3 Software Tools

INTRODUCTION

This chapter discusses the functionality of the software tools, and will aid in understanding the characteristics of some of the most common tools used for the BIM process. The author does not sponsor any specific software product, and the companies whose products are represented here are not the only ones producing software of this type. These descriptions are not included from a marketing standpoint, but as guidelines for the reader to compare different products available in the marketplace. It is important for the potential BIM manager to gain enough software familiarity to be able to select the most appropriate tools and deal with the vendors and technical support for these tools. It is also essential for all people involved in the application of the BIM processes to familiarize themselves with the software capabilities and to stay current in this field. Software changes continuously, and it is important to remain informed of the current state of the art of this technology. All software is based on a particular *architecture*, e.g., an approach to dealing with the organization of the information in the BIM; this software architecture changes less frequently than the particular functionality of the software. The descriptions provided by the software developers in their marketing materials will help the reader to become aware of some of the differences among these tools, and reading this book will give a good background for making a decision as to which tool is best suited for a particular purpose. It requires a good deal of research to update oneself with the state of the art of BIM software development, and due to its temporal nature, books will generally not be the place to look for the most current information about the functionality of a particular tool.

The software industry is a business, it is for profit, the software companies will develop what they think they can sell, and consequently they will also not develop anything they think they will not be able to sell. This puts the burden on the users (designers, fabricators, contractors, etc.) to educate the software industry about their needs. It also means that "what you see is what you get," and there is no guarantee about that which is promised by the software companies. Shop well, and keep searching for practical solutions that solve the problems at hand.

A huge area that is frequently overlooked is the training of software operators. This book addresses training in BIM processes; it does not address training in specific software tools, although this clearly is a requirement for anyone planning to use such tools. Most modeling tools will be intuitive enough to allow even the inexperienced user to build relatively simple projects in a fairly short time. The young generation just starting professional careers has been raised on video games and Internet exploration. This is both good and bad. The good part is that they are generally fearless about computers and will keep hitting various keys and clicking dialog boxes until some result is achieved; this, however, also includes the less desirable side of this approach—it often lacks discipline. There is only one way to master a modeling tool, and that is through discipline; without the necessary rigorous exercises the most one can achieve is advanced amateur status. Now it is fair to say that unless one is interested in becoming professional, there is nothing wrong with being an amateur. As a BIM manager, however, it is important to know the difference. An amateur model may be remarkably different from a professional one; inaccuracies or unclear organization in a model can make them virtually unusable for any number of analytic processes. It is thus imperative that proper training in a specific software tools be part of the preparation to implement BIM on a project. This training will frequently have to be provided by the software developer or an outside consultant. A word of caution is needed here: Make sure that what is promised by sales representatives will actually be deliverable; not all companies have a very good follow-up support record. Not all software is equal when it comes to learning how to use it either; the training of staff will generally require far more of the company's resources than the actual software purchase itself. Just as software salespeople are likely to promise more than they can deliver, software operators often represent themselves as being more capable than they are; in fairness to them, they do not know what they do not know, but that does leave the burden on the BIM manager who hires the operators for the performance of specific tasks.

Interoperability is a concept that is mentioned several times throughout this text. A separate book could be written about this subject; it is important, it is complex, and it is continually changing. This discussion does not address the subject in detail, but the BIM practitioner needs to become informed about the interoperability of the specific tools used on any given project; not just looking at the specification sheets of the software, but actually making the processes work on several files is a necessity.

It is shocking to realize how dependent everyone is on the various products that have been developed for use. For most users it is simply not realistic to customize such software.* The only hope is that the software developers will address the needs of the field, or (as is most often the case) the field will be able to adjust its practices to the capabilities of the available tools. Both possibilities may lead to unintentional but positive surprises, as well as to frustrations and setbacks.

*Frank Gehry is an exception—Gehry Industries developed a special version of CATIA, a French aeronautical design software, to address the architectural design challenges of Frank Gehry's work. This tool is now commercially available for those who wish to design these types of architectural forms.

MODELING TOOLS

The kind of information that is desired from the BIM will determine the type of model that is required, which in turn will lead to the nature of the modeling tool that can accomplish this. It may even be advantageous to compose the project of several models to be able to take advantage of different modeling tools' strengths.

Models are by definition **simulations**; a model represents something—it simulates it—it is an **abstraction** of the "real object." This also means that the model is *not real*; it is an abstraction, and there will *always* be a difference between the model and reality. It is extremely important to continue to remember this when using a model to facilitate the decision-making process in a project; sound decisions can be made only when the level of abstraction and the reliability of a model are clearly understood. *The implications of construction tolerances are important relative to model accuracy; tolerances are how reality deals with abstraction.* Generally 3D models do not address tolerance; they are exact, just as 2D drawings state exact dimensions for components. In reality, however, where the real components have to fit, tolerance is required because reality never matches the degree of perfection of the abstraction in a model.

A mathematical model can be composed of a structure of mathematical formulas representing a certain process, which can be studied by manipulating the variables in the formulas or by adjusting the formulas in relation to one another. Generally a model is created to aid in the visualization and understanding of a process, or an object. Models are used to represent, analyze, communicate, etc. If "a picture is worth thousand words," then it could perhaps be said that "a 3D model is worth a thousand pictures."

Non-3D Element Modelers (Process Modeling)

These types of models have been in use for a long time and can be seen as visual aids in process planning. Examples are Gantt charts for a schedule; graphs, bar or pie charts, etc., for spreadsheet information, drawings, and diagrams, and endless other model representations of information. The model represents the information in such a way that it allows the viewer to visualize its impact or relevance.

This does not refer to the modeling of non-3D elements in a BIM, such as project inspections, submittal approvals, etc. There are various methods to represent these sorts of events by hyperlinking placeholder 3D elements in the BIM to information of a non-3D nature.

Surface Modelers

Surface models consist only of surfaces (that have no thickness) to give a 3D look to their "hollow" forms (see Fig. 3.1). These models are generally only used for visualization, and typically they contain only visual information. Software programs such as SketchUp, Maia, etc., make surface models.

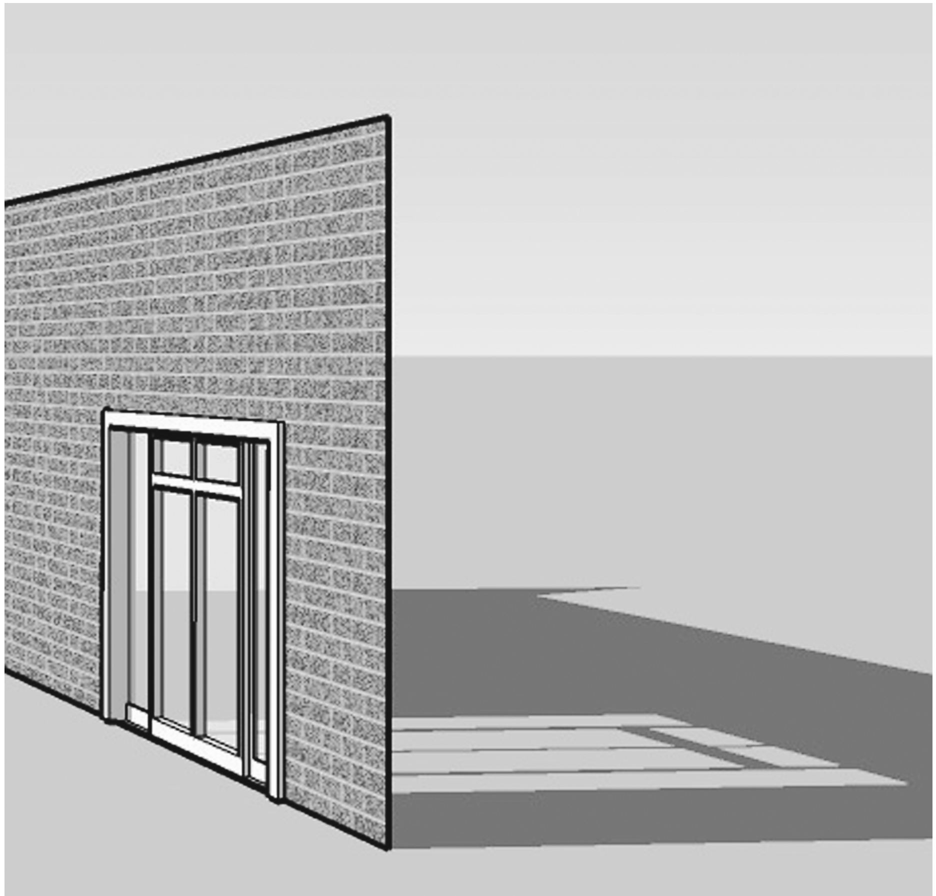
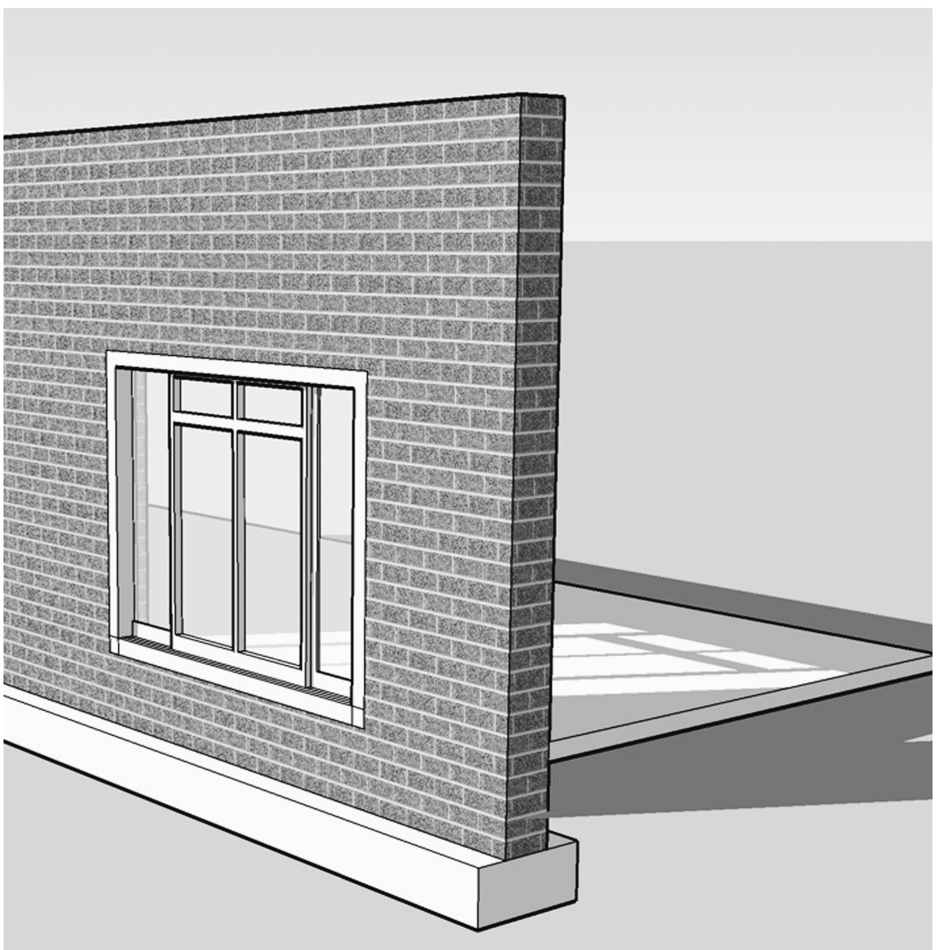


Figure 3.1 The most basic surface model, just one plane representing a wall, modeled in SketchUp.

Most of the software tools used to make 3D images and animations are surface modelers; the results may be anywhere from very sketchy to photorealistic quality, and can be very useful for communication. See Fig. 3.2. Models produced in this manner may be able to help in BIM; i.e., it is possible to create a surface model of a HVAC ducting system and use it in a clash analysis in NavisWorks. A surface model will not be usable, however, for the fabrication of ductwork; a solids modeling tool will generally be used for that purpose. It may thus be more efficient to model the ducts with a tool that is designed to aid in the fabrication process in the first place. This also encourages collaboration with the mechanical subcontractor, so that he or she can produce the fabrication model that can then be used for a coordination/clash analysis. A word of caution about the timing of model production: A detailed production model can only be created once the necessary information for this level of detail is available. Therefore a certain type of model may not be available until a specific stage of the project's development; i.e., if a model is needed for clash detection between MEP and structural systems before the fabricators have produced the models, it will be necessary to perhaps have the designers produce the models. Only some, and not all modeling tools facilitate the development of a design model into a fabrication (or production) model.

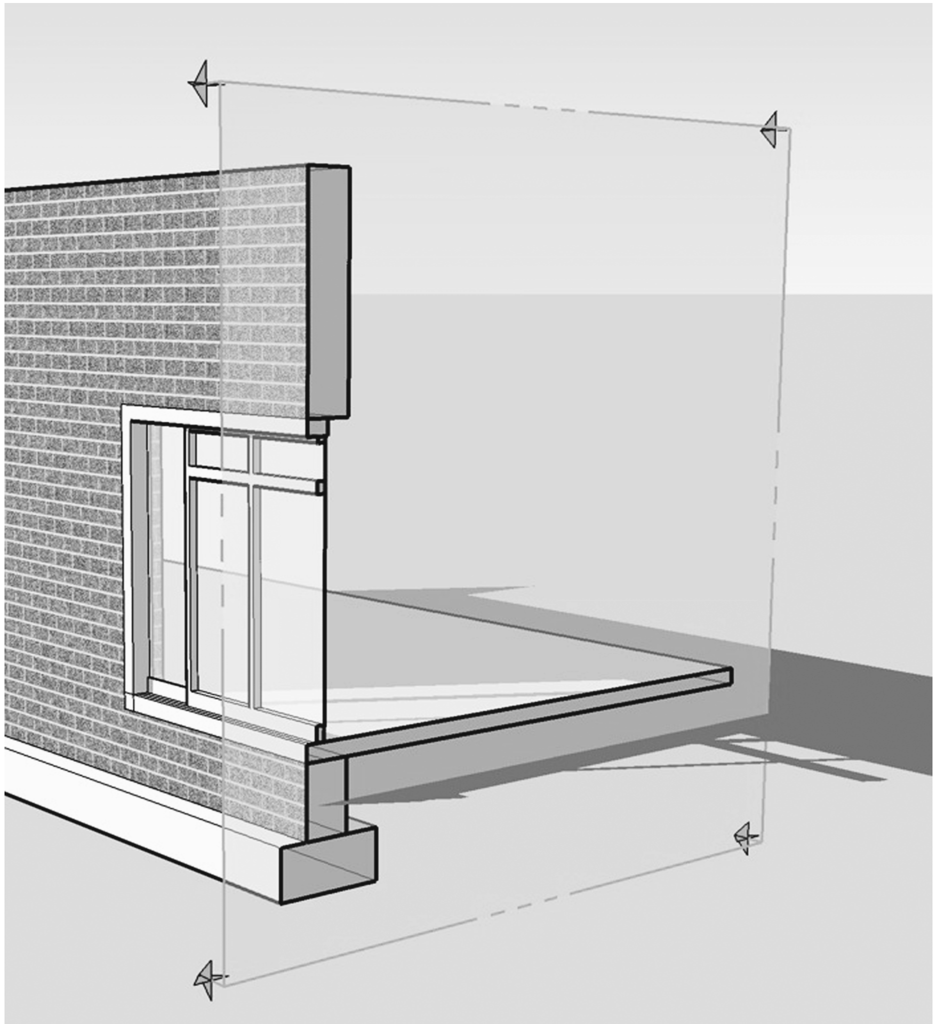
A viewer such as NavisWorks will turn a solid model into a surface model; i.e., the solids will be represented by all their surfaces only. This "translation" allows the software to manipulate the information faster, thus making navigating through the model in NavisWorks faster and more practical for the user. The import of model data from a solids modeler will thus be limited to the 3D information that is critical to the functionality of NavisWorks; the nature of import and export functions between software tools is often limited, and relates to interoperability. It is the translator that determines exactly what information is brought across to the other platform, and how it is manifested in that new platform. Trial is the only reliable test of the utility of the translated format. This is why it is so critical to run a test simulation of all the processes anticipated during a BIM project; there can be no certainty



(*a*)

Figure 3.2
A slightly more complex surface model. The wall and slab are now represented by two surfaces (and ends), as can be seen in the section view of a SketchUp model. (*a*) Note that the model without the section cut looks like a solid model, and (*b*) that the wall and slab are hollow in the section cut view.

Figure 3.2 (*Continued*)



(*b*)

about the ability to perform certain analysis until it has been tried and proved (see Chap. 2). The fact that NavisWorks translates all models into surface models also means that models made by any other surface modeler, e.g., SketchUp, will also be usable for clash detection, sequence analysis, etc., in NavisWorks.

Surface Model Abstraction

The *abstraction* of the model refers to the nature of the model components; it is the way in which reality has been abstracted in the model. With surface modelers there is a specific abstraction already inherent in the nature of the modeling tools, namely, that all volumes are represented by their surfaces. The level of abstraction results directly from the requirement for detail in the model components. The more details are desired, the more realistic (less abstract) the model will have to be. This of course also directly effects the amount of modeling effort required to produce this level of detail.

Surface Model Organization

The *organization* of the model refers to the manner in which the model components are organized within the model. SketchUp, e.g., uses layers that can contain any components or parts of components. The layers can be shown or hidden, locked or unlocked, and greatly aid in the flexibility of editing the model. Due to its nature a surface of a 3D element in SketchUp consists of a plane with edges that are all individually selectable; it is actually possible for some of the edges and surfaces of a component (a form) to be in a different layer from the other edges and surfaces, something that can cause confusion with the novice modeler. Most surface modelers also permit the grouping of surfaces into objects (or components) that are not editable (other than scaling) until they are "exploded" into their constituent edges and surfaces again. When manipulated as a component or group, such elements can also be scaled in any or all dimensions to become functional in the project model. For example, a drawing of a floor plan or site plan can be imported as an object and then scaled to actual size to build the 3D model on its lines. See SketchUp exercises in Chap. 4. A door or window can be placed as a component into a model and scaled to reflect the desired size of that door or window.

Surface Modeling Tool Characteristics

Surface modelers create *only* edges, from which all planes and forms are constituted; all elements in such a model are represented by these edges. Of course the planes can have color and 2D texture that give them the look of a particular material. It is very similar to scene design for a play, or to constructing everything from paper, even the sticks that hold up the assembly. To build the more complex model parts, specific groups of edges and planes can be saved as components that may look similar to the "library parts" of a solid model. These model parts are not parametric, however, and still are simply collections of edges, with little ability to hold information other than their physical size and look. Groups of edges can generally only be scaled to become larger or smaller in relation to any of or all the *X, Y,* and *Z* axes, meaning that they can be stretched (or shrunk) in one, two, or three directions proportionally. When plan or section views of a surface model are viewed, they are collections of lines with empty space in between; they cannot show a distinction between the mass of a solid component and the space between the components. Thus if a drawing is needed, it will generally be necessary to do a bit of editing to turn it into an understandable document, with the necessary graphic information.

The 3D model, however, can look like any other and can be visually indistinguishable from a solid model. Thus it is often the tool of choice for the creation of flexible 3D models where the emphasis is on the visual information only, especially since the surface modelers are often faster and simpler to use.

Solid Modelers

Solid models are actual representations of real objects in 3D space, having the correct dimensions, location, and ability to contain other information about the object characteristics; e.g., a wall can consist of various thicknesses of specific materials that can then be used to calculate material quantities from the model. Solid objects represent the real objects both inside and out; in other words the walls have an actual thickness and "look solid" in a section view. See Figs. 3.3 and 3.4. A model is an abstraction, however, and there is a limit to the detail that can (and should) be represented in a model. Solid model components have volume, even the thinnest elements will have a measurable thickness (unlike the surfaces in a surface model).

Solid Model Abstraction

The abstraction of the model will largely depend on the stage of project development and the level of detail required from the model for the analysis. A very schematic (massing) model may look at an entire story as one thick solid slab, while only modeling the basic structural elements will already provide much more information in a model that is still at a schematic level of development. The level of model abstraction thus reflects the project, its stage of development, and its needs; it is less dependent on the software tool chosen to create the model. The questions related to determining the level of model abstraction are as follows:

- What needs to be represented in the project model?
- How will the components of the project be represented in the model?
- How can the required information be attached to these components? See "The Nature of Information" under "Specifying the Model" in Chap. 2.

These three questions will enable the project team to determine the level of abstraction and detail that needs to be addressed in the model. As was noted in Chap. 2, however, a lot of planning will be required before the answers to these three questions can be clear in the minds of the project team members.

Most modeling tools will specialize in a particular level of detail. Solid models, since they can be designed to be useful for fabrication instructions for component manufacturing, are often of the most detailed nature. Whether the same software tool will also be fast and flexible enough to be used for the design phase of the same components will depend on the particular software characteristics. It is important to note here that the nature of the software tool is also heavily dependent on the discipline that it serves. The mechanical and structural fields are fortunate enough to have a fairly limited range of materials, forms, sizes, etc., that are used to generate their building systems. This comparatively narrow range of components makes it much more likely for software in these disciplines to be more advanced in its functionality. Tekla produces software tools that will let the engineer introduce loading configurations, lay out structural steel systems, analyze them for stress, deflection, etc., and then design and model the steel members and their detailed connections. The engineer continues to use the same file to either generate shop drawings for fabrication or send the model to fabrication, where the manufacturing equipment is set up to take its instructions from such a model. This is

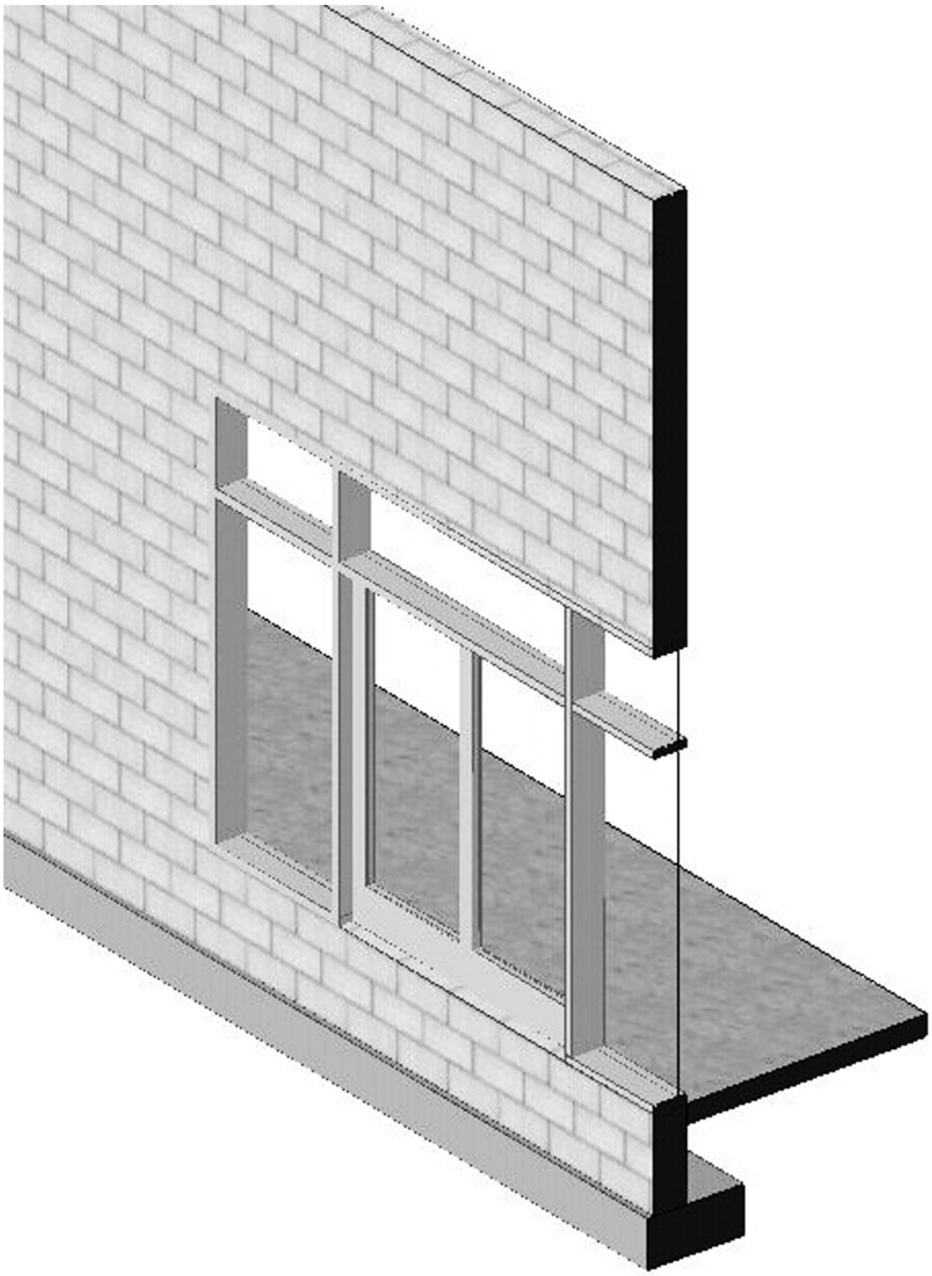


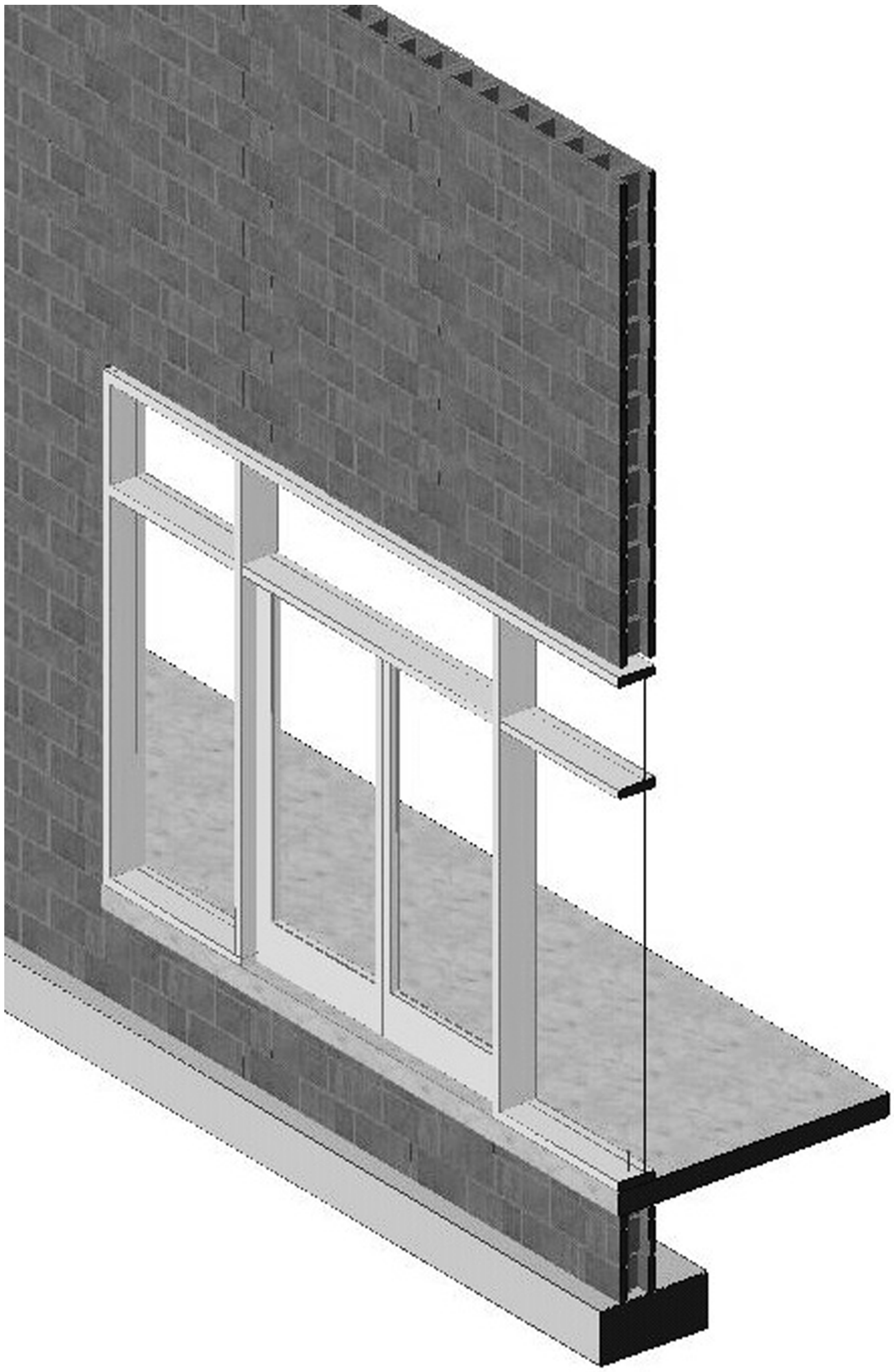
Figure 3.3
A section cut view through the wall showing solid elements as wall, slab, and window mullions, modeled in ArchiCAD.

referred to as Direct Digital Exchange (DDE) in the industry. The model may also be used during assembly to help establish the steel erection sequence. There exist mechanical HVAC ductwork and piping software tools that have similar functionality.

Tekla or other specialty modelers can also create a basic assembly of shapes, similar to what Revit Structure or Vico Constructor could produce. Such a simplified

Figure 3.4

(*a*) A section cut view through the wall constructed with individual concrete block library parts and some steel rebar, modeled in ArchiCAD. This model can show a partially constructed wall. (*b*) Note that the partially constructed wall is not a section cut and looks like it could also be a surface model.



(*a*)

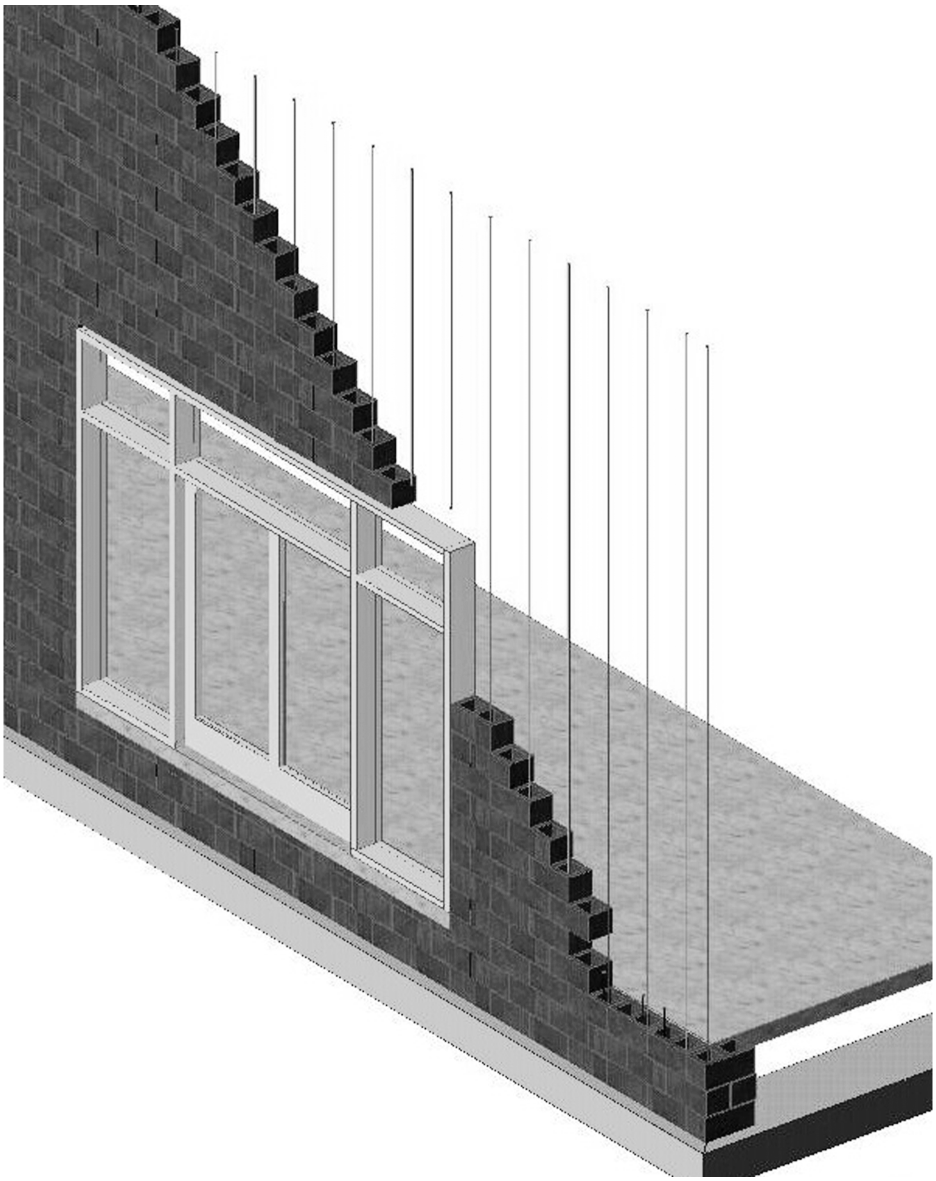


Figure 3.4
(*Continued*)

(*b*)

model would be usable, e.g., for developing floor-to-floor heights during design development, or to study the effects of alternative lateral bracing systems (at a schematic level) on the architectural development of a project. Once a structural system has been designed, a basic assembly of steel shapes can also be used to do coordination (clash analysis) with the various other building systems. Neither Revit nor Vico, however, (at this time) could develop its structural models to a greater level of detail that addresses fabrication requirements, or the design and analysis needs of the structural engineer. It also calls attention to the fact that if the structural engineer produces the initial design model in Tekla, it will then be useful in all other phases of the planning and construction process. The level of detail in the model can be developed based on the stage of the project and the available information. On the other hand, if such a model were produced in Revit or Constructor, it would not be able to be used for either structural analysis or fabrication purposes; and should the fabricator wish to use a 3D model, she or he will have to produce one of her or his own. This also holds true for the mechanical models of HVAC or piping systems. The idea of producing models from already completed designs by the consultants only makes sense when the consultants are unable to deliver such models and there is still value in conducting systems coordination exercises. It makes no sense at all for an engineer to make a model of a system that cannot be developed into a production model; it is simply the wrong software tool for such an application.

At this time it seems highly unlikely that a software tool that does not have the functionality to address both design and fabrication requirements should be recommended for use in a specialty field, particularly because there are very capable software tools on the market that do just that. It also does need to be noted that current standards of interoperability generally do not address the use of sophisticated production models for use in most other modeling software tools. The ability to analyze a "foreign" model is an extremely rare option, and the best that can be hoped for is the use of such a model in NavisWorks for viewing and clash detection against other models. The utility of an "abbreviated" tool such as Constructor or Revit MEP or structural is mainly for persons outside the specialty field, e.g., the architect or general contractor modeling the structural or mechanical systems strictly for his or her own purposes. While this may be the best alternative in the interim and not everyone on a project team may have modeling capabilities, it is clearly an inefficient way to address the overall project. In general it will be in the project team's best interest to have the specialty consultants and subcontractors model their areas of responsibility in their particular design to production ready software tools.

In the architectural area, it is the wide variety of materials, forms, objects, connections, and custom elements that makes modeling all this to a greater level of detail a large challenge. Curtain wall design appears to be the next area that may be evolving into a specialty area in relation to 3D modeling. However, a significant number of issues are part of the architecture of a project that are difficult to address in a model. This creates a different attitude toward the architectural model than toward the structural or mechanical models. In the mechanical and structural steel specialties there is a well-developed mechanism that takes a design into fabrication. There are almost no other areas in construction where this process has been developed to this extent due to the uniqueness of building projects and the variety in most other building parts. There is little opportunity to make an architectural model useful for fabrication purposes in general, and thus architectural models will generally show a much lower level of detail. There are some exceptions to this generalization; some general contractors are now trying to develop refinements for their self-performed work and analyze construction and assembly processes through the use of 3D models. This has parallels to the simulation of a manufacturing process to develop improved methods.

There are a number of sources for prefabricated 3D model components, many with a high level of detail. These are good labor savers in realistic models; some care has to be taken, however, to provide for artistic consistency in the model by not combining areas with high and low levels of abstraction and potentially creating confusion for the viewer. This phenomenon has similarities to 2D art and graphic design; an understanding of design principles and an aesthetic eye will help the delivery of a clear message with the 3D model.

Solid Model Organization

The *organization* of the model refers to the manner in which the model components (and connected information) are organized in the software file. Generally this organization will be related to the actual project, i.e., number of stories, number of wings, etc. The model organization is characteristic of the software, however; and it is the architecture of the software that will determine how a project model and its relevant information can be organized within the software file. Almost all software tools that create 2D and 3D objects and images allow the layering of the file contents. The layers are transparent sheets with names that contain various 2D and 3D content that can be shown or hidden, and locked or unlocked. Thus combinations of layers can provide selected views of the contents of the model that are relevant for any particular reason; e.g., showing only the structural layers will reveal views of the structural framing of a project, so when the architectural layers are also turned on, the model views will show both structure and architecture. The layers are also a means to select particular elements in groups. Generally layers can be edited, and content can be moved or removed from layers at any time. Use of the layer structure is one of the primary means of viewing and selecting/editing elements in a model. In general the layer structure will stay with the model contents when it is translated into another format for use in another software. It is thus advisable to choose an easy-to-learn naming system for the layers, so that people unfamiliar with the file can find their way around in it. A major difference between Revit and the other primary architectural modeling tools is the absence of layers in Revit. Elements are instead created as families and types and can also be selected as such. The project browser and the visibility graphics override determine what is shown or hidden, e.g., the electrical plan or the framing plan of the project.

Especially in the development phase of a project, the editing of both the components themselves and the attached information is continual. The components of the model therefore need to be well organized in layers (and/or stories, zones, etc.), so that at any given time they can be easily selected in the model. All components need to be organized so that multiple objects can be selected by type and edited in a single operation. At the same time the organization has to remain simple enough not to slow down the file manipulation, especially by others who will need to learn that organization before being able to manipulate the file freely.

When sequence studies are made, it is again necessary to be able to select groups of components to attach time-related information to them from the construction schedule. This operation can be tedious and needs to be properly prepared by an appropriate structure of the model parts.

This is similar to use of the model for cost analysis; specific cost information will have to be linked to each model component so that the quantitative information from the model part can be combined with the cost information from the database. A good model organization will simplify this task, and the model component structure has to be compatible with the cost database; e.g., the costs for material, labor, preparation work, etc., all have to be assigned to the specific appropriate model component in order to generate a realistic cost estimate.

Exercise projects are essential to develop a sense for a practical approach to model structure and object organization. This type of knowledge is best gained through hands-on experience. The planning discussions in Chap. 2 explain the need for preparing all the analytic activities for the BIM so that the models can be structured to best accommodate these tasks.

Solid Modeling Tool Characteristics

Most solid modeling tools have a lot in common. They all provide ways to manage building information. The nature of the specific information and the ways in which that information is connected to the model entities may differ a little depending on the software emphasis, e.g., whether it is an architecturally, structurally, or mechanically oriented design package. The user will create actual 3D components in the software and assemble them into a single model, of which multiple views are possible. The views do not exist independently—they are actual views of the single model entity; thus when a change is made in any of the views, it will actually be a change in the model, and therefore be reflected in *all* views of that model. Drawings become annotated views of the model, and the same holds true for changes made to the model; these are also directly manifested in all the drawings, thus ensuring an automatically updated project information database. Caution is required with the protocol regarding updating the various other forms of the project information, e.g., downloaded files, printed drawings, etc., which may or may not be the latest available version. When one is creating updates of files, it is best to choose a file name that will remain the same throughout a longer part of the (if not the entire) project. This has several advantages; first and foremost, when files are connected to one another, they find one another by name (and location), and thus it is very inconvenient when files are frequently renamed and all links have to be reestablished. Second, it is frequently unclear whether there exists a later version of a particular file that may be overlooked in that moment. It is the *archived* version of a file that needs to be identified as having a particular date or phase related to the project, but the working file should only be kept under its own—unchanging—name and should be the *only* working file for the project. This is a management challenge and a critical one to handle properly.

Most modeling tools will have a series of object tools to create the basic components that will comprise the bulk of the model for that particular functionality. The objects that cannot be directly created by these tools will need to be made up of several smaller components or imported as premade objects (library parts). Most software will come with such a parts library, and more and more manufacturers of construction materials are beginning to make "virtual catalog parts" available for the modeling profession. These library parts are often parametric and lend greater realism and accuracy to the

BIM; e.g., a piece of equipment will contain its connection and support points so that these details can be addressed accurately in the BIM. For example, door or window companies can create virtual models of their products that can then be inserted in the project model. This clearly will require interoperability since each modeler will have its own native file format, and to introduce an imported object will require a compatible format. There are also websites that sell model parts for specific software platforms; these premade library parts can make modeling easier and more realistic for the user. For a project with a specific need, it will be important to check on the availability of existing objects, as this could have a large effect on the required modeling effort. It is also important to verify the behavior of such objects and their ability to represent the actual needs of the BIM, as well as their ability to accept attached information for further analysis, such as cost estimates or sequence schedules.

Most solid modelers are also **parametric** modelers. This means that the model will contain parametric components that are characterized by programmable entities and may also have some intelligence in relation to one another. Both the elements created by the modeling tools such as walls, slabs, and roofs as well as the library parts, such as trusses, beams, and fixtures from a component library can be parametric objects. On the other hand, an object does not need to be parametric to be functional in a BIM. It is typically more convenient to model with parametric objects than to create unique objects for each element in a model. (However, even in a surface modeler, premade objects can be created to simplify multiple use of a particular model element.) For example, if a steel beam is a parametric component, it can be programmed to be whatever size it needs to be and can take on those characteristics in the 3D model; in addition, it may recognize the column to which its end is attaching and generate the appropriate connection (also programmed parametrically) in a semiautomatic fashion. Parametric components are powerful in their flexibility and ease of use; it is one of the primary means by which a BIM contains information and can function “intelligently.” Another example of parametric qualities is the ability of a structural steel frame in a Tekla model to be stretched and have all the components (columns, beams, braces, and connections) automatically adjust themselves to the change.

The modeling tools also have means to use already completed work for multiple projects. Entire component assemblies can be turned into editable library parts and used in other locations or projects. A whole building could be a single component in a campus containing many buildings, thus becoming a 3D referenced object that exists independently in a separate file (where it can be edited). When one is creating complex project models, it is well worth it to plan the assembly of the project file carefully to optimize the production effort.

MODEL PRODUCTION

The actual creation of the model can be accomplished in a variety of ways. The best method is to have a design model produced by someone intimately involved with the design process. This will ensure that the understanding developed through the making of the model will benefit the project team the most, by keeping the project learning within the project team. A schematic model can usually best be produced by a single person; but the amount of effort necessary to produce a more detailed (advanced) model usually necessitates more persons. It is of course possible to have the model produced by a "model shop." In this case a set of documents describing the project will have to be provided so that someone unfamiliar with the project can build the simulation. Producing such a set of drawings will require enough time and effort that a model could have been produced by a project team and provided many additional advantages as well.

It is clear that a designer will derive much less benefit from having a model produced by an outside consultant. The outsourcing of modeling is primarily used by construction companies that lack the necessary resources to produce models in-house. It will quickly become clear that a lot of communication is required to be able to create an accurate simulation, and that this may be expedited by having the persons with the greatest project knowledge guiding those who are creating the models. Many questions (about the project) only arise at the moment someone tries to model it, because accurate visualization is required before modeling can take place. The modeling process is parallel to the actual construction (fabrication) process, in that the simulation is a representation of the actual project, and the whole reason to create a simulation is to find all those instances that had not been anticipated (or fully understood). A general observation is that if some aspect of a project is difficult to model, it will likely also be difficult to construct (but the reverse is not necessarily true—the fact that it is easy to model may not imply that it will be easy to build).

A BIM is likely to consist of a variety of models produced by different persons, possibly with different levels of detail, and with diverse software tools in unlike formats. Examples of models that may be produced to become part of the BIM include the following:

- Site model (context—land, buildings, landscape, etc.)
- Architectural model (walls, floors, roof, circulation, special objects, etc.)
- Structural model (structural systems)
- MEP models (mechanical, electrical, plumbing)
- FP model (fire protection)
- Specialty models (equipment, finishes, temporary construction—scaffolding, formwork, trenching, etc.)

Not only do these models represent a wide variety of types of information, but also it is possible to combine models with different levels of development; e.g., a schematic site model may provide the context for a production level MEP model together with a less detailed architectural model. Due to the variety of this list it is helpful to establish some ground rules for all these various models. First and foremost, a compatible file format has to be established. *Interoperability is critical.* This means that those models that need to be combined in a particular software have to be provided in a compatible format.

NavisWorks saves the day in this regard; it seems to be able to see almost any 3D file format and combine it readily with all the other formats it can read. This does depend, however, on whether what needs to be accomplished can in fact be done in NavisWorks—and NavisWorks does *not* do everything. It is in all cases well advised to run a test file through all the potential combinations and software tools, to make sure that the modeling is not in vain.

The second important ground rule is to establish an official origin for the project so that all models will register properly in 3D space. A practical way to accomplish this is to create a registration file that contains a cube with one corner as the established origin for the BIM. If available, it may be helpful to include already established grid lines and floor levels to help the other modelers begin their work. This registration file will then become the starting point for everyone else's modeling effort.

The third important ground rule is for all models to be as accurate as realistically possible; guessing is permitted only when there is no other alternative, and when something is guessed, it needs to be flagged so that everyone else will know it and not rely on its being accurate. Many unnecessary difficulties are introduced by careless inaccuracies.

MODEL ANALYSIS

The advantages of having the project team produce the whole BIM are immense, because it facilitates the resolution of many problems that need to be addressed before the model is considered ready for analysis. Viewing the model will be a predominant activity, since communication is one of the primary purposes for the model. Special model viewing software has been developed that allows the import of models from a large variety of other software products. These viewers permit the combination of many models into one viewable whole. Some of these viewers also allow other analysis to be made or other information to be connected to the model components (such as time-related information).

There exists another group of tools that perform quantitative analysis of a model; some analyze cost for a project estimate, others analyze construction time for scheduling, or energy consumption, or natural and artificial lighting levels, etc.

Most modeling software can be purchased "bundled" with compatible analysis software. There are also a number of independent software developers who have developed software tools that can import various 3D models and perform analysis on them. It is essential to verify that *all model sources* on a given project will provide models that can be imported into the analysis tools scheduled to be used for the project.

Qualitative Analysis

Qualitative analysis considers the nature of the issues, often irrespective of the quantities associated with it. The list of such processes is primarily visual in nature.

- *Communication and marketing illustrations and movies*. This is strictly visual content intended to give others an impression of the project. Almost all modelers will address this function; both surface and solid modelers will generally create good 3D images from a model. See Figs. 3.5 and 3.6a and b.
- *Constructability analysis*. Constructability refers to a visualization of the methods necessary to construct (assemble) a project; it is an inspection for practicality and is intended to spot potential difficulties. A certain level of detail will be required from the model to be able to visualize issues of interest. This process can be implemented with either solid or surface models. See Fig. 3.7.
- *Systems coordination and clash detection*. This is probably the most popular application of BIM models at this time. The clash analysis will find objects in the 3D model that take up the same space; thus they "clash." This can be a duplicate of the same object or one object touching or running through another one. This can be done with either solid or surface models. See Fig. 3.8.
- *Energy*. Energy analysis borders on the quantitative end and generally will require a solid model due to the information that needs to be available about the materials used for construction. The nature, size, and location of *zone boundaries* need to be calculated to generate heat gain and loss for each zone in the model. The model may be able to visually show hot or cold spots in the project through the simulation of certain circumstances and conditions.

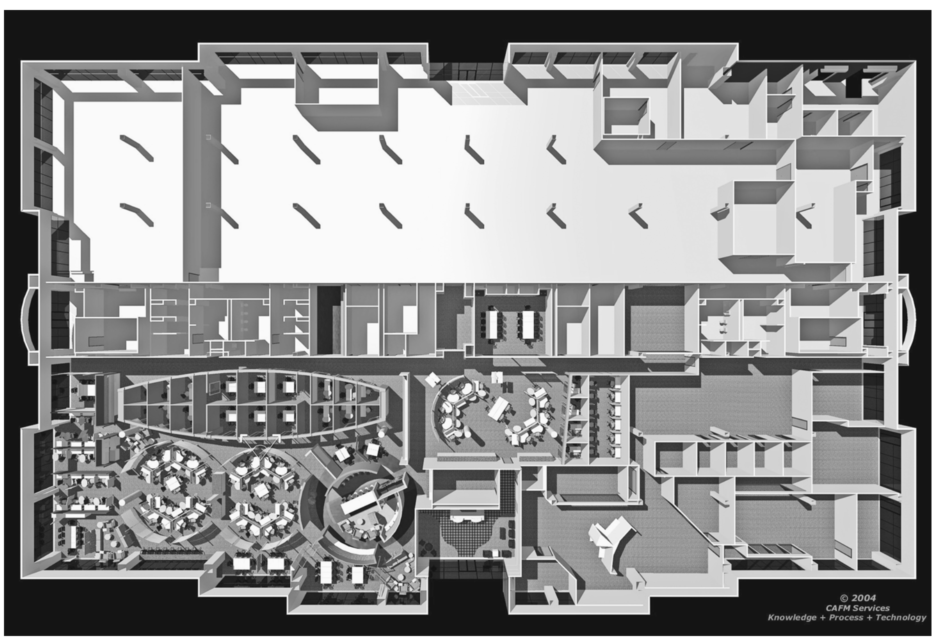


Figure 3.5
A 3D floor plan for communication purposes, modeled and rendered in MicroStation TriForma. (*Image courtesy of Design Village.*)

(*a*)

(*b*)

Figure 3.6
Interior renderings for communication purposes, modeled and rendered in MicroStation TriForma. (*Images courtesy of Design Village.*) See also color insert for image *a*.

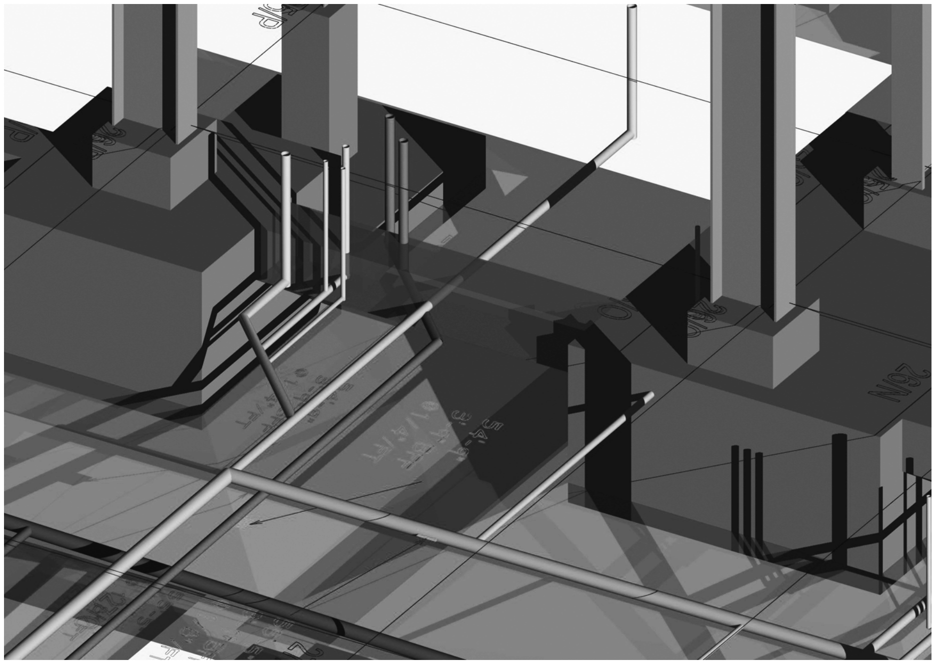


Figure 3.7
A detail of Fig. 2.6 showing constructability of underground utilities with respect to foundation location. (*Image courtesy of RQ Construction.*)

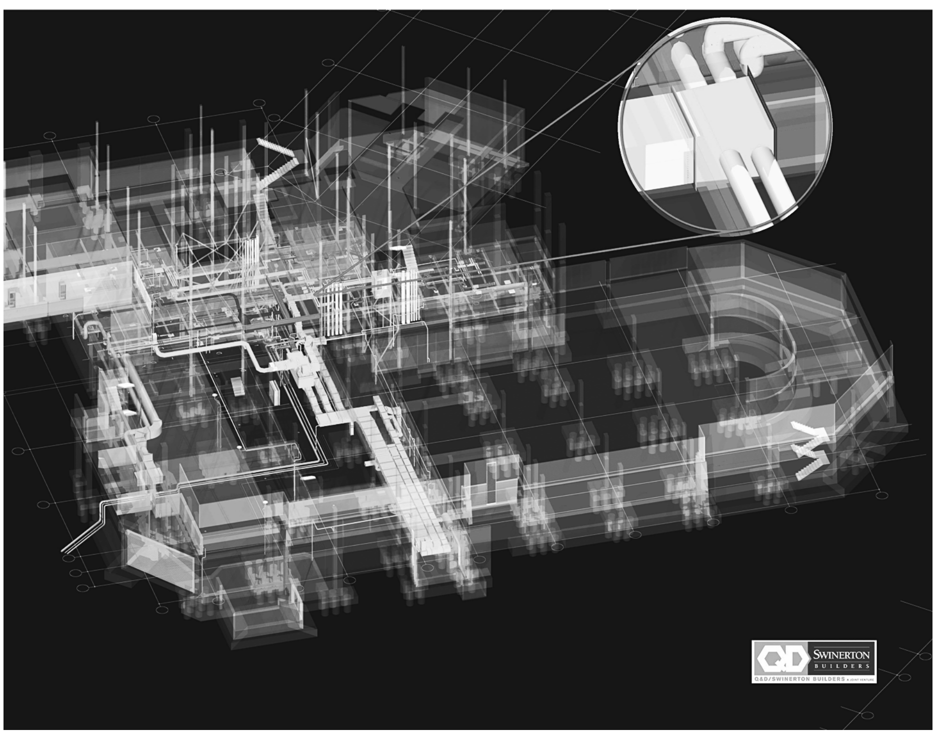


Figure 3.8
Coordination image of Ritz Carlton project in NavisWorks. Note the transparency of all architecture and structure for visibility of the MEP systems. (*Image courtesy of Q&D Construction and Swinerton Builders.*) See also color insert.

Sequential Analysis

Sequential analysis refers to studies that include time, both in duration and in sequence. These studies are very visual, although they also include a large amount of quantitative information, e.g., the amount of time required for specific tasks. The visual nature of the model provides a location-based element to the study; this will permit evaluation of issues such as work area definitions and crew interferences (and the rate of movement of a work area through the project).

- *Assembly and installation sequences*. Such sequences are visual interpretations of the model where components show up in time according to the construction schedule. This yields a movie showing the assembly of the project (or part thereof). Both solid and surface models can be used for this type of analysis.
- *Construction schedule and sequence*. This is primarily visual, although time is treated as a quantity. The construction sequence can be created with either solid or surface models. The link between the time information from the schedule and the model is typically one-way; changes in the model can generally not be sent back to update the schedule. Creating a schedule from the quantities in the model and linking it to productivity rates are particular to Vico Project Control at this time. This is a scheduling tool that has a bidirectional link with the model and is totally interactive; it requires a solid model produced by Constructor (Vico), or ArchiCAD (Graphisoft). Revit also advertises to have a bidirectional link to MS Project; the author has not yet verified this functionality.

Quantitative Analysis

Quantitative analysis involves measuring the amount of something in the project model and often combining it with other information. Most of the list is not visual in nature, but can often be represented in a spreadsheet or database format.

- *Quantity takeoff*. By virtue of the physical information inherent in the model components, the quantities of the various materials in the model can be extracted. This requires a solid model and interpretive information regarding the constituents of each model component.
- *Construction cost estimate*. The cost estimate is the product of the model quantities with the cost from a database. This also requires a solid model. Depending on the software, the nature of the link between the model and the database will vary.
- *Cash flow analysis*. Once a cost link has been established, the model can be used to track cash flow as construction progress is tracked in the model. This is a combination of the cost-estimating and the sequence-scheduling functions possible with a solid model.
- *Life cycle cost analysis*. This relatively new application of BIM is connected to cost control and energy consumption; and solid models are necessary. Both the operating cost (utility use, depreciation, etc.) of the project and the maintenance expenditures are tied to the model components to arrive at a life cycle forecast for the project.

There is clearly some overlap among these lists, and it is not intended to be exhaustive. This organization of the analysis functions helps to better consider them individually. Often the project team is interested in a particular analysis for the BIM; this list will identify some other related activities that may be relatively easy to accomplish because something similar is already being planned. It is a little like checking out what might be worthwhile to see in a city we are planning to visit for other reasons. Figs. 3.9 and 3.10 show the revealing nature of model images that may have been created for a variety of other purposes.

Figure 3.9 Presentation images of the Ritz Carlton project, modeled and rendered in Constructor. (*Images courtesy of Q&D Construction/ Swinerton Builders.*)

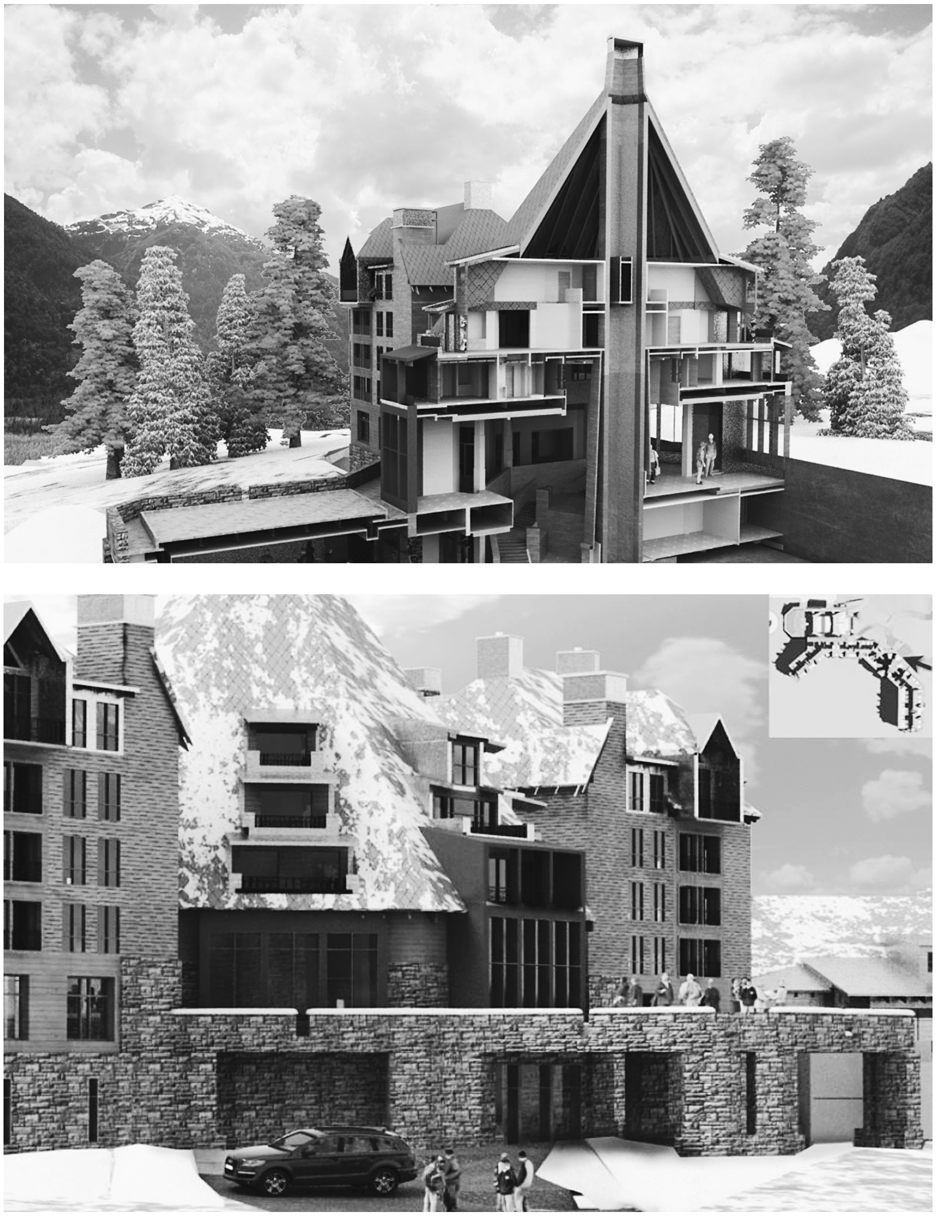


Figure 3.9
(*Continued*)

SPECIFIC SOFTWARE OPTIONS

Preparing for the Purchase

There are several major software developers that supply products with functionality in the BIM world; these represent the primary BIM tools for the construction industry. It is important to realize that there are many companies competing in this market and that all the various marketing claims will need to be substantiated carefully before purchasing decisions are made by the customer. Just as a buyer of an automobile would need to assess the need for a car to determine which model will suit that purpose, so will a software purchaser need to assess the requirements of the tasks carefully to choose among the various products addressing these needs. It is all too common that a specific product is purchased primarily on hearsay or assumed reputation, just so that the necessary research that would actually result in a more satisfactory purchase can be avoided. In an unfamiliar field it is

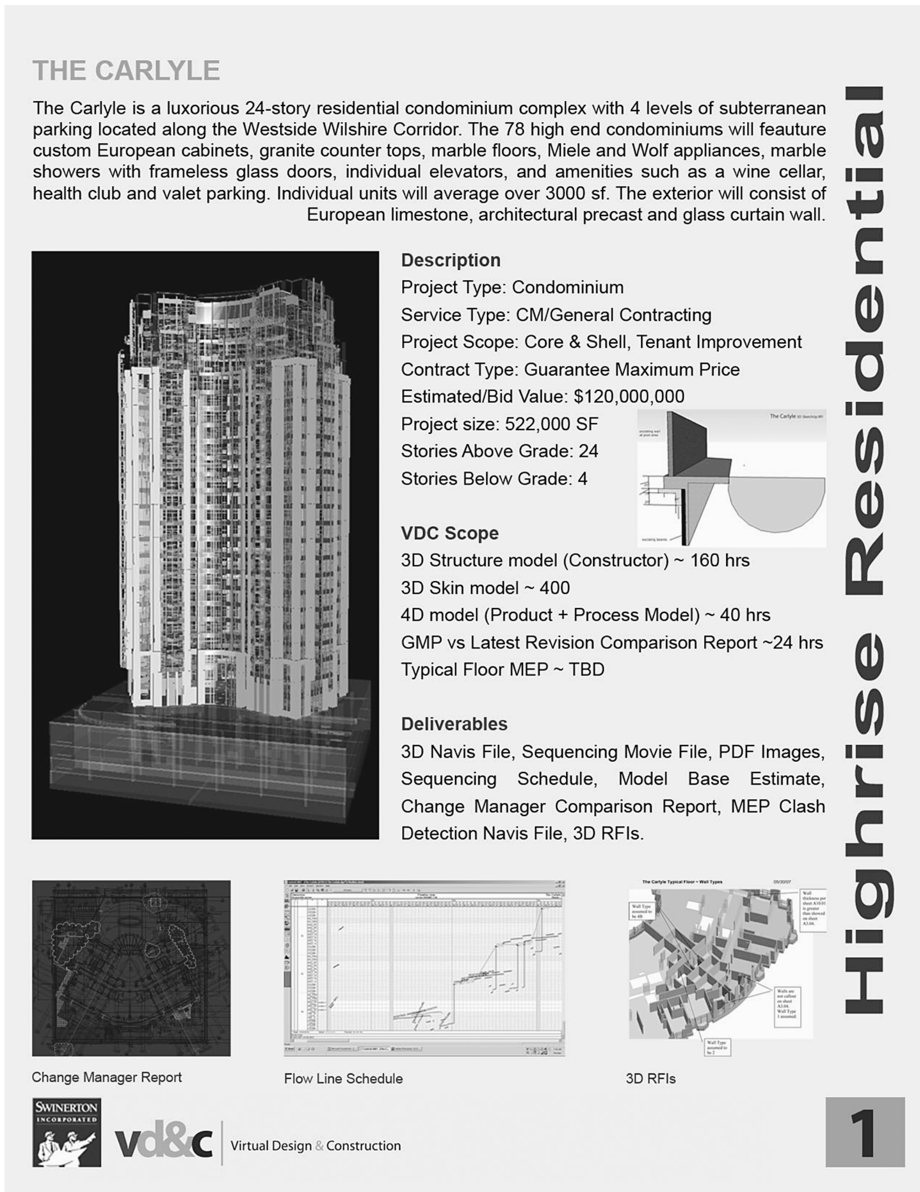
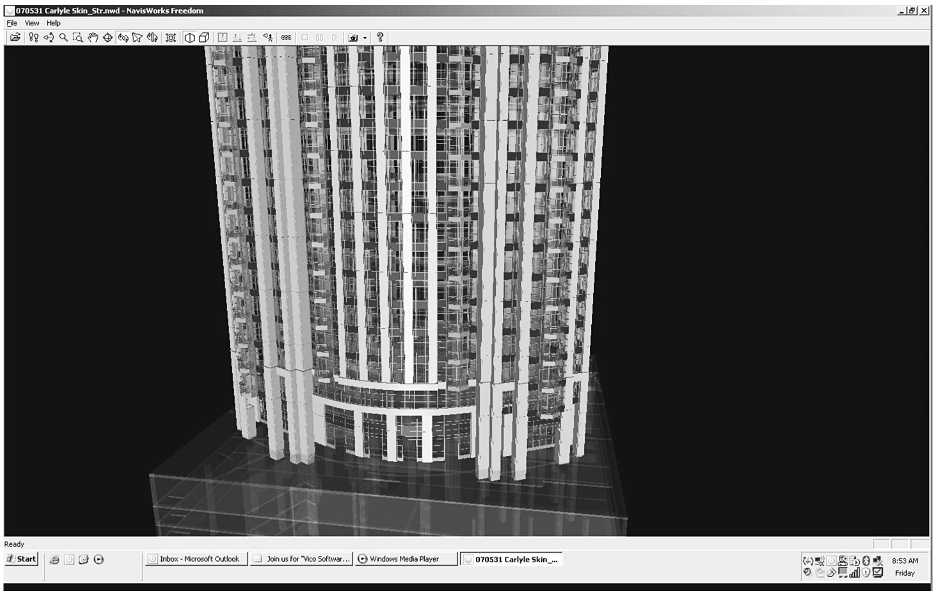


Figure 3.10 Presentation images of the The Carlyle project modeled in Constructor, rendered in NavisWorks. (*Images courtesy of Swinerton Builders.*)



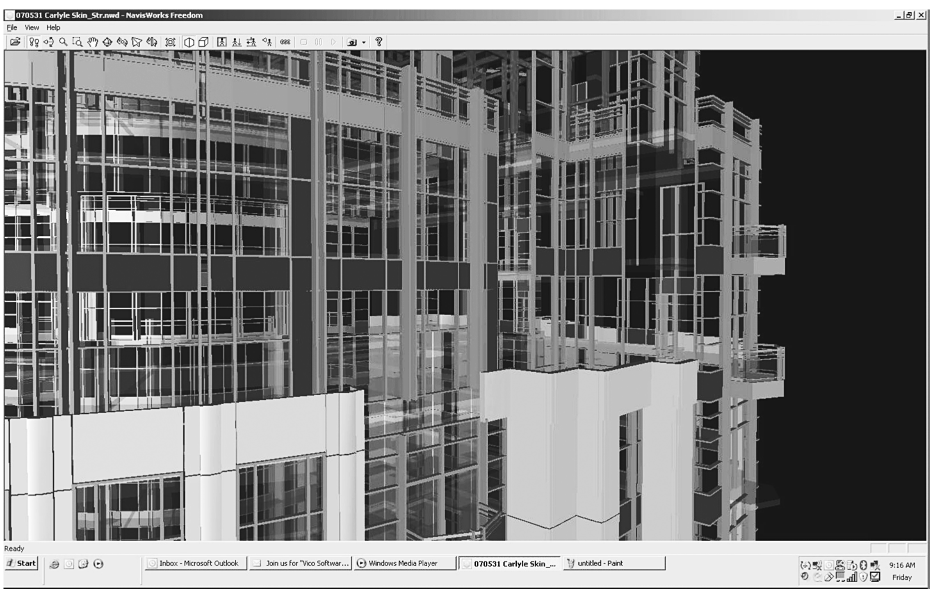


Figure 3.10 (*Continued*)

difficult to learn about a product and research its specifications; most potential buyers simply want the assurance that it will work, and they will try to just begin using it. This approach could easily lead to unanticipated problems, particularly in the area of BIM tools. With its relatively short history and complexity, building information modeling has created many unrealistic promises and expectations, on the parts of both the users and the developers of the software tools. Expecting to save money by upgrading an existing software license rather than investigating other potential tools could easily become one of the most expensive mistakes of embracing BIM technology. *The importance of a basic understanding of the concepts and applications of the BIM processes, to make intelligent choices among the available products, cannot be overstated.*

Once an understanding for BIM has been developed, answers to the following questions will help in choosing a particular product:

- What is the purpose of the software? Will it be used to make models (what will need to be modeled)? Will it be used to manage and view models? Will it be used to analyze models? What about the model is critical to its use, i.e., bill of quantities, visual 3D forms, central database, etc. (look at the list of benefits of BIM)?
- Who will be using the software? How will the software be learned? How long does it take to master the software? How often does the developer update the software? How much changes in each update? How long has the particular product been on the market, and how much has it changed over the last 10 years (not the company, but the software!)?
- What file formats are easily derived from the software? What is its native format? How can other file formats be imported (exported) from the software?
- Are there case study examples from other users who have done something close to what is required from the software? What exactly did it take to accomplish this? What were the major difficulties these users encountered? When there is a problem using the software, exactly how can it be resolved with the help of the software company (or other consultants)?

At this time there is no relationship in the author's mind between reliability and market share of the various software companies. Each product presented here is functional and needs to be evaluated on its own merits. Software vendors will frequently make unsubstantiated claims about their own or the competitors' products. The best advice is to question everything—ask for examples. If a vendor cannot specifically demonstrate something, it will warrant more research. (Many vendors are not particularly proficient in the use of the software they sell.) Most software companies will promise support and training; it behooves the buyer to check some references before making final decisions. The fact that everyone is using brand X means very little in the current BIM climate; interoperability is becoming a reality, and a product such as NavisWorks takes a lot of concern out of "being able to see someone else's model properly." It is useful to look at the history of the software company. A lot more can be learned about reliability of a product by looking at the track record than by listening to the sales pitches. Look at examples of the work produced by a tool, and make sure it is clear how that result was achieved exactly! Ask for demonstrations! Ask for a test drive! Do not buy from fear, buy from understanding! Committing to a particular software is like starting a relationship; it is difficult to just walk away from it, because it requires an investment that is far greater than the purchase price of the software boxes.

Software Descriptions

The following descriptions are just a sampling of the characteristics of some of the software companies and tools. It is by no means exhaustive either in relation to the companies producing software that may be used for building information modeling or in relation to the products available from any of the companies discussed in this section. It is also necessarily the case that even as this is being written, new functionality is continually being developed, and these descriptions of specific software characteristics will soon be out of date.

It is interesting to observe that in the marketing materials prepared by the software companies, it is generally the benefits of the BIM process that are being sold as if they were the advantages of the particular software. It is in fact difficult to distinguish among some of the features of these tools, and it is more important to understand the concepts of the processes, so that the user can recognize what the claims of the vendors actually apply to. Many of these claims in the description of a particular software tool can in fact be applied to almost any of the other software tools as well. This is like shopping for any item, e.g., a car; many of the differences among products may be very subtle. *It is thus the author's strong recommendation to test drive before committing to purchase. Put the software to a rigorous test, see if it actually does everything that is expected from it.*

NavisWorks

NavisWorks is the best place to begin the initial explorations into building information modeling. This is a viewer of models and has many useful applications in almost all phases of the use of the BIM. For anyone who has not been exposed to 3D models, it is a great place to begin to learn to view, navigate, and understand virtual environments. NavisWorks functions much as a video game, and since it is not a modeler, it also limits the severity and number of things that can go wrong in a BIM analysis.

The primary function of NavisWorks is to provide 3D model interoperability for the building design and construction field. Many different software tools are being used by many different disciplines that all produce 3D models in different file formats. Most of these tools do not import or export one another's native file formats, thus NavisWorks has provided a model viewer that can read almost any 3D file format. Not everyone will ever be using the same software for everything; thus the need for interoperability is fundamental to the successful implementation of the BIM process. A project team using BIM is faced with four major challenges that NavisWorks addresses: it can read different file types from various sources, it can import and handle huge files, it will combine different file types into the same file together successfully, and it facilitates graphical communications across the entire project team. At this time there is no other software that does this as well as NavisWorks, but there are rumors that several other companies are working on similar functionality.

One reason that NavisWorks can handle huge files and navigate through the virtual environments so effortlessly is that all models are translated into surface models. This necessarily removes some of the information (and most of the intelligence) from the original model, but that is generally not a particular problem. What is left is all the surface and spatial information, and that is enough to maintain all the visual data and perform sequence and clash analysis. The NavisWorks suite contains Roamer, which is the basic "engine" for NavisWorks and allows model combining and viewing. A free viewer called *Freedom* is also available; this can look at already prepared composite (or single) models in the correct file format. Freedom is for users who do not wish to analyze or manage projects, but simply wish to have visual access to the models. Special functionality can be added to Roamer with Clash Detective for coordination clash analysis, Publisher for providing files to be viewed with Freedom (the free viewer), Presenter for preparing high-end renderings of model views, and Time Liner for the creation of construction sequence analysis.

Clash Detective is the most popular of the functionalities of NavisWorks and the one that provides a quick return on investment. It is capable of finding and identifying all instances where model parts clash (take the same space in the model). This is invaluable for the coordination among building systems. The clashes not only are found and listed, but also can be managed through the same software until they are dismissed or resolved. Time Liner is very useful in providing a simulation of the construction (or installation) sequence of a project. By either importing a construction schedule from an outside software or building a new schedule in Time Liner, the 3D model components can be linked to a scheduled task, and thus can be seen appearing (or disappearing) in timed sequence. This is an excellent way to communicate construction progress visually. Autodesk purchased NavisWorks in 2007, and the entire construction community is watching what exactly will happen to NavisWorks now.

Many of the images in this book have been rendered in NavisWorks, and the two attractive images in Figs. 3.11 and 3.12 also attest to its capabilities in this area. There are of course software packages on the market that specialize in creating high-end

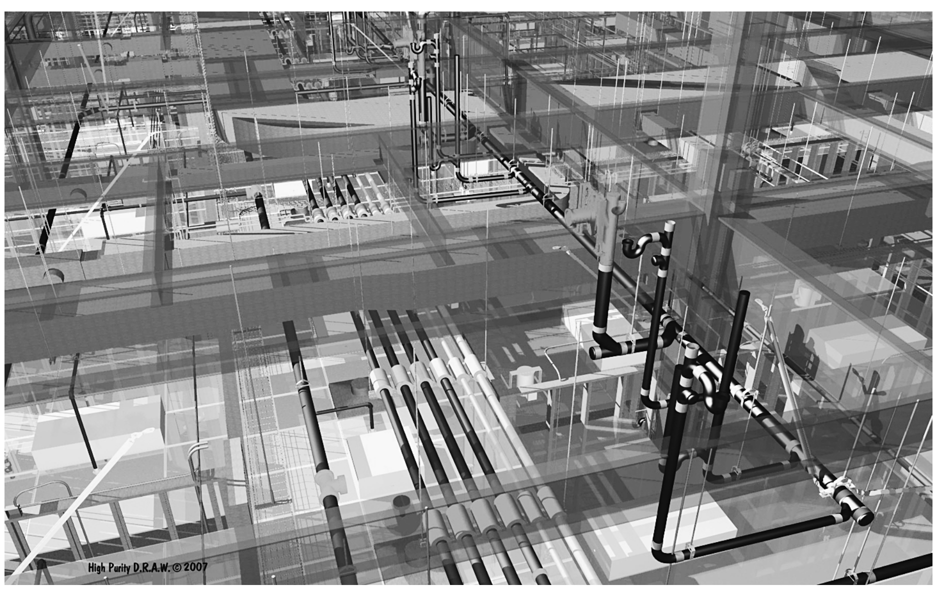


Figure 3.11
A sample of a detailed model "above the ceiling" in a medical facility, rendered in NavisWorks. (*Image courtesy of NavisWorks.*)

Figure 3.12
A sample of a model that has been rendered in NavisWorks JetStream Presenter. (*Image courtesy of NavisWorks.*) See also color insert.

rendered images from 3D models, but for its ease of use and high-quality output, many professionals whose purpose is related to the design and construction of the project (and not the rendering of it) seem to choose NavisWorks for the pretty images.

Google—SketchUp

The original SketchUp was developed by @Last Software and has swept the design industry by storm. It is almost irresistible, so simple, so powerful, so affordable. Now Google owns the software, and it appears to be a supportive marriage. SketchUp is a surface modeler; it is not trying to be a BIM tool, but it can be used as one anyway. Of course this means that its limitations have to be kept in (a "broader") mind. The limitations are primarily related to the type of information that can be contained in the model itself; that information is mostly related to size, location, and "look." It is not an object modeler, and thus it cannot be treated like one; the components only look like objects, but actually are just collections of surfaces (and can easily fall apart). The ways in which SketchUp *can* be a BIM tool lie in its phenomenal ability to quickly convey the essential information about a situation (mostly related to size, location, and look) into a 3D model. This model does not always need to be part of or be attached to a more complex BIM; it may simply be a communication tool for a specific issue. A SketchUp model can be imported into NavisWorks and seen together with any other model that may also be imported into the viewer. Once in NavisWorks, it is even possible to run a Clash Detection with the SketchUp model, or to use it in Time Liner; but again, its limitations have to be kept in mind, for it is not meant to be an information-rich modeler. The author encourages the reader to analyze the images in this book and study the uses for the various modeling tools to be able to develop a feel for the flexibility in this area. See Fig. 3.13.

Bentley

Bentley has produced software for design, fabrication, and construction for a long time. Presently the main product is called MicroStation TriForma, an extremely robust and stable 3D platform that addresses all the needs of the various disciplines required to develop and assemble construction projects. The company has a history of providing excellent support; and the well thought out software solutions leave the professional user little room for criticism. The only criticism that is sometimes heard is that the software is difficult to master, and that it requires a serious IT department to manage a

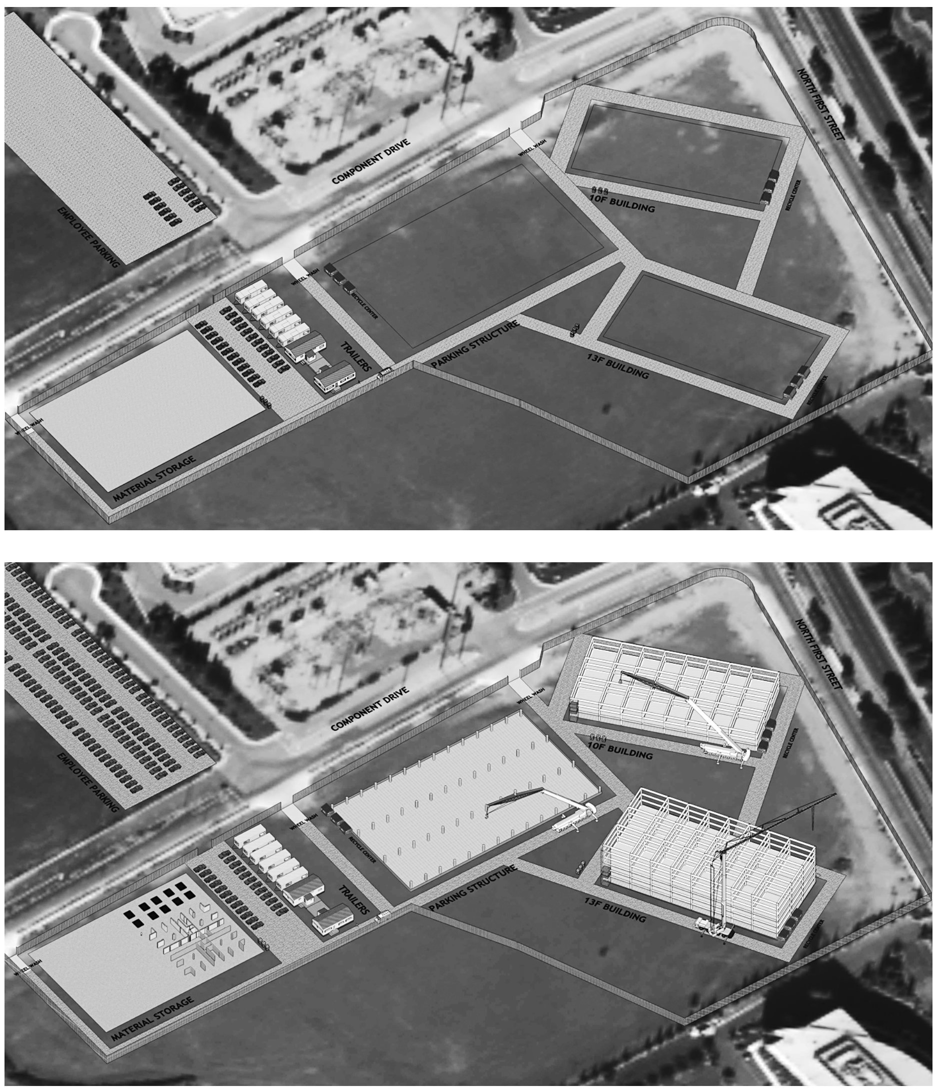


Figure 3.13 Bea Campus, San Jose, proposal modeled in SketchUp with Google Earth background. (*Images courtesy of Swinerton Builders.*)

Figure 3.13
(*Continued*)

network operating the Bentley's MicroStation software. For a professional, however, this is not a criticism, but a confirmation that the software is a robust, reliable high-end product. Many of the tools that are easier to learn and manage often are so at the expense of stability and functionality. Nevertheless, Bentley customers are generally large firms that build complex projects; thus for a smaller company it can appear too demanding to implement TriForma.

Bentley has thought through the BIM approach very carefully and recommends an evolutionary approach for its clients to fully transit to BIM (from the traditional 2D environment). Upgrades in the Bentley product line have never demanded large-scale changes or adaptations from the users; Bentley has developed its products and maintained stability and reliability throughout.

Bentley Building has chosen to address the fragmentation of the construction industry as the critical problem.[1] This fragmentation is experienced in the project teams that consist of disconnected people, in the construction processes that are fragmented into disconnected tasks, and in the fragmented tools evolved out of the disconnected construction disciplines. It is the cause of much wasted time, risked quality, and limited profitability and competitiveness. Bentley developed the "Build As One" motto, based on the use of a BIM as the hub for a collaborative approach to planning and construction. (See Benefits in Chap. 2.) Bentley also advises that starting over with a new, incompatible platform, as Autodesk suggests with Revit, is unnecessary to achieve the goals envisioned for the BIM approach to planning and construction.[2] The evolutionary path to software tool development serves the MicroStation TriForma users better, most have large investments in both training and software, and discontinuities in technology are undesirable. A second point on which Bentley disagrees with Autodesk is that of data management. Of the two possible approaches, with one being a **federated database** and the other being a **centralized database**, Bentley has chosen to develop the first option.

They found that developing a centralized database throughout all the phases of a project's life cycle is too risky to be reliable. Even though the centralized database is an attractive alternative for smaller projects, it quickly becomes unmanageable for larger, more complex ones where it ceases to be an option. When the various experiences of actual projects by different companies are taken into consideration, it becomes clear that the idea of a truly centralized database is fairly theoretical, it is a tempting idea, but always remains out of reach and cannot be implemented as would be expected; so far it remains intriguing but quite impractical.

Bentley focuses on supporting its products with a single comprehensive unchanging platform, on continually extending and improving its functionality, and on augmenting the software as necessary with collaborative products. It has evolved its CAD applications into BIM applications in a relatively seamless fashion. This will typically not be the experience of an Autodesk customer.

The database issue is a significant one and relates directly to planning. It is quite human to quickly get started and then solve the problems as they present themselves. The greater experience someone has in a particular area, the more planning will precede the actual start of the project and the more problems will be anticipated while there is still an opportunity to address them. The database problem is huge, and it behooves anyone contemplating a BIM approach to a project to carefully consider the potential difficulties of managing the project-related data throughout the life cycle of the project. *Experience in this particular area is worth a lot more than naïve enthusiasm.*

For examples see Figs. 2.14, 2.37, and 3.6 which are modeled in MicroStation TriForma.

Autodesk

Autodesk's main BIM product is Revit. This is probably the most widely used of the modeling tools; although it is also the youngest of the ones discussed in this book as well as the least mature. Autodesk's strength clearly is marketing; their market share with AutoCAD is enabling them to simply offer Revit as the next upgrade for their customers, hence its popularity. The fact that there is almost no continuity from previous attempts to address 3D modeling seems irrelevant—it shows a certain lack of consideration for the customer, which Autodesk can apparently afford. Many Autodesk customers are often simply unaware of any other software possibilities. Nevertheless, Revit is a serious BIM tool (it was already a modeler with good potential before Autodesk purchased it and when no one seemed to have heard of it), and the large user base will undoubtedly be very helpful in its further development. As mentioned in the Bentley description, Autodesk is billing Revit as a modeler with a centralized database; fortunately this is probably only wishful thinking at this stage. There is very little evidence that the data in a Revit model are any more centralized than those in a TriForma or Constructor model. When one is dealing with information in any BIM, it still needs to be managed wherever it resides, and simply having links to other locations does not centralize those data. Revit has very similar functionality to the other major solid modelers; in other words, the user can probably model just about anything in any of the software tools. There are various "bells and whistles" that may distinguish one modeler from another, but by and large the actual modeling experience of a seasoned user will not be that different from one software tool to the next. A greater concern is the ability to organize and manage the information that is collected in the BIM over the evolution of the project. Here also Revit is a serious contender, and even though it is particularly in the management of data that the modelers vary, all three are robust enough packages that it will take a lot of experience to understand which tool is the best for the specific projects and methods of a given design, management, or construction company.

Revit, e.g., does not use layers to organize its model components. It is unique in this feature, and the jury is still out on whether that is actually a feature or a shortcoming. On one hand, simpler is better, or is it? When a project becomes complex enough, it may just be an advantage to have additional means to sort its elements. It is useful to fully understand the nature of the tools used for these processes, so that solutions can be approached creatively and the characteristics of a specific tool do not become an obstacle.

Revit is able to link to MS Project and exchange scheduling information bidirectionally. See Fig. 3.14. It appears that the components in the model can be linked to multiple

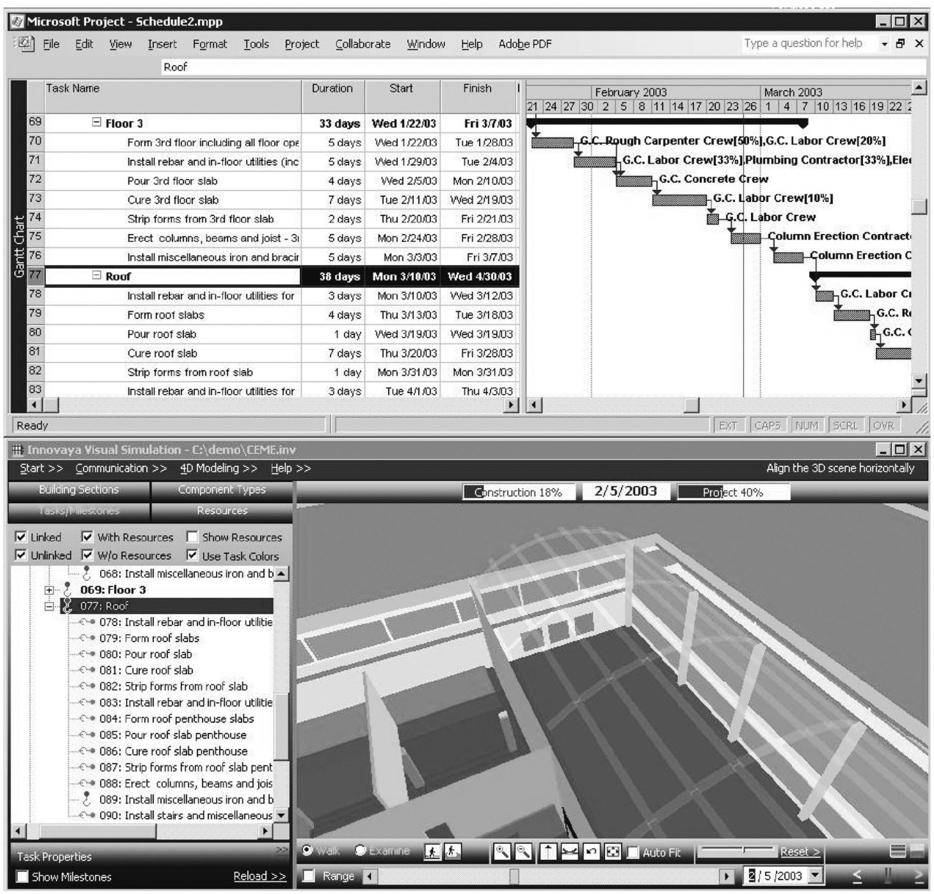
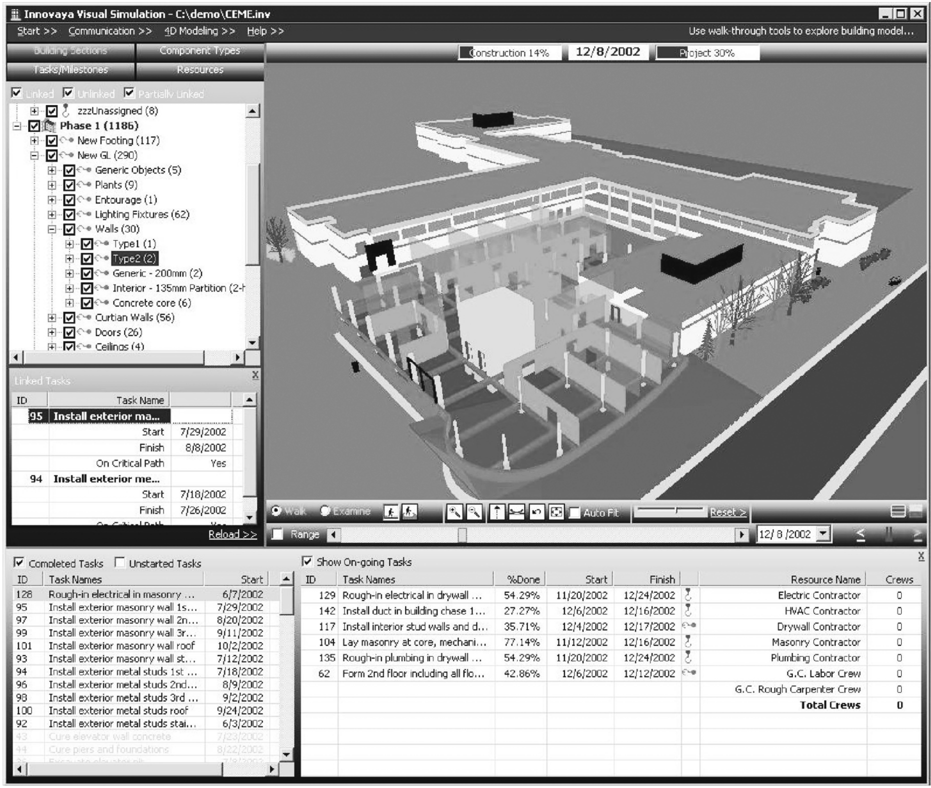


Figure 3.14 Linking between the Revit model and scheduling software. (*Images courtesy of Autodesk.*)

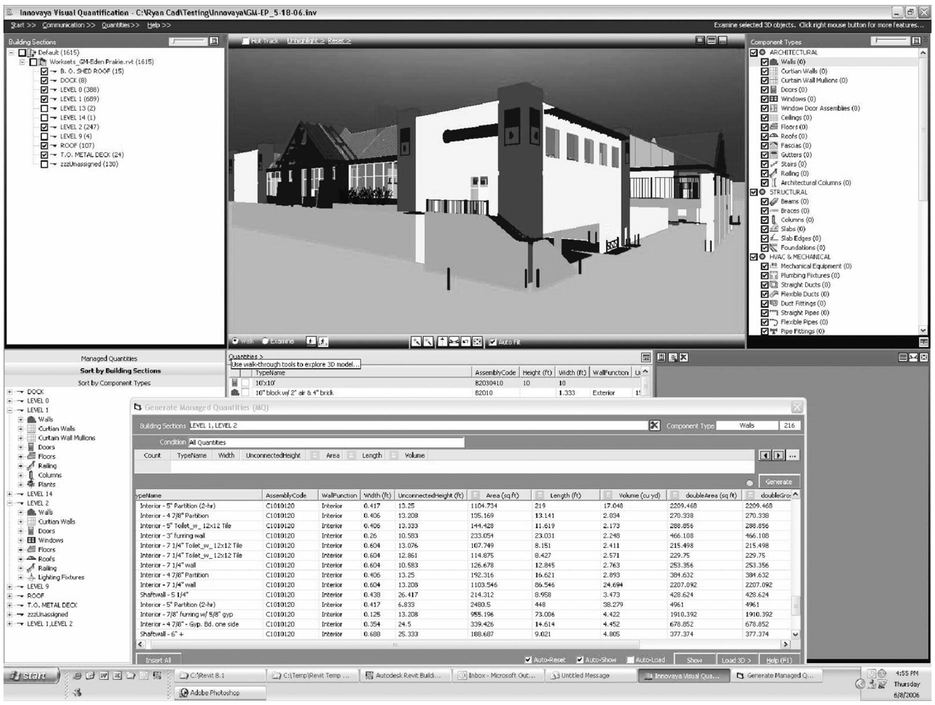
Figure 3.14 (*Continued*)



tasks in the schedule; e.g., a wall component may need to reflect the framing, the sheetrock installation, the door and window installation, and the finishing, all as tasks in a schedule.[3]

Revit also has the ability to export its model quantities to cost-estimating software. See Fig. 3.15. Due to the nature of the model, the quantities can be very accurate and thus reflect the status of the project design reliably. It appears that the connection is a one-way transfer of quantitative information and that all interpretations of this information are made in the cost-estimating software. Some questions are arising from users about the functionality of this link to estimating at this time.

Autodesk has also augmented its modeler with Revit Structures and Revit MEP. Both of these modules are designed to create specialty components to address the representation of the components for these disciplines. Whether the engineering design community will adopt these tools professionally remains to be seen. Revit Structures is not anything like Tekla, and there are any number of MEP modelers that will take these systems from the design through the fabrication phase. At this time the Revit modules for structural steel and MEP systems seem to be most attractive to project teams, where these systems models are not provided by the design consultant or the fabricator, and cannot be well coordinated without a 3D model (clearly a very justifiable use).



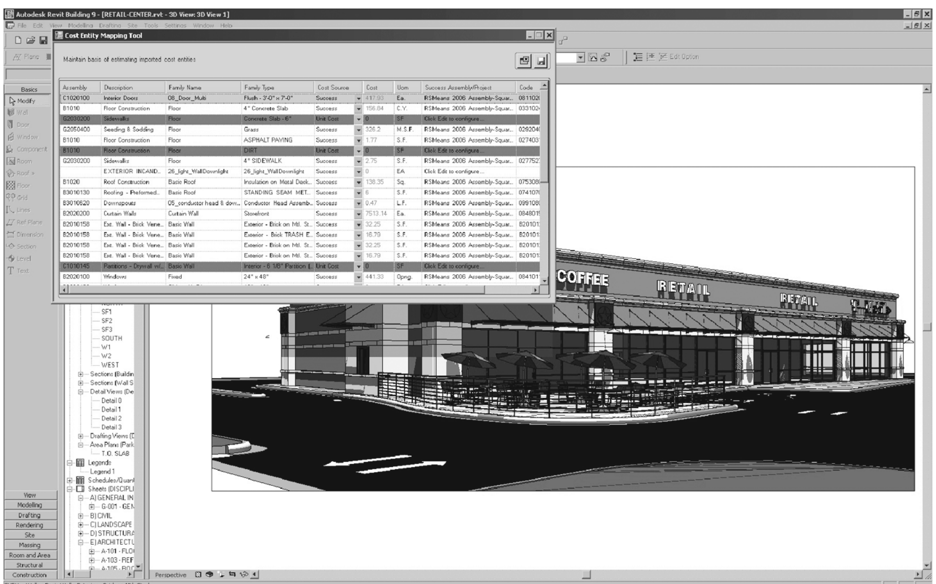


Figure 3.15
Linking between the Revit model and estimating software. (*Images courtesy of Autodesk.*)

In summary, Revit is a young, but potentially powerful tool for the planning and management of construction projects. Only time will show whether Autodesk can develop Revit to keep pace with the demands of the design and construction industry.

The images in Fig. 3.16 show a Revit model addressing the GSA BIM requirement. As of 2007 the Government Services Administration (GSA) is requiring a BIM on all new projects. This requirement consists of an area analysis to facilitate space management in the planning phase and after construction of the project. This is an excellent way to introduce a complex concept in a simple fashion to the real estate industry. The GSA is the largest "real property" owner in the world, thus space management is a top priority for this organization.

Vico

Vico is a new company, but the engine behind its Constructor is ArchiCAD. In 2007 Graphisoft sold ArchiCAD to a German software developer and Constructor, its construction industry software suite, spun off to Vico Software, a newly formed company with the design and construction industry as its primary focus. The suite consists of the modeling engine ArchiCAD, which has been a professional solid modeler since the mid-1980s, and several modules that facilitate construction project management; Estimator, which is a cost database, a Line of Balance scheduling software called *Project Control,* and 5D Presenter which facilitates project presentations, all with a bidirectional link to each other and the model. As mentioned in the previous descriptions, the modeler is very similar to the other software modelers—test drive it! As a

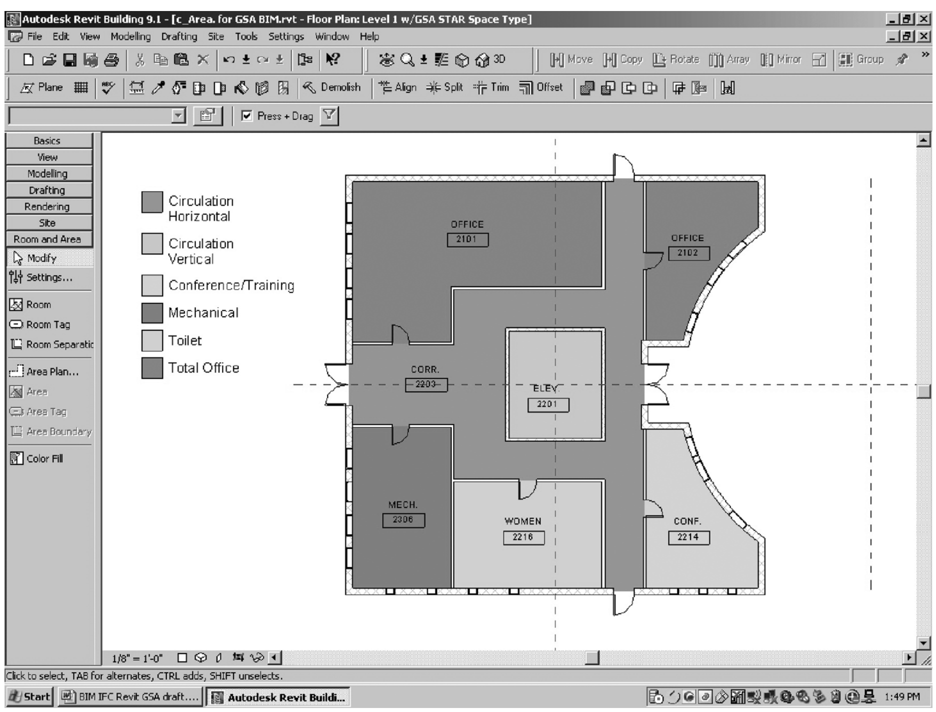


Figure 3.16 A space (floor area) analysis in the Revit model (format based on the GSA mandate). (*Images courtesy of Autodesk.*)

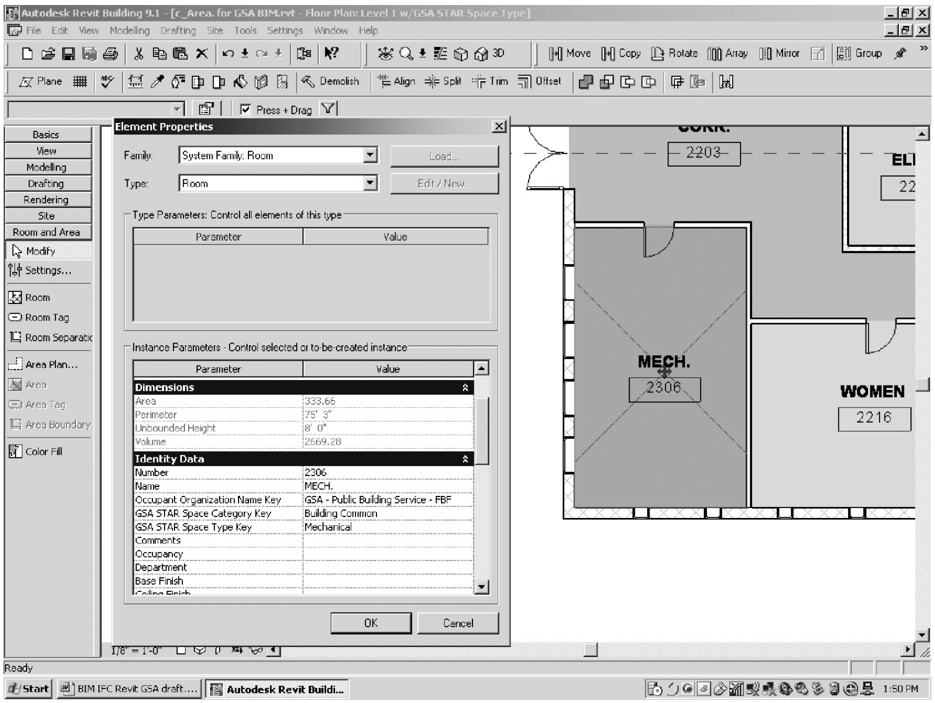


Figure 3.16 (*Continued*)

modeler ArchiCAD and Constructor are equal with the exception of some specific functionalities and object libraries that focus on either the architect's practice or the construction industry. The modeling is simple and straightforward to learn; the file structure is based on layers and stories that contain all the objects either created by the modeling tools or imported from object libraries. This is similar to most other modeling software. The Navigator is very effective in taking the user to any view of the model that has been saved (this applies to both 2D and 3D) and to any layer combination, etc. Since the software has had many years of development, it is very refined and has not lost the intuitiveness it was known for in the early 1990s, when it was one of the few professional 3D modelers in the marketplace (that still exists). It is easy to create custom objects with the modeling tools and save them as library parts. It is also not difficult (relatively) to create objects by writing code in GDL, which is not approachable in most other software tools. When one is making objects with modeling tools, they cannot easily become parametric objects; with GDL, however, there are almost no limits to the intelligence that can be written into the code of the objects.

The other components of the Constructor suite[4] are, however, the most interesting for their usefulness to construction project planning and management. Cost calculations can be created from the model by attaching a link from a recipe to a model part. A recipe is a description of the materials, labor, and resources that are required for a specific building component; it is part of the cost database. See Fig. 3.17. The object then provides the quantities that result in the cost calculation of that component. A recipe consists of one or more methods (project tasks like placing concrete, building formwork, etc.),

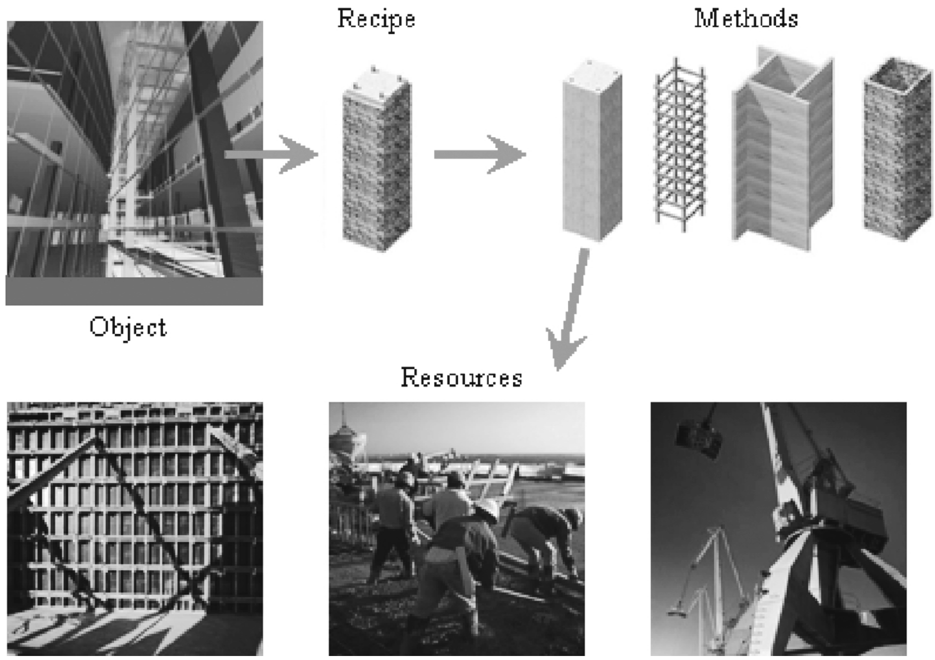


Figure 3.17 The structure of a recipe. (*Image courtesy of Vico.*)

and a method consists of one or more resources (materials, labor, etc.). This gives recipes a flexibility that allows them to be used for almost any application, from representing a floor area with an overall square foot cost, to the analysis of a concrete element with all the details of formwork, reinforcing, finishing, etc. Such flexibility also permits the cost analysis of a project at any stage, thus facilitating the tracking of the cost changes as the design evolves. The refinement of both the recipes and the model parts over the course of the project's development provides a good cost management tool. Even though the management of the recipes can be tedious, it works well and once it is set up, it will remain functional for various projects where similar recipes can be used.

The scheduling tool in Constructor is called *Project-Control,* it is a Line of Balance Scheduler. A *Line of Balance* is a line that represents a task in a project, and its slope indicates the productivity rate of the task. The line thus shows both time and quantity of work as well as the location where the work takes place see Fig. 3.18. It is far more visual than a conventional Gantt chart, where the same information is not graphically represented, but buried in a dialog box. The Empire State Building was built in 13 months using this type of scheduling. Control connects the quantitative information from the model through the recipes in Estimator to the lines of balance in Control, both the quantity of materials and the time required to complete the task. Control then plots the lines and permits the manipulation of these lines to edit the time element of the tasks (not the material quantities, they can only be changed in the model); thus when a task starts and how long it takes can be adjusted until the whole project has been optimized

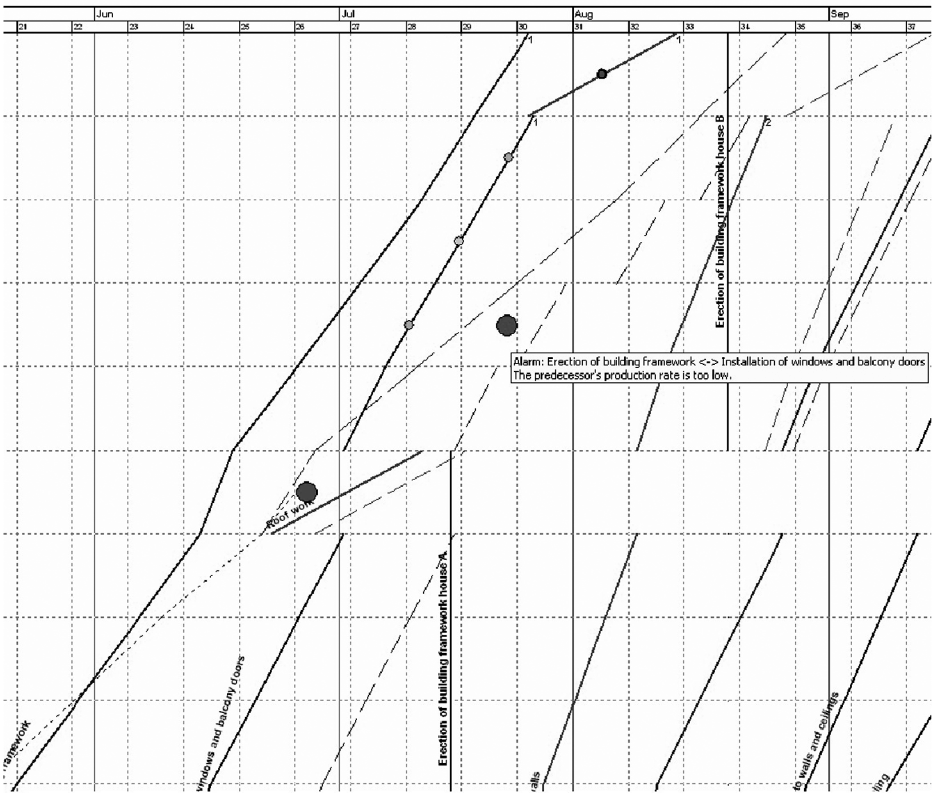


Figure 3.18 Line of Balance Schedule. (*Image courtesy of Vico.*)

in relation to productivity of crews and work locations in the project. Design updates are synchronized with the schedule by reimporting the model data from Estimator. The schedule can then be taken to the 5D viewer that will simulate it in a construction sequence.

Vico also describes its Design to Build and Build to Design ideas for the 3D model.[5] Design to Build represents building a model of the proposed project that reflects the actual construction techniques, including the actual tolerances of the objects in the model. This will facilitate proper coordination of the trades without incurring undue attention to unnecessary detail and accuracy. Build to Design then refers to the key model points being compared to the actual dimensions in the project by laser survey (or other means) to check the design tolerance compliance and ensure that prefabricated elements will fit.

The entire process of creating a 4D simulation can be described in steps.[6] The first step is represented in Fig. 3.19, the connection of the geometric model data to the recipe describing its methods and resources. The next step is the semiautomatic generation of a cost estimate and a schedule for the project based on the links between the information in the recipe and the model components, as shown in Fig. 3.20.

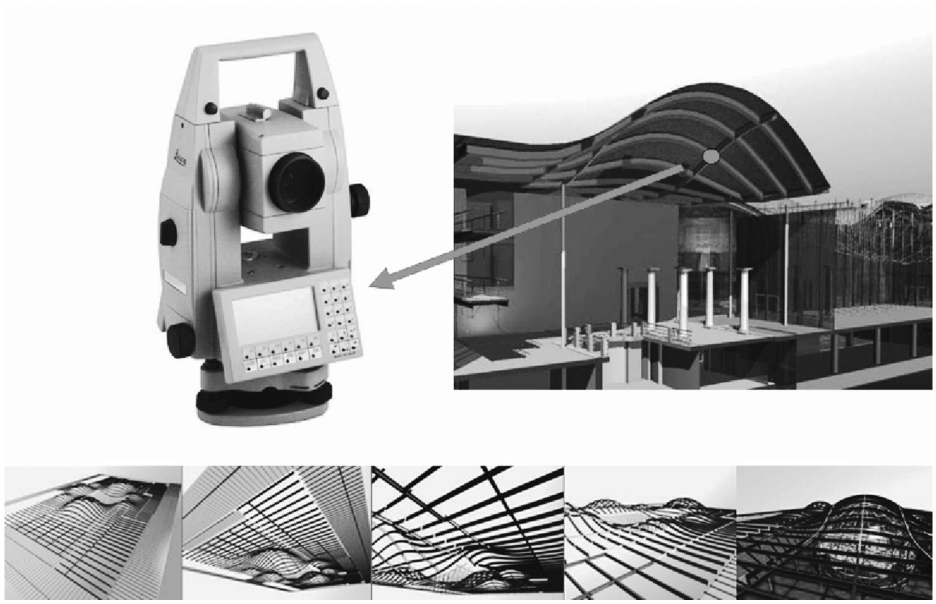


Figure 3.19
Build to Design. (*Image courtesy of Vico.*)

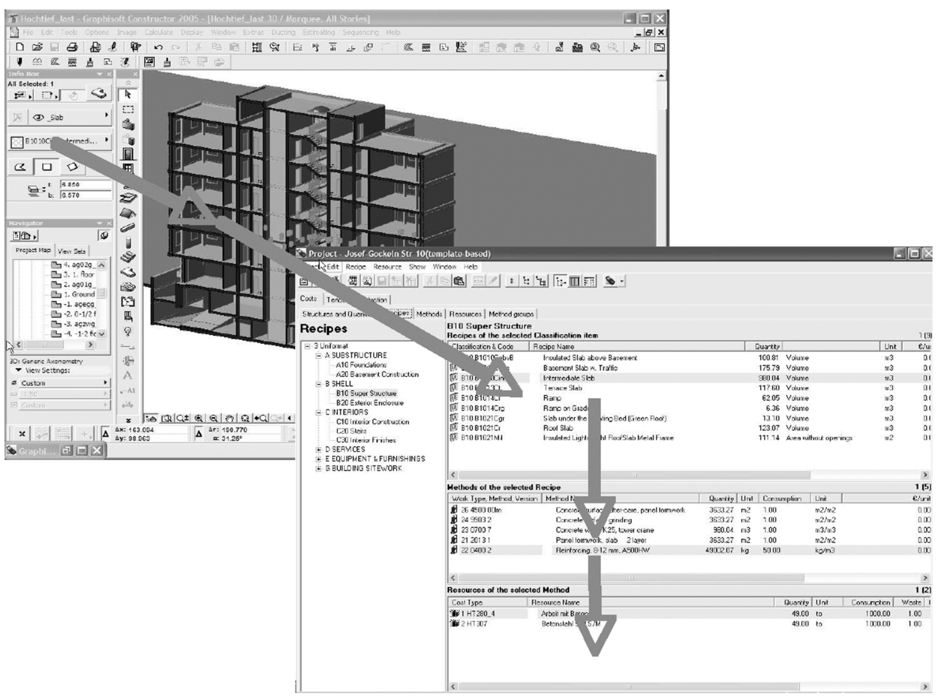


Figure 3.20
The 3D model screen shows the recipe connected to the selected objects; the imported model data in Estimator (right) shows recipes on top, methods in the middle, and resources below. (*Image courtesy of Vico.*)

When design, cost, or time is changed, the other two are automatically updated. Few tools have this level of sophistication in their links.

Since the tasks of a construction schedule typically contain work that is to be done to multiple objects, i.e., a certain number of columns on a floor, etc., it is necessary to create zones to describe the locations in which the tasks are to be completed. This is step 3 in the process. See Fig. 3.21.

Once the zones have been described in the model by a line, the tasks and their respective recipes are automatically assigned to the proper zone. This now results in a *work breakdown structure* (WBS) that reflects the quantitative information of the project by zone or location, and this in turn is the basis for the *Line of Balance* schedule. Quantities from the methods and resources are the input for the calculations of task durations. The schedule results in activity durations, set by location (zone), assigned to the work breakdown structure. All 3D elements in the WBS have task connections, and the resulting 4D model can now be used for schedule simulation and analysis.

The different trades can be organized into task groups and identified by color in the 4D model, so that the activity can be visualized in the simulation. This level of visualization provides many possibilities to refine the construction process planning.

Tekla

Tekla is a Finnish software developer who addresses structural steel, steel reinforcing in concrete, and precast concrete modeling. The software is capable of taking the

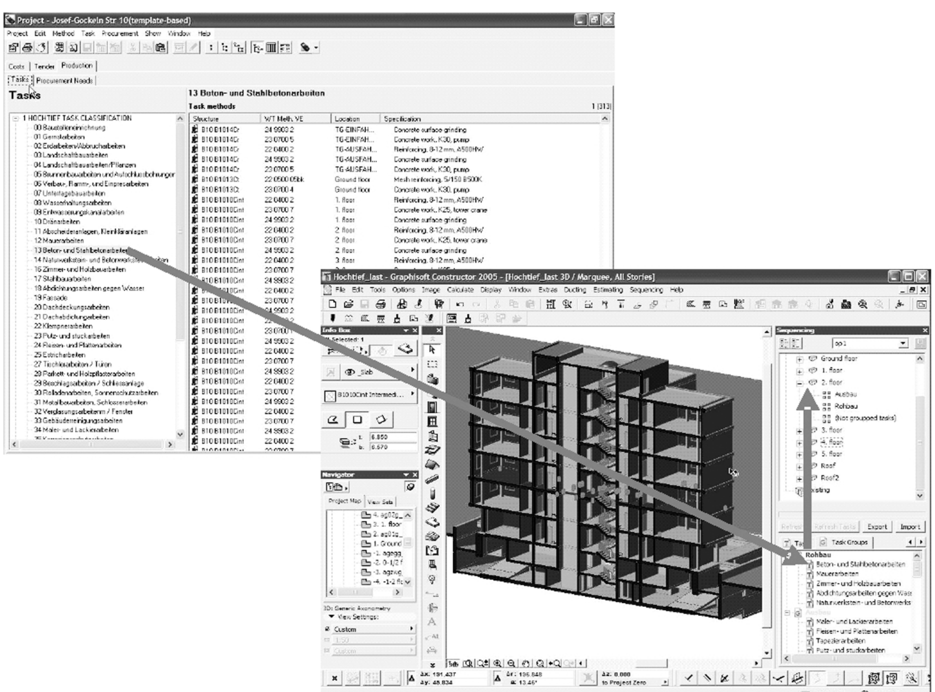


Figure 3.21
Tasks are mapped to objects, which belong to a construction zone, defined in the 3D model. (*Image courtesy of Vico.*)

project from the design phase, through detailing, into production and assembly. The engineer can create the model that begins the design and structural analysis process and pass it on for use in fabrication and installation. The software also has the ability to model reinforcing steel and precast concrete components for concrete construction. Tekla is a very focused tool, applied to a well-defined discipline; consequently it performs extremely well. Tekla models can of course be combined with models created in other software packages in viewers such as NavisWorks, so that the Tekla model can be used for construction coordination.

The modeling in Tekla is parametric; this means that the components of the model can be customized and edited at any time to suit the requirements of the project. This facilitates the model building or editing, and together with its graphical interface, Tekla is intuitive and not difficult to learn. There are three modules that can be added to Tekla Structures (the standard design configuration of Tekla), namely, steel detailing, precast detailing, and reinforced concrete detailing. Any and all drawings that are to be produced from the intelligent model will automatically represent the current version of the model and not require manual updating. Any of the building components can be modeled, from the steel connections and fasteners, to the railing, stairs, and trusses; if the available library components do not match the desired objects closely enough, then custom objects can be created from simpler elemental parts and saved as library parts also.

An advanced graphical input interface makes Tekla an excellent modeler and model manager, as well as making it very effective for navigation through the model and its various views in model or drawing format. The construction schedule can be simulated visually by the model and connected to both the location and time quantities of the model components. See Fig. 3.22.

The Tekla model file also includes interfaces to the most typical file formats and permits links to some of the other modeling tools for the transfer and integration of model information. The design engineer can begin the analysis of a steel structure by assigning the loading conditions to the 3D model; the analysis will result in the design of specific members based on the assigned loading; and when the loading or the model configuration changes, the analysis will automatically update the required structural calculations for the members. See Figs. 3.23 and 3.24. Connections are

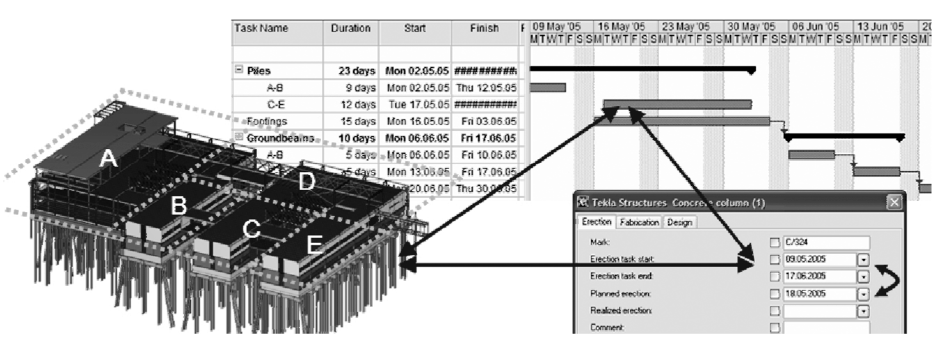


Figure 3.22 Construction schedule to model connection in Tekla. (*Image courtesy of Tekla.*)

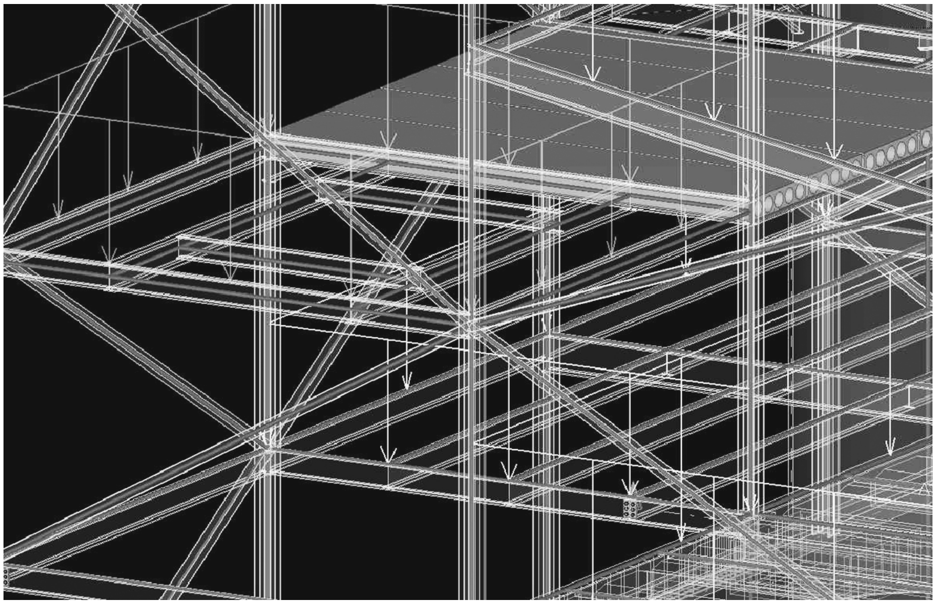


Figure 3.23
Loading assignments in structural model. (*Image courtesy of Tekla.*)

(semi) automatically created by parametric design specifications and will update themselves with layout (configuration) changes.

This same model can be developed into the fabrication model for the project, and then continue its service during the erection process. The model use for fabrication is referred to as Direct Digital Exchange. Either the design or the fabrication model will

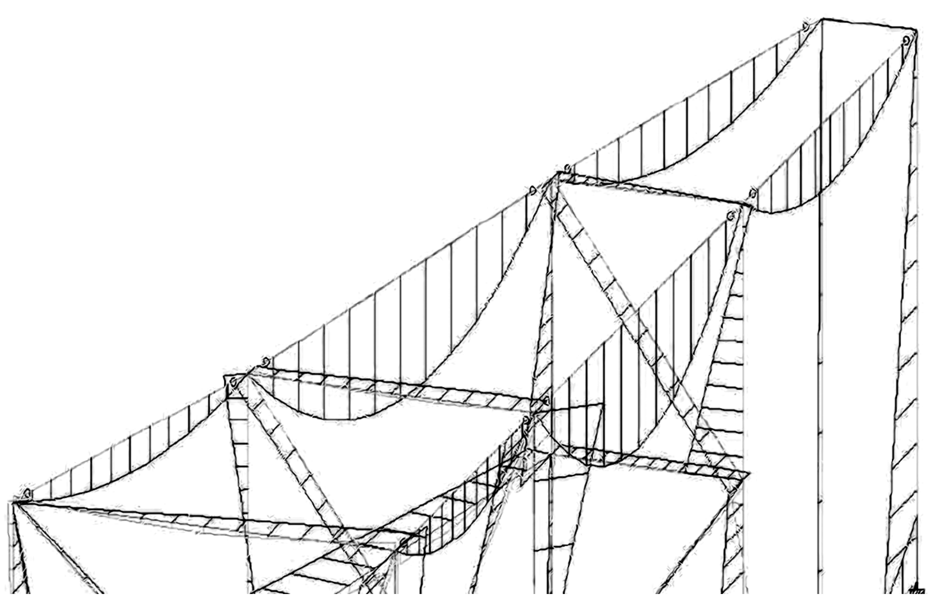
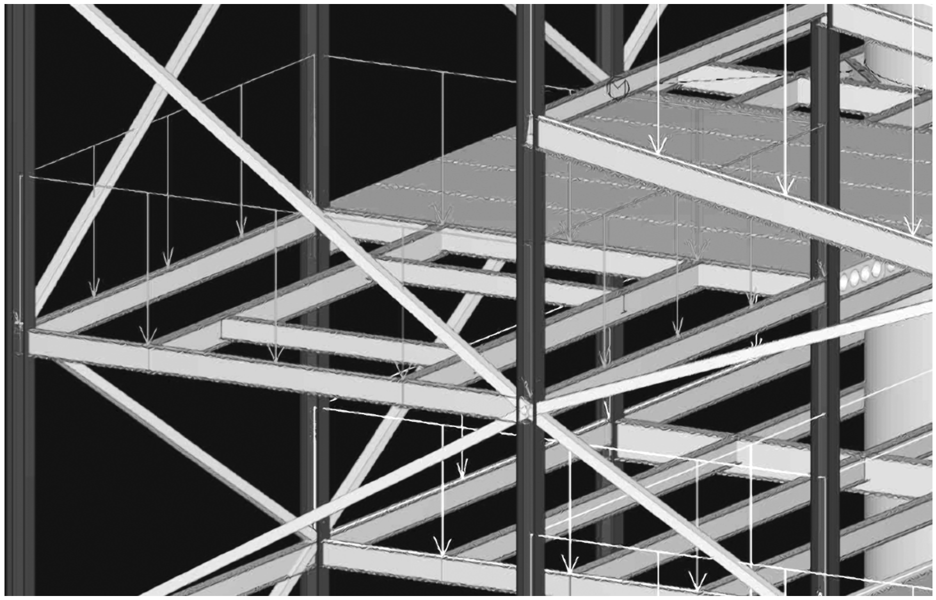


Figure 3.24
Structural analysis in alternative views. (*Image courtesy of Tekla.*)

Figure 3.24 (*Continued*)



be able to function in the systems coordination process as well. Tekla's strength is software design and customer support. The software suite is soon to be expanded with various construction management modules.

REFERENCES

1. "Build as One" by Huw W. Roberts, AIA, CSI, Global Marketing Director, Bentley Building.
2. "Does the Building Industry Really Need to Start Over?", by Keith Bentley, CTO, and Brad Workman, AIA, VP, Building Vertical.
3. BIM and Project Planning, by Autodesk.
4. "Virtual Construction Benefits," by Dominic Gallello, CEO Graphisoft.
5. Ibid.
6. "Integrating Product Modeling with the Line of Balance Scheduling Technique in Virtual Construction," by Clay Freeman and Olli Seppanen of Vico.
7. The websites of the software companies may be visited for more information:
   NavisWorks—www.navisworks.com
   SketchUp—www.sketchup.google.com
   Bentley—www.bentley.com
   Autodesk—www.autodesk.com/revitarchitecturesuite
   Vico—www.vicosoftware.com
   Tekla—www.tekla.com