



A COMPUTATIONAL FRAMEWORK FOR HOLISTIC LIFE-CYCLE DESIGN OF BUILDINGS

TERJE HAUKAAS and STEVAN GAVRILOVIC

Dept of Civil Engineering, University of British Columbia, Vancouver, Canada

This paper combines building information modeling with advanced structural modeling to optimize the design of building components using a life-cycle perspective. A new object-oriented software framework that extends an existing program is presented. It creates information-rich finite element models from industry foundation class files that are exported from building information programs. In addition to models for structural responses, the new components contain models for construction cost, repair cost, environmental impact cost, and human health cost. Random variable objects represent uncertainties in material properties, hazards, and cost models. Similarly, design variable objects represent decisions, such as structural dimensions and materials. All costs are summed and the expected value of this sum is employed as a decision criterion. This paper presents the software framework and a demonstration application.

Keywords: Structural design, Cost estimation, Object-oriented programming, Building information modeling, Probabilistic models, Finite elements.

1 INTRODUCTION

The objective in this paper is to provide models and computer algorithms to enable holistic design optimization of buildings and their components. In practice, structural components are usually designed to meet prescriptive code requirements. Because the primary objective behind building codes is life safety, the cost of repairs after extreme loading events and the environmental impacts of construction are often not explicitly considered. The themes of this conference, resilient designs and sustainable construction, acknowledge this issue and this paper presents tools to complement current design procedures.

The resulting computer program is intended for use in design but also in research to obtain unbiased estimates of life-cycle performance and cost. The contributions presented in this paper consist of an object-oriented software framework with 1) detailed finite element models of the building and its components; 2) information-rich components imported from building information models; 3) probabilistic hazard models for climatic loads and earthquake ground motion; 4) probabilistic models for a matrix of costs; 5) parameterization that employs random variable objects to characterize uncertainties and design variable objects to characterize values that are at the discretion of the decision-makers; and 6) algorithms for structural analysis, reliability analysis, and optimization analysis. The framework is an extension of the computer program Rt (Mahsuli and Haukaas 2013). An important purpose of Rt was to facilitate probabilistic analysis with many interacting models for hazards, buildings, and costs. Until now Rt has been used for regional seismic risk analysis with relatively simple models for each building. The extended program presented here is labeled Rts and addresses detailed building analysis.

At the foundation of Rts is a risk-based design philosophy in which the decision criterion is the minimization of the total expected cost. All concerns, ranging from construction cost to environmental impacts and occupant wellbeing are translated into cost values. The use of a single objective formulated in terms of cost has a long tradition in the structural reliability community (Benjamin and Cornell 1970, Enevoldsen and Sørensen 1994, Roynet and Polak 2004, Liang *et al.* 2007). Since the mid-1990's the earthquake engineering community has also devoted substantial efforts at estimating repair cost and using it as a design criterion (Cornell and Krawinkler 2000, Yang *et al.* 2009, Federal Emergency Management Agency 2012). Efforts have also been made at establishing cost-based frameworks with multiple costs (Frangopol *et al.* 1997, Sánchez-Silva and Rosowsky 2008, Gardoni *et al.* 2016). Although some of the early research efforts do not include environmental impact models and detailed structural models, they do emphasize the importance of holistic life-cycle considerations in structural design.

2 SOFTWARE ARCHITECTURE

Rts is programmed in C++ using an object-oriented approach. This means that all models and algorithms are organized into a hierarchy of classes from which objects are created at run-time (Deitel and Deitel 2002). The organization of algorithms into classes is an important part of object-oriented programming and this section provides an overview of how it is implemented in Rts. The object-oriented approach has the advantage in that it makes it easier to share parts of code. An objective in this work is to develop a collection of algorithms and classes that can be readily shared and improved in a joint effort with other research groups.

The user interface of Rts is shown in Figure 1 with oval labels to identify the four panes. The objects pane displays a list of the objects that the user has created, either in this interface or by loading an input file. The objects are separated into four categories: parameters, models, methods, and components. Each of those has a corresponding base class in the code. Subclasses of the parameter class include random variable, design variable, constant, and response. Subclasses of the method-class include linear solvers and optimization algorithms. The focus in this paper is on the models and components. The other panes in Figure 1 are the properties pane, which shows attributes of selected objects, the output pane, which displays messages during the analysis, and the visualization pane.

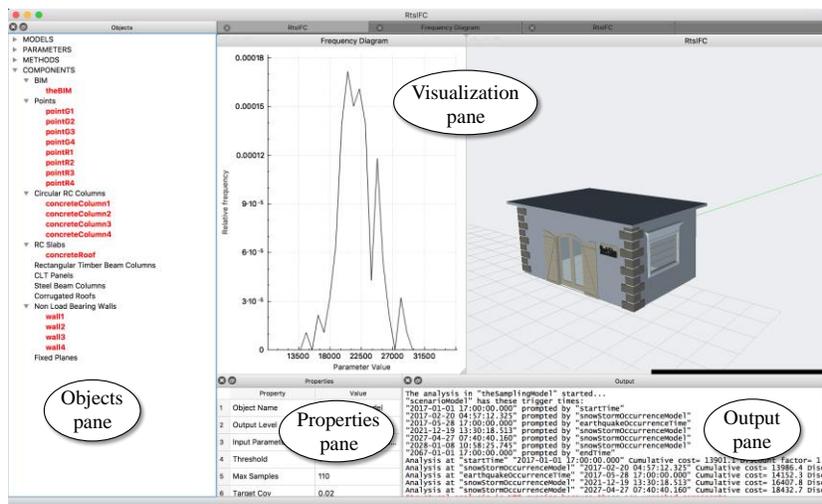


Figure 1. User interface.

Figure 2 displays the class map for a detailed building analysis in Rts. All classes above the top horizontal dashed line are models of the type that existed in Rt. This means that responses flow from the hazard models on the left-hand side through the building model ultimately providing the total lifetime cost to the optimization model. In fact, the building model outputs the cost at a given time instant to the scenario model, which simulates a scenario realization from construction to demolition. The scenario model calls the building model only at certain times, such as the time of construction, the time of an earthquake event, the time of demolition, etc. The scenario model outputs the total lifetime cost for one realization of the random variables. Figure 2 shows that this response may be input to the sampling model in order to obtain the relative frequency diagram for the total cost that is shown in Figure 1. Figure 2 also shows that the total cost can be input to the FORM model, which calculates cost-exceedance probabilities using the first-order reliability method. In turn, the risk model employs those exceedance probabilities to calculate the expected cost using a numerical integration scheme that yields exact derivatives of the cost used by the gradient-based optimization algorithm (Haukaas and Gavrilovic 2016).

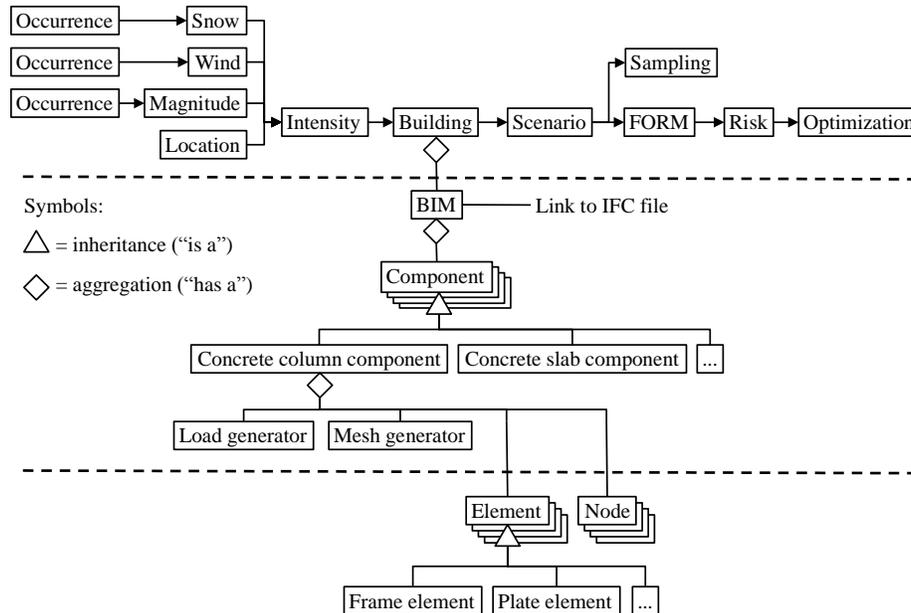


Figure 2. Organization of classes.

The classes below the bottom dashed line in Figure 2 are standard finite element classes found in most object-oriented finite element programs (McKenna *et al.* 2009). It is the classes between the two horizontal dashed lines in Figure 2 that are novel in this paper. The class labeled BIM takes an industry foundation class (IFC) file as input and creates the building components. IFC is an object-oriented data model governed by the ISO standard 10303, informally known as STEP. Rts uses the open source library IfcPlusPlus to read STEP files (Gavrilovic and Haukaas 2017).

The load and mesh generators shown in Figure 2 are outside the scope of this paper; it is the propagation of information from the components through the BIM object to the building model that is a focus here. The interface of these three classes is shown in Figure 3. The arrows indicate how information flows from the components to the building model, which has the evaluateModel() method that all models in Rts have. For example, to obtain the total volume of

concrete in the building, which is needed by the model presented in the next section, the building model calls `getVolumeOfConcrete()` of the BIM object. The BIM object then loops over all the components and calls `getVolumeOfConcrete()` to obtain that result. It is the component classes that epitomize the information-rich building modeling in Rts. They contain models for construction cost, cost of repair, and an array of random variables that describe model uncertainty and uncertainties in the material properties. Both structural and non-structural components are currently implemented in Rts.

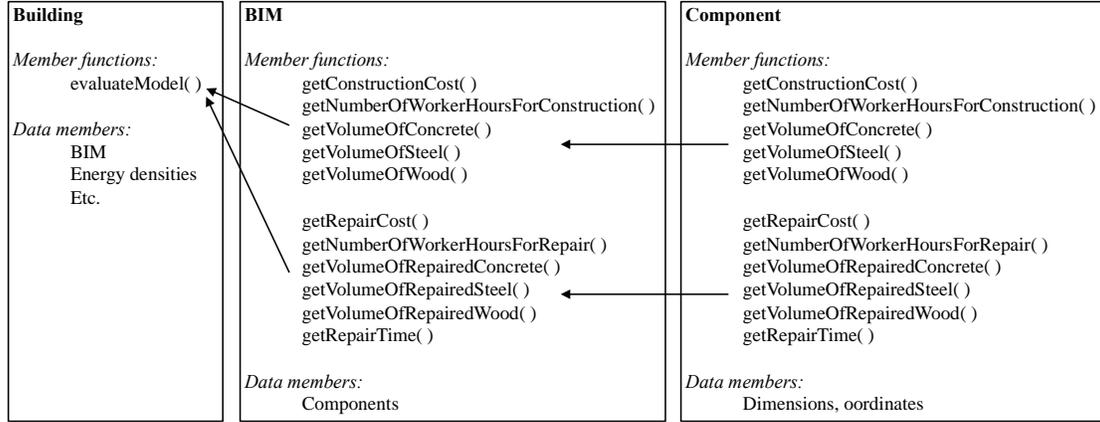


Figure 3. Class interfaces.

3 COSTS

Many concerns are translated into cost in Rts and Figure 4 organizes these costs into a matrix. Each row in this matrix addresses one phase in the life of the building, while the columns divide each phase into different cost types. Let c_{ij} denote the cost of phase $I = 1, 2, 3, 4, 5$ and type $j = a, b, c, d, e$. According to Eq. (1) the total cost is:

$$c_{\text{total}} = \sum_{i=1}^5 \sum_{j=a}^e c_{ij} \times e^{-r \cdot t} \quad (1)$$

where the exponential factor discounts the cost to present time with r as the real interest rate. Due to space limitations the detailed models are presented elsewhere (Haukaas *et al.* 2017) but the model for environmental and health impacts of manufacturing the materials is provided here for illustration. Adopting the notation from Figure 4 and using Eq. (2) this cost is:

$$c_{1bc} = \sum_{m=1}^M \left(w_m \cdot i_m \cdot \sum_{n=1}^N \left(f_{n,m} \cdot \frac{g_n}{r_n} \right) \right) \quad (2)$$

where M = number of materials, such as wood, steel, concrete, glass, aluminum, plastic, and copper, N = number of energy sources, such as diesel, gasoline, fuel oil, liquid petroleum gas, electricity, natural gas, and coal, w_m = total weight of material m in kg, i_m = energy intensity required for extraction and manufacturing of material m in J/kg, $f_{n,m}$ = fraction of energy source n that is employed in the extraction and manufacturing of material m , adding to unity for each m , γ_n = conversion from fuel amount or electricity amount to environmental cost in \$/L for fuel and

in \$/kWh for electricity, including health impacts, and ρ_n = energy density obtained from energy source n in J/L for fuel and in J/kWh for electricity. This model is implemented in the building model in Figure 2 and the BIM object provides the material volumes as described above.

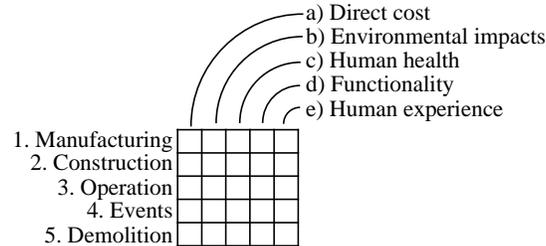


Figure 4. Cost matrix.

Each component in Rts, whether structural or non-structural, is meshed with finite elements. At the occurrence of a hazard, e.g., a windstorm or an earthquake, the structural analysis is triggered. Based on the resulting displacements, strains, stresses, and accelerations, the visual damage in each component is estimated (Javaherian Yazdi 2015). Based on the visual damage the repair actions are predicted, and in turn the cost and time of repair, i.e., costs 4a and 4d in Figure 4, respectively.

4 EXAMPLE

The example considered in this paper is the building shown in Figure 1. The building information model is created in Archicad, a program developed by Graphisoft. The IFC file exported from Archicad is imported into Rts, creating the components that are schematically visualized in Figure 5 and listed in the right-hand side of the figure. The building is a very simple example intended to run an analysis to demonstrate the new implementations. As indicated in Figure 5, three material choices are considered for this building: concrete, wood, and steel. For each material choice, continuous design variables represent the column dimensions and roof thickness. A sampling analysis conducted with Rts, using the classes presented previously and including a matrix of costs, gives the probability density function of total cost shown in Figure 1. It has a relatively jagged shape because only 100 realizations of a 50-year time-period, from construction to demolition, were created.

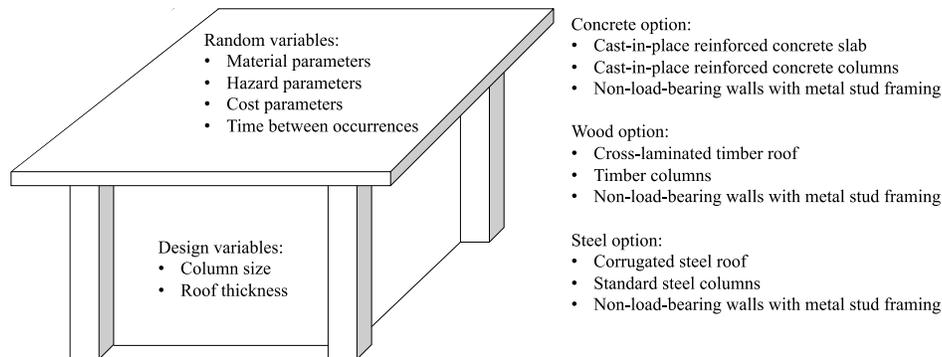


Figure 5. Ingredients of the example.

5 CONCLUSIONS

This paper presents an extension of the computer program Rt, a program for probabilistic analysis with many interacting models. The extended program, named Rts, employs information-rich building components meshed with finite elements and imported from building information programs to obtain the total life-cycle cost of buildings. A comprehensive array of costs is predicted, including direct cost of construction, environmental impacts, damage due to earthquakes, and cost of operation. With the implementations presented here the optimization algorithms in Rts can minimize the mean, i.e., expectation of the life-cycle cost of the building.

References

- Benjamin, J. R., and Cornell, C. A., *Probability, Statistics, and Decision for Civil Engineers*. McGraw-Hill, 1970.
- Cornell, C. A. and Krawinkler, H., “Progress and challenges in seismic performance assessment”, *PEER Center News*, Vol. 3 No. 2, 2000.
- Deitel, H. M. and Deitel, P. J., *C++ How to program*. Prentice Hall, 2002.
- Enevoldsen, I., and Sørensen, J. D., “Reliability-based optimization in structural engineering”, *Structural Safety*, 15, 169–196, 1994.
- Federal Emergency Management Agency, *Seismic Performance Assessment of Buildings - methodology. Fema P-58-1*, 2012.
- Frangopol, D. M., Lin, K.-Y., and Estes, A. C., “Life-Cycle Cost Design of Deteriorating Structures”, *Journal of Structural Engineering*, 123(10), 1390–1401, 1997.
- Gardoni, P., Guevara-Lopez, F., and Contento, A., “The Life Profitability Method (LPM): A financial approach to engineering decisions”, *Structural Safety*, Elsevier Ltd, 63, 11–20, 2016.
- Gavrilovic, S. and Haukaas, T., “Reliability-Based Design Optimization Using Building Information Models”, *Proceedings of the 12th International Conference on Structural Safety & Reliability, ICOSSAR 2017*, Vienna, Austria, 2017.
- Haukaas, T., and Gavrilovic, S., “Using FORM for Minimizing the Uncertain Cost of Structural Designs”, *Edited Volume in Honor of Prof. Der Kiureghian*, 2016.
- Haukaas, T., Gill, G., and Gavrilovic, S., “Probabilistic Cost Models and Computational Framework for Life-cycle Design of Buildings”, *Proceedings of the 12th International Conference on Structural Safety & Reliability, ICOSSAR 2017*, 2017.
- Javaherian Yazdi, A., “Damage Modelling for Performance-based Earthquake Engineering”, The University of British Columbia (UBC), Vancouver, 2015.
- Liang, H., Haukaas, T., and Royset, J. O., “Reliability-based optimal design software for earthquake engineering applications”, *Canadian Journal of Civil Engineering*, 34, 856–869, 2007.
- Mahsuli, M., and Haukaas, T., “Computer program for multi-model reliability and optimization analysis”, *Journal of Computing in Civil Engineering*, 27(1), 87–98, 2013.
- McKenna, F., Scott, M. H., and Fenves, G. L., “Nonlinear finite-element analysis software architecture using object composition”, *Journal of Computing in Civil Engineering*, 24(1), 95–107, 2009.
- Royset, J. O. and Polak, E., “Reliability-based optimal design using sample average approximations”, *Probabilistic Engineering Mechanics*, 19(4), 331–343, 2004.
- Sánchez-Silva, M. and Rosowsky, D. V., “Risk, reliability and sustainability in the developing world”, *Proceedings of the ICE - Structures and Buildings*, 161(4), 189–197, 2008.
- Yang, T. Y., Moehle, J., Stojadinovic, B., and Der Kiureghian, A., “Seismic performance evaluation of facilities: methodology and implementation.” *Journal of Structural Engineering*, 135(10), 1146–1154, 2009.