CHAPTER 2

BIM Tools and Parametric Modeling

2.0 EXECUTIVE SUMMARY

This chapter provides an overview of the primary technology that distinguishes BIM design applications from other CAD systems. Object-based parametric modeling was originally developed in the 1980s. It does not represent objects with fixed geometry and properties. Rather, it represents objects by parameters and rules that determine the geometry as well as some non-geometric properties and features. The parameters and rules allow the objects to automatically update according to user control or changing contexts. In other industries, companies use parametric modeling to develop their own object representations and to reflect corporate knowledge and best practices. In architecture, BIM software companies have pre-defined a set of base building object families for users, which may be extended, modified, or added to. An object family allows for the creation of any number of object instances, with forms that are dependent on parameters and relationships with other objects. Companies should have the capability of developing user-defined parametric objects and corporate object libraries for customized quality control and to establish their own best practices. Custom parametric objects allow for the modeling of complex geometries, which were previously not possible or simply impractical. Object attributes are needed to interface with analyses, cost estimations, and other applications, but these attributes must first be defined by the firm or user.

Current BIM tools vary in many ways: in the sophistication of their pre-defined base objects; in the ease with which users can define new object families;
in the methods of updating objects; in ease of use; in the types of surfaces that can be used; in the capabilities for drawing generation; in their ability to handle large numbers of objects and their interfaces with other software.

Most architectural BIM design tools let users mix 3D modeled objects with 2D drawn sections, allowing users to determine the level of 3D detailing while still being able to produce complete drawings. Objects drawn in 2D are not automatically included in bills of material, analyses, and other BIM-enabled applications. Fabrication-level BIM tools, alternatively, typically represent every object fully in 3D. The level of 3D modeling is a major variable within different BIM practices.

This chapter provides an overall review of the major BIM model generation tools and some functional distinctions that can be used for assessing and selecting among them.

2.1 HISTORY OF BUILDING MODELING TECHNOLOGY

2.1.1 Early 3D Modeling of Buildings
The modeling of 3D geometry was a broad research goal that had many potential uses including movies, design, and eventually games. The ability to represent a fixed set of polyhedral forms—shapes defined by a volume enclosing a set of surfaces—for viewing purposes was developed in the late 1960s and later led to the first computer-graphics film, *Tron* (in 1987). These early polyhedral forms could be used for composing an image but not for designing more complex shapes. In 1975, the easy creation and editing of arbitrary 3D solid shapes was developed separately by three groups, Ian Braid at Cambridge University, Bruce Baumgart at Stanford, and Ari Requicha and Herb Voelcker at the University of Rochester (Eastman1999; Chapter 2). Known as *solid modeling*, these efforts produced the first generation of practical 3D modeling design tools.

Two forms of solid modeling were developed and competed for supremacy. The boundary representation approach (B-rep) defined shapes using operations of union, intersection, and subtraction—called Boolean operations—on multiple polyhedral shapes and also utilized refining operations, such as chamfering, slicing, or moving a hole within a single shape. A small set of such operators is shown in Figure 2-1. The sophisticated editing systems developed from combining these primitive shapes and the Boolean operators allowed generation of a set of surfaces that together were guaranteed to enclose a volume.

In contrast, Constructive Solid Geometry (CSG) represented a shape as a tree of operations and initially relied on diverse methods for assessing the final
shape. An example is shown in Figure 2-2. Later, these two methods merged, allowing for editing within the CSG tree (sometimes called the \textit{unevaluated shape}) and also changing the shape through the use of general-purpose B-rep (called the \textit{evaluated shape}). Objects could be edited and regenerated on demand. Figure 2-2 depicts a CSG tree, the unevaluated shapes it references, and the resulting evaluated shape. The result is the simplest of building shapes: a single shape hollowed with a single floor space with a gable roof and door opening. Notice that all locations and shapes can be edited via the shape parameters in the CSG tree, however, shape edits are limited to Boolean or other editing operations shown in Figure 2-1. First generation tools supported 3D faceted and cylindrical object modeling with associated attributes, which allowed objects to be composed into engineering assemblies, such as engines, process plants, or buildings (Eastman 1975; Requicha 1980). This merged approach to modeling was an important precursor to modern parametric modeling.

Building modeling based on 3D solid modeling was first developed in the late 1970s and early 1980s. CAD systems, such as RUCAPS (which evolved into Sonata), TriCad, Calma, GDS (Day 2002), and university research-based systems at Carnegie-Mellon University and the University of Michigan developed their basic capabilities. (For one detailed history of the development of CAD technology see http://mbinfo.mbdesign.net/CAD-History.htm.) This work was carried out in parallel with efforts in mechanical, aerospace, building and electrical product design, where early concepts of product modeling and integrated analysis and simulation were developed. Early conferences in computer-aided design were integrated across all areas of engineering and design, resulting in high levels of synergy. For example: Proceedings 7th–18th Annual Design Automation Conference (ACM 1969–1982); Conferences on Engineering and Scientific Data Management (NASA 1978–1980); Proceedings of CAD76, CAD78, CAD80 (CAD 1976,1978,1980).

Solid modeling CAD systems were functionally powerful but often overwhelmed the available computing power. Some aspects of production, such as
drawing and report generation, were not well developed. Also, designing 3D objects was too conceptually foreign for most designers, who were more comfortable working in 2D. The systems were also expensive, costing upward of $35,000 per seat. The manufacturing and aerospace industries saw the potential benefits in terms of integrated analysis capabilities, reduction of errors, and the move toward factory automation. They worked with CAD companies to resolve the technology’s early shortcomings. Most of the building industry did not recognize these benefits. Instead, they adopted architectural drawing editors, such as AutoCAD® and Microstation® that augmented the then-current
2.1.2 Object-Based Parametric Modeling

The current generation of BIM architectural design tools, including Autodesk Revit® Architecture and Structure, Bentley Architecture and its associated set of products, the Graphisoft ArchiCAD® family, and Gehry Technology’s Digital Project™ as well as fabrication-level BIM tools, such as Tekla Structures, SDS/2, and Structureworks all grew out of the object-based parametric modeling capabilities developed for mechanical systems design. These concepts emerged as an extension of CSG and B-rep technologies, a mixture of university research and intense industrial development, particularly by Parametric Technologies Corporation® (PTC) in the 1980s. The basic idea is that shape instances and other properties can be defined and controlled according to a hierarchy of parameters at the assembly and sub-assembly levels, as well as at an individual object level. Some of the parameters depend on user-defined values. Others depend on fixed values, and still others are taken from or relative to other shapes. The shapes can be 2D or 3D.

In parametric design, instead of designing an instance of a building element like a wall or door, a designer defines a model family or element class, which is a set of relations and rules to control the parameters by which element instances can be generated but will each vary according to their context. Objects are defined using parameters involving distances, angles, and rules like attached to, parallel to, and distance from. These relations allow each instance of an element class to vary according to its own parameter settings and contextual relations. Alternatively, the rules can be defined as requirements that the design must satisfy, allowing the designer to make changes while the rules check and update details to keep the design element legal and warning the user if these definitions are not met. Object-based parametric modeling supports both interpretations.

While in traditional 3D CAD every aspect of an element’s geometry must be edited manually by users, the shape and assembly geometry in a parametric modeler automatically adjusts to changes in context and to high-level user controls.

One way of understanding how parametric modeling works is by examining the structure of a wall family, including its shape attributes and relations, as shown in Figure 2-3. We call it a wall family, because it is capable of generating many instances of its type in different locations and with varied parameters. While a wall family may focus on straight and vertical walls, varied geometric capabilities are sometimes desired, including those with
curved and non-vertical surfaces. A wall shape is a volume bounded by multiple connected faces, some defined by context and others defined by explicit values. For most walls, the thickness is defined explicitly as two offsets from the wall control line, based on a nominal thickness or the type of construction. Walls with tapered or varying thicknesses have multiple offsets or possibly a vertical profile. The wall’s elevation shape is defined by one or more base floor planes; its top face may be an explicit height or possibly defined by a set of adjacent planes (as shown here). The wall ends are defined by the wall’s intersection, having either a fixed endpoint (freestanding) or associations with other walls. The control line of the wall (here shown along the bottom) has a start and end point, so the wall does too. A wall is associated with all the object instances that bound it and the multiple spaces it separates.

Door or window openings have placement points defined by a length along the wall from one of its endpoints to a side or to the center of the opening with its required parameters. These openings are located in the coordinate system of the wall, so they move as a unit. A wall will adjust its ends by moving, growing, or shrinking as the floor-plan layout changes, with windows and doors also moving and updating. Any time one or more surfaces of the bounding wall changes, the wall automatically updates to retain the intent of its original layout.

A well-crafted definition of a parametric wall must address a range of special conditions. These include:

- The door and window locations must check that they lie completely within the wall and do not overlap each other or extend beyond the wall boundaries. They typically display a warning if these conditions fail.
A wall control line may be straight or curved, allowing the wall to take varied shapes.

A wall may intersect floor, ceiling, or side walls, any of which are made up of multiple surfaces and result in a more complex wall shape.

Walls may have tapered sections, if they are made of concrete or other malleable materials.

Walls made up of mixed types of construction and finishes may change within segments of a wall.

As these conditions suggest, significant care must be taken to define even a generic wall. It is common for a parametric building class to have over one hundred low-level rules for its definition. These rules also explain why users may encounter problems with unusual wall layouts—because they are not covered by the built-in rules—and how easy it is to define wall definitions that may be inadvertently limited.

For example, take the clerestory wall and the windows set within it shown in Figure 2-4. In this case, the wall must be placed on a non-horizontal floor plane. Also, the walls that trim the clerestory wall ends are not on the same base-plane as the wall being trimmed. Early BIM modeling tools could not deal with this combination of conditions.

In Figure 2-5, we present a simple sequence of editing operations for the schematic design of a small theater. The upper left view in Figure 2-5 shows the theater with two side walls angling inward toward the stage and a wall separating the back of the theater from the lobby. In terms of boundary associations, the theater side walls are initially attached to the ceiling and floor, and their ends are attached to the rear of the lobby and front wall of the stage. The sloping theater floor is attached to the side walls of the building.

In Figure 2-5, upper right, the lobby side walls are detached from the rear wall and moved part-way open, allowing the lobby to flow around their edges.
Notice that the ceiling arc is automatically trimmed. In the lower left figure, the side walls of the theater are re-angled, with the theater rear wall automatically trimmed to abut the walls of the theater. In lower right figure, the back wall is moved forward, making the theater shallower. The back wall automatically shortens to maintain its abutment to the side walls, and the theater and lobby floor automatically adjust to remain abutted to the back wall. The significant point is that the adjustments of the side walls to the roof and sloped theater floor are completely automatic. Once the initial spatial configuration is defined, it is possible to make quick edits and updates. Notice that these parametric modeling capabilities go far beyond those offered in previous CSG-based CAD systems. They support automatic updating of a layout and the preservation of relations set by the designer. These tools can be extremely productive.

2.1.3 Parametric Modeling of Buildings

In manufacturing, parametric modeling has been used by companies to embed design, engineering, and manufacturing rules within the parametric models of their products. For example, when Boeing undertook the design of the 777, they defined the rules by which their airplane interiors were to be defined, for looks, fabrication, and assembly. They fine-tuned the outside shape for
aerodynamic performance through many hundreds of airflow simulations—
called Computational Fluid Dynamics (CFD)—linked to allow for many alter-
native shapes and parametric adjustments. They pre-assembled the airplane
virtually in order to eliminate more than 6,000 change requests and to achieve a
90% reduction in spatial re-work. It is estimated that Boeing invested more
than $1 billion dollars to purchase and set up their parametric modeling
system for the 777 family of planes. A good overview of the Boeing effort, its strengths
and shortcomings, is available (CalTech 1997).

In a similar way, the John Deere company, working with LMS of Belgium,
defined how they wanted their tractors to be constructed. Various models were
developed based on John Deere’s design-for-manufacturing (DfM) rules. (www.
lmsintl.com/virtuallab). Using parametric modeling, companies usually define
how their object families are to be designed and structured, how they can be
varied parametrically and related into assemblies based on function, and other
production criteria. In these cases, the companies are embedding corporate
knowledge based on past efforts on design, production, assembly, and mainte-
nance concerning what works and what does not. This is especially worthwhile
when a company produces many variations of a product. This is the standard
practice in large aerospace, manufacturing, and electronics companies.

Conceptually, building information modeling (BIM) tools are object-based
parametric models with a predefined set of object families, each having behav-
iors programmed within them, as outlined above. A fairly complete listing of
the pre-defined object families provided by major BIM architectural design
tools is given in Table 2-1 (as of early 2007). These are the sets of pre-defined
object families that can be readily applied to building designs in each system.

A building is an assembly object defined within a BIM system. A building
model configuration is defined by the user as a dimensionally-controlled para-
metric structure, using grids, floor levels, and other global reference planes.
Alternatively, these can simply be floor planes, wall centerlines, or a combina-
tion of them. Together with their embedded object instances and parametric
settings, the model configuration defines an instance of the building.

In addition to vendor-provided object families, a number of Web sites
make additional object families available for downloading and use. These are
the modern equivalent of drafting block libraries that were available for early
2D drafting systems – but of course they are much more useful and powerful.
Most of these are generic objects, but a growing capability is the provision of
models of specific products. These are discussed in Section 5.4.2 and some
of the sites are listed.

There are many detailed differences between the specially-developed para-
metric modeling tools used in BIM and those used in other industries. Buildings
are composed of a very large number of simple parts. Its regeneration dependencies are more predictable than for general mechanical design systems; however, the amount of information in even a medium-sized building at construction-level detail can cause performance problems in even the most high-end personal computers. Another difference is that there is a broad set of standard practices and codes that can be readily adapted and embedded to define object behaviors. These differences have resulted in only a few general-purpose parametric modeling tools being adapted and used for Building Information Modeling.

### Table 2-1 The built-in base object families in major BIM tools.

<table>
<thead>
<tr>
<th>BIM Tool</th>
<th>ArchiCAD v10</th>
<th>Bentley Architecture v8.1</th>
<th>Revit Building v9.1</th>
<th>Digital Project r5.v3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Objects</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solid model w/ features</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Site model</td>
<td>•</td>
<td>•</td>
<td>(Contoured model)</td>
<td>(Toposurface)</td>
</tr>
<tr>
<td>Space definition</td>
<td>Manual</td>
<td>Manual</td>
<td>Room (automatic)</td>
<td>Room (automatic)</td>
</tr>
<tr>
<td>Wall</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Column</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Door</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Window</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Roof</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>Custom object</td>
</tr>
<tr>
<td>Stair</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>Custom object</td>
</tr>
<tr>
<td>Slab</td>
<td>•</td>
<td>•</td>
<td>Floor</td>
<td>•</td>
</tr>
<tr>
<td>Wall End</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>•</td>
<td>Ceiling</td>
<td>Ceiling</td>
<td></td>
</tr>
<tr>
<td>Beam</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique Objects for each System</td>
<td>Skylight, Corner window</td>
<td>Shaft</td>
<td>Floor, Curtain System, Railing, Mullion, Brace, Foundation</td>
<td>Opening, Contour opening</td>
</tr>
</tbody>
</table>
One functional aspect of BIM design tools that is different from those in other industries is their need to explicitly represent the space enclosed by building elements. Environmentally conditioned building space is a primary function of a building. The shape, volume, surfaces, and properties of an interior space are a critical aspect of a building. Previous CAD systems were not good at representing space explicitly. It was generally defined implicitly, as what was left between walls, floor, and the ceiling. We owe thanks to the General Services Administration (GSA) for demanding that BIM design tools be capable of deriving ANSI/BOMA space volumes beginning in 2007. This capability was conveniently overlooked in most BIM systems until GSA mandated it in the GSA BIM Guide, stating that the amount of space produced in government buildings must be accurately assessed. Today, all BIM design tools provide this capability. The GSA BIM Guide is available online at www.gsa.gov/bim. The GSA expands its information requirements annually.

Parametric modeling is a critical productivity capability, allowing low-level changes to update automatically. It is fair to say that 3D modeling would not be productive in building design and production without the automatic update features made possible by parametric capabilities. Each BIM tool varies with regards to the parametric object families it provides, the rules embedded within it, and the resulting design behavior. These important differences are taken up in Section 2.2.

2.1.4 User-Defined Parametric Objects
While each BIM design tool has a growing set of pre-defined parametric object families (see Table 2-1), these are complete only for the most standard types of construction. They are incomplete in two ways:

- Their built-in assumptions about design behavior for the pre-defined object families are normative and do not address some special cases encountered in real world contexts.
- The base object families include the most commonly encountered ones but omit those needed in many special types of construction and building types.

Another perspective is that the base object families in a BIM design tool represent standard practice, as does Ramsey and Sleeper’s Architectural Graphic Standards (Ramsey et al. 2000). While standard practice reflects industry conventions, best practice reflects the adjustments to details, the experience a designer or firm has acquired with respect to how elements are to be detailed. Best practices distinguish the quality of design offered by most successful design practices. That is, the predefined objects that come with a
BIM design tool capture design conventions rather than expertise. Any firm that considers itself BIM-capable should have the ability to define its own libraries of custom parametric object families.

All BIM model generation tools support the definition of custom object families. If a needed parametric object family does not exist in the BIM tool, the design and engineering team has the option of either laying out the object instance using fixed B-rep or CSG geometry and remembering to update these details manually or alternatively defining a new parametric object family that incorporates the appropriate design rules and automatic updating behaviors. This embedded knowledge captures, for example, how to frame a particular style of stairway, how to detail the joining of different materials like steel and concrete or synthetic stucco and aluminum extrusions. These objects, once created, can be used in any project in which they are embedded. Clearly, the definition of details is an industry-wide undertaking that defines standard construction practices and a firm-level activity that captures best practice. Detailing is what academics such as Kenneth Frampton have referred to as the tectonics of construction (Frampton et al. 1996). It is an essential aspect of the art and craft of architecture.

If a firm frequently works with some building type involving special object families, the added labor to define these parametrically is easily justified. They provide automatic insertion of company best practice in the various contexts found in different projects. These may be at a high-level for layouts, or for detailing. They may be at various levels of sophistication.

Stadium seating is an example that justifies parametric layouts. It involves constraints that deal with seating capacity and sight lines. An example is shown in Figure 2-6. The two slightly different seating configurations at the top were
generated from the same object model, which defines a section profile in terms of seating tread width, sightline target, and visual clearance above the tread below. The extruded section is then swept along a three-part path. The seating on the right allows for sightlines at closer points to the playing field. The dialog box for the sightline settings is shown at the bottom. This layout type is straightforward and relies on older solid modeling capabilities. This example was implemented in Architectural Desktop.

A more elaborate custom parametric model for the shell of a stadium is shown in Figure 2-7 and Figure 2-8. The project is for a 50,000-seat soccer and rugby stadium in Dublin, Ireland and was undertaken by HOK Sports. The initial stadium geometry and form was developed through a series of models in Rhino. Defined by a set of site constraints and sightline regulations, a parametric model was constructed in Bentley’s Generative Components, using an Excel-based file to store the geometric information. This data was used as a path between Buro Happold and HOK Sports, allowing Buro Happold to develop a roof structure and HOK to develop a facade and roofing system, both in Generative Components. Having established a series of rules for the parametric model, both offices worked to develop the construction information using Generative Components and Excel (as a coordination tool), which is

---

**FIGURE 2-7**
The control curves generated in Generative Components. Image provided courtesy of HOK sports.
shown in the top of Figure 2-8. HOK Sports further developed the model in Bentley’s Generative Components for the production of a louvered facade system, which allowed ventilation through the stadium shell, as shown. A rendering of the stadium’s final design is also shown in the lower part of Figure 2-8. More adventurous architectural firms are using such customized parametric models to lay out and manage complex geometries for single projects, resulting in a new range of building forms that were hardly possible before.

The standard method for defining most custom parametric object families is by using a sketch tool module that is part of all parametric modeling tools. These
are primarily used for defining swept shapes. Swept shapes include: extruded profiles, such as steel members; shapes with changing cross sections, such as ductwork fittings; shapes of rotation, such as to sweep a dome section profile around a circle; and other shapes. Sweeps are the most general tool for creating custom shapes. Combined with relations to other objects and Boolean operations, a sketch tool allows constructing almost any shape family.

The sketch tool allows a user to composed of draw a 2D closed section made up of lines, arcs or higher level curves between points, not necessarily to scale, then dimension the sketch and apply other rules to reflect the design intent in terms of the parametric rules. A swept surface may be defined with multiple profiles, either interpolating between them or in some cases with step changes. Four examples from different BIM design tools are shown in Figure 2-9. Each tool has a different vocabulary of rules and constraints that can be applied to a sketch and the operations that can be associated with its parametric behavior.

When developing custom object families using B-rep or parameters, it is important that the objects carry the attributes necessary for the various assessments that the object family’s instances must support, such as cost estimation and structural or energy analyses. These attributes are also derived parametrically. These issues are taken up in Section 2.2.2.

**FIGURE 2-9**
A figure-eight sketch, swept along a path made up of an arc and a straight segment to create a complex form. Examples are generated using all four of the major BIM design tools.
2.1.5 Design-for-Construction

While all BIM tools allow users to assign layers to a wall section in terms of a 2D section, some architectural BIM authoring tools include parametric layout of nested assemblies of objects, such as stud framing, within generic walls. This allows generation of the detailed framing and derivation of a cut lumber schedule, reducing waste and allowing for faster erection of wood or metal stud framed structures. In large-scale structures, similar framing and structural layout options are necessary extensions for fabrication. In these cases, objects and rules deal with objects as parts and their composition into a system – structural, electrical, piping, etc. In the more complex cases, each of the system’s parts are then internally composed of their constituent parts, such as the wood framing, or steel reinforcing in concrete.

BIM tools for building design focus on architectural-level objects, but a different set of authoring tools have been developed for modeling at the fabrication level. These tools provide different object families for embedding different types of expertise. Early examples of such packages were developed for steel fabrication, such as Design Data’s SDS/2®, Tekla’s X-steel®, and AceCad’s StruCad®. Initially, these were simple 3D layout systems with predefined parametric object families for connections, copes that trim members around joining steel sections, and other editing operations. These capabilities were later enhanced to support automatic design based on loads, connections, and members. With associated CNC cutting and drilling machines, these systems have become an integral part of automated steel fabrication. In a similar manner, systems have been developed for precast concrete, reinforced concrete, metal ductwork, piping, and other building systems.

Recent advances have been made in concrete engineering with cast-in-place and precast concrete. Figure 2-10 shows precast reinforcing embedded to meet structural requirements. The layout automatically adjusts to the section size and to the layout of columns and beams. It adjusts for reinforcing around connections, irregular sections, and cut-outs. Parametric modeling operations can include shape subtraction and addition operations that create reveals, notches, bull-noses, and cut-outs defined by the placement of other parts. A precast architectural facade example is shown in Figure 2-11, in terms of the 3D model of the piece and the piece mark (the drawing that describes it). Each building sub-system requires its own set of parametric object families and rules for managing the layout of the system; the rules define the default behavior of each object within the system.

Parametric rules are beginning to encode large amounts of modeling expertise within each building system domain regarding how parts should be laid out and detailed. Current parametric object families provided as base objects
in BIM tools provide similar information to that provided by *Architectural Graphic Standards* (Ramsey et al. 2000), however, in a form that supports automatic piece definition, layout, connection, and detailing within the computer. More ambitious efforts are now underway among several construction material associations, such as the *American Institute of Steel Construction’s Steel Design Guide* (AISC 2007), which now encompasses 21 volumes, and the Precast/Pre-stressed Concrete Institute’s *PCI Design Handbook* (PCI 2004). Consortiums of members within these organizations have worked together to draft specifications for defining the layout and behaviors of objects in precast and steel design. Use of these tools by fabricators is discussed in more detail in Chapter 7. It should be noted that despite the fact that fabricators have had a direct hand in defining these base object families and default behaviors, they often need to be further customized so that detailing embedded in the software reflects the company’s engineering practices. It should also be noted that architects have not yet taken this road but have relied on BIM design tool developers to define their base object families. Eventually, design handbooks will be delivered in this way, as a set of parametric models and rules.

In fabrication modeling, detailers refine their parametric objects for well understood reasons: such as to minimize labor, to achieve a particular visual appearance, to reduce the mixing of different types of work crews, or to
minimize the types or sizes of materials. Standard design-guide implementations typically address one of multiple acceptable approaches for detailing. In some cases, various objectives can be realized using standard detailing practices. In other circumstances, these detailing practices can be overridden.
A company's best practices or standard interfacing for a particular piece of fabrication equipment may require further customization.

2.1.6 Object-Based CAD Systems
Several CAD systems in use today are not general purpose parametric modeling-based BIM tools such as those that have been reviewed up to this point. Rather, they are traditional B-rep modelers possibly with a CSG-based construction tree and a given library of object classes. AutoCAD®-based construction-level modeling tools, such as CADPipe, CADDUCT, and Architectural Desktop (ADT) are examples of older software technologies. Some Bentley products with fixed vocabularies for object classes are also of this type. Within these CAD system environments, users can select, parametrically size, and lay out 3D objects with associated attributes. These object instances and attributes can be exported and used in other applications, such as for bills of material, fabrication, and other uses.

These systems work well when there is a fixed set of object classes to be composed using fixed rules. Appropriate applications include: piping, ductwork, and cable tray systems for electrical layout. ADT was being developed in this way by Autodesk, incrementally extending the object classes it could model to cover those most commonly encountered in building. ADT also supports custom-defined extrusions and other B-rep shapes but does not support user defined interactions among object instances. New object classes are added to these systems through the ARX or MDL programming language interfaces.

One critical difference with BIM is that users can define much more complex structures of object families and relations among them than is possible with 3D CAD, without undertaking programming-level software development. With BIM, a curtain-wall system attached to columns and floor slabs can be defined from scratch by a knowledgeable non-programmer. Such an endeavor would require the development of a major application extension in 3D CAD.

Another fundamental difference is that in a parametric modeler, users can define custom object families and relate them to existing objects or control grids, also without resorting to computer programming. These new capabilities allow organizations to define object families in their own way and to support their own methods of detailing and layout. Such capabilities were critical for manufacturing applications, such as those that dealt with divergent fabrication processes and product designs. In building, these capabilities are similarly well-utilized by fabrication-level tools, such as those that allow steel fabricators to define connection details and precast and cast-in-place concrete detailers to define connection and reinforcing layouts. Parametric modeling transforms modeling from a geometric design tool to a knowledge embedding tool. The
implications of this capability in building design and construction are only beginning to be explored.

2.2 VARIED CAPABILITIES OF PARAMETRIC MODELERS

In general, the internal structure of an object instance as defined within a parametric modeling system is a directed graph, where the nodes are object families with parameters or operations that construct or modify an object and links in the graph reference relations between nodes. At this level, systems vary in how features are pre-defined and embedded in an object (such as a steel connection) and whether parametric objects can be nested into a larger parametric assembly and then further into even larger assemblies, as needed. Some systems offer the option of making the parametric graph visible for editing. Early parametric modeling systems rely on a full re-building of a part-and-assembly model by traversing the complete graph in response to model edits. Modern parametric modeling systems internally mark where edits are made and only regenerate affected parts of the model's graph, minimizing the update sequence. Some systems allow for the optimization of the structure's update graph based on changes made and can vary the sequence of regeneration. Other systems, called variational systems, use simultaneous equations to solve equations (Anderl et al. 1996). These capabilities all result in varied performance and scalability for dealing with projects involving a large number of object instances and rules.

The range of rules that can be embedded in a parametric graph determines the generality of the system. Parametric object families are defined using parameters involving distances, angles, and rules, such as attached to, parallel to, and distance from. Most allow 'if-then' conditions. Their definition is a complex undertaking, embedding knowledge about how they should behave in different contexts. If-then conditions replace one object family or design feature with another, based on the test result of some condition. These are used in structural detailing, for example, to select the desired connection type, depending upon loads and the members being connected. Such rules are also needed to effectively lay out plumbing and duct runs, by automatically inserting the correctly specified elbows and tees.

Some BIM design tools support parametric relations to complex curves and surfaces, such as splines and non-uniform B-splines (NURBS). These tools allow complex curved shapes to be defined and controlled similarly to other types of geometry. Several major BIM tools on the market have not included these capabilities, possibly for performance or reliability reasons.
The definition of parametric objects also provides guidelines for their later dimensioning in drawings. If windows are placed in a wall according to the offset from the wall-end to the center of the window, the default dimensioning will be done this way in later drawings. The \textit{wall control line} and \textit{endpoint intersections} define the placement dimension of the wall. (In some systems, these defaults can be overridden).

A basic capability of 3D parametric modeling systems is conflict detection of objects that spatially interfere with each other. These may be hard interferences, such as a pipe that hits a beam, or soft interferences of objects that are too close together, such as reinforcing in concrete that is too close for aggregate to pass through or a steel beam or pipe that has insufficient clearance for insulation or a concrete cover. Objects that are automatically placed via parametric rules may take interferences into account and automatically update the layout to avoid them. Others do not. This issue varies according to the specific objects being laid out and the rules embedded in them.

Enclosed spaces are the primary functional units realized in most building construction. Their area, volume, surface area composition, and often their shape and interior layout are among the most critical aspects for fulfilling the objectives of the building project. Spaces are the voids within the solid objects of construction and are usually derived from the bounding solids. They are derived in current tools in different ways: automatically defined and updated without intervention by the user; updated on demand; generated in plan defined by a polygon, then extruded to the ceiling height. These methods provide various levels of consistency, management by the user and accuracy. At the same time, space definitions contain the locations of most building functions and many analyses are applied to them, such as for energy, acoustic, and air flow simulations. While the US General Services Administration realized a first-level of capability in this area, this ability will become increasingly important for some uses discussed in Chapter 5. The capabilities of BIM design tools for space definition (as of early 2007) are shown in Table 2-1.

Parametric object modeling provides a powerful way to create and edit geometry. Without it, model generation and design would be extremely cumbersome and error-prone, as was found with great disappointment by the mechanical engineering community after the initial development of solid modeling. Designing a building that contains a million or more objects would be impractical without a platform that allows for effective low-level automatic design editing.

2.2.1 Topological Structures
When we place a wall in a parametric model of a building, we automatically associate the wall to its bounding surfaces, its base floor planes, the walls its
ends abut, any walls butting it, and the ceiling surfaces trimming its height. It also bounds the spaces on its two sides. When we put a window or door in the wall, we are defining a connection relation between the window and the wall. Similarly, in pipe runs, it is important to define whether connections are threaded, butt welded, or have flanges and bolts. Connections in mathematics are called topology and—distinct from geometry—are critical to the representation of a building model and one of the fundamental aspects of parametric modeling.

Connections carry three important kinds of information: what can be connected; what the connection consists of; and how the connection is composed in response to various contexts. Some systems restrict the object types that an object can be connected to. For example, in some systems walls can connect to walls, ceilings, and floors, but a wall-edge may not connect to stairs, windows (perpendicularly), or a cabinet. Some systems encode good practice to exclude such relations. On the other hand, prohibiting them can force users to resort to work-arounds in certain special circumstances. Connections between objects can be handled in different ways in building models. The nailing of sheetrock or wood studs to a base plate are seldom detailed but are covered by a written specification. In other cases, the connection must be defined explicitly with a detail, such as the embedding of windows into precast architectural panels. (Here we use the word connection generically to include joints and other attachments between elements.) Topology and connections are critical aspects of a BIM tool that specify what kinds of relations can be defined in rules. They are also important as design objects and often require specification or detailing. In architectural BIM tools, connections are seldom defined as explicit elements. In fabrication-level BIM tools, they are always defined as explicit elements.

2.2.2 Property and Attribute Handling
Object-based parametric modeling addresses geometry and topology, but objects also need to carry a variety of properties if they are to be interpreted, analyzed, priced, and procured by other applications.

Properties include: material specifications needed for fabrication, such as steel or concrete strength and bolt and weld specifications; material properties related to different performance issues such as acoustics, light reflectance, and thermal flows; properties for assemblies like wall and floor-to-ceiling systems or steel and precast concrete assemblies based on weight, structural behavior, etc; and properties for spaces such as occupancy, activities, and equipment needed for energy analysis.

Properties are seldom used singularly. A lighting application requires material color, a reflection coefficient, a specular reflection exponent, and possibly a texture and bump map. For accurate energy analysis, a wall requires a different
set. Thus, properties are appropriately organized into sets and associated with a certain function. Libraries of property sets for different objects and materials are an integral part of a well-developed BIM model generation tool and the environment in which the tool resides. The property sets are not always available from the product vendor and often have to be approximated by a user, the user’s firm or from the American Society of Testing and Materials data (ASTM). Although organizations such as the Construction Specifications Institute are looking at these issues, the development of property sets for supporting a wide range of simulation and analysis tools has not yet been adequately addressed and is left to users to set up.

Current BIM generation tools default to a minimal set of properties for most objects and provide the capability of adding an extendable set. Several existing BIM tools provide Uniformat™ classes to associate elements for cost estimation. Users or an application must add properties to each relevant object to produce a certain type of simulation, cost estimate, or analysis and also must manage their appropriateness for various tasks. The management of property sets becomes problematic because different applications for the same function may require somewhat different properties and units, such as for energy and lighting.

There are at least three different ways that properties may be managed for a set of applications:

- By pre-defining them in the object libraries so they are added to the design model when an object instance is created.
- By the user adding them as-needed for an application from a stored library of property sets.
- By the properties being assigned automatically, as they are exported to an analysis or simulation application.

The first alternative is good for production work involving a standard set of construction types but requires careful user definition for custom objects. Each object carries extensive property data for all relevant applications, only some of which may actually be used. Extra definitions may slow down an application’s performance and enlarge its objects’ sizes. The second alternative allows users to select a set of similar objects or property sets to export to an application. This results in a time-consuming export process. Iterated use of simulation tools may require the addition of properties each time the application is run. This would be required, for example, to examine alternative window and wall systems for energy efficiency. The third approach keeps the design application light but requires the development of a comprehensive material tag...
that can be used by all exporting translators to associate a property set for each object. The authors believe that this third approach is the desired long-term approach for attribute handling. The necessary object classifications and name tagging required of this approach must still be developed. Currently, multiple object tags must be developed, one for each application.

The development of object property sets and appropriate object classification libraries to support different types of applications is a broad issue under consideration by the Construction Specification Institute of North America and by other national specification organizations. It is reviewed in more detail in Section 5.3.3. A comprehensive solution does not yet exist but needs to be developed to support full utilization of BIM technologies.

Object libraries, representing company best practices and specific commercial building products, are an important component of a BIM environment. This important facility is reviewed in Section 5.4.

2.2.3 Drawing Generation

Even though a building model has the full geometric layout of a building and its systems—and the objects have properties and potentially specifications—drawings will continue to be required, as reports extracted from the model, for some time into the future. Existing contractual processes and work culture, while changing, are still centered on drawings, whether paper or electronic. If a BIM tool does not support effective drawing extraction and a user has to do significant manual editing to generate each set of drawings from cut sections, the benefits of BIM are significantly reduced.

With building information modeling, each building object instance—its shape, properties, and placement in the model—is defined only once. From the overall arrangement of building object instances, drawings, reports, and datasets can be extracted. Because of this non-redundant building representation, all drawings, reports, and analysis datasets are consistent if taken from the same version of the building model. This capability alone resolves a significant source of errors and guarantees internal consistency within a drawing set. With normal 2D architectural drawings, any change or edit must be transferred manually to multiple drawings by the designer, resulting in potential human errors from not updating all drawings correctly. In precast concrete construction, this 2D practice has been shown to cause errors costing approximately 1% of construction cost (Sacks et al. 2003).

Architectural drawings do not rely on orthographic projections, as learned in high school drafting classes. Rather, drawings such as plans, sections, and elevations incorporate complex sets of conventions for recording design information graphically on sheets of paper. This includes symbolic depiction of
some physical objects, dotted representation of geometry behind the section plane in floor-plans, and very selective dotted-line representation of hidden objects in front of the section plane, in addition to line-weights and annotations. Mechanical, electrical, and plumbing systems (MEP) are often laid out schematically (topologically), leaving the final layout to the contractor, after equipment has already been selected. These conventions require that BIM design tools embed a strong set of representational rules in their drawing extraction capabilities. In addition, drawing conventions of individual firms must be added to the built-in tool conventions. These issues affect both how the model is defined within the tool and how the tool is set up for drawing extraction.

Part of a given drawing definition is derived from how an object is defined, as described earlier. The object has an associated name, annotation, and in some cases line-weights and formats for presentation in different views that are carried in the object library. The placement of the object also has implications. If the object is placed relative to a grid intersection or wall end, that is how its placement will be dimensioned in the drawing. If the object is parametrically defined relative to other objects, such as the length of a beam placed to span between variably placed supports, then the drawing generator will not automatically dimension the length unless the system is told to derive the beam length at drawing generation time.

Most BIM models of buildings do not include 3D and attribute information for all the pieces of a building. Many are shown only in section details. Most BIM design tools provide the means for extracting a drawn section at the level of detail to which they are defined in the 3D model. The location of the drawn section is automatically recorded with a section-cut symbol on a plan or elevation as a cross-reference and the location can be moved if needed. The section is then detailed manually with the needed wood-blocks, extrusions, silicon beading, weather stripping; and associated annotations are provided in the fully detailed drawn section. An example is shown in Figure 2-12, with the figure on the left showing the extracted section and the one on the right showing the detailed section with drafted annotation. In most systems, this detail is associated with the section cut it was based on. When 3D elements in the section change, they update automatically in the section but the hand drawn details must be manually updated.

To produce drawings, each plan, section, and elevation is separately composed based on the above rules from a combination of cut 3D sections and aligned 2D drawn sections. They are then grouped into sheets with normal borders and title sheets. The sheet layouts are maintained across sessions and are part of the overall project data.
Drawing production from a detailed 3D model has gone through a series of refinements to make it efficient and easy. Below is an ordered list of the levels of quality that can now be supported technically, though most systems have not realized top-level drawing generation. We start from the weakest level.

1. A weak level of drawing production provides for the generation of orthographic sections cut from a 3D model, and the user manually edits the line formats and adds dimensions, details, and annotations. These details are associative. That is, as long as the section exists in the model, the annotation set-up is maintained across drawing versions. Such association capabilities are essential for effective re-generation of drawings for multiple versions. In this case, the drawing is an elaborated report generated from the model.

2. An improvement upon 1 (above) is the definition and use of drawing templates associated with elements for a type of projection (plan, section, elevation) that automatically generates dimensioning of the element, assigns line weights, and generates annotations from defined attributes. This greatly speeds-up the initial drawing setup and improves productivity, though set-up for each object family is tedious. Only changes to the presentation of data can be made in drawings; edits to the drawing do not change the model. In these first two cases, report management should be provided to inform the user that model changes have been made, but the drawings cannot automatically update to reflect these changes until they are regenerated.

3. Current top-level drawing functionality supports bi-directional editing between models and drawings. If drawings are a specialized view of the model data, then shape-changes made to the drawings should be
permitted in and propagated to the model. In this case, the drawings are updated. If displayed in windows alongside views of the 3D model, updates in any view can be referenced immediately in the other views. Bi-directional views and strong template generation capabilities further reduce the time and effort needed for drawing generation.

Door, window and hardware schedules are defined in a similar way to the three alternatives described above. That is, schedules are also model views and can be updated directly. A static report generator method is weakest, and a strong bi-directional approach is strongest. Such bi-directionality offers important benefits, including the ability to trade hardware used on a set of doors with hardware recommended on the schedule, rather from the model.

In fabrication-level BIM modeling systems, this mixed system of schematic 3D layout and 2D detailing is not used, and the design is assumed to be generated primarily from the 3D object model. In these cases, joists, studs, plates, plywood sills, and other pieces shown in Figure 2-12 would be laid out in 3D. Line-weights and crosshatching are defined for the piece type and applied automatically. Some systems store and place associated annotations with object sections, though these annotations often need shifting to achieve a well-composed layout. Other annotations refer to details as a whole, such as name, scale, and other general notes and these must be associated with the overall detail. Such capabilities come close to automated drawing extraction, but it is unlikely that automation will be complete.

Drawing sheets typically carry more information than plans, sections, and elevations for a building. They include a site plan, which shows the building’s placement on the ground plot relative to recorded geo-spatial datum. Some BIM design tools have well developed site-planning capabilities, others do not. Table 2-1 shows which BIM design tools include site objects.

An obvious current goal is to automate the drawing production process as much as possible, since most initial design productivity benefits (and costs) will depend on the extent of automatic generation. At some point, most parties involved in the building delivery process will adapt their practices to BIM technology, such as building inspectors, and financial institutions; we are slowly moving to a paperless world. Drawings will continue to be used, but as throw-away mark-up sheets by construction crews and other users. As these changes take place, the conventions regarding architectural drawings are likely to evolve, allowing them to be customized for the specific task in which they are used. This trend is described in further detail in Section 2.3.3.

It should be clear at this point that BIM technology generally allows designers to use 3D modeling to varying degrees, with 2D drawing sections filling in
the missing details. The BIM benefits of data exchange, bills of material, detailed cost estimation, and other actions are lost on those elements defined only in 2D section drawings. Thus, BIM technology allows users to determine the level of 3D modeling detail they wish to use. It can be argued that complete 3D object modeling is not warranted. Few would argue to include nails, flashing, and some forms of vapor barrier as 3D objects in a building model. On the other hand, most projects today are only partially supporting BIM. Fabrication-level models are likely (or should be) full BIM. This mixed technology is also good for firms getting started in BIM, as they can utilize the technology incrementally.

2.2.4 Scalability
A problem that many users encounter is scalability. Problems in scaling are encountered when a project model gets too large for practical use due to its large memory size. Operations become too sluggish, so that even simple operations are unfeasible. Building models are large; even simple 3D shapes take a lot of memory space. Large buildings can contain millions of objects, each with a different shape. Scalability is affected by both the size of the building, say in floor area, and also by the level of detail in the model. Even a simple building can encounter scalability problems if every nail and screw is modeled.

Parametric modeling incorporates design rules that relate geometry or other parameters of one object with those of other objects. Changing one control grid may propagate updates to the whole building. Thus, it is hard to partition a project into parts for separate development. BIM tools developed for architecture do not generally have the means for managing a project spanning multiple object files. Some systems must carry all updated objects in memory simultaneously and are considered memory-based. When the model gets too large to be held in memory, virtual memory-swapping occurs, which can result in significant waiting time. Some systems have methods of propagating relations and updates across files and can open, update, and then close multiple files within the span of a single operation. These are called file-based systems. File-based systems are generally slower for small projects but their speed decreases only slowly as project size grows.

Based on these definitions, Revit and ArchiCAD are memory-based; Bentley, Digital Project and Tekla Structures are file-based. Tool-specific work processes can mitigate some of the problems associated with scalability. These should be discussed with product vendors.

Memory and processing issues will naturally decrease as computers get faster. Sixty-four-bit processors and operating systems will also provide significant help. There will be the parallel desire, however, for more detailed building models. Issues of scalability will be with us for some time.
2.2.5 Open Questions

Strengths and Limitations of Object-Based Parametric Modeling

One major benefit of parametric modeling is the intelligent design behavior of objects. This intelligence, however, comes at a cost. Each type of system object has its own behavior and associations. As a result, BIM design tools are inherently complex. Each type of building system is composed of objects that are created and edited differently. Effective use of a BIM design tool usually takes months to gain proficiency.

Modeling software that some designers prefer, such as Sketchup, Rhino, and FormZ, are not parametric modeling–based tools. Rather, they have a fixed way of geometrically editing objects, varied only according to the surface types used; and this same functionality is applied to all object types. Thus, an editing operation applied to walls will have the same behavior when it is applied to piping. In these systems, attributes defining the object type and its functional intention, if applied at all, can be added when the user chooses, not when it is created. An argument can be made that for design use, BIM technology with its object-specific behavior is not always warranted. This topic is explored further in Chapter 5.

Why Can’t Different Parametric Modelers Exchange Their Models?

It is often asked why firms cannot directly exchange a model from Revit with Bentley Architecture, or exchange ArchiCAD with Digital Project. From the overview discussed previously, it should be apparent that the reason for this lack of interoperability is due to the fact that different BIM design tools rely on different definitions of their base objects. These are the result of different capabilities involving rule types in the BIM tool and also the rules applied in the definition of object families. This problem applies only to parametric objects, not those with fixed properties. These problems could disappear if and when organizations agree on a standard for object definitions. Until then, exchanges for some objects will be limited or will fail completely. Improvements will come about incrementally, as the demand to resolve these issues makes implementation worthwhile, and the multiple associated issues are sorted out. The same issue exists in manufacturing and has not yet been resolved.

Are There Inherent Differences in Construction, Fabrication, and Architectural BIM Design Tools?

Could the same BIM platform support both design and fabrication detailing? Because the base technology for all of these systems has much in common, there is no technological reason why building design and fabrication BIM tools
cannot offer products in each other’s area. This is happening to some degree with Revit Structures and Bentley Structures. They are developing some of the capabilities offered by fabrication-level BIM tools. Both sides address the engineering market and, to a lesser degree, the contractor market; but the expertise needed to support full production use in these information-rich areas will depend on major front-end embedding of requisite object behaviors, which are distinctly different for each building system. Expert knowledge of specific building system object behaviors is more readily embedded when it is codified, as it is, for example, in structural system design. The interfaces, reports, and other system issues may vary, but we are likely to see skirmishes in the middle-ground for a significant period of time, as each product attempts to broaden its market domains.

Are There Significant Differences between Manufacturing-Oriented Parametric Modeling Tools and BIM Tools?

Could a parametric modeling system for mechanical design be adapted for BIM? Some differences in system architecture are noted in Section 2.3.1. Of course, mechanical parametric modeling tools have already been adapted for the AEC market. Digital Project, based on CATIA, is an obvious example. Also, Structureworks is a precast concrete detailing and fabrication product based on Solidworks. In other areas, such as plumbing, curtain wall fabrication, and ductwork design, we can expect to see both mechanical parametric modeling tools and architectural and fabrication-level BIM tools vying for these markets. The range of functionality offered in each market is still being sorted out. The market is the battleground.

This chapter has provided an overview of the basic capabilities of BIM design tools resulting from their development as object-based parametric design tools. We now turn to reviewing the main BIM design tools and their functional differences.

2.3 OVERVIEW OF THE MAJOR BIM MODEL GENERATING SYSTEMS

Below, we summarize the major functional and performance capabilities that distinguish different BIM design systems, as presented in earlier sections of this chapter. The capabilities apply to both design-oriented systems as well as fabrication BIM tools. These distinguishing capabilities are proposed for those wishing to undertake a first-level review and assessment of alternative systems, so as to make a well-informed decision for a platform at the project, office, or
enterprise. The choice affects production practices, interoperability, and to some degree, the functional capabilities of a design organization to do particular types of projects. Current products also have different capabilities regarding interoperability, which may affect their ability to collaborate and can lead to convoluted workflows and replicated data.

We emphasize that no one platform will be ideal for all types of projects. Ideally, an organization would have several platforms that it supports and toggles between for specific projects. At this early date, an effort to adopt any of the available BIM design tools is a significant undertaking and is discussed in later chapters. It involves understanding the new technology, the new organizational skills it presupposes, and then learning and managing those skills. These challenges will recede over time, as the learning curve for one system is scaled. Because the functionality of BIM design tools are changing quickly, it is important to look at reviews of the current versions in AECBytes, Cadalyst, or other AEC CAD journals.

2.3.1 Discriminating Capabilities

Within the common framework of providing object-based parametric modeling, BIM authoring tools embody many different kinds of capabilities. Below, we describe them in rough-rank based on our sense of their level of importance.

**User Interface:** BIM tools are quite complex and have much greater functionality than earlier CAD tools. Some BIM design tools have a relatively intuitive and easy-to-learn user interface, with a modular structure to their functionality, while others place more emphasis on functionality that is not always well-integrated into the overall system. Criteria to be considered here should include: consistency of menus across the system’s functionalities following standard conventions; menu-hiding that eliminates irrelevant actions not meaningful to the current context of activities; modular organization of different kinds of functionality and on-line help providing real-time prompts and command-line explanation of operations and inputs. While user interface issues may seem minor, a poor user interface results in longer learning times, more errors, and often not taking full advantage of the functionality built into the application.

**Drawing Generation:** How easy is it to generate drawings and drawing sets and to maintain them through multiple updates and series of releases? Assessment should include quick visualization of the effects of model changes on drawings, strong associations so that model changes propagate directly to drawings and vice versa, and effective template generation that allows drawing types to carry out as much automatic
formatting as possible. A more thorough review of functionality is provided in Section 2.2.3.

**Ease of Developing Custom Parametric Objects:** This is evaluated with regard to the existence and ease-of-use of a sketching tool for defining parametric objects; determining the extent of the system's constraint or rule-set ('a general constraint rule set should include distance, angle including orthogonality, abutting faces and line tangency rules and ‘if-then’ conditions) its ability to interface the objects into the user interface for easy embedding in a project’) and its ability to support parametric assemblies of objects. These issues are explained further in Section 2.1.4.

**Scalability:** the ability to handle combinations of a large project scale and modeling at a high level of detail. This involves the ability of the system to remain interactive and responsive regardless of the number of 3D parametric objects in the project. A fundamental issue is the degree that the system is disk-based, in terms of data management, rather than memory-based. Disk-based systems are slower for small projects, but their delay time grows slowly as the project size grows. Memory-based systems performance drops quickly once memory space is exhausted. These issues are partially limited by the operating system; Windows XP supports up to 2 GB of working memory for a single process. Sixty-four-bit architectures eliminate the memory use restriction. Graphic card performance is important for some systems. This topic is discussed in Sec. 2.2.4.

**Interoperability:** Model data is generated, in part, to share with other applications for early project feasibility studies, for collaboration with engineers and other consultants and later for construction. It is supported by the degree that the BIM tool provides direct interfaces with other specific products and, more generally, its import and export support of open data exchange standards, which are reviewed in detail in Chapter 3.

**Extensibility:** A BIM authoring tool is both for end-use and for use as a platform for customization and extension. Extensibility capabilities are assessed based on whether they provide scripting support—an *interactive language* that adds functionality or automates low-level tasks, similar to AutoLISP® in AutoCAD—an Excel format bi-directional interface, and a broad and well-documented application programming interface (API). Scripting languages and Excel interfaces are generally for end-users, while an API is for software developers. These capabilities are needed depending on the extent to which a firm expects to customize, particularly in the area of interoperability.

**Complex Curved Surface Modeling:** Support for creating and editing complex surface models based on quadrics, splines, and non-uniform B-splines
is important for those firms that do this type of work or that are planning to. These geometric modeling capabilities in a BIM tool are foundational; they cannot be added on later.

**Multi-User Environment:** Some systems support collaboration among a design team. They allow multiple users to create and edit parts of the same project directly from a single project file and manage user access to these various information parts.

Below, we offer an overview of the current capabilities of the major building model generation platforms. Some reviewed support only architectural design functions, others only various types of fabrication-level building systems, and others both. Each assessment is for the version of the software system noted; later versions may have better or worse capabilities. We review them according to the criteria developed above.

### 2.3.2 BIM Tools for Architectural Design

Each BIM building design platform is introduced in terms of its heritage, corporate organization, the family of products it is a part of, whether it uses a single file or multiple files per project, support for concurrent usage, interfaces supported, extent of the object library, general price class, building classification system supported, scalability, ease of drawing generation, support for 2D drawn sections, types of objects and derived attributes, and ease of use.

As is broadly understood, the acquisition of a software package is very different from most other purchases we make. Whereas the purchase of a car is based on a very specific product and set of features, a software package involves both its current capabilities and the development path of enhancements that are released regularly, at least annually. A purchaser is buying into both the current product and its future evolutions, as projected by the company. One is also purchasing a support system that at least one person in a firm will be dealing with. The support system is an augmentation of the user-provided documentation and on-line support built into the BIM tool.

Apart from the vendor’s support network, a software system owner is also part of a broader user community. Most provide blog communication for peer-to-peer help and open portals for the exchange of object families. These may be free or available at a small cost.

**Revit:** Revit Architecture is the best known and current market leader for the use of BIM in architectural design. It was introduced by Autodesk in 2002 after the company acquired the program from a start-up. Revit is a completely separate platform from AutoCAD, with a different code base and file structure. The version reviewed here is 9.1. Revit is a family of integrated products that
currently includes Revit Architecture, Revit Structure, and Revit MEP. It includes: gbXML interfaces for energy simulation and load analysis; direct interfaces to ROBOT and RISA structural analyses, and the ability to import models from Sketchup, a conceptual design tool, and other systems that export DXF files. Viewing interfaces include: DGN, DWG, DWF™, DXF™, IFC, SAT, SKP, AVI, ODBC, gbXML, BMP, JPG, TGA, and TIF. Revit relies on 2D sections as a way of detailing most types of assemblies.

**Revit's strengths:** It's easy to learn and its functionality is organized in a well-designed and user-friendly interface. It has a broad set of object libraries developed by third parties. It is the preferred interface for direct link interfaces, because of its market position. Its bi-directional drawing support allows for information generation and management based on updates from drawing and model views; it supports concurrent operation on the same project; and it includes an excellent object library that supports a multi-user interface.

**Revit's weaknesses:** Revit is an in-memory system that slows down significantly for projects larger than about 220 megabytes. It has limitations on parametric rules dealing with angles. It also does not support complex curved surfaces, which limits its ability to support design with or reference to these types of surfaces.

**Bentley Systems:** Bentley Systems offers a wide range of related products for architecture, engineering, and construction. Their architectural BIM tool, Bentley Architecture, introduced in 2004, is an evolutionary descendent of Triforma. Integrated with Bentley Architecture are: Bentley Structural, Bentley Building Mechanical Systems, Bentley Building Electrical Systems, Bentley Facilities, Bentley PowerCivil (for site planning), and Bentley Generative Components. These are file-based systems, meaning that all actions are immediately written to a file and result in lower loads on memory. Third parties have developed many different applications on the file system, some incompatible with others within the same platform. Thus a user may have to convert model formats from one Bentley application to another. Currently, Bentley Architecture is in version V8.9.2.42. Interfaces with external applications include: Primavera and other scheduling systems and STAAD and RAM for structural analyses. Its interfaces include: DGN, DWG, DXF™, PDF, STEP, IGES, STL, and IFC. Bentley also provides a multi-project and multi-user model repository called Bentley ProjectWise.

**Bentley System's strengths:** Bentley offers a very broad range of building modeling tools, dealing with almost all aspects of the AEC industry. Bentley supports modeling with complex curved surfaces, including Bezier and NURBS. It includes multiple levels of support for developing custom parametric objects, including the Parametric Cell Studio and Generative Components. Its parametric modeling plug-in, Generative Components, enables definition of
complex parametric geometry assemblies and has been used in many prize-winning building projects. Bentley provides scalable support for large projects with many objects.

**Bentley System’s weaknesses:** It has a large and non-integrated user interface that is hard to learn and navigate; its heterogeneous functional modules include different object behaviors, making it hard to learn. It has less extensive object libraries than similar products. The weaknesses in the integration of its various applications reduce the value and breadth of support that these systems provide individually.

**ArchiCAD:** ArchiCAD is the oldest continuously marketed BIM architectural design tool available today. Graphisoft began marketing ArchiCAD in the early 80s. It is the only object-model-oriented architectural CAD system running on the Apple Macintosh. Headquartered in Budapest, Graphisoft was recently acquired by Nemetschek, a German CAD company popular in Europe with strong civil engineering applications. The current version of ArchiCAD is release 11.0. Today, ArchiCAD continues to serve the Mac platform in addition to Windows and has recently released a Mac OS X (UNIX) version. Graphisoft recently introduced a number of construction-oriented applications on the ArchiCAD platform. In early 2007, after Graphisoft was acquired by Nemetschek, the construction applications were spun off to Vico Software, a new company that is actively marketing them. These applications are discussed in Chapter 6.

ArchiCAD supports a range of direct interfaces, with Maxon for curved surface modeling and animation, ArchiFM for facility management and Sketchup. It has interfaces with a suite of interfaces for energy and sustainability (gbXML, Ecotect, Energy+, ARCHIPHISIK and RIUSKA). Custom parametric objects are primarily defined using the GDL (Geometric Description language) scripting language, which relies on CSG-type constructions and a Basic-like syntax. (Basic is a simple programming language often taught to beginners). It contains extensive object libraries for users and also has an OBDC interface.

**ArchiCAD’s strengths:** It has an intuitive interface and is relatively simple to use. It has large object libraries, and a rich suite of supporting applications in construction and facility management. It is the only strong BIM product currently available for Macs.

**ArchiCAD’s weaknesses:** It has some limitations in its parametric modeling capabilities, not supporting update rules between objects in an assembly or automatic application of Boolean operations between objects (Khemlani 2006). While ArchiCAD is an in-memory system and can encounter scaling problems with large projects, it has effective ways to manage large projects; it can partition large projects well into modules in order to manage them.
Digital Project: Developed by Gehry Technologies, Digital Project (DP) is an architectural and building customization of Dassault’s CATIA, the world’s most widely used parametric modeling platform for large systems in aerospace and automotive industries. DP requires a powerful workstation to run well, but it is able handle even the largest projects. The One Island East case study in Chapter 9 provides an example of DP’s ability to model every part of a 70 story office tower. It is able to model any type of surface and can support elaborate custom parametric objects, which is what it was designed to do. The logical structure of CATIA involves modules called Workbench. Until the third release of Version 5, it did not include built-in base objects for buildings. Users could re-use objects developed by others, but these were not supported by DP itself. With the introduction of the Architecture and Structures Workbench, Gehry Technologies has added significant value to the base product. Although not advertised, DP comes with several other workbenches: Knowledge Expert supports rule-based checking of design; the Project Engineering Optimizer allows for easy optimization of parametric designs based on any well-defined objective function; and Project Manager for tracking parts of a model and managing their release. It has interfaces with Ecotect for energy studies.

DP supports VBA scripting and a strong API for developing add-ons. It has the Uniformat© and Masterformat© classifications embedded, which facilitates integration of specifications for cost estimating. It supports the following exchange formats: CIS/2, SDNF, STEP AP203 and AP214, DWG, DXF™, VRML, STL, HOOPS, SAT, 3DXML, IGES, and HCG. In Release 3, it has IFC support.

Digital Project’s strengths: It offers very powerful and complete parametric modeling capabilities and is able to directly model large complex assemblies for controlling both surfaces and assemblies. Digital Project relies on 3D parametric modeling for most kinds of detailing.

Digital Project’s weaknesses: It requires a steep learning curve, has a complex user interface, and high initial cost. Its predefined object libraries for buildings are still limited. External third party object libraries are limited. Drawing capabilities for architectural use are not well developed; most users output sections to drafting systems for completion.

AutoCAD-based Applications: Autodesk’s premier building application on the AutoCAD platform is Architectural Desktop (ADT). ADT was Autodesk’s original 3D building modeling tool prior to the acquisition of Revit. It is based on solid and surface modeling extensions for AutoCAD and provides a transition from 2D drafting to BIM. It has a predefined set of architectural objects, and while not fully parametric, it provides much of the functionality offered by
parametric tools, including the ability to make custom objects with adaptive behaviors. External Reference Files (XREF) are useful for managing large projects. Drawing files remain separate from the 3D model and must be managed by the user, albeit with a degree of system version control. It relies on AutoCAD’s well-known capabilities for drawing production. Interfaces include: DGN, DWG, DWF™, DXF™, and IFC. Its programming extensions include: AutoLISP, Visual Basic, VB Script, and ARX (C++) interfaces.

Additional 3D applications developed on AutoCAD come from a large world-wide developer community. These include Computer Services Consultants (CSC) which offers a number of structural design and analysis packages, AEC Design Group, which offers CADPIPE, COADE Engineering Software, that offers piping and plant design software, SCADA Software AG, that develops control system software, and other groups that produce 3D applications for piping, electrical system design, structural steel, fire sprinkler systems, ductwork, wood framing and others.

**AutoCAD-based applications’ strengths:** Ease of adoption for AutoCAD users because of user interface consistency; easy use because they build upon AutoCAD’s well-known 2D drafting functionality and interface.

**AutoCAD-based applications’ weaknesses:** Their fundamental limitations are that they are not parametric modelers that allow non-programmers to define object rules and constraints; limited interfaces to other applications; use of XREFs (with inherent integration limitations) for managing projects; an in-memory system with scaling problems if XREFs are not relied upon; need to propagate changes manually across drawings sets.

**Tekla Structures:** Tekla Structures is offered by Tekla Corp., a Finnish company founded in 1966 with offices worldwide. Tekla has multiple divisions: Building and Construction, Infrastructure and Energy. Its initial construction product was Xsteel, which was introduced in the mid-1990s and grew to be the most widely used steel detailing application throughout the world.

In response to demand from precast concrete fabricators in Europe and North America (represented by the ad hoc Precast Concrete Software Consortium), the software’s functionality was significantly extended to support fabrication-level detailing of precast concrete structures and facades. At the same time, support for structural analysis, direct links to finite-element analysis packages (STAAD-Pro and ETABS), and an open application programming interface were added. In 2004 the expanded software product was renamed Tekla Structures to reflect its generic support for steel, precast concrete, timber, reinforced concrete, and for structural engineering.

Tekla Structures supports interfaces with: IFC, DWG™, CIS/2, DTSV, SDNF, DGN, and DXF™ file formats. It also has export capabilities to CNC
fabrication equipment and to fabrication plant automation software, such as Fabtrol (steel) and Eliplan (precast).

**Tekla Structures’ strengths:** Its versatile ability to model structures that incorporate all kinds of structural materials and detailing; its ability to support very large models and concurrent operations on the same project and with multiple simultaneous users. It supports compilation of complex parametric custom component libraries with little or no programming.

**Tekla structures’ weaknesses:** While a powerful tool, its full functionality is quite complex to learn and fully utilize. The power of its parametric component facility, while a strength, requires sophisticated operators who must develop high levels of skill. It is not able to import complex multi-curved surfaces from outside applications, sometimes leading to work-arounds. It is relatively expensive.

**DProfiler:** DProfiler is a product of Beck Technologies, located in Dallas, Texas. It is based on a parametric modeling platform acquired from Parametric Technologies Corporation (PTC) in the middle 1990s, after PTC decided not to enter the AEC market. DProfiler is an application based on a platform called DESTINI that has evolved from the PTC-acquired software. DProfiler supports very quick definition for the conceptual design of certain building types and then provides feedback regarding construction costs and time. For income generating facilities, such as hotels, apartments, and office buildings, it provides a full economic cash-flow development proforma. It supports planning for: office buildings up to 20 stories; one and two-story medical buildings; apartment buildings and hotels up to 24 stories; elementary, middle, and high schools; and town halls, churches, and movie centers and more. With financial and schedule reporting, the user gains a set of concept design drawings. Users can input their own cost data or use data from RSMeans. Their advertising states 5% accuracy in its business calculations, but these are informal assessments. It currently supports Sketchup and DWG export for further development. The Hillwood case study (Section 9.9) describes its use. Its interfaces include Excel and DWG. Other applications being developed on the DESTINI platform include energy analyses.

**DProfiler strengths:** DProfiler is being marketed as a closed system, primarily for preliminary feasibility studies before actual design begins. Its ability to generate quick economic assessments on a project plan is unique.

**DProfiler weaknesses:** DProfiler is not a general purpose BIM tool. Its single purpose (currently) is economic evaluation of a construction project (cost estimating and, where appropriate, income forecasting). Once a model is complete, its interface to support full development in other BIM design tools is limited to 2D DWG files.
2.4 CONCLUSION

Object-based parametric modeling is a major change for the building industry that is greatly facilitating the move from a drawing-based and handcraft technology to one based on digitally readable models that can be exchanged with other applications. Parametric modeling facilitates the design of large and complex models in 3D but imposes a style of modeling and planning that is foreign to many users. Like CADD, it is most directly used as a documentation tool separate from designing. A growing number of firms, however, use it directly for design and for generating exciting results. Some of these uses are taken up in Chapter 5, and the case studies in Chapter 9 provide further examples.

The ability to extract geometric and property information from a building model for use in design, analysis, construction planning, and fabrication, or in operations, will have large impacts on all aspects of the AEC industries. Many of these opportunities are taken up and discussed in the succeeding chapters. The full potential of this enabling capability will not be fully known for at least a decade, because its implications and new uses are discovered gradually. What is currently known is that object-based parametric modeling resolves many of the fundamental representational issues in architecture and construction and allows quick payoffs for those transitioning to it, even with only partial implementation. These payoffs include a reduction in drawing errors due to the built-in consistency of a central building model and the elimination of design errors based on spatial interferences.

While object-based parametric modeling has had a catalytic influence on the emergence and acceptance of BIM, it is not synonymous with BIM tools or the generation of building models. There are many other design, analysis, checking, display, and reporting tools that can play an important role in BIM procedures. Many information components and information types are needed to fully design and construct a building. The authors are of the opinion that many types of software can facilitate the development and maturing of Building Information Modeling. The BIM tools considered here are only the newest in several generations of tools, but in fact they are proving to be revolutionary in their impact.
Chapter 2 Discussion Questions

1. Summarize the major functionalities that distinguish the capabilities of a BIM design tool from 3D CAD modeling tools.

2. Most BIM design tools support both 3D object models as well as 2D drawn sections. What considerations should be made when determining the changeover level of detail, such as when to stop modeling in 3D and complete the drawings in 2D?

3. Why is it unlikely that a single integrated system will incorporate a unified parametric model of all of a building’s systems? On the other hand, what would be the advantages if it could be achieved?

4. In what ways are some of the current popular design tools not BIM tools? Sketchup? 3D Max Viz? FormZ? Rhino?

5. What are the essential differences between a manufacturing parametric modeling tool, such as Autodesk Inventor, and a BIM design tool, such as Revit?

6. Do you think there may be additional manufacturing oriented parametric modeling tools used as a platform to develop BIM applications? What are the marketing costs and benefits? What are the technical issues?

7. Suppose you are a Chief Information Officer for a medium-sized architectural firm (with fewer than 25 employees). The firm specializes in school buildings. Propose an outline structure for the firm’s custom object library. Relate to the list of built-in objects in Table 2-1 when considering your answer.

8. You are part of a small team of friends that has decided to start an integrated design-build firm comprised of both a small commercial contractor and two architects. Lay out a plan for selecting one or more BIM-model creation tools. Define the general criteria for the overall system environment.