

Computer Representation to Support Conceptual Structural Design within a Building Architectural Context

Rodrigo Mora¹; Hugues Rivard²; and Claude Bédard³

Abstract: Computer support for conceptual design of building structures is still ineffective, mainly because existing structural engineering applications fail to recognize that structural design and architectural design are highly interdependent processes. This paper describes a computer representation called StAr (structure-architecture), aimed to act as a common basis for collaboration between architects and engineers during conceptual structural design. The StAr representation describes the structural system as a hierarchy of entities with architectural counterparts, which enables the direct integration of the structural system to the building architecture as well as engineering feedbacks to the architect at various abstraction levels. The hierarchical structural description implements a top-down design approach where high-level structural entities, which are defined first, facilitate the configuration of lower-level entities whose functions in turn contribute to those of the higher-level wholes that they belong to. The representation has been built on top of a geometric modeling kernel that allows reasoning based on the geometry and topology of the design model, which is paramount during early design stages. A proof-of-concept software prototype, called StAr prototype, has been developed and a test example demonstrates how the representation can support the different activities that take place during the conceptual design of building structures.

DOI: 10.1061/(ASCE)0887-3801(2006)20:2(76)

CE Database subject headings: Computer aided design; Conceptual design; Structural design; Architecture.

Introduction

Building design is an iterative process that follows three stages of refinement: conceptual design, preliminary design, and detailed design. Conceptual design is the first stage where the most important characteristics of the building are defined. Major decisions are made regarding the building architecture, such as the internal configuration of spaces and physical elements that give shape to the building form, as well as the salient features of the engineering support systems, such as their type, material(s), and layout. These decisions have great impact on the constructability, cost, and overall performance of a building (Fenves et al. 2000). Consequently, multiple conflicting building design criteria need to be considered during conceptual design coming from the

different disciplines involved in the building design process. This research project focuses on the interactions between the structural and architectural domains during conceptual design.

There is a consensus among building design practitioners that better engineering feedback to the architect is required early on during the design process. Computer programs for performing structural analysis and detail design calculations have been on the market and used by practitioners for many years. However, there is still a lack of computer programs available to properly assist engineers in the conceptual design of building structures. First, these applications support a constructive bottom-up approach for structural model generation where the model of the structural system is synthesized element by element. This approach is not suited for conceptual building design since it neglects design refinement from systems to elements. Second, most of these programs enable the generation of a model of the structural system in isolation, with minimum regard to the building architecture and other disciplines. One explanation for this lack of support is that during conceptual design the engineer's work flow is highly dependent on the amount, quality, and type of information that is obtained from the architect. This in turn affects the quality and effectiveness of the engineering feedback provided to the architect.

The goal of this research is to improve collaboration between architects and engineers during conceptual structural design. The main premise of this work is that in order to improve such collaboration, computers should allow engineers to lay out the structural system within a building architectural context. A computer representation, called StAr (structure-architecture), has been developed to support this premise. The StAr representation is expected to act as a common basis for collaboration between architects and engineers during early building design.

¹Research Associate, Dept. of Construction Engineering, ETS, 1100 Notre-Dame St. West, Montreal PQ, Canada H3C 1K3; formerly, Research Assistant, Dept. of Building, Civil, and Environmental Engineering, Concordia Univ., 1455 de Maisonneuve Blvd. West, Montreal PQ, Canada H3G 1M8. E-mail: rodrigo_mora@sympatico.ca

²Professor, Dept. of Construction Engineering, ETS, 1100 Notre-Dame St. West, Montreal PQ, Canada H3C 1K3. E-mail: hugues.rivard@etsmtl.ca

³Professor and Dean of Research and Technology Transfer, ETS, 1100 Notre-Dame St. West, Montreal PQ, Canada H3C 1K3. E-mail: claude.bedard@etsmtl.ca

Note. Discussion open until August 1, 2006. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on September 16, 2004; approved on October 27, 2004. This paper is part of the *Journal of Computing in Civil Engineering*, Vol. 20, No. 2, March 1, 2006. ©ASCE, ISSN 0887-3801/2006/2-76-87/\$25.00.

This paper is organized as follows. The first section provides an overview of previous research in developing computer representations to support collaborative structural design. The following section describes the envisioned process of conceptual structural design within a building architectural context. Next, the StAr representation is described to support this process. Then, a proof-of-concept software prototype, called the StAr prototype, is briefly presented along with a test case to validate the approach.

Literature Review

A design representation for supporting conceptual building design must satisfy the following requirements (Rivard and Fenves 2000): (1) integrate multiple views for the different participants involved in the design process; (2) support design refinement/evolution; and (3) favor the exploration of alternative design solutions (e.g., enable iterations and changes in the design). Several computer representations have been developed for supporting building design. However, most of these representations are not specifically tailored toward supporting conceptual design. Those representations vary in scope as well as on the emphasis they place on a particular stage of the building life cycle. The scope of the representation dictates whether a semantically explicit representation (i.e., exhaustively incorporating the concepts that are relevant to the domains) is more convenient than a syntactic representation (i.e., generically defining the data structures and the rules for organizing the information), or vice versa. A new trend aims toward representational flexibility by means of high-level languages and constructs for building and extending representations.

Semantically Explicit Representations

Several semantically explicit representations have been proposed for supporting structural design. Most of these representations model structural entities exclusively while including some strictly necessary architectural entities, such as stories and walls. Some relevant examples are Law et al. (1990), Biedermann and Grierson (1995), and Fuyama et al. (1997). Only few representations provide more detailed descriptions of the building architecture and its relation to the structure: these are from Björk (1992), Dias (1996), Khemlani et al. (1998), and Sacks and Warsawski (1997). The representation proposed in this paper is akin to those of the latter group. Björk, Dias, and Khemlani et al. tackle the problem of formalizing the relationship between building spaces and their boundaries. All of them assume that structural elements always lie along the boundaries of spaces, which is a limiting assumption since structural elements also often lie within spaces. In addition, none of these representations consider the role of high-level structural entities (e.g., structural massings, subsystems, and assemblies) in specifying and laying out lower-level structural elements. As a consequence, they support neither abstraction nor decomposition of the structural system for design refinement.

Sacks and Warsawski (1997) proposed a representation that incorporates product and process considerations, and therefore is called project model (i.e., a product model plus a process model). It relates three main hierarchies that represent: spaces, functional systems, and construction processes. The representation describes various types of spaces and decomposes functional systems into assemblies and elements. Spaces contain assemblies and elements that are installed by construction activities. This representation

supports design refinement through functional systems, assemblies, and elements. It also supports multiple views of the design since a functional system can be structural, lighting, mechanical, etc. The main limitation of this representation for conceptual structural design comes from its correspondence between the decomposition of spaces and construction activities. Hence, the representation does not accept overlapping space aggregations, which are useful for providing multiple design views. For example, architectural concerns may require the distinction between private and public zones, which may be described as a single zone for structural purposes.

Syntactic Representations

Turk (2001) argues that representation models are subjective (i.e., respond to the developer's partial understanding of reality) and hinder creativity (i.e., force designers to use predefined semantic structures). Thus, he advocates for generic (syntactic) models because these impose no semantic structures on designers. Rivard and Fenves (2000) developed a syntactic representation for conceptual building design. The representation supports hierarchical decomposition of the design problem as well as two reasoning mechanisms: heuristics and case-based reasoning. The representation consists of three layered models: an underlying object-oriented data model, a generic information model built on top of the data model, and a conceptual model incorporating the semantics for a particular design domain. Similarly, Datta and Woodbury (2004) developed a syntactic construct, called Feature Node, which aims at capturing the rationale underlying the design process. The Feature Node construct emphasizes the support for design exploration and therefore captures the exploration history as a collection of ancestor and progeny feature nodes. The above generic representations exhibit good capabilities for supporting conceptual design; however, they overlook the role of spatial information in conceptual building design.

Standardization Efforts

The Industry Foundation Classes [International Alliance for Interoperability (IAI 2005)] information exchange standard for the architecture, engineering, and construction (AEC) industry uses an object-based representation model (IFC model) with object descriptions encompassing the different domains involved in building design, fabrication, construction, and operation. The IFC model provides a layered architecture with generic entities specialized into domain-specific semantic ones. In the structural engineering domain, there are four completed IFC projects: steel frame constructions (ST-1), reinforced concrete structures and foundation structures (ST-2), precast concrete construction (ST-3), and the structural analysis model and steel constructions (ST-4). For the IFC ST-1 project, a liaison was made between the IAI and the developers of CIMsteel Integration Standard version 2.0-CIS/2 (SCI 2005). CIS/2 provides a semantically explicit representation for describing steel structures.

The work described in this paper provides support that comes upstream to the IFC model and therefore can feed this model for integrating conceptual structural design. This type of support is still lacking because current IFC descriptions of the structure are too detailed as they are geared toward detailed design and construction (or fabrication and erection for steel structures). For example, such components describe reinforcing bars for concrete and provide detailed specifications for steel connections. Extensive information results in unnecessary model complexity, and

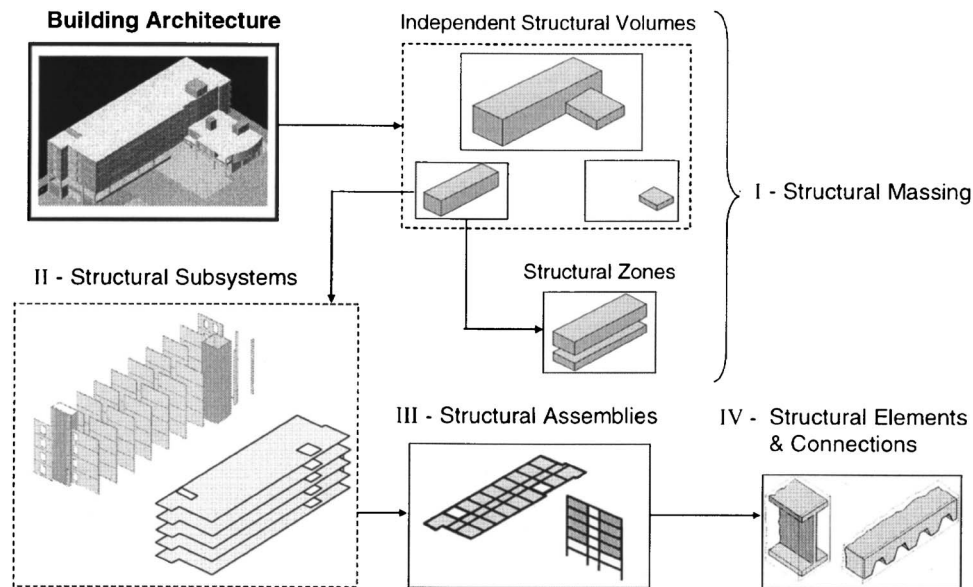


Fig. 1. Top-down structural design model

adversely affects design exploration during the early stages. As a consequence, their component descriptions are limited to tangible building entities (e.g., columns, beams, and connections) not being able to model purely functional building entities, such as structural subsystems and assemblies. Unlike tangible building entities, purely functional entities do not have actual physical embodiment. Such entities are particularly important during early design stages because they let designers express their concerns about structural functions while facilitating the subsequent structural configurations (Björk 1989; Rosenman and Gero 1996). In this paper, the term “function” is used in the sense given by Rosenman and Gero—to reflect what an object is intended to do (e.g., to transfer loads) in response to a purpose by the designer (e.g., to design a safe, serviceable, and efficient structural system).

Flexible and Extensible Representations

Despite the standardization efforts and support for integrated building representations, some researchers criticize integrated representation models as being rigid since they impose fixed organizational structures to the information that give no room for design creativity and exploration (Turk 2001; Haymaker et al. 2002; Stouffs and Krishnamurti 2004). Haymaker et al. developed an approach that seeks to augment integrated representation models and render them more flexible. They developed a so-called “perspective approach” for augmenting representation models. The approach allows new entity types and dependencies to be defined by the user during building design and/or construction. It incorporates a spatial reasoning mechanism called “perspectors” that, given a spatial criterion (e.g., beams at a 30° angle with slabs), searches the model looking for entities in the model fulfilling that criterion and defines explicit relationships among them. Stouffs and Krishnamurti (2004) developed a schema for constructing representations that is based on a construct called “sort.” The whole schema is supported by an algebra that allows designers to build and alter representations based on their own needs. Rather than a fixed data schema, the aim is to develop an extensible vocabulary of data components that can be composed into a representational language. In the field of precast concrete,

Sacks et al. (2004) define a construct called “abstract functional objects” (AFOs) that represent, store, and implement the functional properties of building assemblies and parts, and carry their fundamental parametric behavior and design intent. Therefore, a three-dimensional parametric modeling system could be viewed as collections of predefined AFOs that include an initial library of the common physical objects that embody them. In any particular design situation, a designer is able to define new physical objects that incorporate the functionality of existing objects by using the AFOs. Although the above approaches underline the importance of more flexible and extensible representations, none of them has been tested with conceptual structural design problems.

In this paper a semantically explicit representation has been developed for supporting conceptual design of building structures within a building architectural context. The StAr representation provides two views of the conceptual design problem, namely: the architectural view and the structural view. The initial aim is to support conceptual structural design for most typical, non-sculptural buildings, namely: office, apartment, educational, and mixed-use buildings. Therefore, it stands upstream from current structural engineering applications that are tailored toward supporting subsequent design stages. The semantic concepts that are incorporated in the representation are well defined in the architectural and structural engineering domains, and therefore can be explicitly represented without limiting the flexibility and extensibility of the representation. Such a representation that is tailored made for conceptual structural design offers the advantage of eliminating unnecessary concepts, entities, and relationships that are required only at later building lifecycle stages. Representational simplicity also facilitates design refinement and exploration because it enables fast design iterations.

Conceptual Structural Design Process

The goal of the engineer during conceptual structural design is to devise and lay out feasible structural systems that transfer loads to the ground efficiently and harmonize with the building architecture while responding to the design intent of the architect and the

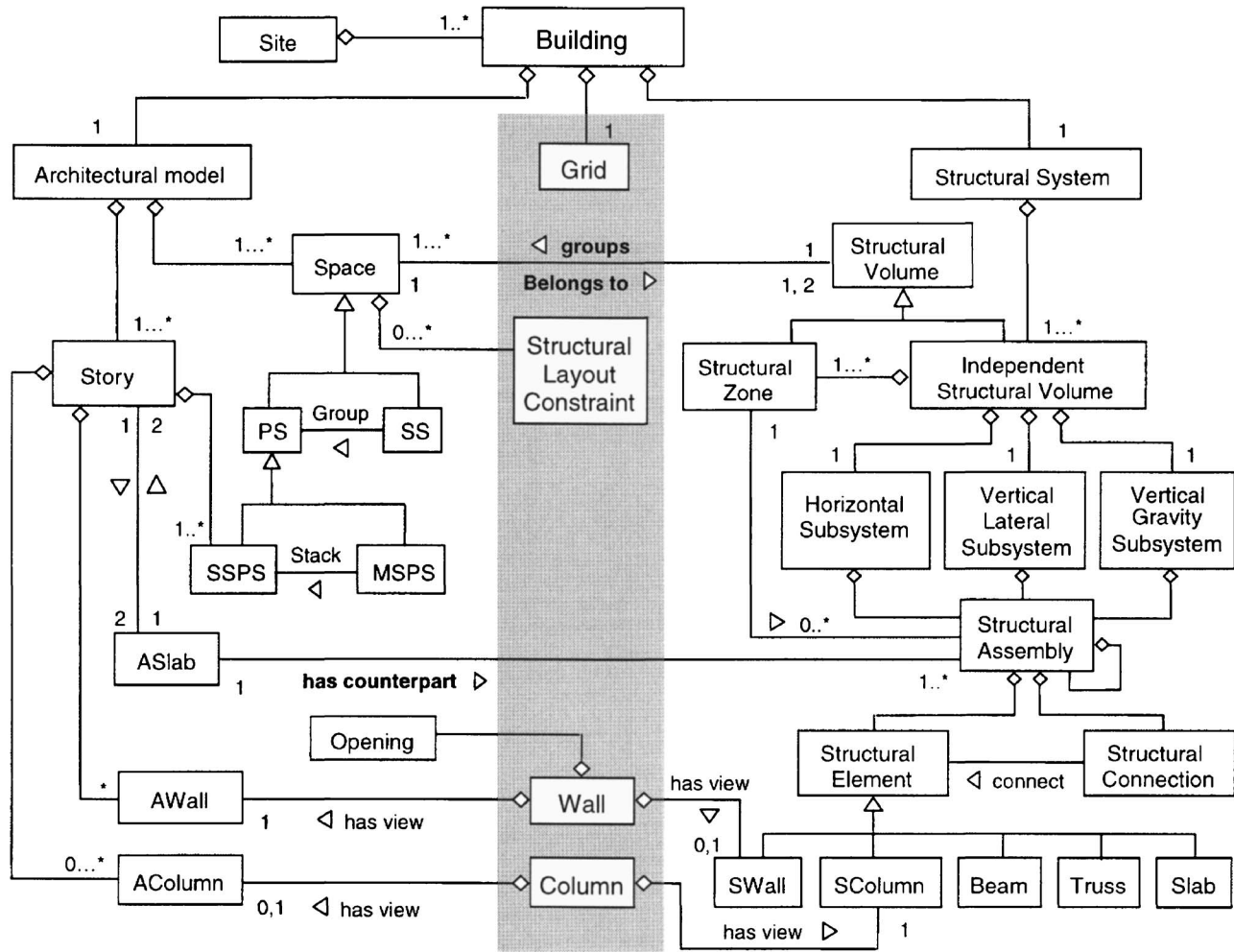


Fig. 2. StAr representation

client. A feasible structural system layout provides continuous load paths to the ground in a way that makes it structurally stable and realizable as a whole. The structural elements that make up the structural form are arranged, sized, and connected in such a way that acting together contribute to the intended functions of the structural subsystems and assemblies to which they belong. Consequently, a central premise of this research is that the conceptual design process of building structures should follow a top-down approach where purely functional entities (see the previous section) are initially defined to describe the type, layout, material(s), and other salient characteristics of the structure. These purely functional entities are subsequently refined by the constituent components. In this way, such entities become contexts for thinking about local issues of detail component interactions, and compatibility is assured between functional concepts and component definitions.

Several researchers have proposed process models that apply to conceptual structural design (Sause et al. 1992; Sacks and Warszawski 1997; Rivard and Fenves 2000; Scherer and Gehre 2000). These models share several characteristics, including following a top-down approach. However, only few of these models attempt to link decisions made on architectural spaces to decisions concerning subsystems, assemblies, and elements (Sacks and Warszawski 1997; Rivard and Fenves 2000).

This research project uses the model for conceptual structural design proposed by Rivard and Fenves (2000), which is inspired

from a top-down approach developed by Lin and Stotesbury (1988). In this model, the first level of top-down decomposition is the structural massing level. Thus, the engineer first breaks down the structural system into independent structural volumes that are assumed to behave as structural wholes (see Fig. 1). Independent structural volumes are in turn subdivided into smaller subvolumes called structural zones. These are introduced in order to allow the definition of structural requirements that correspond to architectural functions (e.g., applied load patterns, vertical supports, and allowed floor spans are related to the space functionality). Independent structural volumes are also decomposed into four structural subsystems—the foundation, the horizontal, the gravity, and the lateral subsystems. The foundation subsystem is not considered further in this paper. Each of the three other subsystems is further refined into structural assemblies. Finally, these are in turn decomposed into structural elements and their connections.

StAr Representation

The StAr representation combines concepts from the architectural domain and the structural domain in an integrated representation scheme (Fig. 2). The representation has been built on top of a geometric modeling kernel for representing spatial information

and enabling reasoning in terms of the geometry and topology. It explicitly defines object-oriented relationships among entities (i.e., inheritance, aggregation, and association), while spatial relationships such as adjacent, overlap, and intersect are not explicitly defined since they can be derived from the underlying geometric modeler. The unified modeling language (UML) notation is used in Fig. 2 (Booch et al. 1999) to describe the entities of the representation and their interactions.

The more abstract entities in the representation, which are defined by the architect, are the site and the building. The site may contain several buildings. The building entity includes general information such as its intended use(s) and the number of stories. The building also includes a reference grid. The grid entity has grid lines that are defined by the architect to act as a reference base for the configuration of spaces, but may be adjusted or augmented as a result of the structural engineer's feedback. For the moment, the grid lines serve merely for visual reference but are not explicitly related to other entities in the design model. In the future, they are expected to play a more active role in the design. The building incorporates two domain models—the architectural model and the structural system—which are in turn decomposed into their constituent entities. The central shaded portion of Fig. 2 shows key entities and relationships that link the two domain models. These domain models and their links are explained in the following sections.

Architectural Domain Representation

The architectural domain representation includes purely functional entities as required to model the architect's intentions, as well as physical entities that substantiate them. The guiding principle behind the representation is that the architectural model is mostly configured story-wise (Leclercq 1996). Therefore, the majority of relationships and constraints between architectural entities take place within stories. Accordingly, as shown in Fig. 2, a functional entity called story acts as a main organizer of the architectural model; all other entities in the architectural model are directly or indirectly related to a story. Space is another key concept in the architecture. There are two complementary ways of defining a space (Björk 1992): implicitly through its boundaries and/or explicitly as a functional entity. The proposed approach adopts the functional view which is important for architects during early design. A space is therefore a functional entity that is specialized into primary space (PS) and secondary space (SS). A (PS) is the smallest space that can be occupied at any given time. SSs group PSs that share some functionality (Thiel 1963). For example, they can be used for zoning the building in order to differentiate private zones from public ones. PSs are further specialized into single-story primary spaces (SSPS) spanning a single story and multistory primary spaces (MSPS) spanning several stories. Even though most of the spaces in a typical building are SSPSs, MSPSs are necessary for modeling large open PSs such as building atriums and conference rooms. In addition, MSPSs are also used for modeling staircases and other vertically continuous shafts. This space specialization allows the later association of structural entities and functions to architectural spaces.

A story is delimited from above (i.e., ceiling) and below (i.e., floor) by physical entities called ASlabs (i.e., the architectural view of the slab). For simplicity, each story accepts only one

ASlab as floor and one as ceiling; however, in the future, stories with many ASlabs at varying elevations are expected to be modeled. Stories also have AWall and AColumn entities. These are physical entities that correspond to architectural views of walls and columns. ASlabs, AWalls, and AColumns are defined by the architect without necessarily being concerned about structural aspects. AWalls encapsulate attributes that specify if they are permanent or removable, translucent, or opaque. These attributes facilitate the work of the engineer in selecting the walls that may play a structural role. However, since these entities are not laid out as part of a functional structural whole they need to be later verified and confirmed by the engineer and then integrated to the structural system. Finally, walls have openings representing windows and doors. These openings are defined by the architect.

Note that unlike Björk (1992), Dias (1996), and Khemlani et al. (1998), this representation does not include explicit relations between spaces and their boundaries. Therefore, ASlabs, AWalls, and AColumns are not related to spaces. Similarly, in the representation AWalls are not explicitly related to each other. Imposing a rigid data-structure limits design flexibility and applicability of the representation for solving generalized problems. For example, the representation acknowledges that AWalls and AColumns not only enclose spaces but they may also partition spaces. Furthermore, in many buildings structural elements and assemblies are positioned inside spaces with reference to their boundaries (offset). Not considering such relationships explicitly also makes the representation simpler. In this way the design model does not get unnecessarily overcrowded with relationships that can be derived from the geometric model using spatial operations.

This approach does not go against the architect's design intent since it has been well established in practice that walls in buildings have the main architectural function of defining and separating spaces (Rosenman and Gero 1996). When implicit, the bounding relationship between walls and spaces, which is apparent from the architectural model, can become explicit on demand using spatial operations. Based on the underlying capabilities of the geometric modeling kernel and those of the StAr representation, such operations use the partial or complete geometry and topology of the design model for constructing new geometry and topology, and for verifying the model. Thus, whenever required, space-enclosing AWalls can always be derived from enclosed spaces or vice versa. AWall adjacencies and intersections can also be evaluated using spatial operations. By contrast, such relationships can become explicit in building environmental design where the quality of the indoor environment is directly related to the characteristics of the ASlabs and AWalls that surround the spaces. For structural engineering, however, maintaining these relationships is expensive and fruitless.

In addition, when modifications to the architecture take place, model consistency can also be maintained through spatial operations that capture the spatial relationships between entities before the modifications take place and modify spatially "related" entities accordingly. For example, if the architect moves an AWall, a spatial operation checks if that AWall was bounding one or more spaces (i.e., before the change took place), then another operation adjusts the geometry of the "related" spaces accordingly.

Structural Domain Representation

The structural domain representation includes both functional and physical entities as required for conceptual structural design, as described earlier. As shown in Fig. 2, independent structural

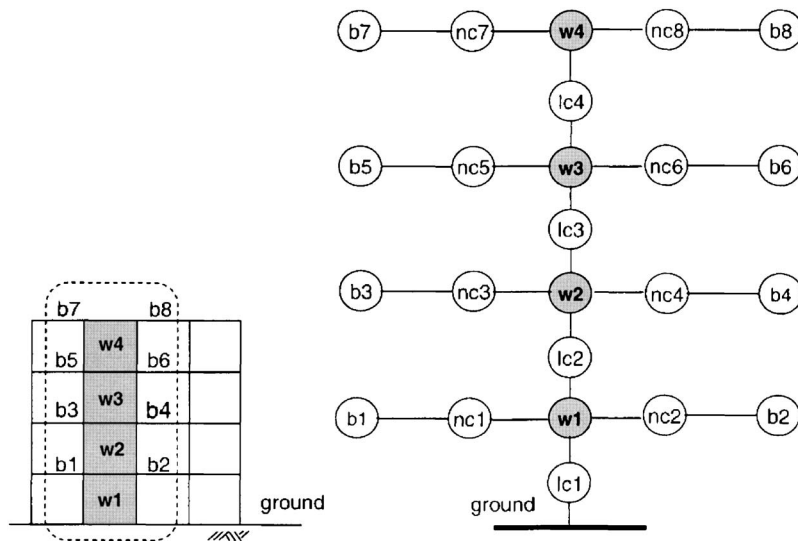


Fig. 3. Partial topology of a frame assembly with a shear wall

volumes are functional entities that act as the main organizers of the structural system. The structural system is subdivided into one or several independent structural volumes, each one containing an independent system of load paths to the ground (i.e., a self-contained structural skeleton). Independent structural volumes also aggregate structural zones, which are functional entities that group spaces and specify additional structural requirements, such as prescribed loads and column spacing. Independent structural volumes and structural zones are volumetric entities and therefore are specializations of structural volumes. Independent structural volumes are also subdivided into three types of structural subsystems: horizontal subsystem, vertical lateral subsystem, and vertical gravity subsystem. Structural subsystems are functional entities that are composed of structural assemblies, which are arrangements of structural elements, and may be composed of other structural assemblies (subassemblies). Structural assemblies are specialized into more specific types such as frame assembly, floor assembly, wall stack, and column stack, which are not shown in Fig. 2 for clarity. The representation includes five types of structural elements, namely: SWall (i.e., the structural view of a wall), SColumn (i.e., the structural view of a column), beam, truss element, and slab element, which is a planar horizontal element that supports gravity forces directly. Structural elements are joined together through structural connections. Hence, at the structural element level, the structural hierarchy expands into a network of interconnections. This type of network is represented as an adjacency structure (Meyer 1995), which is essentially a graph with attributes and behavior at each node. The graph nodes are structural elements and structural connections, which are the indivisible atomic units of the representation hierarchy. Two types of structural connections are incorporated in the representation depending on the dimensionality of the structural entities to be connected—node connection and line connection (these two types are not shown in Fig. 2 for clarity). Fig. 3 shows the partial topology of a frame assembly with a shear wall. In Fig. 3, “w” stands for wall, “b” stands for beam, “lc” indicates line connection, and “nc” indicates node connection.

Integration of the Building Architecture and the Structural System

The integration between the architectural model and the structural system is carried out through a set of entities and relationships that link the two models at different levels of refinement (see the gray zone in the center in Fig. 2). The first link between the architectural model and the structural system takes place between purely functional entities—that is, between spaces and structural volumes. When spaces are grouped into structural zones the functionality of the spaces is translated into structural requirements and constraints that limit the synthesis of structural configurations. This is achieved through structural layout constraints that allow the architect to restrict the layout of structural elements within certain spaces. Examples of structural layout constraints are the column_free constraint, the beam_free constraint (e.g., limiting visible beams spanning MSPSPs), and the floor_to_ceiling_height constraint (i.e., limiting the depth of floor assemblies). These constraints incorporate some architectural design intent into the model that may not be obvious to the structural engineer otherwise.

The next link takes place at the structural assembly level. From the architectural view of a slab (i.e., the ASlab) the engineer generates one or more floor assemblies. Usually, the geometry of a floor assembly is limited by the independent structural volume where it belongs. However, if an ASlab overlaps structural zones with dissimilar support requirements, then it may require division by the engineer into various floor assemblies.

The lowest links between the two disciplines take place through physical entities. Walls are architectural elements that can be given structural roles, and therefore have two views. The architectural view (AWall) holds information that is relevant to the architect—for example, linking to a given story—while the structural view (SWall) holds structural information only, as well as its relationship to a wall stack. Columns are structural elements (SColumn) that may also play an architectural role (AColumn); therefore they also have two views. Similarly, when AColumns are tentatively laid out by the architect, if they are structurally relevant once verified by the engineer, they can be aggregated into column stacks and integrated to the structural system.

Representation of Spatial Information

Building elements need several spatial representations depending on their degree of refinement, as well as their domain view (Martini and Powell 1990; Zamanian 1992; Rosenman and Gero 1996). IFC (IAI 2005) and CIS/2 (SCI 2005) standards provide multiple geometric representations for the entities they represent. For example, for any structural element and connection CIS/2 provides one geometric description for structural analysis, another description for design, and yet another description for fabrication. One approach for handling the multiple geometric descriptions of an entity is to have a single fundamental representation from which all required spatial abstractions can be derived. Zamanian (1992) calls this single fundamental representation a primary spatial representation, while the spatial abstractions are called secondary spatial representations. The primary spatial representation captures the topologic relations among facility entities, which are invariant over the different spatial abstractions of those entities. The primary spatial representation, also called a skeletal representation, can be regarded as a mixed-dimensional wire-frame, where two-dimensional slabs and walls coexist in the same model with one-dimensional beams and columns. Since the skeletal representation captures the topologic relations of entities it facilitates reasoning about topology. The skeletal representation also facilitates reasoning about geometry at the conceptual stage since structural elements usually have one or two salient dimensions that are sufficient for characterizing them within the structural configuration—that is, the cross section of structural elements does not affect their relative position in space.

Modern commercial structural engineering applications provide from the outset detailed geometric descriptions of structural elements and connections. Maintaining these detailed geometric descriptions is computationally expensive and fruitless for conceptual structural design where simplified geometric descriptions are sufficient. Martini and Powell (1990) proposed a top-down approach for the description and refinement of structural elements that starts as a skeleton of line segments, then the beam schedule fills in depths, but leaves the component and end conditions undefined; finally, the details define the specific shapes of components.

Consequently in this research, architectural and structural building elements are spatially represented with a mixed-dimensional primary spatial representation, which is sufficient for conceptual structural design. Considering architectural entities, spaces are represented using a primary spatial representation as 3D volumes, walls and slabs are represented as bounded 2D planes (planes bounded by space enclosures), and AColumns are represented as 1D wires. Correspondingly, the decomposition of structural entities is also supported by different geometric representations: structural volumes are represented as 3D entities; structural subsystems do not have geometry of their own, but their geometry is defined through aggregation of their constituent structural assemblies, which are represented as bounded 2D planes (planes bounded by the enclosures of structural volumes) for wall stacks, floor assemblies, and frame assemblies; and 1D wires for column stacks. Therefore, structural assemblies have an abstract geometry (a bounded plane) that defines their position, orientation, and extension, and a specific geometry that is defined by the geometry of their constituent structural elements. At the structural element level, slab elements and walls are represented as 2D planar entities, while beams and columns are represented as 1D wires.

Knowledge and Constraints

Structural entities incorporate a basic description of themselves and how they interact with other entities. So far, structural entities incorporate knowledge about the structural form, while knowledge about structural function and expected behavior is not included. The initial intent is therefore to provide a basic level of assistance to competent structural engineers that have a thorough understanding of the structure's function and behavior, which they use to produce the structural form. Once configured by a competent structural engineer, a structural form can be transferred to structural analysis programs for assessing its behavior.

Structural entities incorporate knowledge about the structural form as object-oriented methods that constrain their configuration as well as the configuration of related structural entities, thus reducing the range of configuration possibilities (Gero 1990). Such methods rely mostly on geometric and topologic concerns and on generic structural engineering knowledge. The types of constraints that are encapsulated by the entities in the representation are feasibility or hard constraints because they are enforced in order to produce feasible structural solutions. Since the structural system is hierarchically described, entity interactions take place vertically between abstract entities and their constituents, and horizontally among entities at each hierarchical level. Vertically, parent entities verify that their child entities are well connected and well positioned within the spatial limits defined by the parent entity. Thus, the integrity of the structural system is guaranteed mostly by part-whole constraints that are supported via spatial operations, which rely mostly on the low-level non-regularized Boolean operations available in the underlying geometric modeling kernel. First, independent structural volumes constrain the geometry of structural assemblies so that they lie within the volume that they contain. Structural subsystems limit the types of structural assemblies that are required to make the subsystem behave as expected. Structural zones constrain the type and location of structural assemblies that they enclose. At lower levels of structural decomposition, a frame assembly restricts the location of its constituent structural elements to be within its plane (geometric, planarity constraint). In this way, vertical structural assemblies provide vertical continuity constraints that connect stories at suitable locations. Horizontally, structural elements also incorporate structural knowledge to verify if they are properly related to other structural elements (i.e., properly supported). Hence, vertical structural assemblies enforce valid arrangements of the structural elements that make them up, while respecting the horizontal interaction constraints coming from their constituent elements. For example, a joist may be directly supported by a column; however, in a three-level (slab/joist/beam) floor assembly, joists must be supported by primary beams (topologic and associativity constraint). In addition, according to its type, each particular structural assembly requires specific types of structural connections as well as allowable spans for its structural elements.

StAr Prototype

The StAr prototype has been implemented in Visual C++. It includes the following components in over fifty object-oriented classes:

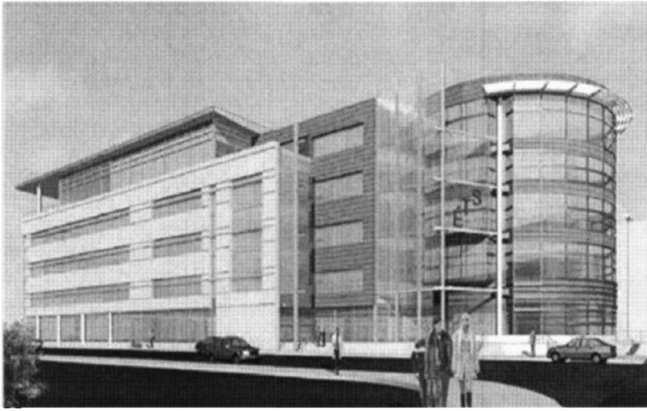


Fig. 4. Architectural rendering of the ETS building

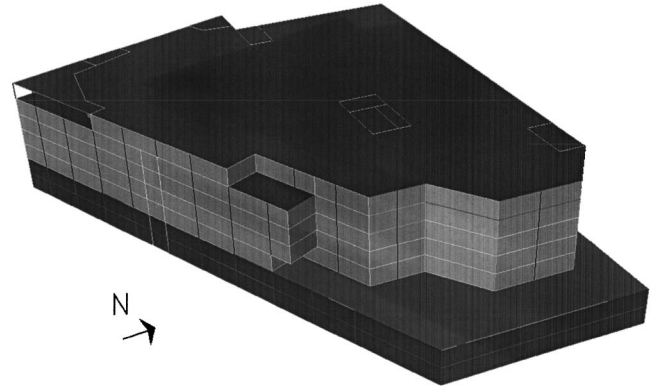


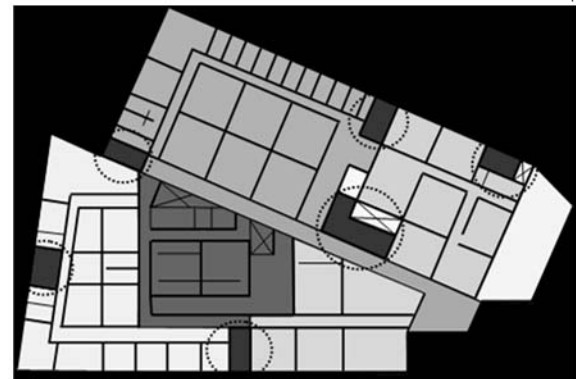
Fig. 5. Architectural model generated using the software prototype

- *Alphanumeric user interface*: allows the architect to enter the early architectural design and the engineer to define the higher-level entities of the structural system hierarchy via input text file or an alphanumeric menu;
- *Top-down design manager*: keeps track of the sequence of design activities for enabling top-down design synthesis;
- *StAr representation* (cf. Fig. 2): core component of the prototype that is used as a schema to produce the design model of the building integrating the architectural model and the structural system;
- *Geometric modeling kernel*: the entire system is built on top of ACIS (Spatial Corp. 2005), which incorporates low-level data structures and algorithms to support mixed-dimensional geometric modeling;
- *Synthesis algorithms*: assist the engineer in synthesizing the structural system; they rely on geometric modeling techniques and on knowledge encapsulated in the entities of the StAr representation to allow the engineer to reason based on the geometry and topology of the model being created (For example, the algorithms assist the engineer while inspecting the architectural model in verifying the vertical continuity of walls and columns, detecting large setbacks and cantilevers, etc. They also provide assistance in finding supports for structural elements and in verifying gravity and lateral load paths to the ground. Furthermore, based on the entities instantiated at higher levels of the structural hierarchy, synthesis algorithms generate structural elements and structural connections under the guidance and supervision from the engineer. This paper, however, does not elaborate more on the synthesis algorithms since its focus is on the representational aspects of the conceptual design problem.); and
- *Visualization interface*: commercial application HOOPS (TSA Inc. 2005) is used for 3D visualization.

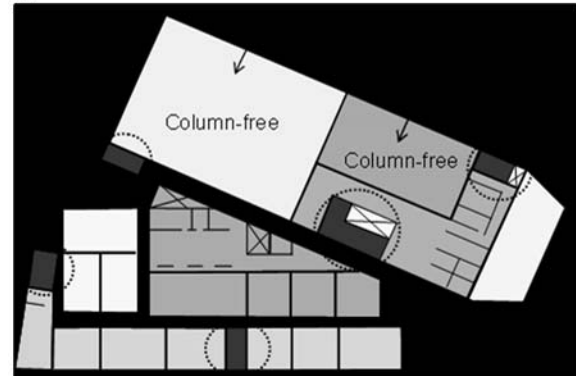
In the next section the prototype StAr is used to demonstrate how the computer representation can support the different activities that take place during conceptual design of building structures.

Test Example

The test example is the new building for the *École de Technologie Supérieure* (ETS) in Montreal. The building has five stories above ground (which are labeled ground, first, second, third, and fourth), occupying an area of 20,400 m², and two parking stories below ground. The building houses several activities, such as



(a)



(b)

Fig. 6. ETS building: (a) second story and (b) third story

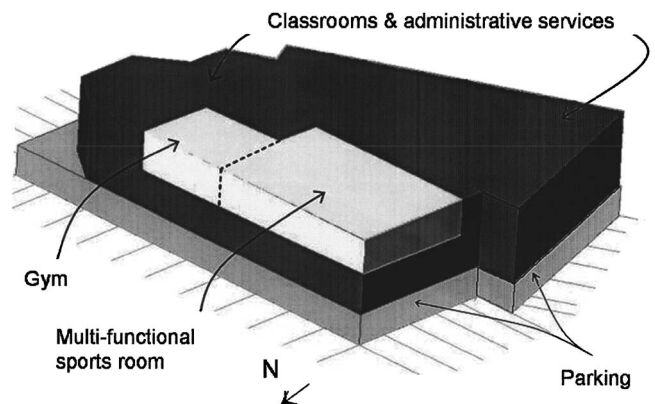


Fig. 7. Structural zones

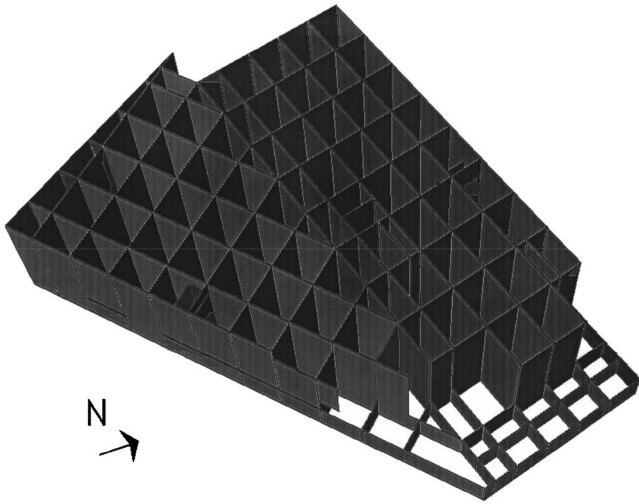


Fig. 8. Abstract geometry of frame assemblies

classrooms, computer rooms, sports facilities, a cafeteria, and various administrative services. Fig. 4 shows an architectural rendering of the building.

For the purpose of testing StAr, the shape of the circular façade had to be represented by a number of straight lines because the prototype is currently limited to polygons. The adjusted architectural model is illustrated in Fig. 5. The model has been constructed as described in StAr representation section of this paper. The model also incorporates the reference grids as initially defined by the architect.

Fig. 6(a) shows a simplified view of the second story of the building, and Fig. 6(b) shows the third story, as entered in StAr. In Fig. 6, circles are used to illustrate core AWalls that have been selected for structural purposes.

A gym and a multifunctional sports room are located in the third story, but they actually span the third and the fourth stories. These two rooms are large and column-free requiring particular structural considerations, which must integrate well with the rest of the structure.

The structural system is a reinforced concrete structure with a vertical gravity subsystem consisting of columns and walls, a vertical lateral subsystem consisting of rigid frames and shear walls acting together, and a horizontal subsystem consisting of flat slabs with drop panels. The building has six vertical circulation cores, which are circled in Fig. 6(a) and used for structural purposes. All the core walls rest on the foundations located at the second basement level. However, following architectural functional requirements, some of these walls are interrupted before reaching the last story. The two parking levels below ground are surrounded by perimeter retaining walls. The roof of the building is a steel structure supported by beams and girders, except for the roof of the gym and the multifunctional sports room that is supported by long-span trusses.

After inspecting the architectural model (size, shape, proportions, space patterns, and dimensions), the engineer decides that it can be supported by a single structural skeleton. Therefore, one independent structural volume is created by grouping the spaces in the building that make up its entire volume. Then, structural subsystems are defined, namely: horizontal subsystem, vertical gravity subsystem, and vertical lateral subsystem. Each structural subsystem is defined initially by specifying material(s) and types of assemblies, elements, and connections that are most likely

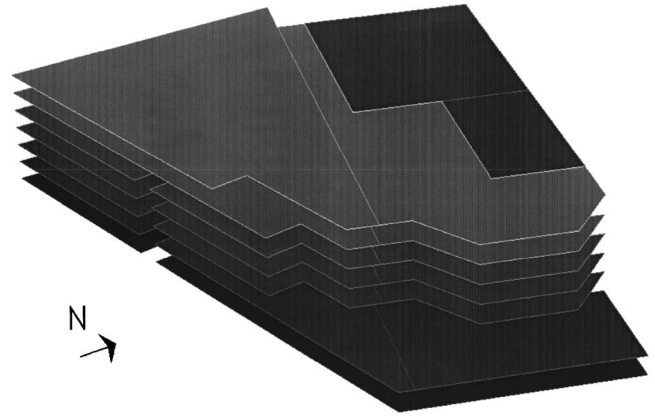


Fig. 9. Abstract geometry of floor assemblies

required for making it behave as expected. Later on each structural subsystem will be populated with its constituent assemblies and elements. Spaces having similar structural characteristics are grouped into structural zones. Structural layout constraints associated to spaces by the architect participate to the definition of structural zones. In Fig. 7, the independent structural volume representing the structural system at this stage is divided into three structural zones: a parking zone, a zone with classrooms and administrative services, and a zone that houses the gym and a multifunctional sports space. Each structural zone is then assigned corresponding applied loads. The last structural zone (indicated with the lighter shade of grey in Fig. 7) groups spaces that are column free. Thus, it carries a column-free structural layout constraint, as specified by the architect, to restrict the layout of the vertical gravity subsystem.

Once structural volumes and structural subsystems are defined, the configuration of structural assemblies begins. Initially, a structural assembly is spatially represented with an abstract geometry, which is a bounded 2D plane (a plane bounded by the enclosure of a structural volume) that determines its layout and extent. Later on in the process, structural assemblies are populated with structural elements and structural connections. Thus, the engineer begins by laying out frame assemblies following closely the patterns in the building architecture and respecting spaces functionality. This is done simply by specifying two x,y points

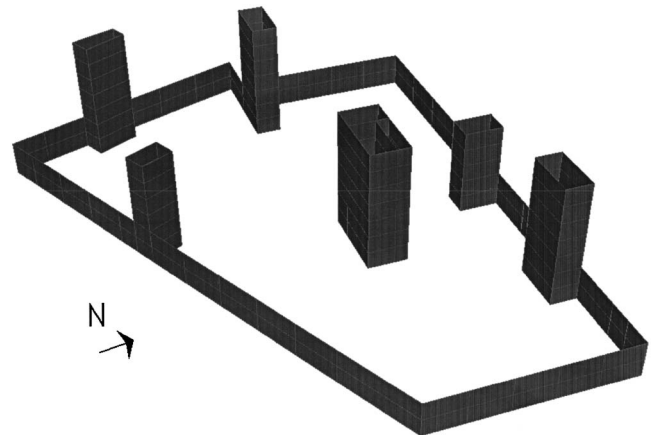


Fig. 10. AWalls from the architecture that become structural SWalls and wall stacks

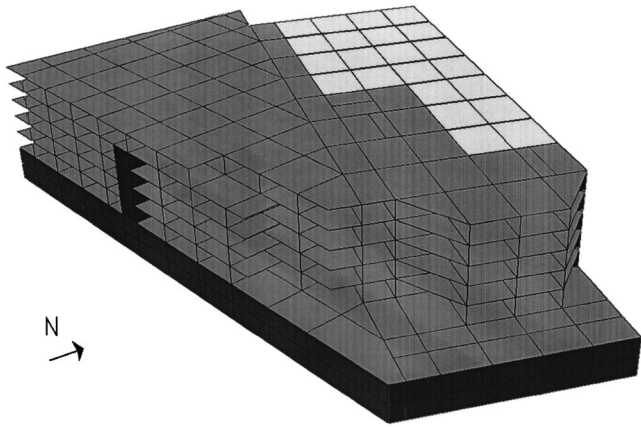


Fig. 11. Resulting structural system

defining a frame alignment. The abstract geometry of each frame assembly, which is a bounded vertical plane, is then defined within the boundaries of its parent independent structural volume. By laying out frame assemblies in both principal structural directions the engineer implicitly defines column lines and structural bays. Once laid out, each frame assembly is then integrated to its corresponding vertical lateral subsystem. Fig. 8 shows the abstract geometry of all frame assemblies.

ASlabs from the building architecture also participate in the synthesis of the structural system. Independent structural volumes and structural zones may break down ASlabs based on structural considerations. In this case, the structural zone that groups the gym and the multifunctional sports room, which are column-free spaces, requires a special type of roof assembly. Floor assemblies are given an abstract geometry initially consisting of a bounded horizontal plane. Once laid out by the engineer, each floor assembly is automatically integrated to the horizontal subsystem. Fig. 9 shows the floor assemblies that make up the horizontal subsystem.

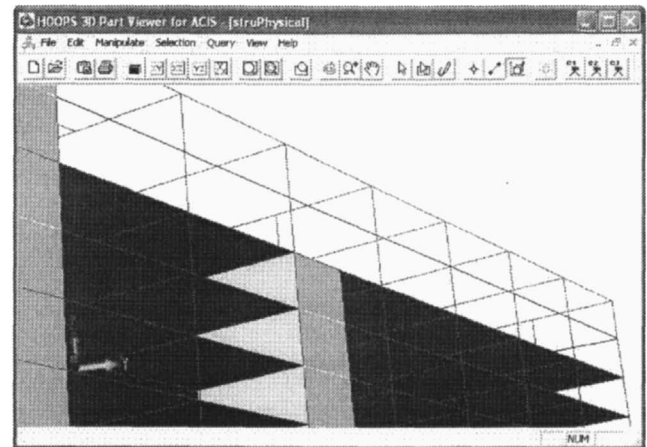
At this point the engineer decides to integrate AWalls to the structural system so that they become SWalls, which are then grouped into wall stacks (see Fig. 6 and 10). Then, the computer integrates each wall stack to a coplanar frame assembly, if applicable, and to the corresponding vertical lateral subsystem and vertical gravity subsystem. Perimeter AWalls at the two basement levels also become part of the structural system as retaining SWalls.

AColumns, if placed tentatively by the architect, also participate in the synthesis of the structural system. These AColumns become structural provided that they are useful to the structural needs, thus becoming column stacks, and are properly aligned horizontally, thus providing feasible spans for supported floor assemblies. In this example, all the columns are assumed to be laid out by the engineer.

Once structural assemblies are laid out and relevant architectural elements have been integrated to the structural system, synthesis algorithms intersect the abstract 2D planar geometries of structural assemblies and generate beams and columns. Vertically continuous columns, which are placed in column lines at the intersection between two frame assemblies, become column stacks, which are part of the vertical gravity subsystem. Primary beams are obtained by intersecting the abstract geometry of the frame assemblies and the floor assemblies. Column stacks, wall stacks, and primary beams make up frame assemblies. Each



(a)



(b)

Fig. 12. Structural system: (a) actual building under construction and (b) partial view generated by StAr

frame assembly is integrated to its corresponding vertical lateral subsystem, which then gets populated. Primary beams are then used by synthesis algorithms for generating slab elements. Fig. 11 shows the resulting structural system, which is described as a hierarchical organization of structural entities that responds to architectural functional and physical design concerns.

Fig. 12(a) shows the structural system under construction with the steel structure for the multifunctional sports room and the gym supported by a concrete structure underneath. It shows large steel trusses spanning a column-free structural zone. Fig. 12(b) shows the same portion of the structure as generated by the StAr prototype under the engineer's guidance and supervision.

Evaluation

The StAr representation fulfills the requirements for conceptual design as follows:

1. It integrates multiple functional views for the different participants involved—the representation integrates both architectural and structural views;
2. It supports design refinement/evolution by means of a hierarchical structural decomposition; as a consequence, the representation allows reasoning in terms of functional subsystems and assemblies, as required during conceptual structural design; and

3. It facilitates the exploration of alternative design solutions by supporting hierarchical structural synthesis and by allowing the engineer to backtrack to any level of the structural hierarchy to follow another path.

Communication/coordination errors are also minimized because the representation lets the engineer directly inspect the building architecture and use relevant architectural entities and constraints for structural synthesis while providing informed feedback to the architect at each level of the structural hierarchy. Furthermore, enabling the structural system to be synthesized within a building architectural context allows the engineer to become more easily aware of architectural constraints and potential structural opportunities. Since architectural functional requirements become explicit (through structural layout constraints) these cannot be overlooked and therefore must be taken into consideration by the engineer.

The main limitation of the current representation is that it lacks mechanisms for maintaining the consistency of the design model when changes are required as a result of the two-way feedback between architects and engineers. In particular, the following capabilities are required:

- A parametric mechanism for incorporating the architect's intent that is not apparent from the geometry and topology of the design model, and therefore cannot be derived from the model through spatial operations only; and
- A reasoning mechanism for maintaining the intent of both the architect and the engineer, whether explicit or implicit in the design model, as well as the consistency of the entire model when changes take place.

These will be explored in future research. In its current state, changes to the building architecture need to be done manually by the architect while maintaining the intended functionality and consistency of the affected architectural entities. When the structural system is being partially or completely integrated to the building architecture, changes demand either backtracking or reinitiating the whole structural top-down process. Backtracking is necessary when a conflict, resulting from a design change, cannot be handled at a current hierarchical level, and therefore needs to be solved at a higher level. Reinitiating the structural system layout process is necessary when drastic changes to the architecture take place. This is facilitated by the top-down approach in which the engineer needs only to redefine higher-level structural entities while the computer handles the tedious task of configuring structural elements.

Considering the structural system geometry, even though the skeletal representation is adequate for conceptual design, improved support for decision making between architects and engineers can be achieved by providing approximate dimensions of structural members. A mechanism is therefore required for the expansion of the primary spatial representation into a simplified secondary spatial representation as proposed by Zamanian (1992). A design knowledge system is currently being developed that will take into consideration geometry, applied loads, material(s), cost, compatibility, and constructability for the selection and initial sizing of feasible structural assemblies and elements.

Future Work

In the next stage of this research, two-way translations between the StAr representation for conceptual structural design and the IFC model will be developed with the following purposes:

- To receive architectural models from early architectural design applications; and
- To transfer structural entities to subsequent structural analysis and design applications.

On the one hand, for the building architectural domain, most entities from the StAr representation have corresponding definitions in the IFC model: site (*ifcSite*), story (*ifcBuildingStory*), space (*ifcSpace*), wall (*ifcWall*), column (*ifcColumn*), and slab (*ifcSlab*). In addition the IFC model provides an entity (*ifcZone*) that aggregates spaces. The grids also have their corresponding IFC definitions (*ifcGrid*). Still, structural layout constraints are not explicitly defined in the IFC model. Constructing the architectural model from the IFC model without structural layout constraints is expected to be straightforward; however, as discussed previously, these constraints are paramount for transferring architectural design intent to the engineer. Further research is required to study if IFC classes could be utilized to derive those constraints.

On the other hand, for the structural system domain only the physical structural entities from the StAr representation have corresponding definitions in the IFC model. Thus, considering the physical structure, the IFC model describes structural elements (*ifcStructuralMember*) and structural connections (*ifcStructuralConnection*), including their materials (*ifcMaterial*). Besides the topological description of the physical structure, the IFC model also defines explicitly a set of entities for describing the structure's function (*ifcStructuralLoad* and *ifcStructuralAction*) and behavior (*ifcStructuralReaction*) as required for structural analysis. Considering purely functional structural entities, structural volumes and structural subsystems are not currently used by subsequent structural analysis and design software and therefore they do not need to be translated to the IFC model. However, structural assemblies are usually required for detailed design and fabrication (for steel structures). Further research is required to verify if structural assemblies can be derived from the IFC model using its underlying capabilities for defining and relating new entities through inheritance. In general, conceptual design descriptions of structural entities are simplified and therefore these descriptions, for the entities that need translation to analysis and design packages, are expected to be a subset of the subsequent more specialized analysis and design descriptions. Due to the above considerations, the required two-way information exchange between the StAr representation and the IFC model is expected to be straightforward.

Conclusions

A computer representation called StAr has been described that aims to act as a common ground for collaboration between architects and engineers during conceptual structural design. The StAr representation is semantically explicit since it incorporates essential concepts from the building architecture and the structural system as required during conceptual structural design. On the one hand, the description of the building architecture minimizes explicit relationships among entities for simplicity and representational flexibility, while taking advantage of the geometric operations provided by the underlying geometric modeling kernel. On the other hand, the hierarchical description of the structural system enables a top-down design approach, where purely functional entities facilitate the spatial configuration of the structural elements. The structural hierarchical description also facilitates the integration of the structural system to the building

architecture at various levels. This approach is in sharp contrast to current computer applications for structural design, as well as current AEC information-exchange standards, where the emphasis is placed on the detailed modeling of physical elements and connections. With this work, it is therefore possible to feed current AEC information exchange standards to integrate conceptual structural design and pass the results to existing structural engineering applications.

The StAr representation has been successfully tested to demonstrate its capabilities in assisting activities that take place during conceptual design. However, further work is required to tackle the following issues: (1) facilitating changes as a result of the two-way feedback between architects and engineers; and (2) providing more complete descriptions of structural form and function for more informed decision making during conceptual structural design. Research is currently underway in order to come up with a more intuitive and collaborative conceptual structural design environment.

Acknowledgments

The writers wish to thank the Natural Science and Engineering Research Council of Canada (NSERC), the Fonds Québécois de Recherche sur la Nature et les Technologies (FQRNT) of the province of Quebec, as well as the collaborative research project Emerging Information Technologies for Architecture, Engineering, Construction, and Facilities Management in Canada (ITAC), funded by NSERC, for financially supporting this research.

References

- Biedermann, J., and Grierson, D. (1995). "A generic model of building design." *Eng. Comput.*, 11(3), 173–184.
- Björk, B. C. (1989). "Basic structure of a proposed building product model." *Comput.-Aided Des.*, 21(2), 71–78.
- Björk, B. C. (1992). "A conceptual model of spaces, space boundaries and enclosing structures." *Autom. Constr.*, 1(3), 193–214.
- Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The unified modeling language user guide*, Addison-Wesley, Reading, Mass.
- Datta, S., and Woodbury, R. F. (2004). "Feature nodes: An interaction construct for sharing initiative in design exploration." *Proc., Design Computing and Cognition'04*, J. S. Gero, ed., Kluwer, Dordrecht, The Netherlands, 23–36.
- Dias, W. P. S. (1996). "Multidisciplinary product modeling of buildings." *J. Comput. Civ. Eng.*, 10(1), 78–86.
- Fenves, S. J., Rivard, H., and Gomez, N. (2000). "SEED-config: A tool for conceptual structural design in a collaborative design environment." *Artif. Intell. Eng.*, 14(3), 233–247.
- Fuyama, H., Law, K. H., and Krawinkler, H. (1997). "An interactive computer-assisted system for conceptual structural design of steel buildings." *Comput. Struct.*, 63(4), 647–662.
- Gero, J. (1990). "Design prototypes: A knowledge representation schema for design." *AI Mag.*, 11(4), 26–36.
- Haymaker, J., Fischer, M., and Kunz, J. (2002). "Perspectors." *Proc., Artificial Intelligence in Design'02*, J. S. Gero, ed., Kluwer, Dordrecht, The Netherlands, 593–615.
- International Alliance for Interoperability (IAI). (2005). "Extension projects for IFC." (<http://www.iai-international.org/projects/extensionprojects.html>) (December 2005).
- Khemlani, L., Timerman, A., Benne, B., and Kalay, Y. (1998). "Intelligent representation for computer-aided building design." *Autom. Constr.*, 8(1), 49–71.
- Law, K., Barsalou, T., and Wiederhold, G. (1990). "Management of complex structural engineering objects in a relational framework." *Eng. Comput.*, 6(2), 81–92.
- Leclercq, P. (1996). "Environnement de conception architecturale préintégré: Eléments d'une plate-forme d'assistance basée sur une représentation sémantique." PhD thesis, Faculty of Applied Sciences, Univ. de Liège, Liège, Belgium.
- Lin, T. Y., and Stotesbury, S. D. (1988). *Structural concepts and systems for architects and engineers*, 2nd Ed., Van Nostrand Reinhold, New York.
- Martini, K., and Powell, G. H. (1990). "Geometric modeling requirements for structural design." *Eng. Comput.*, 6(2), 93–102.
- Meyer, S. (1995). "A description of the structural design of tall buildings through the grammar paradigm." PhD thesis, Engineering Design Research Center, Dept. of Civil Engineering, Carnegie Mellon Univ., Pittsburgh.
- Rivard, H., and Fenves, S. J. (2000). "A representation for conceptual design of buildings." *J. Comput. Civ. Eng.*, 14(3), 151–159.
- Rosenman, M. A., and Gero, J. S. (1996). "Modeling multiple views of design objects in a collaborative CAD environment." *Comput.-Aided Des.*, 28(3), 193–205.
- Sacks, R., Eastman, C. M., and Lee, G. (2004). "Parametric 3D modeling in building construction with examples from precast concrete." *Autom. Constr.*, 13(2), 291–312.
- Sacks, R., and Warszaiwki, A. (1997). "A project model for an automated building system: Design and planning phases." *Autom. Constr.*, 7(1), 21–34.
- Sause, R., Martini, K., and Powell, G. H. (1992). "Object-oriented approaches for integrated engineering design systems." *J. Comput. Civ. Eng.*, 6(3), 248–265.
- Scherer, R. J., and Gehre, A. (2000). "Approach to a knowledge-based design assistant system for conceptual structural system design." *Proc., ECPPM 2000, Product and Process Modeling in Building and Construction*, R. Gonçalves, A. Steiger-Garçao, and R. J. Scherer, eds., Balkema, Rotterdam, The Netherlands, 229–238.
- Steel Construction Institute (SCI). (2005). *Online previews of CIMsteel integration standards release 2*, 2nd Ed., (http://www.cis2.org/documents/cis2_docs.htm) (December 2005).
- Spatial Corp. (2005). "3D components for modeling, interoperability, and visualization." *Spatial Corp.*, (<http://www.spatial.com>) (December 2005).
- Stouffs, R., and Krishnamurti, R. (2004). "Data views, data recognition, design queries and design rules: Representational flexibility for design." *Proc., Design Computing and Cognition'04*, J. S. Gero, ed., Kluwer, Dordrecht, The Netherlands, 219–238.
- Thiel, P. (1963). "Notes on environmental space and elementary space notation." College of Architecture and Urban Planning, Univ. of Washington, Seattle.
- TSA Inc. (2005). "Graphic toolkits for engineering applications." Tech Soft America, (<http://www.hoops3d.com/>) (December 2005).
- Turk, Z. (2001). "Phenomenological foundations of conceptual product modeling in architecture, engineering and construction." *Artif. Intell. Eng.*, 15(2), 83–92.
- Zamanian, M. (1992). "Modeling and communicating spatial and functional information about constructed facilities." PhD thesis, Dept. of Building, Civil, and Environmental Engineering, Carnegie Mellon Univ., Pittsburgh.