# INTERACTIVE COMPUTER SUPPORT FOR THE CONCEPTUAL DESIGN OF BUILDING STRUCTURES

**Mora R.[1], Bédard C.[2], Rivard H.[3]**

## ABSTRACT

This paper proposes an interactive computer application to support conceptual structural design. It follows a synthesis-by-problem-decomposition approach in which a structural model is created by progressively refining a design solution from a global abstract level to well-defined sub-problems. Interactivity is achieved by enabling manipulation of building architectural design and interactive decision-making by the engineer at different levels of structural refinement, so that initial abstract decisions lead to subsequent detailed decisions. A first software prototype called StAr (Structure-Architecture) has been developed that provides basic support for interactive conceptual structural design. A second version of the prototype is being developed that includes a design knowledge manager and a graphical user interface (GUI). This paper focuses on the GUI design.

## KEYWORDS

Conceptual structural design, computer-assisted building design, interactive design, GUI

## INTRODUCTION

Computer support for conceptual design of building structures is still in its infancy. On the one hand, commercial structural design applications fail to provide adequate support because they operate only at a detailed level and minimize interactions with the building architectural design. On the other hand, AI (artificial intelligence)-based applications tend to work at a planning level and automate design decisions. This paper proposes a third alternative, an interactive computer application to support conceptual structural design. It follows a synthesis-by-problem-decomposition (i.e. top-down) approach in which a structural model is created by progressively refining a design solution from a global abstract level to well-defined sub-problems. Interactivity is achieved by enabling manipulation of building architectural design and interactive decision-making by the engineer at different levels of structural refinement, so that initial abstract decisions lead to subsequent detailed decisions.

The paper is organized as follows: the next section summarizes relevant research in assisting conceptual structural design. Then, the proposed approach for interactive support is presented, followed by a description of the current prototype implementation and the components required for proper engineer-computer interactions. Next, the functional design

---

[1] Postdoctoral Fellow, Dept. of Construction Engineering, ETS, 1100 Notre-Dame St. West, Montreal, H3C 1K3, Canada, Phone 514/396-8667, FAX 514/396-8584, rodrigo_mora@sympatico.ca

[2] Professor and Dean of Research and Technology Transfer, ETS, 1100 Notre-Dame St. West, Montreal, H3C 1K3, Canada, Phone 514/396-8829, FAX 514/396-8525, claude.bedard@estmstl.ca

[3] Professor, Dept. of Construction Engineering, ETS, 1100 Notre-Dame St. West, Montreal, H3C 1K3, Canada, Phone 514/396-8667, FAX 514/396-8584, hugues.rivard@estmstl.ca

of the graphical user interface is presented followed by a scenario of interactive support for conceptual structural design.

## LITERATURE REVIEW

Over the last three decades researchers have applied artificial intelligence (AI) techniques to assist engineers in exploring design alternatives over a vast array of possible solutions under constraints. Relevant techniques and examples are the following: expert systems (Maher 1988, Bédard and Ravi 1991), formal logic (Jain, Krawinkler and Law 1991, Eisfeld and Scherer 2003), grammars (Meyer 1995), case-based reasoning (CBR) systems (Bailey and Smith 1994, Kumar and Raphael 1997), evolutionary algorithms (Sisk, Miles and Moore 2003, Rafiq, and Mathews and Bullock 2003) and hybrid systems that combine AI techniques such as a CBR system with a genetic algorithm (Soibelman and Peña-Mora 2000).

The impact of AI-based methods in design practice is negligible mainly because many of the proposed systems are standalone with no interactions with design representations currently employed in practice, such as building information models (BIM). In fact, only three of the above research projects (Meyer 1995, Bailey and Smith 1994 and Kumar and Raphael 1997) use architectural models with 3D geometry as input for structural synthesis. In the absence of such models, only “rough” approximate gravity and lateral load transfer solutions can be explored to satisfy overall building characteristics and requirements. However, these solutions need actual architectural models to be substantiated and validated.

Several of the research projects described above rely on a top-down design approach. However, the top-down design approach proposed in this paper differs from those of previous research projects in that the proposed approach has been conceived to enable the conceptual structural design process to be carried out within a building architectural context. This approach is described in the next section.

## INTERACTIVE SUPPORT FOR CONCEPTUAL STRUCTURAL DESIGN

Engineering design is a synthesis, analysis and evaluation (SAE) process. During conceptual structural design, the engineer assembles and compares feasible structural solutions to transfer loads to the ground safely and efficiently within a building architectural context (synthesis). The decisions made by the engineer are based mostly on incomplete knowledge about structural behaviours and experience on the applicability of available construction technologies and materials to different design situations. Thus, due to the limited availability of time and the inherent vagueness of information, knowledge and experience suffice at this stage, and the evaluation of design alternatives is mostly based on weighting factors and simplified analysis. Conceptual structural design therefore involves mainly synthesis, with simplified analysis and evaluation.

Three main sub-processes are carried out by engineers for the synthesis of structural solutions during conceptual structural design: (1) inspection of the building architecture, (2) configuration, and (3) verification of structural solutions. During the inspection of the building architecture the engineer searches for load paths to the ground and detects problems and constraints in the architectural design. Aware of opportunities, problems and constraints in the architectural design the engineer performs the configuration of structural solutions. A top-down process model is used for structural system configuration which is based on a model proposed by Rivard and Fenves (2000). To implement this approach the structural system is described as a hierarchy of entities where abstract functional entities, which are defined first, facilitate the definition of the constituent entities. After complete structural solutions have been synthesized, the computer verifies the gravity and lateral load paths to the ground.

Figure 1 illustrates the configuration of structural solutions. Activities are shown in rectangles, bold arrows pointing downwards indicate a sequence between activities, arrows pointing upwards indicate backtracking, and two horizontal parallel lines linking two activities indicate that these can be carried out in simultaneously. For clarity, courier bold 10 point typeface is used to identify structural entities. As shown in Figure 1, the structural engineer first defines independent structural volumes holding self-contained structural skeletons that are assumed to behave as structural wholes. These volumes are in turn subdivided into smaller sub-volumes called structural zones that are introduced in order to allow the definition of structural requirements corresponding to architectural functions (i.e. applied loads, allowed vertical supports and floor spans). Independent structural volumes are also decomposed into three structural subsystems, namely the horizontal, the vertical gravity, and the vertical lateral subsystems (the foundation subsystem is not considered in this research project). Each of these structural subsystems is further refined into structural assemblies (e.g. frame and floor assemblies), which are composed of structural elements and structural connections. The arrangement of structural elements and structural connections makes up the "physical structural system".

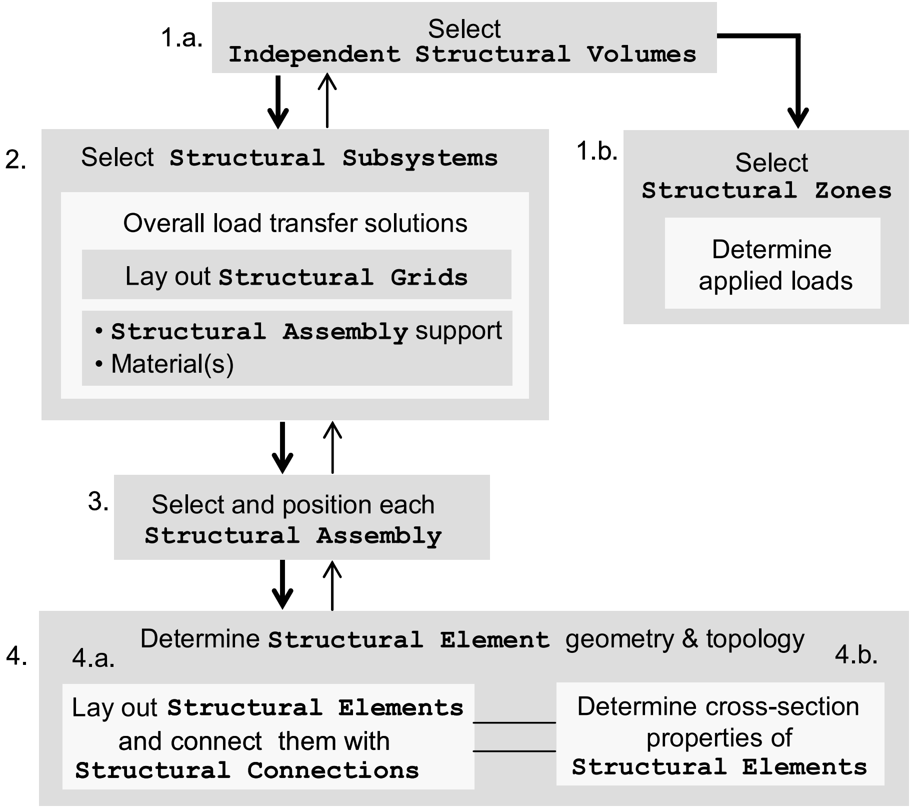

Figure 1. Configuration of structural solutions

During activity number 2 in Figure 1 (i.e. select structural subsystems), the engineer defines overall load transfer solutions described in terms of supporting structural assemblies with corresponding material(s), located on tentative structural grids. An example of a structural solution at the subsystem level is the following: for a 9 m by 12 m structural grid, provide steel rigid frames for lateral support in the long building direction, steel braced frames for lateral support in the short direction, columns for vertical gravity support, and composite steel deck on W shape beams for horizontal gravity support. Structural grids determine tentative vertical supports (at gridline intersections), structural bays, likely floor framing directions, and floor spans. At this stage, the engineer seeks to define, if possible, uniform structural grids for the entire building.

Interactivity is intended between the structural engineer and a simplified (for conceptual design) model of the building architecture and the structural system, architecture-structure model (ASM). Such interactivity is required during each of the abovementioned processes, sub-processes and activities, so that from a simplified building architectural model (AM) the engineer is able to integrate structural solutions with the progressive use of knowledge. Evaluation of structural solutions takes place mainly during activities 2, 3, and 4 of the configuration sub-process.

## IMPLEMENTATION OF THE PROPOSED APPROACH

The implementation of the approach is based on an existing prototype for conceptual structural design which is called StAr (Structure-Architecture), a prototype system that assists engineers in the inspection of a 3D architectural model (e.g. while searching for continuous load paths to the ground) and the configuration of structural solutions. Assistance is based on geometrical reasoning algorithms (GRA) (Mora et al. 2006B) and an integrated architecture-structure representation model (ASM) (Mora et al. 2006A). The building architecture in the ASM describes architectural entities such as stories, spaces and space aggregations, as well as space establishing elements such as walls, columns and slabs. The structural system is described in StAr as a hierarchy of entities to enable a top-down design approach. The GRA use the geometry and topology of the ASM representation to construct new geometry and topology, and to verify the model. These capabilities are provided by an underlying geometric modeling kernel. GRA are enhanced with embedded structural knowledge regarding layout and dimensional thresholds of applicability for structural assemblies made out of cast-in-place concrete. However, the current version of StAr provides still limited capabilities for supporting conceptual structural design because it provides no assistance for the exploration of structural alternatives and the evaluation of rough alternatives where analysis is not possible. The following components are still required for complete support by StAr:

(1) Design knowledge manager (DKM) to advise engineers on decision making, suggest alternative solutions, and perform evaluations. This component is currently being developed;

(2) Design history manager (DHM) to record the sequence of decisions taken to arrive at a given design solution. This component will be the subject of future research;

(3) Design alternative manager (DAM) to keep track of alternatives explored by the engineer at different refinement levels of the design solution. This component will also be the subject of future research.

(4) Graphical user interface that enables the engineer interact with a three-dimensional architecture-structure model (3D-GUI). This component is currently being developed.

This paper focuses on the 3D-GUI design. The following alternatives were considered for the development of the 3D-GUI in the context of StAr:

1. Build on top of a generic, general purpose CAD platform (e.g. AutoCAD).
2. Add to a platform specifically designed for building modeling (e.g. ArchiCAD).
3. Develop from scratch, on top of a geometric modelling kernel using a GUI toolkit.

Options 1 and 2 were explored together since they are based on similar CAD technologies. In general, most CAD platforms enable the development of vertical solutions to extend their core capabilities for specialized applications. However, access to the underlying geometric modeling kernel and algorithms is restricted, and modeling is permitted only using regularized operations on three-dimensional solids. Therefore, option 3 was selected given that it is the only one that provides the required full modeling flexibility through the kernel's API. The following systems are thus selected for the 3D-GUI implementation:

- ACIS geometric modeling kernel (currently being used by StAr) (Spatial 2006);
- HOOPS 3D application framework (TechSoft America 2006);
- Qt application development framework (Trolltech 2006).

These three systems complement each other, have already been tested separately in robust prototype and commercial applications and have also been used together successfully. Furthermore, the creators of these systems have developed interfaces so that the three systems can work together seamlessly.

## FUNCTIONAL DESIGN OF THE GRAPHICAL USER INTERFACE

The main functional requirements for the design of a 3D-GUI in a tool to support conceptual structural design lead to the development of five interface components (see Fig. 2):

***(1) Direct model interaction*** – Direct engineer interaction with an architectural-structural model is required to support the generation, selection, visualization, and query of the model. The interaction capabilities required are comparable to those of typical CAD applications. This is the most ubiquitous and salient feature of the GUI since the main two-way communications between the engineer and the computer take place through this component. It includes three sub-components: (i) a graphic 2D/3D modeling window for engineer-model interaction in 3D, plan, section and elevation views; (ii) a model hierarchy browser for navigating through the hierarchical organization of the model, and (iii) a message area for posting messages from the computer in response to the engineer's actions.

***(2) Efficient interaction mechanisms*** – These are necessary to facilitate the manipulation of the architecture-structure model, given the geometric complexity of the building and the great number of systems and subsystems it contains. These mechanisms permit the engineer to save views and switch between building views easily, and indicate preferences to better visualize and manipulate the model such as setting transparencies for entities or groups of entities, and filters for manipulating only the entities that correspond to the task being performed.

***(3) Context*** – This component comes from an important requirement of conceptual structural design: it has to be carried out within a building architectural context and respect the overall project requirements. The context is therefore divided into two sub-components: (i) the project context that provides mostly textual project information to the engineer (e.g. budget, construction area, building height, location, etc.), and (ii) the building architectural context that provides both textual and graphic information (i.e. architectural design information). The graphic information is displayed in the direct model interaction component.

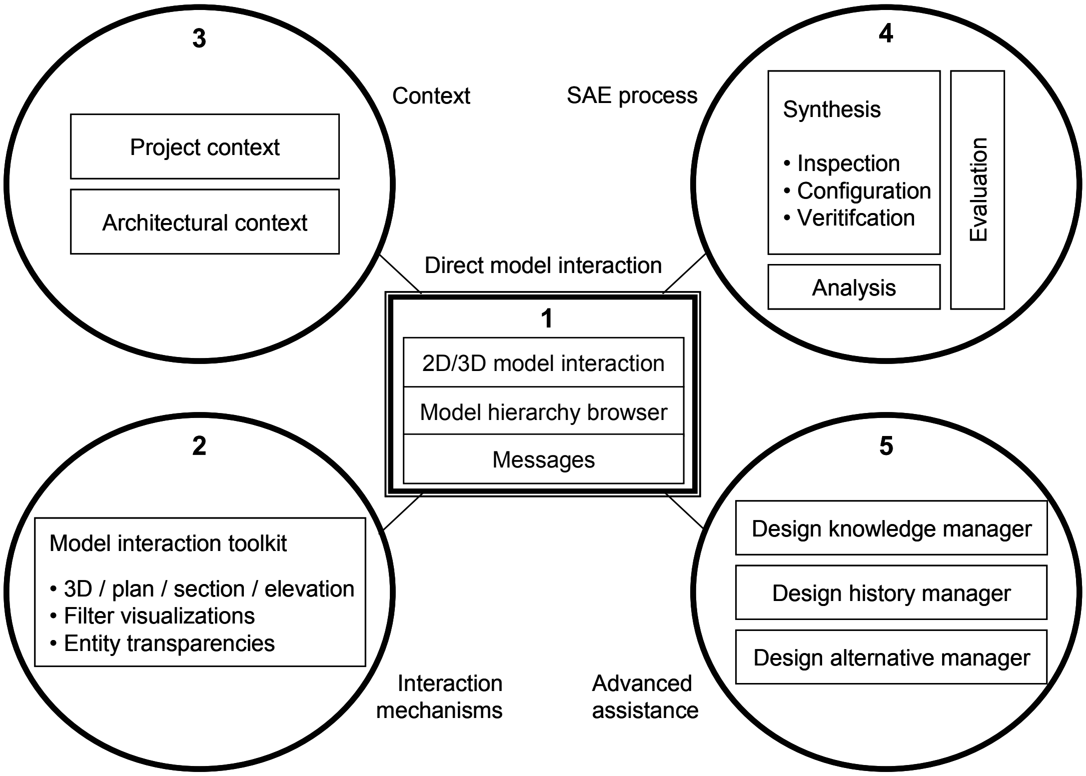

Figure 2. Main functional components of the graphical user interface.

***(4) SAE process*** – Within a project and a building architectural context, the engineer performs conceptual structural design by following the synthesis, analysis and evaluation (SAE) process described above. The main sub-process is the synthesis of structural solutions that is carried out by architectural inspection, configuration and verification. Evaluation is carried out at each level of structural abstraction, and through partial or complete analysis of structural solutions. Finally, analysis is performed at the element level by exporting the partial or complete structure to a commercial structural analysis package.

***(5) Advanced assistance*** – The above components are sufficient to assist expert structural engineers in producing structural solutions quickly for simple buildings. Additional assistance can be provided by three interface sub-components that correspond to the ones described in the previous section: (i) a design knowledge manager (DKM), (ii) a design history manager (DHM), and (iii) a design alternative manager (DAM).

During a typical design session, the architectural model is displayed in transparent mode. The engineer can also use the architectural context and/or entity filters to display, highlight and hide architectural entities for inspection and selection. In addition, during a design session the engineer can switch between views (3D, plan, section, and elevation) at will and check model dimensions.

## SCENARIO OF INTERACTIVE SUPPORT FOR THE SAE PROCESS

The following design scenario illustrates interactive support for the inspection and configuration sub-processes. It shows only one top-down exploration path. It includes interactions with most of the components and sub-components presented above.

### INSPECTION

The following options are given to the engineer to assist in the inspection of the building architecture: *"Verify column/wall continuity"*, *"Find supports from the architecture"*, *"Verify piecewise continuity"*, and *"Check structural layout constraints"*. For the first three options the computer uses a color degradation scale to indicate the length of the continuities. Piecewise continuous walls (i.e. walls with continuity interrupted in some stories) are also indicated in discontinuous stories so that architectural elements nearby can be detected and possibly moved to enable continuity. Once continuity is verified, the engineer can then select any continuous wall or column to become structural. When continuity is interrupted, the engineer can search within a storey for nearby architectural elements that could be moved to provide continuity. Considering the structural layout constraints, the computer presents an interactive list of spaces with associated constraints for the structural system layout. The engineer can click on any constrained space within the list for the constraint to be highlighted in the model.

### CONFIGURATION

The configuration process is carried out using the activities given in Figure 1 as follows (the steps in the process have been numbered for convenience):

1. *Engineer* – Opts for *"**Select independent structural volumes**"* in the menu.
2. *StAr* - Two options are given to the engineer: (1) define one independent structural volume for the entire building and (2) define more than one independent structural volumes. If the former option is selected, the computer creates an independent structural volume automatically by grouping all the spaces in the building. If the latter option is selected, the engineer selects the spaces that make up each independent structural volume and give a name to each independent structural volume.
3. *Engineer* – Selects option 1.
4. *Engineer* – Opts for *"**Select structural zones**"* in the menu.
5. *DKM* – Obtains the applied load for each space according to its function (the roof is treated as a terrace-type space). Calculates wind loads on façade surfaces, snow loads on roof slabs, and seismic loads on floors and roof slabs.
6. *StAr* – Displays floor slabs with colors indicating the intensity of the applied gravity load on each slab area.
7. *Engineer* – Uses coloured slab areas to define structural zones. Takes into consideration support constraints in spaces (e.g. column-free spaces and varying space dimensions).

8. *Engineer* – Opts for *“**Select structural subsystems**”* to explore overall load-transfer solutions.
9. *StAr* – Enables the engineer to select structural subsystems in three stages: (1) lay out structural grids, (2) define the main material for the entire structure, and (3) specify the structural subsystems according to the type of assembly support (the structural material can be replaced in any supporting assembly type). Depending on the flexibility provided by the architectural design, the engineer may explore several layout alternatives at this point or simply rely on existing architectural grids.
10. *Engineer* - For stage (1) the engineer lays out a 9 m by 12 m grid in a plan view and gives a name to the grid layout; for stage (2) the engineer selects steel as main structural material; for stage (3) the engineer defines an overall load transfer solution by selecting each structural subsystem from a list as follows (subsystem alternative 1):
    - horizontal subsystem: steel deck/open web/WF, estimated floor depth = 550 mm;
    - vertical gravity subsystem: column stacks and core walls;
    - vertical lateral subsystem: braced frames.
11. *StAr* – Checks for architectural constraints that may restrict the applicability of the alternative (e.g. column-free spaces, floor depths, modular dimensions, unobstructed views, etc). StAr indicates that the alternative is valid from an architectural standpoint.
12. *Engineer* – Opts again for *“**Select structural subsystems**”* to define another structural alternative at the subsystem level using the same 9 m by 12 m grid layout (subsystem alternative 2). Both alternatives are validated architecturally by StAr and stored.
13. *Engineer* – Asks the DKM to validate the subsystem alternatives.
14. *DKM* – Both alternatives are structurally valid according to the knowledge in the DKM.
15. *Engineer* – Asks the DKM to suggest other feasible structural alternatives for the 9 m by 12 m layout proposed in stage (1), step 10.
16. *DKM* – Suggests another feasible load transfer solution using concrete as main material (subsystem alternative 3):
    - horizontal subsystem: two-way flat slabs, estimated floor depth = 350 mm;
    - vertical gravity subsystem: column stacks and core walls;
    - vertical lateral subsystem: rigid frames.
17. *Engineer* – Asks the DKM to rank all the structural alternatives at the subsystem level.
18. *DKM* – Ranks all the alternatives considering overall structural cost and weight, constructability, soil conditions, foundations type and cost, etc.
19. *StAr* – Presents a table to the engineer with all the alternatives ranked.
20. *Engineer* – Ask the DKM to elaborate on how the ranking was obtained.
21. *DKM* – Explains the factors with their weights considered in the ranking.
22. *Engineer* – Selects subsystem alternative 1 (step 10) and explains the decision to the architect using the table with alternatives ranked and the explanation given by the DKM in step 21.
23. *Engineer* – Opts for *“**Select and position structural assemblies**”*.
24. *StAr* – Allows the engineer to specify structural assemblies individually or by groups.
25. *Engineer* - Specifies and positions frames along the grid lines. When specifying each frame, the engineer decides whether or not it is braced since for lateral support not all frames need to be braced.
26. *StAr* – Groups bay units by dimensions and shows them to the engineer with different slab colors in the model, using a list e.g. bay 9 m x 12 m red…, bay 9 m x 10 m blue

and so on. Note that while at the subsystem level the most repeated typical bay is considered, at the assembly level all bays need to be treated.

27. *Engineer* – Specifies floor and roof assemblies. Selects architectural slabs to specify these assemblies May determine floor framing directions for typical structural bays or asks the DKM to determine these directions automatically.
28. *Engineer* - Based on structural zones and support conditions, the engineer may divide floor slabs into two or more slab assemblies that must be specified accordingly.
29. *Engineer* – If special structural zones are found with few supports available, strong load vibrations or heavy loads the engineer devises local assembly solutions for those zones. S/he may ask the DKM to suggest possible assembly solutions.
30. *Engineer* – Opts for *"**Generate physical structure**"* to determine structural element geometry and topology.
31. *StAr* – Generates a wire-frame of connected primary structural elements with 2D slab elements and walls. In doing so StAr considers architectural constraints.
32. *StAr* – When special geometric conditions exist that force structural members to fall outside the structural grids, StAr does not generate supports. Instead StAr asks the engineer to place supports at the end.
33. *Engineer* – Places special supporting elements.
34. *Engineer* – Determines floor framing directions for non-typical floor assemblies or structural bays, or asks the DKM to determine these directions automatically.
35. *DKM* – Specifies sections and sizes for decks, secondary, and primary structural elements. Ranks feasible deck and secondary element sections versus spacing between secondary elements (based on floor assembly cost and overall floor depth).
36. *Engineer* – Selects sections of decks, secondary and primary structural elements.
37. *StAr* – Displays the resulting structural system.

Note that at in this design scenario overall structural alternatives have been explored and evaluated at the structural subsystem level only. However, structural alternatives can also be explored locally at the structural assembly and element levels.

## CONCLUSIONS

This paper has presented a computer-based approach to support conceptual structural design. The approach attempts to provide complete and effective support for the entire SAE process, with emphasis on the synthesis sub-process. A first prototype called StAr provided basic support capabilities. However, it lacked a graphical user interface (GUI) to enable testing by practicing engineers. This interface is currently being developed. It consists of a set of components aimed at providing full functionality consistent with the requirements of conceptual structural design; that is to provide interactive assistance for the synthesis, analysis and evaluation of conceptual structural solutions within a building architectural context, and considering overall project requirements. Assistance is knowledge-based with limited analysis but frequent evaluations. Given the explorative nature of the process, alternatives need to be managed and decisions need to be tracked. A graphical user interface and a design knowledge manager will enable the enhanced StAr prototype and the entire approach to be evaluated by practitioners in actual building designs.

## ACKNOWLEDGMENTS

The authors wish to acknowledge the financial support for this research from the Natural Science and Engineering Research Council of Canada (NSERC) and the Canada Research Chair Program (for the Chair in Computer-Aided Engineering for Sustainable Building Design held by Dr. Rivard).

## REFERENCES

Bailey S. and Smith I. (1994). “Case-based preliminary building design”, ASCE J. of Computing in Civil Eng., 8(4), 454-467.

Bédard C. and Ravi M. (1991). “Knowledge-based approach to overall configuration of multistory office buildings”, ASCE J. of Computing in Civil Eng., 5(4), 336-353.

Eisfeld M. and Scherer R. (2003). “Assisting conceptual design of building structures by an interactive description logic based planner”, Advanced Engineering Informatics, 17(1), Elsevier, 41-57.

Jain D., Krawinkler H., Law K. (1991). “Logic-based conceptual structural design of steel office buildings”, Report No 49, CIFE, Stanford University.

Kumar B. and Raphael B. (1997). “CADREM: A Case-based system for conceptual structural design. Engineering with Computers”; 13(3), 153-164.

Maher M. (1988). “Expert systems for structural design”, ASCE J. of Computing in Civil Eng., 1(4), 270-283.

Meyer S. (1995). “A Description of the structural design of tall buildings through the grammar paradigm”, Ph.D. Thesis, Dept. of Civil Engineering, Carnegie Mellon University, Pittsburgh.

Mora R., Rivard H., Bédard C. (2006A). “A computer representation to support conceptual structural design within a building architectural context”, ASCE J. of Computing in Civil Eng., to be published in the March issue.

Mora R., Bédard C., Rivard H. (2006B). “Geometric modeling and reasoning for the conceptual design of building structures”, submitted for publication to the Journal of Advanced Engineering Informatics, Elsevier.

Rafiq M., Mathews J., Bullock G. (2003). “Conceptual building design – evolutionary approach”, ASCE J. of Computing in Civil Eng., 17(3), 150-158.

Rivard H. and Fenves S.J. (2000). “A representation for conceptual design of buildings”, ASCE J. of Computing in Civil Eng., 14(3), 151-159.

Sisk G., Miles J., Moore C. (2003). “Designer centered development of GA-based DSS for conceptual design of buildings”, ASCE J. of Computing in Civil Eng., 17(3), 159-166.

Soibelman L. and Peña-Mora F. (2000). “Distributed multi-reasoning mechanism to support conceptual structural design”, ASCE J. of Comp. in Civil Eng.,126(6), 733-742.

Spatial (2006). 3D ACIS modeller, by Spatial Inc. company, Internet website: http://www.spatial.com/company/org/, last visited: January 2006.

Techsoft America (2006). HOOPS 3D application framework, by TechSoft America, Internet website: http://www.spatial.com/company/org/, last visited: January 2006.

Trolltech (2006). Qt Application development framework, by Trolltech, Internet website: http://www.trolltech.com/products/qt/index.html, last visited: January 2006.