From Parametric to Meta Modeling in Design

Marcelo Bernal, Ph.D.
Universidad Técnica Federico Santa María, Chile
marcelo.bernal@usm.cl

Abstract

This study introduces the Meta-Modeling process adopted from the Model Based System Engineering field (MBSE) to explore an approach for the generation of design alternatives beyond the restrictions of the Parametric Models that mainly produce geometric variations and have limitations in terms of topological transformations during the exploratory design tasks. The Meta-Model is the model of attributes and relationships among objects of a particular domain. It describes objects and concepts in abstract terms independent from the complexity of the geometric models and provides mapping mechanisms that facilitate the interfacing with parametric parts. The flexibility of these computer-interpretable and human-readable models can contribute to creatively manipulate the design knowledge embedded in parametric models.

Keywords: Parametric Modeling; Meta-Modeling; Model Based System Engineering; Modeling Languages; Systems Integration.

Introduction

The most common understanding of the modeling activity in architecture is as the representation of geometry and related attributes of an aspect of interest of a project, for such a purpose, according to the early definition of computer models of by Kalay (1989), they use symbolic structures that not only allow the representation, but also the manipulation of these aspects. The parametric modeling technology, developed in the 70's and currently the dominant modeling paradigm in design, relies on Constructive Solid Geometry (CSG) and Boundary Representation (B-Rep) for representing the geometric relationships in a hierarchical binary tree data structure and visualizing the updated geometry when changes in the parameters values occur, respectively (Eastman, 1999; Mantyla, 1988). Parametric Modeling (PM) captures inputs, constraints, conditionals, attributes and functions that control features of parts and assemblies that define the design space (Clevenger & Haymaker, 2011) derived from the range of all possible geometric variations of the model.

Although current PM systems can capture best practices and facilitate the generations of design alternatives, parametric models are not capable to support variations beyond the scope of their hierarchical structure of the geometric relationships prematurely limiting the potential design space. The topological structure of the resulting models that enable the geometric variations relies on vertically sharing parameters and reference geometry. While creating the model, for every dependency relationship between objects a new node on the tree is created. This data structure implies that adding or deleting objects during topological transformations break the dependencies among objects of the current design configuration negatively affecting the behavior of the parametric model.

To tackle this limitation and better support design exploration, this study explores the next steps of the modeling activity (Bernal, Haymaker, & Eastman, 2015) from the perspective of the Model-Based System Engineering (MBSE) that addresses the problem of modeling the structure, behavior and requirements of a system (Reichwein & Paredis, 2011) using Meta-Models (MM), or the model of the attributes and relationships of the objects of the system of interest (Kühne, 2006) that avoids the complexity of CAD data structures.

Parametric Modeling in Design

Despite limitations in terms of design exploration, the parametric modeling technology facilitates how designers manipulate geometric information, capture design knowledge for repetitive tasks, adapt previous solutions for new problems, perform evaluations and search through design spaces. The question is how preserving all these features while extending the technology to support radical topological changes accordingly to the behavior of expert designers, who explore parallel alternatives rather than a single one in early stages (Lawson & Dorst, 2010). Often these alternatives differ dramatically in terms of configuration and even typology and require transformation or generation mechanisms not supported by current PM technology.

Parametric models embedding design expertise

Research efforts in describing and standardizing parametric modeling practices (Lee, Sacks, & Eastman, 2006) are having an important impact on sharing and reusing adaptable objects. Besides the standardization by manufacturers of commercial products at defferent levels of detail (Gentry, Sharif, Cavieres, & Bigg, 2016), custom parametric models can capture firm best practices either into reusable and adaptable objects or algorithms that capture and embed design knowledge (Gómez

& Swarts, 2014) for automation of more sophisticated tasks such as automatic generation of preliminary designs based on application of rules (Bernal & Eastman, 2011). This expertise can determine the behavior of a single part, assemblies of them, or multiple nested assemblies in large models. Even though PM allows the creation of all kind of assemblies, once the instantiation is complete any further transformation of its topological structure must be manually done, which is a complex and distracting task in large scale models.

Expert systems for automatic detailing

These systems stand for technical domains (e.g. Structure, MEP, or HVAC) for automatic specification and detailing. Based on well-defined decision rules distilled from expert knowledge, they select, specify, insert and adapt parametric objects according to various conditions. Examples of ontologies, rules, and even functions to define tolerances can be found in the precast concrete industry (Eastman, Sacks, & Lee, 2003). The automation of detailing usually uses auxiliary geometry or massing conceptual models as inputs. Again, any change of the input geometry implies the need of the re-arrangement of the detailing, and the modification of the binary tree structure acknowledging that some objects remain, while others have been either deleted or added.

Generative systems for design exploration

This approach is partially supported by PM. It is based on the definition of fitness functions, generation mechanisms and evaluation of the outcomes (Frazer, Tang, & Sun, 1999). Through the iterative cycle of generation and evaluation eventually generative systems can lead to unexpected results, interpreted as apparently creative (Lawson, 2004). The emergence of new features not intentionally predefined is derived from the redefinition of the design rules that determine the topological composition of the designs. Nevertheless, PM mainly supports the readjustments of the parameters rather than the rules, reducing the scope of the impact to geometric and not topological variations.

Case based design and implicit knowledge

Case-based Design (CBD) is the adoption of Case-based Reasoning (CBR) developed by the Artificial Intelligence, a subarea of the Computer Science. CBD attempts to use valid previous solution for recognizable problems (Goel & Chandrasekaran, 1992). The new solution is based on adapting or combining old proven solutions by satisfying constraints in order of importance, relaxing constraints and preventing failures (Kolodner, 1992). Every new solution and its history is stored in a repository. The larger the repository of solutions and problems of the design domain, the more effective is the reasoner. The reasoner assumes that some aspects of the solution are implicitly embedded in the invoked case, which could be either a complete product or a single part. Despite some success during mid 90’s, the complexity of PM in terms of its internal data structure is the major impediment to extend the impact of CBD that seems to learn and remember successful relationships problem-solution.

Multi-objective optimization

Multi-objective optimization (MOO) methods basically search through large design spaces guided by objective functions (Kasik, Buxton, & Ferguson, 2005). The generation of these search spaces in the design field relies on PM by continuously updating the inputs and improving the model, or automatically producing an entire set of all possible alternatives within the range of the parameters. While the first method constantly compares the current alternative against the new one and choses the closest to the objectives, the second sorts the alternatives according the same criteria. Either method requires parametrization of the initial model to create the alternatives. Linear growth of variables or ranges triggers an exponential growth of the resulting design space. Subdividing the problem, prioritizing objectives, sequentially satisfying these objectives, or tracking the impact of sensitive variables (Kleijnen, 1997) are some of the techniques to process the larger design spaces produced by taking advantage of the constantly growing computational power. Nevertheless, the two methods produce mostly geometric variations rather than topologically different configurations due to the limitations of PM.

The Meta Modeling Engineering Approach

The purpose of the Meta Model (MM) is capturing domain-specific semantics, attributes, and relationships across parts in very abstract terms (Eck & Schaefer, 2011) without any mediation of geometric models to facilitate the reutilization and extension of the knowledge repository from project to project. The adoption of the Meta-Modeling approach from the Model Based System Engineering (MBSE) process acknowledges the limitation of the current parametric modeling technology to proliferate configurations with different topologies. MBSE is an interdisciplinary process to enable the realization of products and systems (Friedenthal, Moore, & Steiner, 2011). Unlike the interoperability approach of Building Information Modeling (BIM) based on file exchanges, MBSE is based on system integration. For such purpose it uses computer-interpretable MMs to capture, structure, share and reuse domain knowledge across different systems. Since from the perspective of the MBSE, the MM is an abstraction of the structure, behavior, and requirements of any system.

The meta-model

MBSE uses MMs as the main design domain knowledge repositories and the means of communication across systems and related disciplines. The MM captures the domain knowledge through multiple kinds of representation: class definitions, associations, sequences of operation, description of activities, typical use cases and parametric relationships. The Meta-modeling process supports the formalization of requirements, the development of designs, and analysis for verification. It starts in the preliminary design stage, continues through design development and often supports monitoring the life cycle of engineering products. On top of the well-known Kalay’s properties of computational models: completeness, generality, well-formedness and efficiency, MBSE adds some precisions in terms of avoiding ambiguity, achieving accuracy in the object representation and precision in the level of detail.

The MM should preserve the high level design terminology and remain as abstract and general as possible to represent a wide variety of cases. For example, from a design perspective, the definition of a precast panel should be based on abstract concepts avoiding any reference to specific disciplines or computational tools. While any designer can recognize what a precast panel is, different systems can represent it with variable levels of detail and in different ways. Therefore, the MM must remain general to gradually allow adding specific definitions, such as the attributes required by engineering analyses tools. In other words, the MM should not include information of any particular mean of representation, and the computational representation does not require to know about the high level concepts captured in the MM. This independency separates the semantics from the representation.

Non-system specific languages

While design concepts are general terms, the different computational tools differ in the way their internal data structures represent the objects and related attributes. The non-system-specific languages provide the neutral platform required to describe objects avoiding the complexity of every mean of representation. BIM also addresses the same issue by proposing the IFC exchange format to facilitate the communication across different tools through a common neutral schema stemmed from the standardization of the building components. By contrast, MBSE attempts to capture in the MMs the concepts instead of the singularities of these building components, and it uses high level object-oriented languages to describe abstract entities (Kifer, Lausen, & Wu, 1995). These languages are originated in the software engineering field that share the same need of structuring high level concepts. The main languages are UML ("Unified Modeling Language®," 2016), and the derived System Modeling Language (SysML) more recently developed for the systems engineering community ("INCOSE," 2016).

MBSE preferably uses SysML for the formalization of meaning and relationships. This language is intended for design, specification of attributes, different types of analyses and verification. SysML provides graphical notation to represent the objects, their actions and behavior within the system of interest, and it adds the representation of the requirements that the system should satisfy. SysML support different types of human-readable diagrams for these aspects. The *Structure Diagrams* such as the *building block diagrams* represent the structure of the assembly of objects, attributes and relationships, the *Behavioral Diagrams* such as the *activity diagram* represent the behavior of the system while executing a function, and the *Requirement Diagrams* such as the *requirement matrix* represent the relationships between objects of group of them and the requirements that they need to satisfy. All of these diagrams can be created using commercial graphical editors such as Magic Draw ("MagicDraw," 2016) or using the language API of the language.

The main component of the SysML vocabulary is the *block* that correspond to a class or an object. A *block* can represent and specify any abstract or material object by specifying attributes, references to other *blocks*, generalizations of categories in a parent child fashion, operations or actions, and variety of associations among objects. By specifying the values of all them, specific Instance Models (IM) can be created from the general MM templates.

Specification and geometric representation

The Meta-modeling process relies on the separation of the specification of the design configuration and the geometric representation (Figure 1). This separation allows specifying the design alternatives or IMs out of the complexity of the binary tree data structures, and later mapping the specification with actual parametric models. In fact, the specification is a set of discrete instructions that do not build dependencies across objects. The IM preserves all the object or blocks, does not matter if they are or not connected. The associations are not nodes on a hierarchical structure that loses branches every time an object is deleted. On the contrary, the elements of the model are totally independent from the relationships built or unbuilt in the diagrams.

One of the additional benefits of having the specification of all the values of the attributes, and relationships with other *blocks* that represent objects, is the possibility of executing preliminary estimations that do not require the geometry of the objects. Material and cost estimations, adjacencies, areas among others can be calculated before any geometric representation takes place.

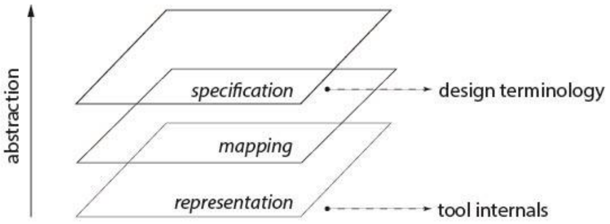

Figure 1: Layers of abstraction

The mapping mechanisms

After specifying the particular values of the IM from the MM, the next step is the geometrical representation. SysML and also UML provide mapping mechanisms to connect the MM with a repository of PMs of parts and assemblies. These resources are extension mechanisms called stereotypes that adds the pointes to the PMs files to the blocks describing objects in the MM. In other words, for every part or assembly of them specified in the MM, a stereotype links the abstract description with the chosen tool. In addition, the IM specifies all the specific values and particular associations. While the

MM is mapped to PMs, the IM provides the values to execute the geometric representation.

The blocks support multiple stereotypes to map different geometric representation of the same object using variety of tools. For example, the same abstract definition of a precast panel can be mapped to either a solid geometry or a b-rep depending on the chosen tool. The stereotypes are the interfaces with all the tools internals, and the blocks capture the design terminology and semantics. Examples of this instantiation mechanisms can be found in the aerospace industry (Bohnke, Reichwein, & Rudolph, 2009; La Rocca, 2011). From the generic MM of the precast panel (Figure 2) topologically different IMs or configurations with different number of windows, frames and glass type can be specified and mapped according to design requirements (Figure 3).

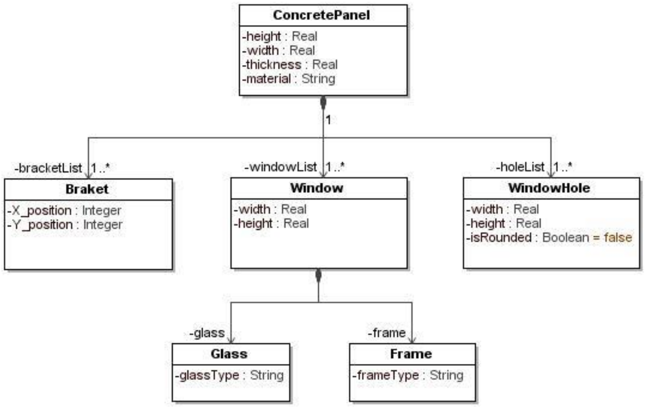

Figure 2: Example of a SysML MM of a precast panel object

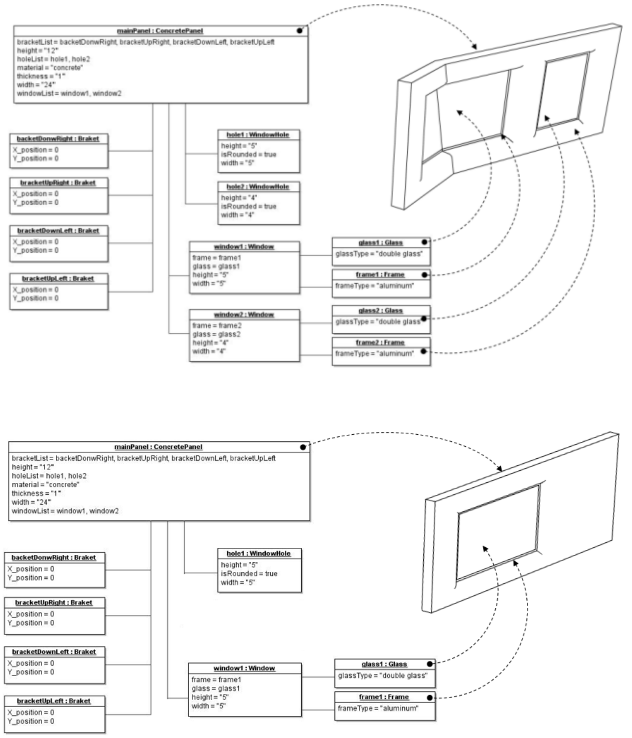

Figure 3: SysML IMs mapped to PMs for geometric representation

Discussion

PM is the dominant technology in computational design. Even though it supports many generative approaches, at the same time implicitly defines strong boundaries of design exploration by prematurely limiting the scope of alternatives to mainly geometric variations. On the other hand, studies on behavior of designers demonstrate that they prefer exploring parallel lines of development early on rather than evolving a particular configuration. Furthermore, the process of questioning the configuration is instrumental in the evolution of the design process. Enabling the generation of parallel configurations better supports the design practice by creating additional design spaces multiplying the universe of possible alternatives. Finding a design solution not only implies geometrical variations, but also constantly changing the very structure of the configuration as well.

The MMs shows an opportunity to develop techniques to produce topologically different alternatives that better suit the trade-off between changing the topology versus changing the geometry of early design stages. The flexibility of the MMs introduces the notion of topological modeling as a long term research effort derived from this study. Although it addressed the problem of the lack of flexibility of the binary trees, still the automation of the mapping and generative processes require far more research efforts. In addition, this endeavor requires deeper understanding of the logic that provides coherence to the arrangements of parts to make some progress in the production of valid specifications.

References

Bernal, M., & Eastman, C. (2011). Top-down Approach to Embed Design Expertise in Parametric Objects for the Automatic Generation of a Building Service Core. Paper presented at the CAAD Futures.

Bernal, M., Haymaker, J. R., & Eastman, C. (2015). On the role of computational support for designers in action. Design Studies, 41, 163-182. doi:10.1016/j.destud.2015.08.001

Bohnke, D., Reichwein, A., & Rudolph, S. (2009). Design language for airplane geometries using the unified modeling language. Paper presented at the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

Clevenger, C. M., & Haymaker, J. (2011). Metrics to assess design guidance. Design Studies, 32(5), 431-456.

Eastman, C. (1999). Building product models: computer environments, supporting design and construction. Boca Raton, Florida: CRC press.

Eastman, C., Sacks, R., & Lee, G. (2003). Development of a Knowledge-Rich CAD System for American Precast Concrete Industry. Paper presented at the Association for Computer Aided Design in Architecture (ACADIA), Muncie, Indiana.

Eastman, C., Teicholz, P., Sacks, R. & Liston, K. (2011). BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors. (second ed.). Hoboken, NJ: John Wiley & Sons Inc.

Eck, O. & Schaefer, D. (2011). A semantic file system for integrated product data management. Advanced Engineering Informatics, 25(2), 177-184.

Frazer, J., Tang, M., & Sun, J. (1999). Towards a generative system for intelligent design support. Paper presented at the Proceedings of the 4th CAADRIA Conference.

Friedenthal, S., Moore, A., & Steiner, R. (2011). A practical guide to SysML: the systems modeling language: Elsevier.

Gentry, R., Sharif, S., Cavieres, A. & Bigg, D. (2016). BIM schema for masonry units and walls. Proceedings of the 16th International Brick and Block Masonry Conference, Padova, Italy, 26-30 June 2016

Goel, A. & Chandrasekaran, B. (1992). Case-based design: A task analysis. Artificial intelligence approaches to engineering design, 2, 165-184.

INCOSE. (2016, 01/12/2016). Retrieved from http://www.incose.org/

Kalay, Y. E. (1989). Modeling objects and environments: Wiley.

Kasik, D., Buxton, W., & Ferguson, D. (2005). Ten CAD challenges. IEEE Computer Graphics and Applications, 81-92.

Kifer, M., Lausen, G., & Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. Journal of the ACM (JACM), 42(4), 741-843.

Kleijnen, J. P. (1997). Sensitivity analysis and related analyses: a review of some statistical techniques. Journal of Statistical Computation and Simulation, 57(1-4), 111-142.

Kolodner, J. L. (1992). An introduction to case-based reasoning. Artificial Intelligence Review, 6(1), 3-34.

Kühne, T. (2006). Matters of (meta-) modeling. Software & Systems Modeling, 5(4), 369-385.

La Rocca, G. (2011). Knowledge based engineering techniques to support aircraft design and optimization: TU Delft, Delft University of Technology.

Lawson, B. and K. Dorst (2009). "Design expertise." Recherche 67: 02.

Lee, G., Sacks, R., & Eastman, C. (2006). Specifying Parametric Building Object Behavior (BOB) for a Building Information Modeling System. Automation in Construction, 15(6), 758-776.

MagicDraw. (2016, 2016). Retrieved from http://www.nomagic.com/products/magicdraw.html

Mantyla, M. (1988). An Introduction to Solid Modeling.

Reichwein, A., & Paredis, C. J. (2011). Overview of Architecture Frameworks and Modeling Languages for Model-Based Systems Engineering. Paper presented at the Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.

Gómez, P. & Swarts, M. (2014). Campus Information-and-knowledge Modeling: Embedding Multidisciplinary Knowledge into a Design Environment for University Campus Planning. International Journal of Architectural Computing 12(4): 439-457.

Unified Modeling Language®. (2016, 02/01/2016). Retrieved from http://www.uml.org