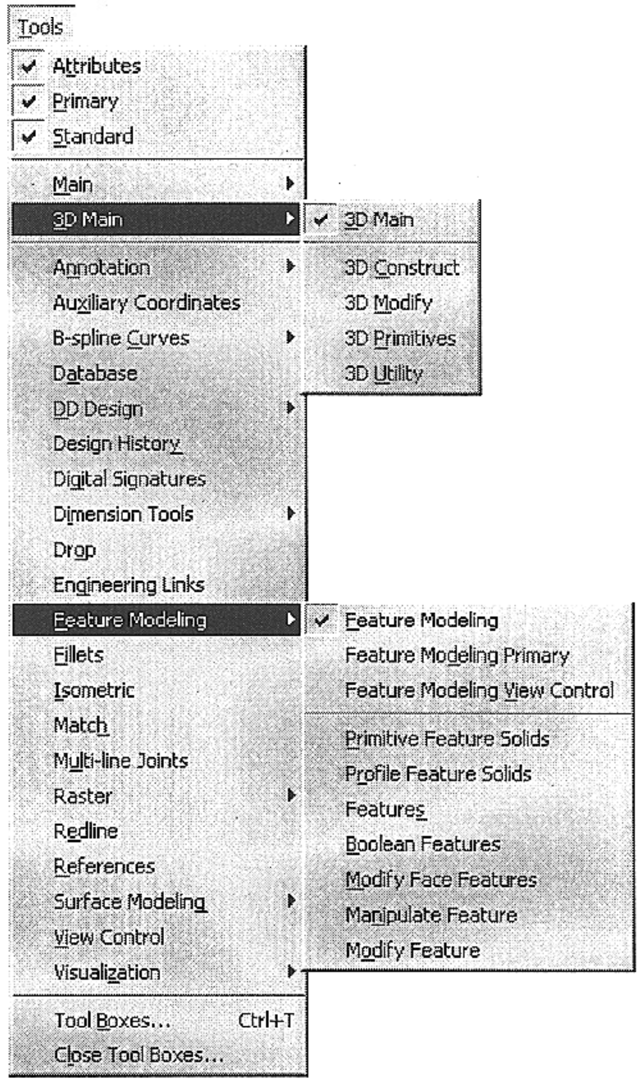


Figure 2 - 33. Modeling menu options in the Tools menu.

Both the *3d Main* and the *Feature Modeling* tools are available with the basic MicroStation foundation product. 3d surface and 3d solid primitives are placed in the model with the tools in the *3d Primitives* tool box. Parametric, feature-based solids are placed in the model with the tools in the *Primitive Feature Solids* tool box.

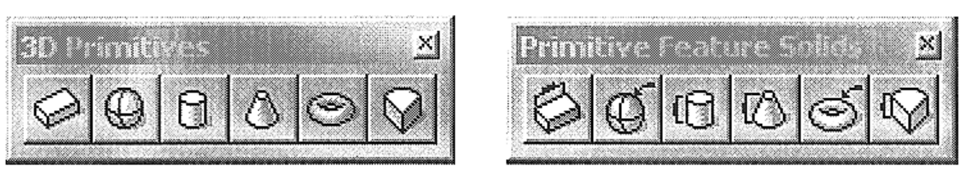


Figure 2 - 34. The 3d Primitives and Primitive Feature Solids tool boxes.

The 3d Primitives tool box is available in **Tools > 3d Main** and the Primitive Feature Solids is available in **Tools > Feature Modeling**. Many of the advanced 3d features are available only when using the feature modeling tools. Solids created with the 3d Primitives tools are referred to as *SmartSolids*. Those created with the Primitive Feature Solid tools are referred to as *Feature Solids.* MicroStation converts any *SmartSolid* to a *Feature Solid* when one of the feature modeling editing tools is applied to a *SmartSolid.* Generally, feature modeling tools create solids that are more flexible because they are easily modified using defining parameters for the solid.

8. Again select the Element Information tool.

   One at a time, pick the two objects at the bottom in *3dmodels.dgn.*

Notice the top line or header in the Element Information identifies the object. The header for the object at the bottom left indicates a *solid* and the header for the item at the bottom right indicates it is a *feature* or a *solid header* (XM) comprised of *slab.*

Feature-based solid modelers emphasize tools that focus on the features rather than each element and the individual steps need to make the feature. For example, a feature-based modeler may have a command to construct a hole with a countersink. To create a hole in a model with a countersink would take several steps if using standard modeling tools. A feature-based modeler incorporates all those steps into a command that focus on the feature rather than the steps.

9. Select the Hole Feature tool in the 3d Design tool frame.

   Hole Feature > Identify solid

   Change the following in the Hole Feature dialog:

Hole Type	**Countersink**
Direction	**Face Normal**
Diameter	**.5000**
Csink Diameter	**1.000**

   Again, pick any edge on the object at the lower left corner (yellow object) in *3dmodels.dgn.* Move the cursor to different surfaces on the model. Note the hole orients (direction) itself relative to the surface face.

   Position the hole and click accept (left mouse button) to place the hole. Once you have placed the hole, click reset (right mouse button) to cancel the command.

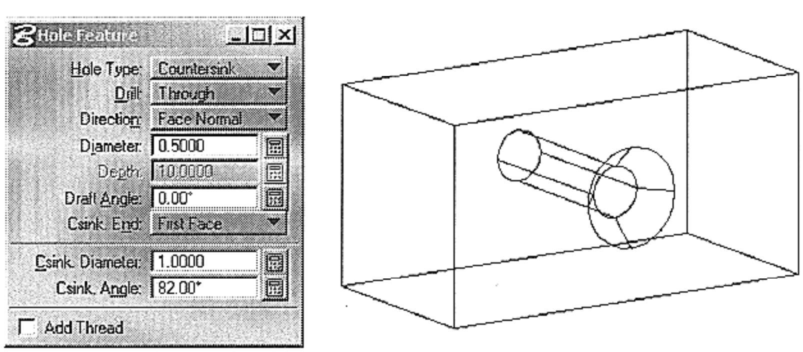


Figure 2 - 35. Hole feature.

10. Again select the Element Information tool.

    Pick the last object in which you placed the hole (lower left corner - yellow object). Notice that the object has been converted from a *SmartSolid* to a *Feature Solid.* The object contains a feature slab and a feature hole. The model was automatically converted when the feature hole was added.

Not all commands work on all 3d elements. For example, the hole command would not have worked on the object without it first being converted to a feature model. There are also tools to convert geometry from one form to another.

11. Select the 3d Convert tool. It is located in the Modify Surfaces tool box. To display the tool box, select **Tools > Surface Modeling > Modify Surfaces**. In the settings box that is displayed, set the *Convert to:* to *Solid.*

    Pick any edge on the object at the upper right corner (red object) in *3dmodels.dgn*. Pick the edge and then click the data button a second time away from any element to accept the selection. This converts the surface model to a solid model.

12. Again pick the **3d Convert** tool.

    Pick any edge on the object located in the lower left corner (yellow object) in *3dmodels.dgn*. Note that the command does not work, since the solid model was converted to a feature model when a hole was placed in the object.

Throughout the remainder of the book, the solid modeling operations are done using the feature modeling tools.

Parametric Modeling

Many CAD packages also incorporate the ability to alter various parameters as part of the functionality of the software. Parametrically-based software stores parameters that define the model as part of the design file. For example, a cube would be identified by height, width, and depth parameters. The cube could then be altered by changing any of the parameters. The software may also allow you to constrain one or more parameters. For example, a constraint could be established that required the height to always be twice the length. A change in length would automatically be reflected in the associated change in height. MicroStation feature modeling places solids that are parametric.

13. Select the Element Selection tool in the main tool frame.

    Alternately pick any edge on the objects at the upper right corner (red) and lower right corner (light blue) in *3models.dgn*.

Notice that the solid and the feature solid appear different when highlighted. The feature solid includes a triad showing the parameters that can be modified. In the XM edition, the handles of the selected objects appear different. In both cases, this provides a visual cue that while both are referred to as solids, they are different.

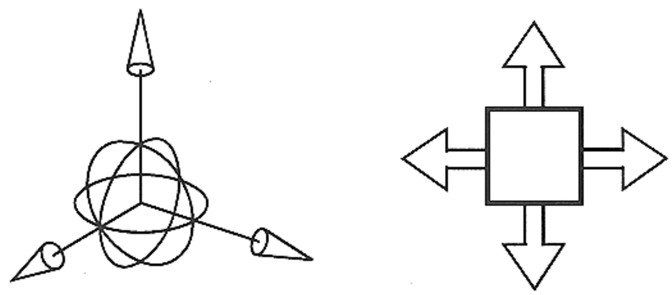


Figure 2 - 36. Triad or handle (XM) that appears when selecting a feature model.

14. Select the Modify Solid Or Feature tool in the 3d Feature Modeling tool frame.

Modify Parametric Solid or Feature > Identify feature

Pick any edge on the object at the lower right corner (light blue) in *3models.dgn*.

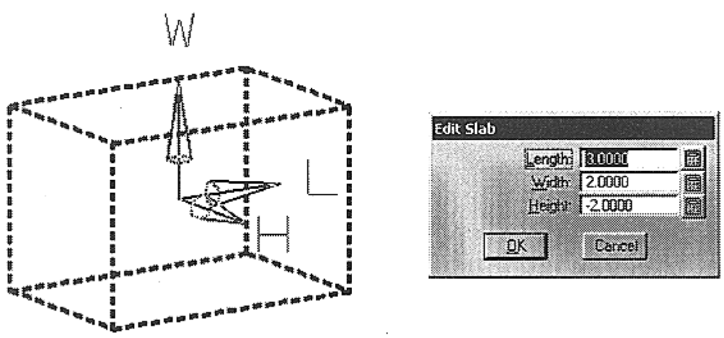


Figure 2 - 37. Edit the parameters of a feature solid.

As mentioned earlier, the slab selected is a parametric model created using the Slab Feature tool.

Change the *Length* to **4.00** and select **OK.**

The parameters (length, width, and height) stored with the model can each be modified. MicroStation even allows you to link the parameter to a variable (i.e. length = width/2). You can define your own variables or use one of the "local" variables maintained by the system. For this model, you could use one of the "local" variables by clicking on the Equation tool next to *Length* in the Edit Slab dialog.

15. After selecting the Modify Solid or Feature tool, select the Equation Button next to the *Length* parameter.

In the entry area for defining the parameter, you see displayed the variable "_SLAB1_Length. You are going to establish a relationship between two of the system or "local" variables.

Display the variables by enabling the *Local* check box. Enabling the *Local* setting shows the available system variables maintained by the software.

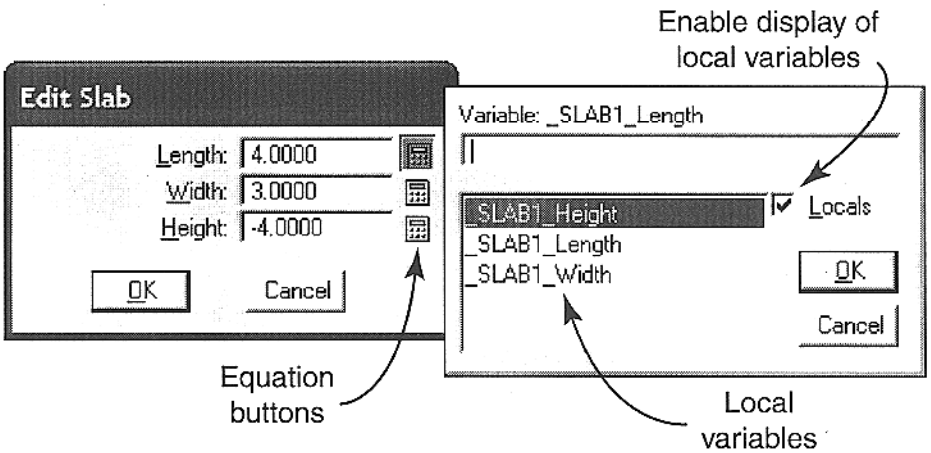


Figure 2 - 38. Enabling the display of local variables.

16. Double clik the " _SLAB1_Width" local variable. This should place the variable name in the entry area. Edit the area by dividing the local variable by two (/2). The completed entry would be "_SLAB1_Width/2". This indicates the length (__SLAB1_Length) of the slab is one-half the slab's width (_SLAB1_Width).

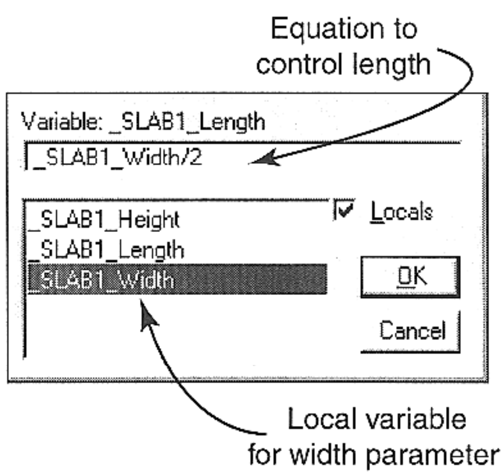


Figure 2 - 39. Using a local variable to control a parameter of a feature solid.

Select **OK.**

Note the results of the slab adjust based on the new parameter settings. In this case, the length of the slab remains half the width of the slab.

17. Close the *3dmodels.dgn* file. Do not save any of the changes to the file.

You can now reload the the *3dmodels.dgn* file.

1. Again, load the *3dmodels.dgn* file.
2. Use **File > Save As** command.

   Change the *Select Format to Save* to the **AutoCad Drawing files (.DWG)** format.

   Select **OK**.

3. Select the Element > Information command from the menu or the **Element Information** tool on the Primary tool box.

   Element information > Identify Element

   Pick any edge on the object in the lower right corner (light blue).

   Select the *Details* tab in the Element Information dialog. Notice that the feature solid in the Microstation DGN file format has been converted to a mesh in the AutoCad DWG file format.

4. Close the file.

Review Questions

1. What distinguishes two-dimensional and three-dimensional computer-aided design software?
2. What is meant by describing the MicroStation design file as a "container."
3. Describe the four primary options available for creating a new model in a MicroStation design file,\
4. What is the difference in creating a "design" model and a "design from seed" model?
5. What is the primary purpose of using the annotation scale?
6. Where and how do you enable the use of an annotation scale?
7. What is the difference in design model and a sheet model?
8. There are several difference approaches computer-aided design software can use to create three-dimensional models. Describe the three primary approaches.
9. What distinguishes the use of tools found in the 3d Primitives tool box from those found in the Primitive Feature Solids tool box?
10. What is a local variable?