FUNCTIONAL MODELING IN PARAMETRIC CAD SYSTEMS

CHARLES EASTMAN[1], RAFAEL SACKS[2] AND GHANG LEE[1]
[1] *College of Architecture, Georgia Institute of Technology*
[2]*Civil and Environmental Engineering, Technion, Israel Institute of Technology*

Abstract. We review the transition now being undertaken from 2D drafting to 3D parametric modeling and the development of Building Information Modeling (BIM). We examine the functionality of BIM platforms, using an example from precast concrete structures. We formulate the development of parametric modeling of building systems from the context of SFB modeling and characterize most parametric modeling as function-form mapping, with optional behavior-driven detailing. We examine the implications of this kind of technology for future building software development and the future delivery of design expertise in building.

1. Introduction

After twenty years of using 2D drafting systems, architects and other parties in the building industry appear to be making a move to production-oriented 3D parametric modeling. While it is still impractical to use 3D modeling for generating contract-level information for more than small-scale projects, there are serious efforts to make 3D modeling of buildings at construction levels of detail a practicality for all scales of building. This goes far beyond the post facto rendering of a design, or the development of a building model as a contract deliverable, after the building has been designed. It involves a revolutionary change in how designs are generated, how the information about a building is represented and how that information is later used in building operations. This step is partly the simple maturation of the industry, ready to take “the next step”. The recognition that major building productivity improvements require machine readable design data (see various articles in *Automation in Construction*, 1992-), such as that needed for design automation and fabrication based on numerical control, and that

3D modeling allows generation of new of architectural forms (Gehry et al, 2003), have also offered motivation to make this unparalleled step. At the same time, the maturation of parametric modeling in mechanical design and advances in hardware have created an opportunity to adapt this technology (with major changes) to building. The major CAD companies seem willing to invest in the education of the industry, as well as development of the technology necessary for this to be achieved. Needing an acronym to go with this change of technology, the name Building Information Modeling (BIM) has become accepted, and we use that term also. Throughout this paper, we will be talking in terms of the building industry. The impact of these changes must be broader than to architecture alone, in that they affect all roles within construction; architects, engineers, contractors, subcontractors and fabricators, real estate developers and building owners and operators (IAI, 2003).

The transition to 3D modeling was first proposed in the 1970s (Eastman, 1975) and will take a generation from now to realize, because of the massive changes in practices required. During that time, many technical issues will have to be resolved. This paper outlines several of the needs and opportunities to achieve this transition. It also places this transition to parametric modeling within the context of design theory, providing a framework for the research issues needing to be resolved.

2. Building Information Modeling

To provide a comprehensive solution, Building Information Modeling must eventually address the full building lifecycle: feasibility planning, design, engineering, construction coordination, shop-level fabrication, commissioning, facility management and operation. People disagree about whether a single integrated model or multiple federated models will be generated in response to the different responsibilities within an overall building project lifecycle. We believe this is a business issue that will be resolved in varied ways, within the contracts and agreements for specific projects. A hospital that is being continuously remodeled will require one level of integration, a housing project another. Also, the issue of whether an as-built archival building model is generated is a different issue from the different BIM modeling tools used during design and construction. There are many different facets to the overall realization of BIM.

Here, we focus on the need of many specialist domains to develop their own BIM tools. We do not expect (or desire) that a monopoly for BIM platform development arise. Rather we expect specialized application tools to be developed on a variety of platforms. Other BIM software will be required to coordinate between the separate models, for example by the contractor coordinating the models of subcontractors, addressing what today is the coordination among shop drawings. We expect to see different BIM tools responding to these different building construction purposes.

Here, we focus on BIM platforms for the development of domain-specific applications. We outlined some requirements for such platforms elsewhere (Sacks et al., 2003). Here we articulate them further:

1. strong *geometric modeling capabilities*, so as to allow full modeling of all building parts, at varied levels of detail. Most construction materials are linear extruded elements or sheet materials (sometimes folded). Building shapes are defined parametrically in terms of these base shapes (sometimes cambered, twisted or spiraled). 2D materials such as paint and other surface treatments applied to 3D surfaces must also be represented. A broad range of geometric modeling capabilities is required;
2. the geometric modeling capabilities must be *parametric*, supporting both automatic layout and updating according to design rules. Large assemblies of elements must be easily edited, allowing intelligent updating and low-level automatic re-design when the context of the building components changes. Parametric modeling has been shown to be necessary, if the design of construction-level models of buildings is to become practical;
3. carefully defined *domain-specific semantics* (e.g., objects, attributes and identification tags), capturing the classification and functional properties of building components. The information carried in these components must in some cases be parametrically defined, related to geometric or other properties that are updated automatically. Semantics also require a large overlapping set of groupings of components by properties, for costing, ordering, delivery, erection and so forth.
4. reliance on a *single integrated model*, allowing all data and relations to be carried in an associative structure, facilitating consistency and integrity management over all data;
5. *automatic report generation* from the building model, allowing all drawings, specifications, NC and other production information, bills of material and other reports to be consistently generated from the integrated building model; automatic version management of reports should allow tracking of obsolete drawings and reports and re-issue of updated ones;
6. *easy import of design model data*, allowing sharing and extension of new design materials, construction methods and parts as they are identified as needed;
7. *easy export of subsets of design data* in a variety of formats for use in downstream activities, such as engineering, contracting, piece fabrication, and erection;

8. *extensibility*, to allow the above capabilities to be easily applied to new classes of design objects and assemblies, easily defined by a designer; no design tool can be closed to new forms, materials or methods of construction;
9. *scalability*, supporting interactive parametric design of 10^5 to 10^7 objects on current standard hardware; architects and other designers regularly take design actions affecting large numbers of parametrically related objects.
10. *concurrent access and management:* only a small project can be designed by one person in a practical timeframe. However, large parametric structures require single-person access to large portions of the building data, to allow parametric updating. Thus elaborated forms of concurrency control are needed to allow teams of designers to work on what functionally is a single large parametric structure.

We list these capabilities as requirements for a BIM system because we cannot see such a system being widely adopted until these capabilities are provided. Currently, no system has effectively realized all of these requirements, though they are being worked on intensely. Some of the requirements are likely to be implemented poorly in the current generation of CAD development efforts. Early adopters are needed to begin using these systems and sort out their limitations before the full functionality exists, so that more mature systems will eventually be developed.

This conference and its theme offer a good setting to reflect on these objectives and to explore some of their implications for future research.

3. Design Automation

A medium-sized building includes more than one millions parts, at construction level of detail. No human can use only interactive computer methods to define and lay out a million individual pieces effectively and correctly in a 3D construction level building model. He or she certainly cannot <u>design</u> using such tools. The time required would be prohibitive and too many layout errors would occur during revisions. Thus automatic shape layout and updating is absolutely required for effective design. We have seen simple versions of this for years: wall editing systems such as those in Triforma® and Architectural Desktop®, (first developed at Carnegie-Mellon in Yacov Yasky's thesis (1980)), piping layout systems that can estimate threaded lengths needed for connections, etc.

The challenge in front of the software companies developing systems for BIM is to provide effective automatic layout and editing tools for all of building system types. Each building system needs to support its own layout and updating behavior, be intuitive and effective to use. Examples of such capabilities are those for steel detailing, such as Xsteel® and SDS/2® and for piping layout and detailing (multiple products are offered by Autodesk, Bentley, CATIA, PTS and others See: (Piping, 2003)). The interactions between systems also need to be addressed automatically. This includes automated routing and subtraction operations to avoid spatial conflicts (pass-throughs for piping, re-routing for electrical) based on precedence ordering to determine which system is to be modified (for an early example, see Gross, 1994). It also includes automatic capture of loads, including structural, plumbing and electrical loads, to make sure that needed system interactions are identified and resolved. A theory of parametric modeling for buildings is needed to provide the appropriate abstractions to allow developers to easily design software systems for these building systems (an example in mechanical design is partially framed by Shah and Mantyla, (1995)).

The development of these capabilities will occupy the CAD system developers for the next 3-5 years, just to cover the different building systems and their varied behavior with first generation capabilities. We are all used to such tools as Visio® and their friendly layout and editing capabilities. What we will see emerge are similar tools that deal with automatic 3D layout, routing and revisions, based on much more complicated, user-controllable rules.

4. Parametric Behavior of Precast Concrete Systems

The authors have recently been involved in the definition of such systems. They advised an industry-based technical group, the Precast Concrete Software Consortium (PCSC), in the development of product specifications for a parametric modeling system for precast concrete design, engineering and fabrication (Eastman et al, 2003a; Eastman et al, 2003b). The detailed level specification was developed jointly with the competitively selected software developer, Tekla Corporation, and is now being embedded into the Xengineer product being beta-tested in incremental quarterly releases. This system will include parametric updating behavior for such systems as:

- column, beam and floor-spanning systems;
- stairwells;
- architectural facades;
- connections across all combinations of piece types;
- rebar mesh and prestress cable layout;
- all embeds, such as connection plates, bars, and pins;
- special surface treatments, joints, and other detailing.

The automated design behavior is achieved by embedding expertise regarding precast concrete design on top of generic system-level parametric modeling capabilities for buildings. We have found that the parametric modeling capabilities needed for buildings have important distinctions from those required for mechanical applications (see Sacks, et al. 2003). Editing behavior of these elements had to respond to structural, aesthetic, precast good practices, building code and other criteria. A goal behind the development of these capabilities was to be able to completely design, engineer and plan production for a simple all-precast structure, such as a parking garage, in less than a week. Currently it takes several months. (It will be interesting to see if the specified tool achieves this goal.) Automated design is an integral aspect of this tool.

In order to examine the structure of the expert knowledge we embedded in the design package, we take one area and look at it in depth.

4.1 FLOOR SPANNING SYSTEMS

Most floor spanning systems in precast concrete are single-direction load transfer systems made of semi-standard components. They are semi-standard in that they are extruded pieces fabricated in standard molds, but they can be customized at their ends and sides by blocking out volumes of concrete, their depth and flange height can also be adjusted in this way. They require custom engineering depending upon the geometry of each piece, its span, loading and its connections. The spanning system interacts with the structural frame that receives its loads.

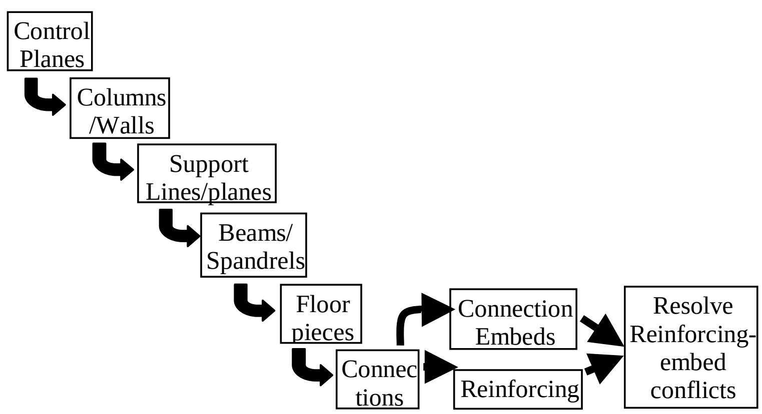


Figure 1: *the hierarchy of updating for a floor assembly system*

Changes are propagated through a unidirectional hierarchy of parametric relationships. The hierarchy for layout and updating that drives the layout of a floor assembly is shown in Figure 1. We walk through each of the steps.

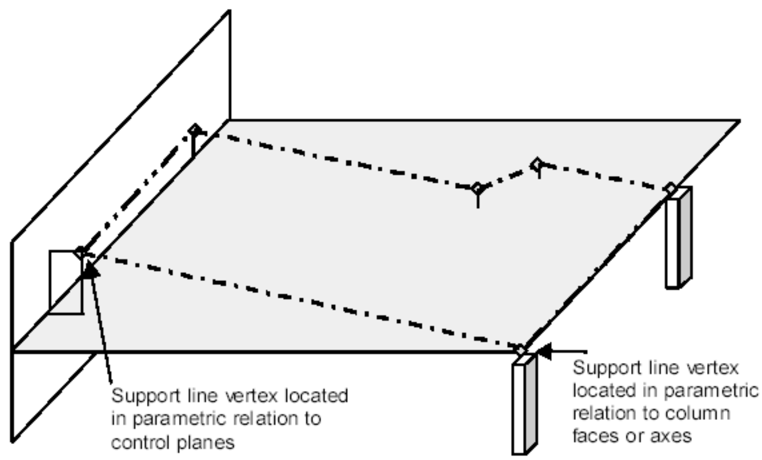


Figure 2: *control planes and vertices for layout of a floor system*

Figure 2 shows two examples of how support lines, controlled by their end vertices, can be located using parametrical relations. The vertices to the left of the figure are related to the control planes. The vertices to the right are related to the columns (the columns themselves may be positioned in turn in relation to control planes). The vertices are all located on or offset from a horizontal control plane.

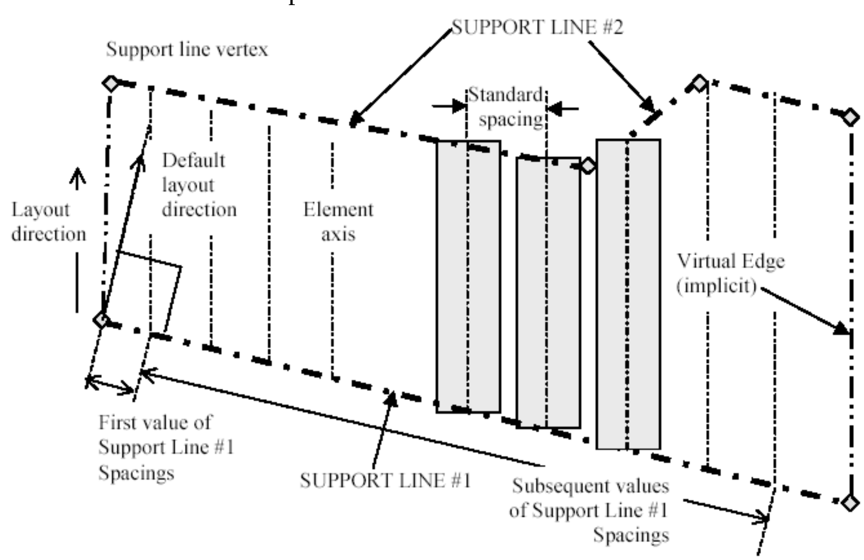


Figure 3: *placement and layout of floor elements.*

The direction of floor panels – here assumed to be double-tees or hollowcore slabs, and their starting placement – is given by a centerline line, as shown in Figure 3. In this case, the widths of floor panels are adjusted to a constant dimension; the edges are trimmed to the spanned area boundary. While Figure 3 is an example of a single span, multiple spans are also supported, usually with separate pieces for each span. Cantilevered layouts can be defined by projecting the spanning area beyond the support lines.

For the application of these capabilities, the user first selects beam sections for the end supports and the floor spanning elements from an existing library, or creates new project-specific elements. Then he or she uses the above parametric tools to quickly generate a design layout. Later, as grid, control lines or parameters change, the floor system elements will be automatically updated.

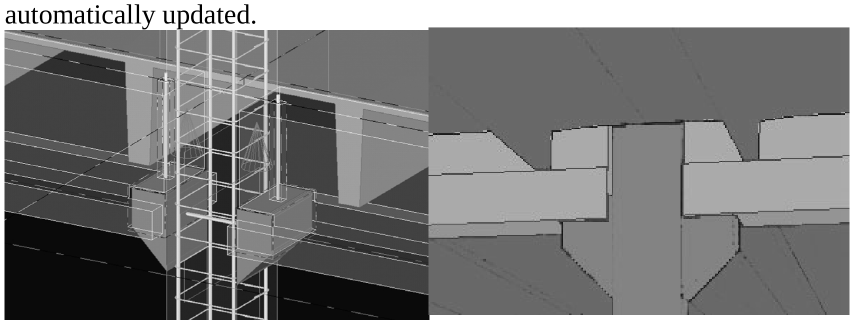
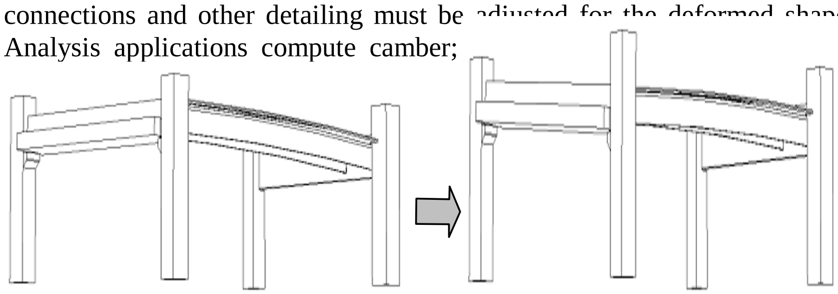


Figure 4: *A haunch style of connection between a column and beam.*

Connections are then placed. The default case is at the end of members. Cantilevers and other special cases are currently defined interactively. The connections are defined at two levels; the first level identifies the topological relation between connected elements. This is automatically followed by assignment of a specific parametric connection. Connections are predefined into a large set defined by the primary and secondary members being connected, then a connection type, then the parametric detail itself. Each parametric connection detail has been defined to adjust to the dimensions and orientation of the members it connects, and includes the subtractive trimming and the addition of shapes applied to the two pieces. The connection also includes all the embedded parts needed for fabrication. It also adds the installation hardware required for erection. An example is shown in Figure 4.

The layout of such floor spanning systems is usually not completely planar. Because of prestressing, a spanning member takes two different forms. In its as-cast form, the member centerline is usually straight (occasionally cambered shapes are cast too). After fabrication and prestressing, the member is cambered because of the eccentric loads from the prestressing, as shown in Figure 5(a). Its final as-fabricated shape must be defined and represented in uncambered form, which often is not eliminated even when its dead and live loads are applied. Placement of connections and other detailing must be adjusted for the deformed shape. Analysis applications compute camber; f



(a) *(b)*

Figure 5: Cambering and warping of precast structural tees in a floor spanning assembly.

In addition, post-cast warping is often purposely applied to each member so they can rest on a support beam that has one corner lower than the other, in response to transitions with ramps in parking garages or to provide drainage. The angular degree of warping is limited to a few degrees (varied limits are used by different companies) and is applied as the members are set in place. The desired design behavior is defined by the slope of the supporting member or surface. A diagrammatic example in shown in Figure 5(b).

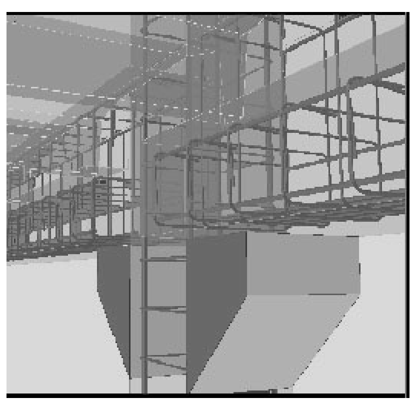


Figure 6: *Reinforcing of a column and beam.*

After the beams and floor elements have been laid out, reinforcing is applied. It is defined as a pattern, with a given number of bars in a prescribed pattern, all defined parametrically. Multiple spacing within a beam and other aspects of current practice are set up for parametric control. An example layout is shown in Figure 6. In the base case, the user assigns the parameters, in response to externally calculated lateral and vertical load conditions, determined by codes and safety factors.

The program is designed to support interfacing with an open set of external structural applications that calculate the resulting reactions from the vertical and lateral loads. These are then automatically fed back into the geometric layout operations. Assuming the initial beam and floor sections are adequate to receive them, the connections are assigned or re-assigned, and the parameters selecting the reinforcing layout pattern and sizes are assigned. One application for doing the structural analysis has already been integrated.

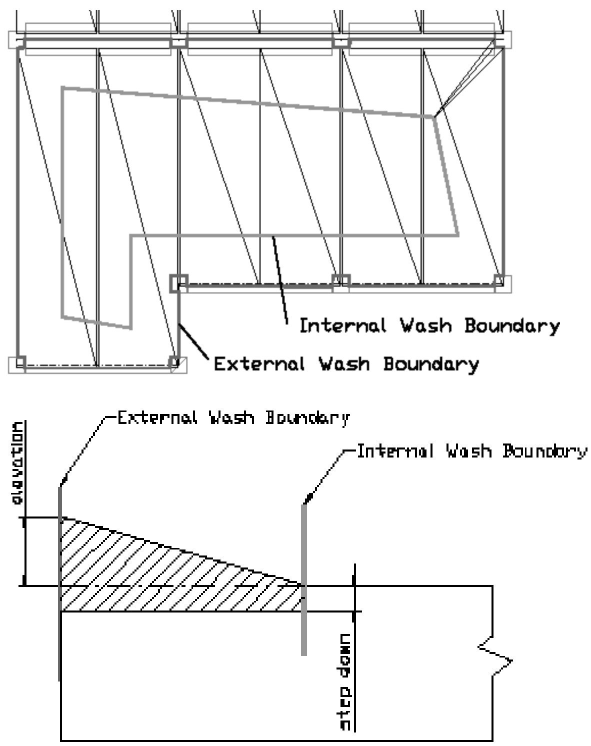


Figure 7: *Definition of toppings for drainage of precast decks.*

The last step in Figure 1 involves resolving possible spatial conflicts arising from the layout of embedded parts and the reinforcing, and the parallel layout of the reinforcing in the members themselves. (These two operations may be done in either order.) We are investigating how they may be addressed automatically; currently, this is an interactive routine.

Floor spanning systems also sometimes require water run-off, requiring additional material to form sloped surfaces. The material must be defined, either for placement as part of the precast production, or as post-erection toppings (called ‘washes’). In either case, the weight of the topping must be considered. Toppings are defined with two or more perimeters, with heights above floor designated for points on the polygons, as shown in Figure 7. Sometimes a “step-down” is required in the precast concrete piece to provide an edge that will not flake.

All the layout operations generate full 3D shapes, sometimes subtracting from existing shapes, other times adding, supporting volumetric and other spatial properties, and spatial conflict testing. From such layouts, assembly and piece drawings can be semi-automatically generated. Company-specific drawing templates define which dimensions and annotations are to be provided. Clean-up of the placement of dimensions and annotations is usually still required. Bills of material are fully automated.

The capabilities for floor systems described above allow quick generation of geometric and topological design candidates and their automatic detailing. The process of layout and regeneration generally follows the process sequence carried out in manual practice, but with each step partially or fully automated, with strong links between them.

The floor spanning system and its design behavior is only one example of over twenty areas of specification now defined in detail for the first production release of the Tekla precast software. This example and others like it suggest how all building-systems need to be configured in the next generation of BIM.

5. Implications for Design Theory

Such capabilities as these will have to be carefully defined and replicated many times to cover the full domain of building design. Similar efforts are required to address different building systems and also to address special user systems involving special technologies – airports, hospitals, laboratories – that also require special equipment, analyses, and design rules. The criteria will need to be developed by domain experts, either through consortia of future users, as developed here, or other means.

In related work, we have identified some of the software system features and capabilities we believe are required to realize such systems (Sacks, et al, 2003). We have also explored the fact that such systems embed expertise and design knowledge, identifying how such knowledge can be elicited from domain experts (Lee et al., 2003). Here, we are interested in the theoretical basis for these efforts. What kind of knowledge are such systems addressing? How will such parametric design tools integrate with other kinds of knowledge-based systems? What is the relation of these systems to our understanding of design?

5.1 INTEGRATING FUNCTION, STRUCTURE AND BEHAVIOR MODELING

In design theory, it has been recognized that engineering design involves at least three kinds of information: regarding the specific *functions* or purposes of the design, the detailed *behaviors* required to achieve those functions, and the different forms or *structures* that support the functions. This line of study, called SFB (or SBF, FSB, FFB, or similar mnemonic), has been studied in the artificial intelligence and engineering design literatures for more than ten years (Chandrasekaran and Josephson, 2000; Goel, 1992; Gero, et al, 1991; Bobrow, 1984). The structure of a design is the geometry and material properties carried in CAD systems. The functions are the human-assigned purposes of the design, both globally and at the detailed level, and also the naturally or environmentally determined functions to which the design must respond. Behaviors are the performance properties that the design is predicted to display in response to its designated functions and in response to external or internally defined loads and conditions.

The goal of this work is to allow computer tools to embed more powerful semantic or reasoning capabilities about designs and to develop richer design representations. SFB modeling research has attempted to operationalize design reasoning processes, mostly along expert system lines (Goel, 1992). Thus a function has interfaces, ports, or connections with other functions. Each of these connections has a property that they transfer or exchange. Thus a bell or buzzer, a switch and a power supply are structures that provide the function of letting a person at some distance know that there is someone at the switch location (see Chandrasekaran and Josephson, 2000) for a detailed description). Here, the measures of the power supply, the loudness of sound, the speed or other properties of the switch, are the behavior properties derived from the structure's designated function.

5.1.1 SFB Modeling of Buildings

Buildings are objects that have a complex set of functions involving safety, support for their particular use, durability, occupancy comfort, material and utility resource utilization, and reflect stylistic and cultural criteria, among possibly many others. The articulation of these different functions has evolved over human history, roughly in parallel with the articulation of means to address them.

Many of these functions are responded to in a building design (and in a building) by *systems*. In fact the building systems exist in associative response to the functions they have been designed to serve. We have structural systems, lighting systems, heating, ventilation and air conditions (HVAC) systems, plumbing, communication and fire safety systems, among others, in response to specific functions of a building.

Each of these systems has an associated set of parts from which they are typically composed, and an associated range of detailed functions that the parts serve to realize the overall function. Structural systems are served by steel, cast-in-place concrete, precast concrete, and sometimes heavy timber materials, that are composed into frames, bearing walls, horizontal decks, trusses and space frames, made up of members, joints, reinforcing, etc. In structures courses, architects and engineers learn the vertical, lateral, static and dynamic behaviors of the systems, the analytical models that are known to predict them, and the physical properties that are required in order to address them. Thus in education, they learn these functions, the systems to resolve them, and the first principles on which the systems are based. They gain this knowledge all together as a large unit of information, to varying degrees of competence. Thus structures and building technologies courses are courses on functions and systems. Other courses deal with other functions and systems.

Most of the intellectual development of systems has migrated from architecture to different areas of engineering, as the relation between specific functions and the behavior of structures that respond to them has become understood and made predictable. Some systems, such as pedestrian circulation systems, food preparation systems in restaurants, and other systems that involve significant human and organizational components, however, are still the province of architects.

It is difficult to rethink from first principles alternative structures that respond to a function. It occurs only rarely in architectural practice (more frequently in structural practice). In almost all cases, we select from the range of existing systems available to respond to a function. For a well-developed system, design consists of element selection, sizing and composition. Almost all of architecture and building design is of this type of activity, even though it is not taught in this manner.

The mapping from function to behavior is a well-developed area of design research (Gotti and Sriram, 1996; Clayton et al, 1999). Parametric CAD systems provide a significant advance because they provide powerful ways to encode the mapping of functionally defined systems into specific form responses within them. They also optionally support behavior-driven design. The examples in the front of this paper are of this kind. They allow the abstract definition of the relevant components making up a building system, their relevant dimensional, topological, and behavioral properties, and the layout and sizing of components to achieve specified behavior.

Notice that the interactive behavior discussed in earlier sections – what we called *design behavior* – is not included in the SFB classification. Rather design behavior addresses how to maintain certain structural relations among components. SFB describes static or state information about a design, while design behavior deals with procedural information.

In the example precast concrete floor spanning system object, the floor spanning components are laid out to span between two end members. Each end member is similarly defined to span between other structural members. Transfer between the floor elements and the end members, and between the end members and their supporting elements, are defined by connections. The layout and updating mechanisms attempt to maintain the topological relations necessary (but not sufficient) for their structural function; i.e. the purpose of the embedded *design behavior* is to maintain the integrity of the system structure, in such a way that after any change, it will still be able to fulfill its system function.

Of course, the floor spanning system does not guarantee that the system is structurally stable (stable under structural loading), only that it is satisfactory topologically. More detailed member sizing and property criteria must be applied: strength of concrete, deck thickness, deck reinforcing, steel area and eccentricity, strength of steel, etc. These are derived from loads and the structural analysis and the results are fed back into the process to assign these properties.

5.1.2 Externalities

The toppings definition and layout is interesting (and was included) because it deals with an issue quite separate from the structural function. It responds to the externally imposed function that rain and other water not accumulate on the exposed roughly horizontal surfaces of a building. Typically, this function is only partially resolved by the precast system. First, drainage of the floor spanning surfaces is a function relevant to all surfaces that are exposed to rain or other water sources. These surfaces are not automatically defined in the current application. Thus a user must identify where the topping layouts are to be applied.

The drainage of rainwater from the lateral surfaces must run toward some drain or gutter that typically is part of the plumbing system, and the responsibility of a different system and contractor. The Tekla application does support the definition of external systems, so as to allow interactive checking of spatial conflicts with these systems, for example. Though not yet developed, it is also possible to check that such drainage capability is provided in all low areas where toppings are applied, as another kind of design check for the precast concrete system.

In general, most building systems interact with each other in only limited and defined ways. Spatial interaction is the most common. In this example, the step-down for the topping to maintain minimal thickness of this later pour removes material from the spanning pieces. Sometimes, this may interfere with embedded objects. However, since it is replaced with additional concrete, it does not negatively affect the required coverage of reinforcing. The other kind of interaction is the additional structural load the topping imposes and is dealt with explicitly in the dead load calculation process.

5.1.3 Abstract Functional Objects

Like any system, the floor spanning system is made up of a number of separate object classes, each with their own associated detailed design behavior. The floor decking, the lateral beams or walls collecting the load transfers, the connections between them, the vertical members that pick up the loads from the lateral beams, the connections between any beams and columns, are all parts of this system. Each of these parts has a load transfer function between the loads they accept and those they transfer, defining its structural topological path. Associated with each of these parts is an analysis method for defining the section properties and bearing surface areas required for the loads it is mandated (by codes) to be able to carry, according to a functional behavior (the B in SFB). The analysis method may be manually executed or automated.

The floor spanning system, like all systems, provides a specific mapping from function to form. A two-way spanning system, such as a cast waffle deck or a space frame, is a slightly different system, with different structural topological paths. They require slightly different analysis methods.

With the built-in specification of these capabilities, any part may be re-shaped, the relations applied to connecting objects can be re-defined, and the system and design behavior will still work. That is, within each of the object parts is an abstract definition of the system's functional behavior. In earlier work, we called this an *abstract functional object* (AFO) (Sacks et al, 2003). AFOs provide wide flexibility in the design of components making up a system.

In the earlier work, we proposed that AFOs make such system design open-ended. Upon reflection, however, it is apparent that the overall system definition, that associates the design behavior and functional behavior of each physical part, limits the range of parts it can accept within its system definition. The limitation places in context the strengths and limitations of parametric modeling. It supports the parametric modeling of building (and other types of) systems. Here a system is not the high level system used when we refer to a "structural system". Rather we a referring to the detailed "one-way precast floor spanning system" described above, possessing a specific topological structure. While the topological structure can vary, it can do so in only limited ways, defined in how load distributions are calculated. A research challenge over time will be to generalize AFOs so they can depict a wider class of designs. We believe this is possible and will be done, but it is unclear how much so without significantly increasing the complexity of designing instances of the system.

In the design tool we have been reviewing, the structural behavior has been used as the primary function for defining the form. How do we build operators that allow form definition responding to multiple functions? Again, the floor topping operators provide an insight. In the Tekla package, the structural function is the primary form generator, and the water drainage is a secondary function. It is secondary because it does not generate the form from scratch[1] but rather modifies an existing form. This suggests a family of secondary operations that need to be associated with a parametric building modeler; these modify an initial design to be responsive to other functions, such as water drainage, thermal insulation, lighting, acoustics, and others. This hierarchy of functions, with a primary form generator and succeeding secondary modifiers of the form, reflect a common pattern of design today that follows the same process.

The hierarchy of function-to-form operators also suggests the wider direction of parametric modeling functionality. We see the eventual provision of multiple parametric modeling systems for a single function, some of these being primary form generators (in our example, for different types of floor spanning systems) and others that can serve as form modifiers. The form modifiers operate on primary forms generated in response to different functions. These could provide designers with different design paths, reflecting different priorities for a particular project.

This discussion has focused exclusively on dealing with functions one at a time. However, an important type of design innovation is the development of a system that responds to new combinations of function. A curtainwall system is an example, which addresses the façade structure, thermal insulation and light transmission. A parametric modeling system should support layout, detailing and analysis responding at least to these three functions, for which it was designed to be responsive. Multiple function-to-form generators, if they are successful, should provide a different space of forms than can be developed in a single function generator with sequential application of function-driven modifiers.

A good example of multiple function design innovation is the stair balustrade in Joseph Hoffman’s house in Vienna at the turn of the century. This art deco architect’s walk-up apartment stair balustrade was designed to also function as a hot water radiator on the ground floor, both warming the space, and providing stair safety. It is clear that the hot water radiator was the secondary function.

[1] We assume such a form generator would only generate sloping surfaces.

If buildings, composed of multiple systems responding to different functions are to be designed with ease, then parametric modeling schemes to support the different systems will be needed. Historically, design handbooks have been developed for many systems. The American Institute of Steel Construction has its LRFD Design Handbook (Hoffman et al, 1996.), The Precast Concrete Institute its Design Handbook (PCI, 2001.). These handbooks identify good practices in the design of one or more systems in response to one or a small set of functions. These handbooks were developed for human interpretation. In the future, this information will be structured as parametric modeling modules, for incorporation into design tools. Other forms of parametric modeling systems, oriented toward different functions, have been proposed. Commercial products exist for piping and HVAC (piping, 2003); and for electrical systems (such as EasyPower® and EDSA®). Shape grammars offer a formalism for aesthetics that could be embedded in a parametric modeling system (Heisserman and Woodbury, 1993).

Also, many issues are not dealt with by defined systems. Human factors and ergonomics are concerns that apply across and between physical systems. Sustainability has very broad concerns that are not easily encompassed as a building system (Streitz et al, 1998).

6. Summary

The new generation of building design systems, developed in collaboration with the Tekla system reviewed here, will change the way information is delivered to the construction industry. It is still largely a handcraft industry, where handcraft as a term signifies that the application of expertise is through its manual application. In this sense architecture has remained largely a handcraft. This new direction will not eliminate the architect's manual application of expertise; handcraft will continue to drive innovation and creativity. It will change dramatically however, the everyday delivery of construction expertise, providing it in the form of software applications, rather then exclusively through experienced draftsmen.

Parametric modeling holds the potential to realize to a significant degree the capabilities posed by Per Galle regarding the definition of what intelligent design is about (Galle, 1995):

- Intelligence: the ability to maintain semantic integrity. The system helps keeping the representation of the evolving design consistent with its meaning.
- Generativity: the system can propose solutions to specified aspects of the design problem.

Acknowledgements

The authors wish to thank the Precast Concrete Consortium for their support for the specification development reviewed above. We also thank Tekla Corporation for their willingness to share these ideas.

References

Automation in Construction, (1992-) Elsevier Science Publishers, Amsterdam, The Netherlands

Bobrow, D.G. (1984). "Qualitative reasoning about physical systems: an introduction", *Artificial Intelligence*, 24 pp. 1-5.

Chandrasekaran B. and John R. Josephson, (2000) "Function in Device Representation," Engineering with Computers, Special Issue on Computer Aided Engineering, 16:162-177

Chandrasekaran B. (1994) Functional representation and causal processes, in Yovits, M. (ed.) Advances in Computers, 38, pp.73-143

Clayton MJ., Teicholz P, Fischer M, Kunz J (1999) Virtual components consisting of form, function and behavior, Automation in Construction 8 1999 351–367

Eastman, C.M. (1975) "The use of computers instead of drawings in building design" J. Amer. Inst. of Architects, (March), pp. 46-50.

Eastman, C. M., R. Sacks and G. Lee (2003). "Development and implementation of an advanced IT strategy for the North American Precast Concrete Industry ".ITCON - International Journal of Construction Information Technology http://www.itcon.org/ 8: 247-262.

Eastman C. M., Sacks R. and Lee G. (2003a), Development and Implementation of Advanced IT in the North American Precast Concrete Industry, Information Technology in Construction, at http://www.itcon.org/. (incomplete)

Eastman, C., Lee, G, Sacks, R. (2003b) Development of a Knowledge-Rich CAD System for the North American Precast Concrete Industry, ACADIA22, Ball State University,

Eastman, C. M., Lee, G., and Sacks, R. (2003c) "Development of a Knowledge-Rich CAD System for the North American Precast Concrete Industry ".ACADIA 22: Connecting-Crossroads of Digital Discourse ,Indianapolis, IN, 208-215.

Eastman, C. M., Sacks, R., Lee, G., and Orndorff, D. (2003d). "PCSC Inform Task Report in Support of Development of Precast Concrete Design and Engineering Software." Georgia Institute of Technology, Atlanta, GA, USA.

Galle, P. (1995). "Towards integrated, “intelligent”, and compliant computer modeling of buildings", Automation in Construction Elsevier Science B.V. Vol. 4 pp. 189-211.

Gehry Frank O., J. Fiona Ragheb, Solomon R Guggenheim Foundation, (2003) Frank Gehry, Architect, Solomon R Guggenheim Museum; (July)

Gero, J. H. Lee and K. Tham. [1991] “Behavior: A Link Between Function and Structure in Design”, In Proc. IFIP WG 5.2 Working Conference on Intelligent CAD, Columbus, Ohio, September 1991, pp. 201-230,

Goel [1992] Goel, Ashok. Representation of Design Functions in Experience-Based Design. In Intelligent Computer Aided Design, D. Brown, M. Waldron, and H. Yoshikawa (editors), pp. 283-308, Amsterdam, Netherlands: North-Holland, 1992.

Gorti SR,and RD Sriram (1996) From symbol to form: a framework for conceptual Design, Compuler-Aided Design, Vol. 26, No. 11. pp. 653-870. 1996

Gross, M. (1994). Avoiding Conflicts in Architectural Subsystem Layout. Concurrent Engineering: Research and Applications, 2, 163-171

Heisserman J. and R. Woodbury (1993). Generating languages of solid models. In *Proceedings of Second ACM/IEEE Symposium on Solid Modeling and Applications*, pages 103–112, May 19-21 1993.

Hoffman, E. et al (1996) *Structural design guide to the AISC (LRFD) specification for buildings*, 2nd ed. New York : Chapman & Hall, 1996.

Lee, G., C. M. Eastman, and R. Sacks. (2003, working paper). BOB: A graphic notation for parametric building object behavior. Georgia Institute of Technology, Atlanta, GA.

PCI, (2001) PCI Design Handbook – Precast and Prestressed Concrete, 6th Edition, Precast/Prestressed Concrete Institute, Chicago, Illinois.

Piping, 2003) See: http://www.processregister.com/Piping Design Software/Suppliers/pid5048.htm)

Sacks, R., Eastman, C. M., and Lee, G. (2003). "Parametric 3D Modeling in Building Construction with Examples from Precast Concrete ".Automation in Construction, in press.

Shah, J. J., and M. Mantyla. (1995). Parametric and feature-based CAD/CAM: concepts, techniques, and applications. New York: John Wiley and Sons.

Streitz, N., Konomi, S., Burkhardt, H.-J. (Eds.), Cooperative Buildings - Integrating Information, Organization, and Architecture. Proceedings of CoBuild'98. Darmstadt, February 1998. Lecture Notes in Computer Science 1370. Springer: Heidelberg, 1998

Yasky, Y. (1980). "Transforming a set of building drawings into a consistent database", Proc. CAD '80 Conf., IPC Press, London, pp. 101-108.