Solid Modelling Techniques

This section serves as an introduction to 3D solid modelling systems. By first looking at 2D systems the reasons for a move to 3D systems becomes clear. The various options for representing 3D data are discussed and finally feature-based and parametric methods of 3D part design are looked at. Also discussed briefly are the two main formats for interaction between the many CAD systems that exist.

2D AND WIREFRAME

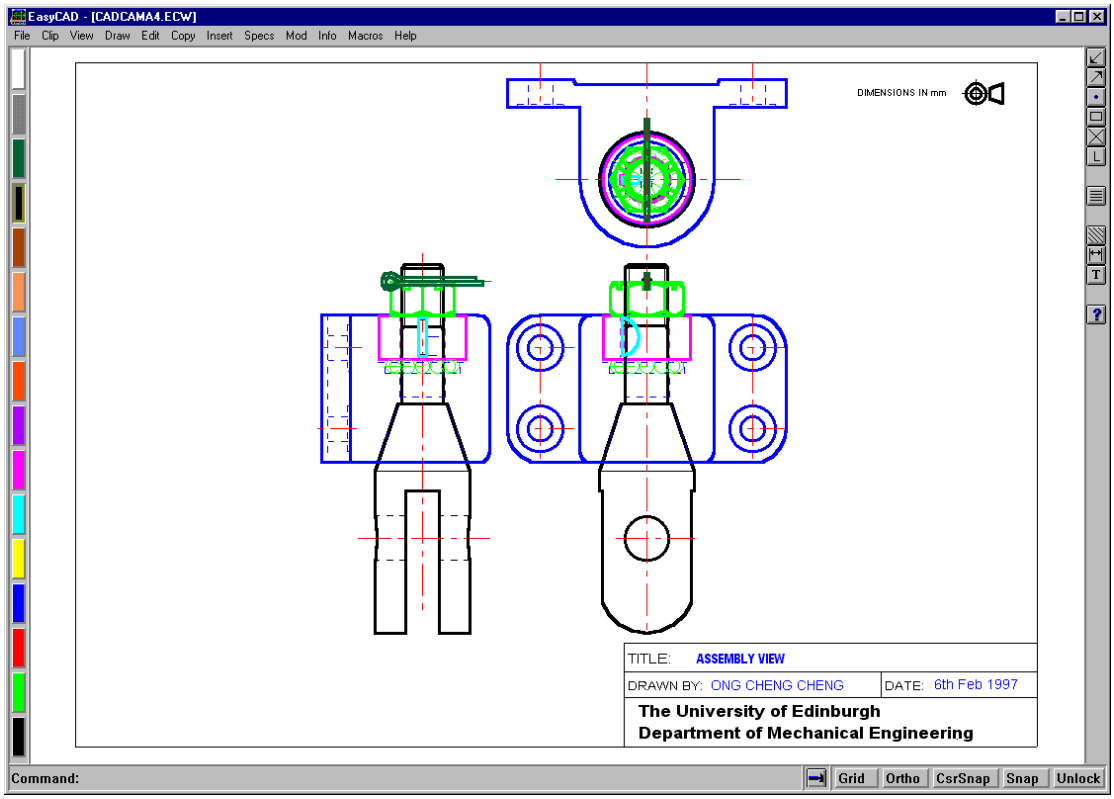

Figure 1: A 2D Drafting Package (EasyCAD)

For many years CAD has been limited to two dimensions, however there are many drawbacks to 2D systems compared to 3D. If we take the typical 2D case (EasyCAD from Evolution Computing is such a system) we are only able to design in terms of lines, arcs and circles, admittedly annotated with dimensions where appropriate (see figure 1). There is no concept of parts within the system. There is no knowledge of the three-dimensional geometry of the system, and indeed there is little knowledge of the two-dimensional geometry of the system in terms of where on the screen the solid material is and is not. The closest we can come to this is the ability to 'area fill' parts of the drawing at will.

These fallbacks can be clearly demonstrated by the ability of these drafting packages to produce drawings of impossible objects such as the ones shown in figure 2 below (prepared in EasyCAD).

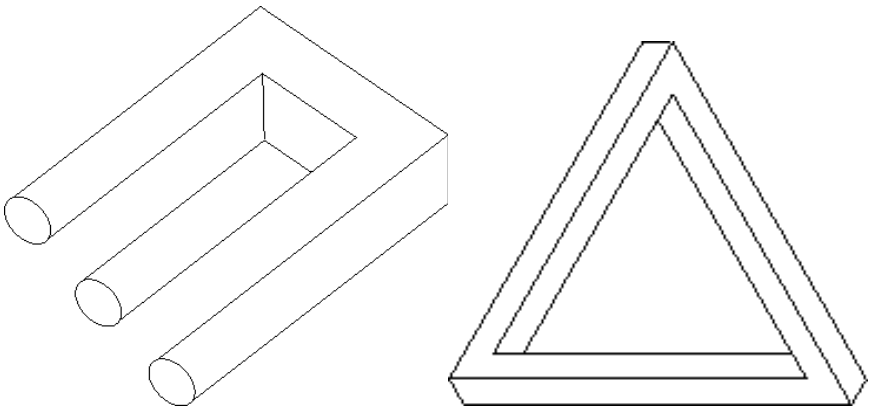

Figure 2: Impossible Objects

As we move to 3D systems, with the advent of increasingly powerful computers and software of ever growing sophistication, we discover similar limitations. The first three-dimensional systems were so-called wire frame modellers. In much the same way as a drawing in 2D is constructed out of line segments, in 3D, wire frame modellers allowed models to be constructed of 3D line segments. Though 3D models could now be produced, these models are inherently ambiguous. It is not possible to tell from a wire frame model where particular surfaces are and whether they exist. Perhaps the simplest example of this is that we cannot in a wire frame model distinguish between a solid cube and an empty box.

To resolve this problem, ways of representing solids were required. Solid modellers can be characterised as representing a point set in Euclidean three-space. Typically further restrictions are placed on the point set to allow us to represent the set to some degree of satisfaction within a finite computer system and in such a way to allow us to manipulate and interrogate this point set in ways deemed useful. Only with a solid model is it possible to check whether any point in space is inside or outside the solid. Because there is no ambiguity in using a solid model to represent a real world object the importance of solid modelling has been recognised. The currently successful methods tend to fall into three major categories:

- Spatial Occupancy methods (Decomposition Models)
- Constructive Solid Geometry
- Boundary Representation

SPATIAL OCCUPANCY METHODS

Spatial Occupancy Methods are often less able to model exact geometry, but can be more compact than other methods. This allows octrees in particular to be used in cases where space is at a premium. Spatial Occupancy methods come in 4 main flavours

- Exhaustive Enumeration
- Cellular Decomposition
- Adaptive Subdivision
- Depth maps

Exhaustive Enumeration

Exhaustive enumeration represents a solid as a collection of identical, smaller 'primitive' objects – voxels - that are assumed to be non-overlapping. This leads only to an approximation of real solids but any kind of solid maybe approximated. Co-ordinates describe the position of each voxel.

Exhaustive enumeration is little used, as it is extremely expensive in memory terms. Consider a component, roughly 200mm by 200mm by 200mm. If we were required to model this part to an accuracy of only 0.1mm then we need 2000 x 2000 x 2000 voxels or 8x10^9 voxels. If we use a single bit for each voxel then, at eight (8) bits to the byte we require approximately 1Gb of space.
However on the positive side the format is extremely simple and it is straightforward to create algorithms to modify the representation.

Cellular Decomposition

In cell decompositions, like exhaustive enumeration, a body is composed of a number of non-overlapping simple cells joined at common faces. However, here the 'primitive' cells can be different within the same solid. Finite element meshes used for stress analysis and computational fluid dynamics are perhaps the most common uses of this technique. The other techniques (CSG and B-rep) are often enhanced with cellular decomposition methods, and this will be discussed later.

ADAPTIVE SUBDIVISION

Adaptive subdivision replaces the regular subdivision of space, as found in the previous two methods, with a more efficient adaptive subdivision. A good example of this is Octree modelling.

Octrees are used for many tasks, often in association with CSG or B-rep modellers (typically for imaging and machining). They split part models up into eight boxes as shown in figure 3.

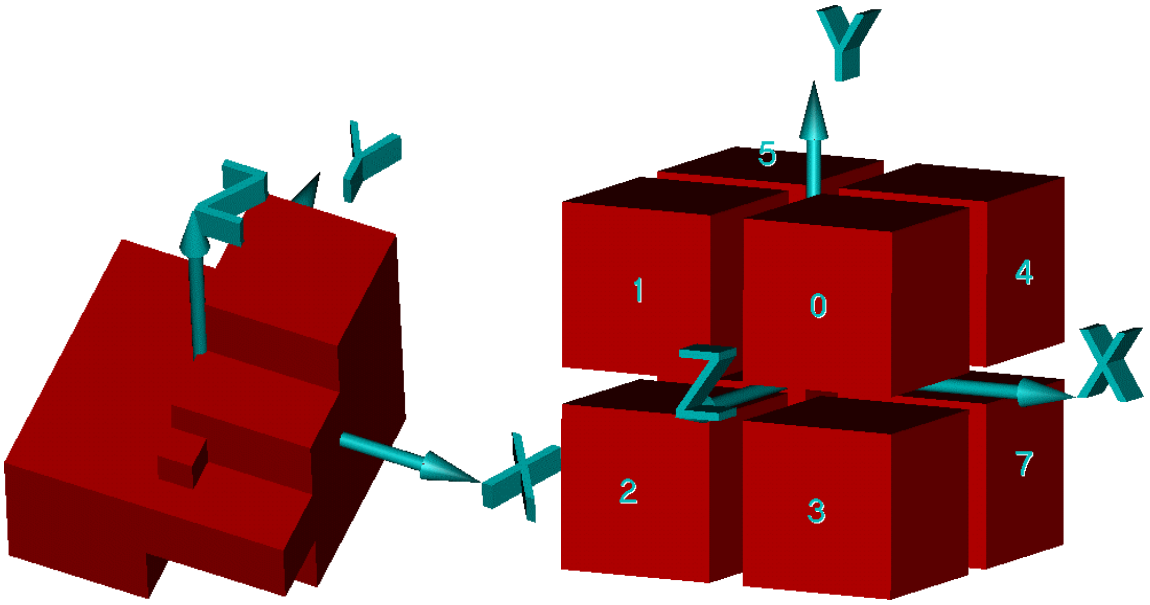

Figure 3: An Example Object and Octant Numbering

Subsequently the individual boxes that have some material in them but which are not 'full' are further decomposed into, again, eight sub-boxes and so on. This process is repeated until completion. A typical data structure is shown in figure 4 below.

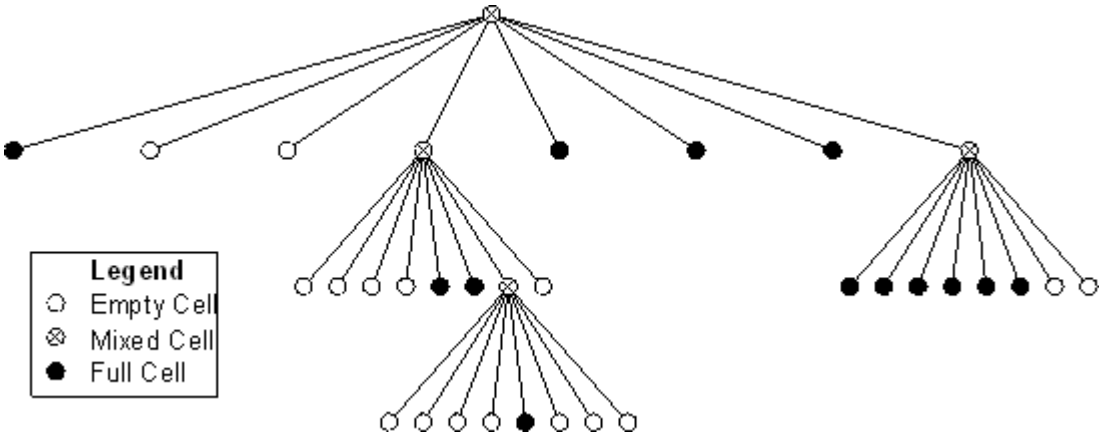

Figure 4: Octree Representation of Example Object

In this octant numbering system, octant 0 is the x,y,z>0 octant and the octants are numbered anticlockwise around the z axis, followed by those in the z<0 halfspace. The octree representation then given is equivalent to the object shown. The initial, top level octants are centred on the origin.

If we took our object 200mm x 200mm x 200mm, with 0.1mm cube missing in a corner (i.e. modelled to an accuracy of 1 part in 2000 or 0.1mm) then to model this we would need an octree up to 11 levels deep. However, we make great space saving over exhaustive enumeration with all the areas that are either entirely within the object or entirely without the object. For our model, each level only need contain eight bits of information thus the file size would only be 11 bytes.

One particular application of octrees is for pre-segmenting a B-rep model to allow rapid ray casting of the model by photorealistic renderers. An octree is made of the B-rep model to some suitable resolution and the nodes of the octree are tagged with those faces of the model that can be found in the appropriate region. This allows rapid ray-firing as precise intersections need only be found for those nodes the ray passes through containing interesting surfaces and not all the empty space.

Depth maps

A common, but limited Spatial Occupancy method that has the advantage of being comparatively compact is the depth-map. Depth maps can only represent components that are single-sided, in machining terms this means that they must be machined only from a single direction. Other single sided objects include terrain maps, making depth maps suitable for Geographic Information Systems (GIS). The output from 3D digitisers, at least in an intermediate form, tends to be as a set of depth maps each from a different viewpoint. A number of 3D depth maps can be knitted together to form a solid 3D object (though not without difficulty and with some limitations). NC simulation programs dealing with purely one-sided 3-axis NC milling can also use a depth map as a suitable representation provided each pixel is small enough to provide satisfactory resolution. Representing a component of dimensions described earlier, but using a single sided depth map takes only 2000 x 1000 pixels. If we model each depth using an 8bit fixed point number, giving us *256* possible depths we require only 2Mb of storage, still large, but a fraction of our previous space. The NC simulation program need only handle 2-dimensional geometrical problems, setting the depth of all pixels a cutter passes over to the cutter height. Simple rendering of depth maps is comparatively easy.

The following is a reconstructed golf club head in ACIS (a solid modeller similar to ParaSolid) from a digitised depth map taken from the laser digitiser of Machine Vision Group of the Department of Artificial Intelligence at Edinburgh University.
Each pixel in the depth map was about 2mm square, and due to specular reflections, some of the data is clearly spurious. The ACIS reconstruction was written quickly by reconstructing a series of slices from the data and uniting them together. No attempt has been made to smooth the resulting body in any way. The resulting body was then rendered in the LightWorks package to produce the image seen. The original head is courtesy of Ben Sayers Ltd, North Berwick.

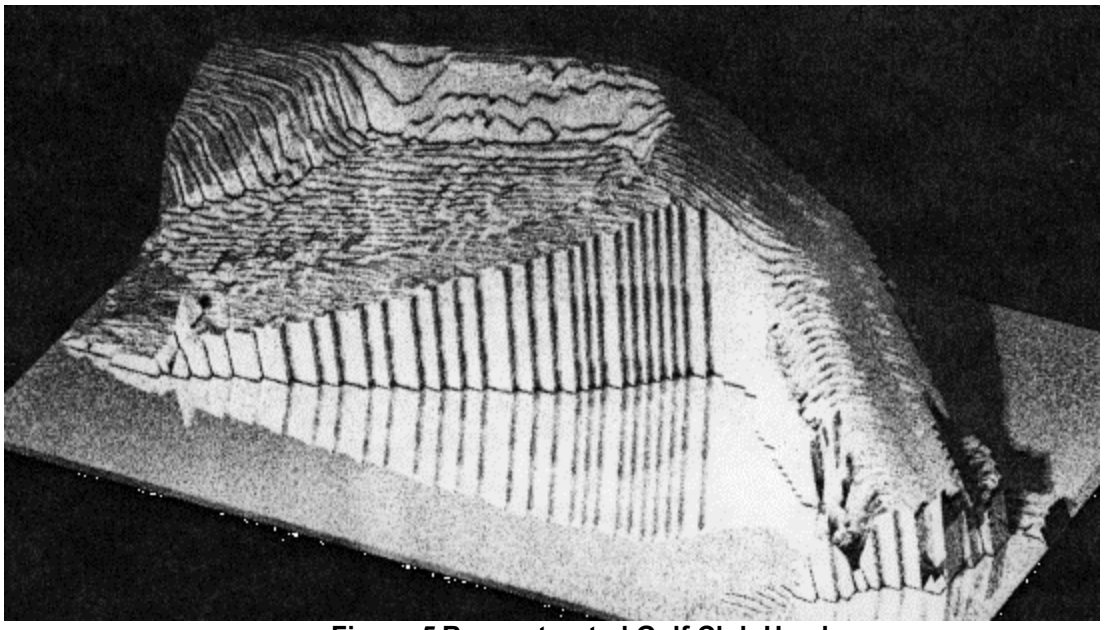

Figure *5* Reconstructed Golf Club Head

CONSTRUCTIVE SOLID GEOMETRY

Constructive Solid Geometry modellers (CSG) also known as set-theoretic modellers describe solids as combinations of primitive sets modified by Boolean operators such as union, subtraction and intersection.

Half-Space

Each of the primitives are described by a combination of so-called half spaces, these are defined by simple functions that separate the world (three-dimensional Euclidean space E3) into in and out, those for which f(P)>0 and those for which f(P)<0. Half spaces can be defined in terms of any real-valued analytic function f(P), P=(x, y, z). E.g.:

ax+by+cz+d>0 planar half space consisting of all the positive points on the plane
$x^2+y^2-r^2<0$ cylindrical half space consisting of all points within the infinite cylinder

For a plane passing through a point P_0=(x,y,z), and with an outward surface normal V=(x,y,z) (figure 6), we can determine whether any point P is inside or outside the half-space by evaluating the function

$(P-P_O)xV$

and noting that if:

(P-PO)xV < 0; point is inside the halfspace
(P-PO)xV =0; point is on the planar surface
(P-PO)xV < 0; point is outside the halfspace

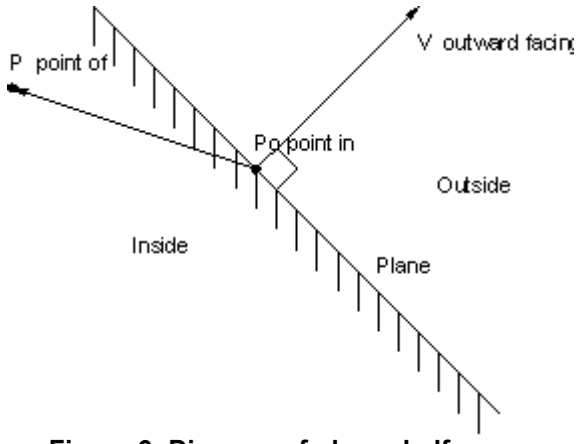

Figure 6: Diagram of planar half space

Similarly, an open-ended (infinitely long) cylinder whose axis passes through a point P_0 and is aligned with a vector V(x,y,z) and with radius r, can be defined with the following equation.

(P-PO) V - r <0; inside the cylindrical surface

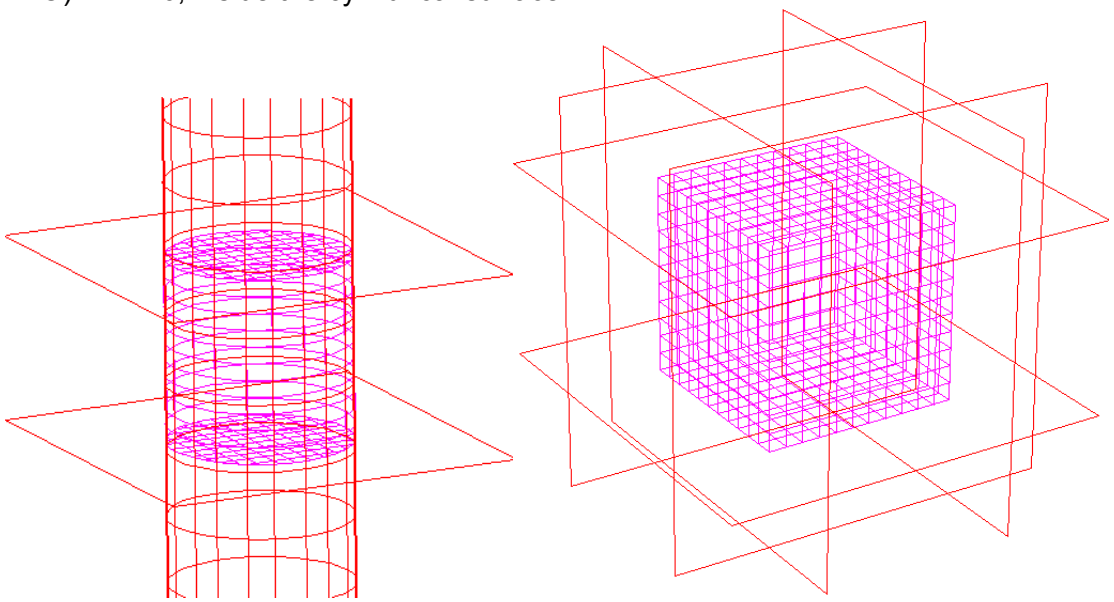

Figure 7: A box and a cylinder in terms of their constituent half space.

We can combine a number of half spaces using set boolean operators such as intersection ($\cap$), union ($\cup$) or subtraction (-) operators to produce more complex bodies. For instance a cylinder, C, of height h, and radius r, centred on the origin (as in Figure 7) can be defined in terms of three half spaces in the following way:

H1: $x^2 + y^2 - r^2 < 0$
H2: $z > 0$
H3: $z - h < 0$
$C = H1 \cap H2 \cap H3$

Though CSG modellers may allow use of the half spaces, most modellers wrap these up into suitable, more user-friendly primitives, such as blocks, cylinders, spheres, cones

and tori. Although a user may be aware that they are developing a CSG tree to describe a component, they may remain blissfully unaware of the primitive half spaces at the bottom of the tree and so a simple component may be modelled as in the following diagram.

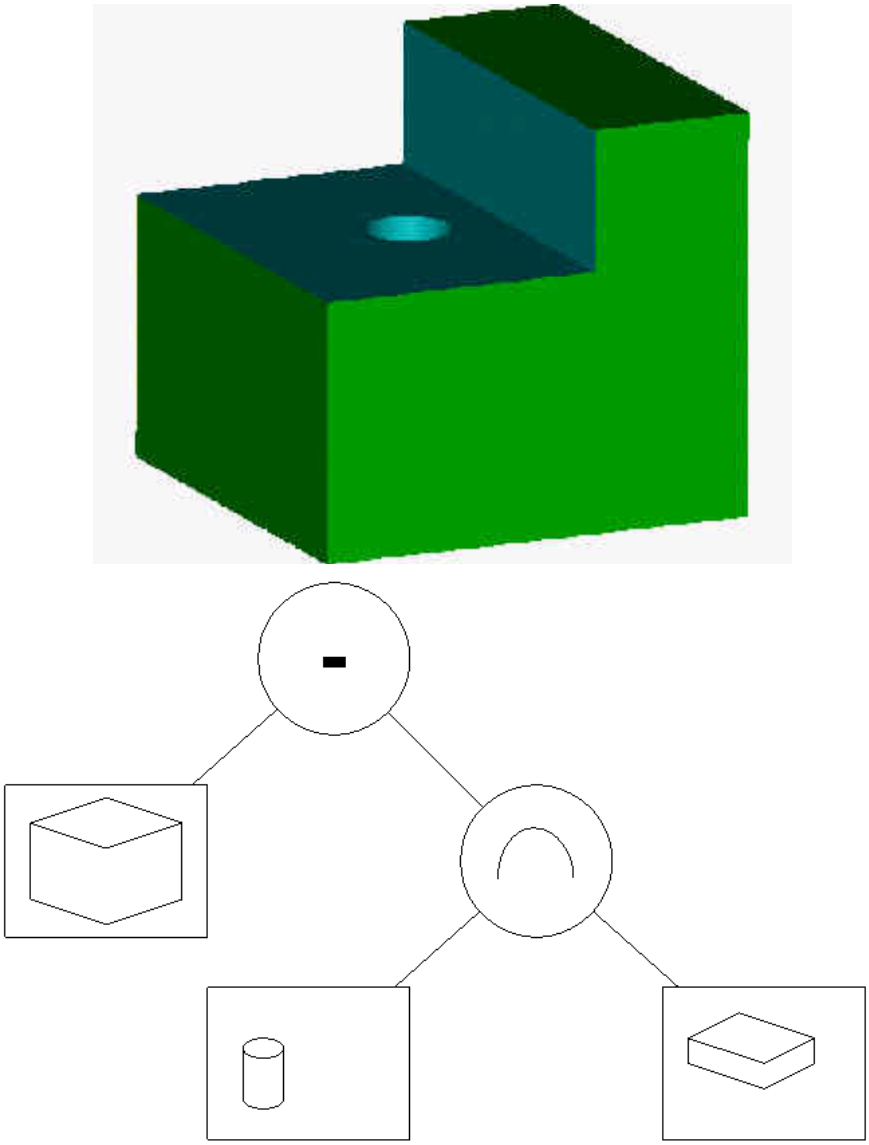

Figure 8: A Simple CSG Tree and Model

So what we end up with is a very user-friendly interface that is easily able to show us the creation steps of the model. However, on the down side it is a slow procedure to produce a rendered image of a model from a CSG tree. This is because most rendering pipelines work on B-reps and the CSG representation has to be converted to this form before rendering. Hence some solid modellers use a B-rep but the user interface is based on the CSG representation.

BOUNDARY REPRESENTATION (B-REP)

B-rep modellers represent a solid as a volume contained in a set of faces together with topological information that defines the relationships between the faces. The storing of the topological data allows solids to be represented as closed spaces in Euclidean space.

The boundary of a solid separates points inside from points outside the solid. B-rep models can represent a wide class of objects but data structure is complex, and it requires a large memory space. The hierarchy of topological elements in ACIS, a commercial B-rep modeller is shown in Figure 9.

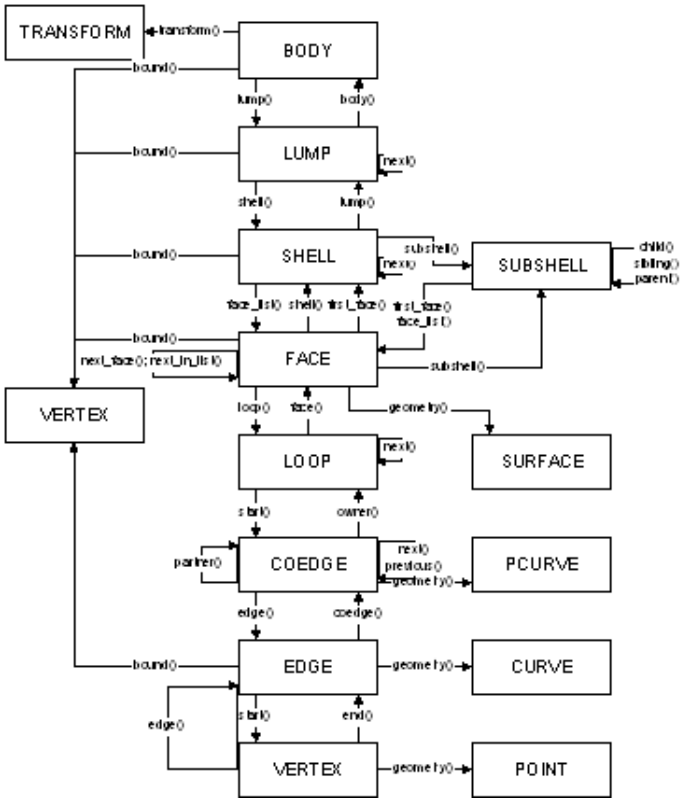

Figure 9: A B-rep hierarchy (ACIS)

A solid object is represented by a collection of faces. Normally a face is a bounded region of a planar, quadratic, toroidal, or sculptured surface. The bounded region of the surface that forms the face is represented by a closed curve that lies on the surface. It is possible for these faces to have several bounding curves to represent holes in a solid. The bounding curves of faces are represented by edges, which are in turn represented by two vertices that represent the portion of the curve that forms the edge. A B-Rep model has to fulfil the following conditions: The set of faces forms a complete skin of the solid with no missing parts, and faces do not intersect each other except at common vertices or edges. Furthermore, the boundaries of faces do not intersect themselves. These conditions disallow self-intersecting and open objects.

Boundary representation can be divided in three classes: facetted, elementary, and advanced B-Rep. In facetted B-Rep, a solid is bounded by planar surfaces. Only points, planes and planar polygons are necessary and are implicitly represented by their vertex points. The surfaces included in elementary B-Rep are planar, quadric, and toroidal surfaces. The bounding curves of the faces are lines or conics. In advanced B-Rep, the surfaces also include B-Spline surfaces in addition to elementary B-Rep. The bounding curves are B-Spline curves.

Boundary representation modellers are currently in the majority in industry. They are perceived as having a number of minor advantages over their CSG compatriots, namely:

- A B-rep model is canonical, there is a unique representation of any solid object, whereas the same solid can (in general) be modelled in an infinity of ways in a CSG modeller. This makes similarity checking very expensive (if not impossible) in a CSG modeller, compared with 'just expensive' in a B-rep modeller.

- The explicit face and edge list contained in B-rep modellers make visualisation routines cheaper for B-rep modellers than for CSG.

These differences are in fact less than might at first appear, as CSG modellers tend to maintain a B-rep model of the object for visualisation, and B-rep modellers tend to allow generation of solids using CSG-like primitives and Boolean operators. The difference is more in the primary representation and the philosophy of the research/developer.

To compute Boolean operations for B-Rep models is expensive and numerical problems may occur. Figure 10 shows a body generated in a demonstration of an early version of FODDS that crashed the system. It is an example of a non-manifold body. Non-manifold bodies are objects where more than two surfaces meet at a single edge. These have been a problem for B-rep modellers for a number of years, as it is quite easy to try to create them. It has been difficult until recent years to represent them, and in particular to perform subsequent operations on the resulting solid model. ACIS 1.3 would allow the above model to be generated, but subsequent Booleans on any portion of the model would cause the modeller to crash. A physical interpretation of non-manifold bodies is difficult. It is not clear from the model whether the portion of space along the rear edge is a very thin wall, or a very thin crack.

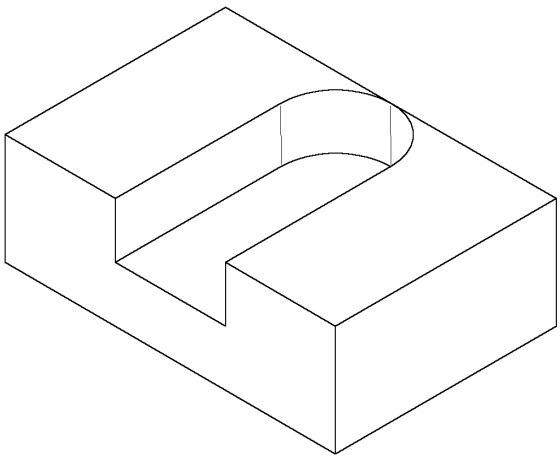

Figure 10 A non-manifold body

CURRENT TECHNOLOGIES IN MIDRANGE CAD SYSTEMS

As has already been discussed modern solid modelling CAD systems, whether based upon CSG or B-rep representations, tend to function as a combination of the two systems. The major core solid modeller in use today, amongst mid-range packages, is Parasolid, which uses Boundary Representation. Parasolid is currently the core modeller within Solid Edge, Solid Works, Pro/Desktop, SolidMAN, ICAD and many others.

As Boundary Representation has become the standard base modeller the key positive functional aspects of CSG have been integrated such as Boolean operations and tree structures. With a stable representation system in place recent developments have been focused upon stretching the practicality of design construction, editing and manufacturing. The key advances in these areas are *parametric* and *feature-based* design. These are aspects that are found in most of today's mid-range modelling packages such as Solid Edge and Solid Works.

Parametric means that the physical shape of the part or assembly is driven by the values assigned to the attributes (primarily dimensions) of its features. The user may define or modify a feature's dimensions or other attributes at any time. Any changes will automatically propagate through the model. By using numeric relations, the software allows the user to change any dimension and the whole model will change to accommodate it, keeping its relative geometry.

Feature-based means that the user creates their parts and assemblies by defining features like extrusions, sweeps, cuts, holes, slots, Therefore, the designer can think of and create a computer model at a very high level, and let the software deal with the low-level geometric detail. Features are specified by setting values and attributes of elements such as reference planes, direction of creation, shape, dimensions, and others. Feature-based methods have emerged over the last twenty years in response to industry's requirements for an integrated solution to design and production, in turn required in order to reduce product lead time. Features enable this reduction in lead-time by supplying the designer with a set of tools at a higher level of abstraction than those of a typical computer-aided design system.

Neutral File Formats

As demonstrated there are many different ways of representing solids and thus neutral files and neutral file interfaces are needed in order to exchange product data between the various CAD systems. Many formats exist but here we will focus on the two most common, IGES and STEP.

IGES

IGES (Initial Graphics Exchange Specification) was the first specification for CAD data exchange published in 1980 as a NBS (National Bureau of Standards) report in USA. IGES version 1.0 was accepted and released in 1981 as an ANSI standard. All major CAD vendors support IGES and it is currently by far the most widespread standard for CAD data exchange.

IGES was originally developed for the exchange of drafting data like 2D/3D wireframe models, text, dimensioning data, and a limited class of surfaces. Due to criticism and bad experience with the data transfer using IGES, the standard has been gradually extended and developed concerning supported entities, syntax, clarity, and consistency. The current version, IGES 5.2, provides the following capabilities:

- *Geometry:* 2D/3D wireframes, 2D/3D curves and surfaces, CSG (since version 4.0 in 1988), B-Rep (since version 5.1 in 1991);
- *Presentation:* Drafting entities for technical drawings;
- *Application dependent elements:* Piping and electronic schematics, AEC elements;
- *Finite Element Modelling:* Elements for FEM systems.

IGES specification defines the format of the file, language format, and the product definition data in these formats. The product definition includes geometric, topological, and non-geometric data. The geometry part defines the geometric entities to be used to define the geometry. The topology part defines the entities to describe the relationships between the geometric entities. The geometric shape of a product is described using these two parts (i.e. geometry and topology). The non-geometric part can be divided into annotation, definition, and organisation. The annotation category consists of dimensions, drafting notations, text, etc. The definition category allows to define specific properties of individual or collections of entities. The organisation category defines groupings of geometric, annotation, or property elements.

The size of IGES files and consequently the processing time are practical problems. IGES files are composed of fixed format records and each entity has to have records in both the directory entry section and the parameter data section with bi-directional pointers. This causes also errors in pre- and post-processor implementations.

STEP

STEP (Standard for the Exchange of Product model data) is a new International Standard (ISO 10303) for representing and exchanging product model information. It

includes an object-flavoured data specification language, EXPRESS, to describe the representation of the data. STEP defines also implementation methods, for instance, a physical transfer file, and offers different resources, e.g. geometric and topological representation. Importantly STEP supports feature data exchange.

The development of STEP started in 1984 as a world wide collaboration. The goal was to define a standard to cover all aspects of a product (i.e. geometry, topology, tolerances, materials, etc.), during its life time. The main parts of STEP are already international standards, while many parts are still under development. The development is performed under the control of the International Standards organisation (ISO).

The use of STEP is growing all the time. The majority of CAD system vendors have implemented or are implementing STEP pre- and post-processors for their CAD systems. STEP is an evolving standard which will cover the whole product life cycle in terms of data sharing, storage and exchange. It is the most important and largest effort ever established in engineering domain and will replace current CAD exchange standards.

The STEP standard is divided into parts:

- *Overview and Fundamental Principles:* defines the principles of STEP;
- *Description Methods:* the EXPRESS data modelling language which represents product information;
- *Implementation Methods:* contain the definitions of the physical representation of product information;
- *Conformance Testing:* a conformance testing methodology and framework;
- *Integrated Resources:* contain the definitions of representation of product information which are common for different application protocols;
- *Application Protocols:* contain the definitions of representation of product information which are specific to a particular application area;
- *Abstract Test Suites:* abstract test cases for an application protocol to support the conformance requirements.