Object Interaction Query:
A Context Awareness Tool for Evaluating BIM Components' Interactions

Carolina Soto Ogueta
Chile
carolinasotoo@yahoo.com

Moa Carlsson
Sweden
moac@mit.edu

Abstract

During the creative process, designers constantly evaluate the relations between objects in space. BIM aids this process by providing a modeling platform where objects are embedded with information, which can be extracted on demand. The object Interaction Query (oIQ) proposes a novel way to query BIM models, not only for geometric properties and dimensions, but also about the relations among the components. By including the queried objects' context and interrelations as part of the computation, the prototype tool is able to provide feedback on complex interactions and conflicts in the design environment. The oIQ approach and its implementation are developed as integral parts of the design process, allowing users to perform customized queries through a GUI in which users can apply their knowledge and design preferences to the model's evaluation.

Keywords: BIM; Object interaction; Context awareness; Rule-based system.

Introduction

Relationships between objects in space are key to the enterprise of design and construction, and designers are constantly evaluating them during the creative process. Building Information Modeling (BIM) aids this process by providing a modeling platform where objects are embedded with information, which can be extracted on demand. The present research, developed in response to an identified need for more design-oriented query systems in BIM, proposes a novel way to query models not only for geometric properties and dimensions (direct query), but also about the relations among the components (indirect query). The oIQ approach and its implementation are developed as an integrated part of the design process, allowing users to perform customized queries in which they can easily apply their expert knowledge and design preferences to the model's evaluation.

Several research initiatives, as well as commercial products, have sought ways to utilize the information embedded in BIM to automate the evaluation of compliance with different building norms. While these can be very powerful tools to avoid design conflicts, two areas have not yet been thoroughly researched. First, the research and commercial tools have been devised to work outside the actual design application through the utilization of the Industry Foundation Classes (IFC) format, moving the evaluation further away from the design process and into the realm of engineering or trade specialists. Second, these tools have mostly focused on evaluating written rules, such as building codes, and not on spatial relations, which are a key part of the design process.

The current paper presents an ongoing research initiated at the MIT Design and Computation Group. As master's degree students from the group, the authors started this work within the context of a 2012 workshop titled *Computational Design Lab: Reinventing BIM*. The paper outlines a novel approach to the evaluation of building models which focuses on enabling context awareness among BIM objects, i.e., access to the spatial relations between objects, through the embedded information available in these environments. The approach is being researched through the development of the *object Interaction Query* (oIQ), a prototype application that enables the inclusion of the user's expert knowledge in the evaluation, integrating the spatial computation in the design process, and providing instant feedback on possible undesired spatial relations in several formats. The main goal of oIQ is to create a platform which enhances the design process by allowing for fast and informed decision-making.

Embedded information in BIM and users' expert knowledge

Embedded information enables BIM to streamline design evaluation and error detection; both tasks which were previously performed manually by domain-specific experts, making them time-consuming, expensive, and error-prone.

BIM components contain embedded information regarding their functions (e.g., walls, floors, and doors), dimensions, materials, and other characteristics, which can be utilized in the evaluation of spatial relations. This embedded information may allow users to conduct specific assessments for the different model objects. As

BIM recognizes the function of a model object, evaluation tools can access specific data regarding that component and then evaluate it in an object-specific way, according to its particular properties. Despite the above, it is important to note that while BIM components contain information about their properties, they do not contain knowledge. Analyzing BIM in the light of Alexander's pattern language, Ozel (2007) distinguishes between information and knowledge, stating that while information in BIM components is intended to be value- and context-free, knowledge, which is not included in BIM components, refers to domain-specific, expert lessons extracted from past experience (p. 458).

Proposal

The current research proposes to integrate a method to enable designers to evaluate complex spatial relations inside the design process. The embedded information contained in BIM positions this technology as a powerful environment in which to conduct this research. Through the inclusion of the user's expert knowledge in the evaluation, the work assesses the possibility of allowing designers to easily evaluate potential conflicts, as well as unwanted behaviors and interactions between architectural components. As will be described in the next section, the presently existing tools allow users to conduct clash detection and rule-based checking on BIM models, enabling users to automatically find geometrical clashes between components as well as report code compliancy violations. The current work investigates how to generate project evaluations that go beyond described geometric clashes and established code to allow for identification of more complex and hard-to-detect conflicts and undesired relations, with the purpose of enhancing the creative process by giving designers a tool to assess projects focusing on their design intent. This design assessment automation is achieved through granting objects the ability to detect and report information about their contexts and the potential conflicts that can result from the interactions with them.

Background Survey

Several approaches have been developed and tested to take advantage of the expansive amount of information available in BIM. Two of them are main reference points for the current research: clash detection and rule-based checking systems. Clash detection does not rely heavily on the model's embedded information, only utilizing it for the purpose of identification of objects and reporting of conflicts. The process of rule-based checking, on the other hand, may rely on the model's embedded information depending on the rules being queried. According to Zhang et al. (2012), currently no commercially available BIM authoring tool includes rule-checking capabilities (p. 4). Nevertheless, a few commercially available software packages (e.g., Solibri) as well as research initiatives (e.g., Borrmann and Rank, 2009) have focused on external rule-based checking for BIM. Another type of rule-checking application, which has not been developed extensively, is the plug-in tool, developed to work inside the BIM authoring application, providing an immediate dialogue between design and evaluation. By putting the evaluation directly inside the design environment this type of tool allows designers to control the evaluation and enables them to use the application as part of their design toolset.

Rule-based checking in BIM has been widely studied and developed by, among many others, Charles M. Eastman (Eastman et al., 2009, 2010; Zhang et al., 2012). In their paper titled *Automatic Rule-Based Checking of Building Designs* (2009), Eastman et al. provide a thorough survey of both the technology and structure of several IFC-based rule-checking efforts. In the paper, Eastman et al. recognize the fact that "almost all efforts in automating rule checking to date have been applied to building code and accessibility criteria" (p. 1012). Among the model evaluation efforts studied three are key to the current research. The first effort, undertaken by Zhang et al. (2012) involves an implemented approach for automated hazard identification and correction. The second study, by Lee J-K et al. (2012) is a project for the GSA and U.S. Courts, investigating design rule checking of federal courthouses using the IFC format. Finally, the third approach is a project developed by Borrmann and Rank (2009) and implemented as a Spatial Query Language (SQL) for 3D buildings and 3D city models. The SQL provides metric, directional, and topological spatial operators and is applied to read the geometric descriptions of objects rather than defining their boundary representation based on information extracted from the embedded properties.

Object Interaction Query (oIQ)

The object Interaction Query is an application which aims to integrate the evaluation of spatial relations between objects inside the design process. By allowing users to easily apply their expert knowledge and design preferences to a model's evaluation, the system enables the computation of complex spatial conditions. Designed as an open framework, oIQ allows users to add their own rules and constraints to a model's evaluation. This user-generated content constitutes pieces that can be plugged into the defined framework. The framework contains the data that determines how these user-generated rules are stored, retrieved, and executed. Following an executed query, the tool can report conflicting conditions in different output formats.

Summarizing the order of operations, designers select objects and, through the application of predefined or custom-made rules, query these objects to evaluate possible spatial constraints; thereafter the tool provides the users with feedback on interactions and conflicts (Fig. 1). The application is being designed to answer queries about spatial constraints, such as: Is there enough clearance around mechanical equipment for installation and repair? Are there any elements obstructing the projected door swing area? Is the desired clearance for stair landings being

applied to all areas of the building? The ability to answer these types of questions, which depend on user knowledge rather than established codes, is accomplished through the inclusion of the Query Volume (QV), which will be explained in the following sections. The tool also allows answering more simple queries such as: Is the width of the door ADA-compliant?

The procedure of oIQ follows the steps identified by Eastman et al. (2009) for rule-based checking applications: first, the *rule translation stage,* where spatial constraints are translated into machine-readable language; second, the *building model preparation stage* where modeled elements are selected and rules and values to test are introduced by the designer through a UI, third, the *rule execution stage* where the evaluation is conducted; and finally the *reporting stage,* when results are reported back to the user.

The main goals of the oIQ development are: (a) including the evaluation inside the design process, (b) taking advantage of the information embedded in BIM components, (c) enabling the inclusion of the user's expert knowledge in the evaluation, (d) allowing rule customization, and (e) reporting query results in user-friendly formats.

Direct and indirect queries

BIM components (e.g., walls, doors, floors) contain different information. Furthermore, the information is organized in very dissimilar ways depending on whether the family is linear-based (e.g., walls), hosted (e.g., doors, windows), component-based (e.g., columns), sketch-based (e.g., floors, ceilings), etc. Therefore, the method oIQ uses to access the embedded information varies accordingly.

The algorithm to access information is pre-coded (hard-coded) to allow that each time a user selects an object to query oIQ identifies the type of object and distinguishes the path to acquire information about the location, orientation, dimensions, performance conditions, and other data of that specific type.

oIQ is designed to conduct two types of queries on BIM objects: direct and indirect. These two types of queries were identified during the development of the plug-in, and the differences they establish are a direct result of the research and coding work carried out. Zhang et al. (2012) in their implementation of an automatic safety checking method for construction in BIM, make a similar distinction - "direct access" and "extended access" - but referring to the ways in which their tool accesses the information in the model, while in oIQ the difference relies on the computation carried out after accessing the information (p. 10).

Direct queries can be answered by accessing the information embedded in elements. For example, as the height of a ceiling is a property commonly contained inside a ceiling object in BIM, querying a ceiling to evaluate its compliancy with a minimum height can be done through a direct query. The application acquires the height information, compares it to a minimum defined by the user according to his/her criteria, and reports whether the minimum condition is being met or not.

Indirect Queries involve not only the information contained in BIM components, but also the conditions generated by the relations established between two or more components in space. An example of this would be the distance between a door and a wall. Because this information depends on the actual design in progress and is contained neither in the door nor in the wall, in order to acquire the distance and then evaluate whether it meets a certain boundary condition, a new process needs to be introduced. Information such as location, orientation, and dimensions of both elements needs to be acquired to execute a more complex evaluation in order to report any violation of a preset minimum distance condition. The information acquisition, as well as the subsequent computation, is conducted through the QV.

Query volume

The Query Volume (QV) represents a transient boundary condition used to fulfill the query of the objects selected at a given time. The QV is used 'behind-the-scene' in the automated computation, to transpose the users' expert knowledge into a computable algorithm. Although, the QV combines that knowledge with information acquired from the component, it is not a mere offset of recognized geometry, but it also adapts to the performance capacities of components, e.g. the swing of a door.

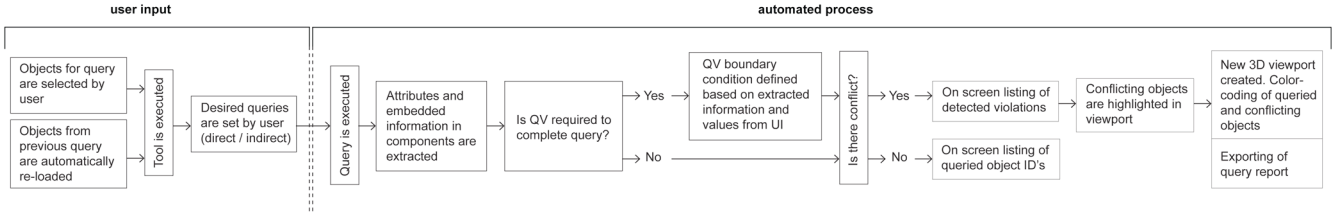

Figure 1: Steps of the query process.

When a query refers to the relation between two or more BIM components (indirect query), the computation is executed through the QV. For example, when the user queries a model to assess whether there is enough clearance around a certain object, the protocol of the query will operate according to following steps:

1 The application acquires the location, orientation, dimensions, and performance conditions of the queried object(s). (e.g. does this door have a swing projection?)

2 The QV, a discrete volume built according to the information acquired from the component, as well as the information input by the user, is introduced in the BIM.

3 The application uses the QV to look for other components that may represent a violation of the rule being queried.

4 Once the computation is done, and the possible conflicts are recorded in the different reporting output formats, the QV is eliminated.

Proof of concept - prototype implementation

In order to test the feasibility of the proposed design-oriented query system, a prototype implementation was developed for Revit using the software's API. The tool approaches oIQ's main goals, previously described as follows:

a. *Including the evaluation inside the design process:* The prototype implementation is a plug-in integrated in the typical BIM workflow. This integration allows the tool to be activated by the designers at any point of their creative process without the need of exporting the model to an external application.

b. *Taking advantage of the information embedded in BIM components:* The information embedded in BIM components is used by the tool in direct and indirect queries to assess the compliance with the conditions set by users.

c. *Enabling the inclusion of the user's expert knowledge in the evaluation:* The tool reads the information input by the designers through the user interface (UI), allowing the designers to evaluate constraints and rules derived from their own experience and knowledge, and from constraints set by clients or design teams for specific projects. In this way, the oIQ allows designers to constantly test the model to assess whether their design intent is being maintained across the different phases of the project.

d. *Allowing rule customization:* The UI allows users to customize the evaluation by: loading existing rules, changing the values assigned to rules, saving rules and or values for future queries, and combining sets of rules.

e. *Reporting query results in user-friendly formats:* Query results are reported graphically and through descriptive text in three separate ways. First, a popup screen opens, listing the objects and rules queried and the encountered conflicts. Second, this list is saved to an external text file that can be used to track conflict evolution. This file is automatically named using the date and time of the query. Third, a color coded 3D view is automatically created. This view shows all the queried elements in green and all the conflicting elements in red. Similarly to the text file, this view is also named using the query's date and time (Fig. 2).

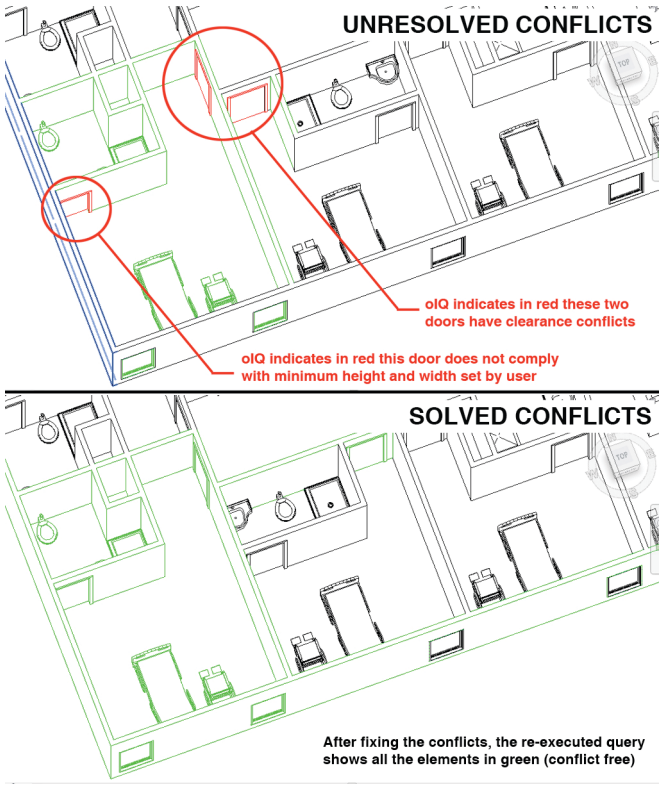

Figure 2: Top: Automatically generated 3D image reports conflicting doors. Bottom: After modification the automatically created 3D view indicates there are no conflicts.

Test scenario

The prototype was used to query doors in order to test the compliancy of three different conditions. The first two – minimum height and minimum width of a door – were direct queries, while the third one – clearance of door swing and projection area – was executed through an indirect query. The rules for evaluating these three queries were hardcoded in the application, while the UI allowed users to input values for minimum door width and height to customize the query. The UI also allowed for storing of these values for future queries.

The tool was tested on a basic BIM model of a residential building. The scene was seeded with different conflicting elements, placed in key positions relating to different doors. The oIQ plugin was able to identify all the conflicts between doors and surrounding objects, as well as violations of height and width minimum conditions. The values input by users through the UI were successfully identified by the application and incorporated in the computation.

After modifying the model to solve the conflicts identified by oIQ, on a re-execution of the same query, the tool indicated that there were no conflicts related to those objects.

Conclusions

The present research, developed in response to an identified need for more design-oriented query systems in BIM, provides a novel way to query BIM models, not only for geometric properties and dimensions (direct query), but also to query models about the relations among the components (indirect query). The oIQ approach and its implementation are developed as an integrated part of the design process and allow users to perform customized queries. The possibility for a user to not only design queries to suit specific and local needs but also to combine rules of different characters, aids the design process by allowing for fast and informed decision making.

The implemented prototype was designed and proved to successfully handle both direct and indirect queries, although operable only within a predetermined, i.e., hardcoded, rule set. The next steps of the research will include further development of: data structure, management of rules, and input and output data. Another key area for future work is the development of an expanded UI to enable users to modify and add new queries more freely, as well as the improvement of the reporting and visualization of query results.

The research presented has shown great potential for becoming an enhanced feature of design in BIM, and the promising results and identified limitations have productively opened up new areas of research, which will be investigated by the authors in future work.

References

Borrmann, A., Rank. E. (2009a). Specification and implementation of directional operators in a 3D spatial query language for building information models. Advanced Engineering Informatics, 23 (1). 32–44.

Borrmann, A., Rank. E. (2009b). Topological analysis of 3D building models using a spatial query language. Advanced Engineering Informatics, 23. 370-385.

Borrmann, A. (2010). From GIS to BIM and back again – A spatial query language for 3D building models and 3D city models. Proceedings of the 5th International 3D GeoInfo Conference. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 38, 19-26.

Eastman, C. M. (2009). Automatic rule-based checking of building designs. Automation in Construction, 18 (8), 1011–1033.

Eastman, C. M., Jeong Y. S., Sacks, R., Kaner, I. (2010). Exchange model and exchange object concepts for implementation of national BIM standards. Journal of Computing in Civil Engineering, 24, (1), 25-34.

Ghang L., Sacks, R., Eastman, C. M. (2005). Specifying parametric building object behavior (BOB) for a Building Information Modeling system. Automation in Construction 15, 758-776.

Lee J-K, Lee J., Jeong Y., Sheward, H., Sanguinetti, P., Abdelmohsen, S., Eastman, C. M. (2012). Development of space database for automated building design review systems. Automation in Construction, 24, 203-212.

Ozel, F. (2007). Pattern language and embedded knowledge in Building Information Modeling. Proceedings of eCAADe 25, 457-464.

Zhang S., Teizer, J., Lee J., Eastman, C. M., Venugopal, M. (2012). Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules. Automation in Construction. Manuscript submitted for publication.