

The Role of Functional Requirements in Multidisciplinary Design: The Case of Application of Photovoltaic Systems to Buildings

Andrés Cavieres

Georgia Institute of Technology, USA

andres.cavieres@gatech.edu

Joseph Goodman

Georgia Tech Research Institute, USA

joseph.goodman@gtri.gatech.edu

Abstract

The paper presents a model-based methodology to support multidisciplinary collaboration for the application of photovoltaic systems to buildings. It focuses on the representation of domain specific knowledge necessary for the design of novel PV racking and mounting structures, based on principles of multi-functionality and functional integration. The proposed representation is based on a language for modeling functional requirements in terms of causal behaviors. These behavioral models provide common ground not only for multidisciplinary design, but also for the elaboration of performance metrics and verification procedures for evaluation of design alternatives. The paper concludes with a discussion on the potential of Model-based approach to support innovation in Design.

Keywords: Knowledge representation; Functional requirement; Model-based systems Integration; Multidisciplinary design; Solar energy.

Introduction

The increasing complexity of contemporary technical artifacts has been partially the result of the enormous progress in computation. Particularly, Computer-Aided Design has allowed the creation of designs that would be very difficult and expensive to produce otherwise. However, much of the advancement has been based on the development of software to represent and control geometry. Other relevant aspects of design, such as the specification, validation and verification of requirements have not enjoyed the same level of progress.

While this situation may not be crucial in projects of low complexity, it is a serious problem in projects characterized by a large number of requirements spanning across the entire lifecycle of a product (Goodman et al. 2011). In these types of projects, the traditional methods of documentation based on written text or spreadsheets are often insufficient (Schamai et al. 2011). Besides being time consuming and prone to error, they can lead to inconsistencies, especially when design changes are being made constantly, and the consequences need to be tracked down and propagated manually (Qamar et al. 2012).

The paper presents a methodology developed as part of the SIMPLE BoS project, intended to capture the requirements associated with the integration of photovoltaic panels in buildings. The general goal of this research project is to design innovative solutions to reduce the Balance of Systems (BoS) costs (i.e. racking and mounting hardware for the attachment PV panels and installation labor) by at least fifty percent of industry best practices.

In order to achieve this goal, the research adopted two main design principles. One has been to take advantage of

opportunities for functional integration. The other has been the development of multi-functional components. The first implies that certain functional requirements should be satisfied by the careful integration of different parts, possibly from different sub-systems. This principle suggests careful attention to the contexts of use. The second principle implies the need for parts and components explicitly designed to support more than one function at the same time. In this way, both principles complement each other. Notice that, in order to apply these principles, every design requirement needs to be treated as, or transformed into, a functional requirement; insofar it may be associated with some type of behavior. In this interpretation of design, conventional categories such as physical, usability or interface requirements are all treated as functional. By adopting these two principles, the research aims to reach its cost reduction targets by proposing racking solutions that not only are simpler and cheaper to produce and install, but which also are more efficient at the whole systems level, adding value to all stages of the product lifecycle, and to a broader set of stakeholders (Cavieres et al. 2013).



Figure 1: Time and motion study for installation of PV panels.

Specific Problem

A fundamental prerequisite prior the application of these design principles is the clarification of the complete scope of requirements associated to the lifecycle of PV systems. These requirements range from manufacturing, storage and transportation, installation and ergonomics, energy production and structural performance, aesthetics, operation and maintenance, reuse and recycling, among others. The reason to tackle such a large scope is that important cost drivers emerge from interactions and dependencies across different kinds of requirements.

However, one of the first challenges faced by the team was the ill-defined nature of some requirements related to PV systems. Few of those requirements, mostly associated with codes and standards, had a clear technical objective and well-defined set of evaluation procedures. Moreover, a significant number of other requirements were found to be ambiguous, based on generalized assumptions regarding the nature of the problem to be solved. In some cases, the assumed technical problem did not have a clear relation to a business goal that would justify the effort to solve it. This kind of ill-defined “floating” requirements made it difficult to identify priorities and to evaluate trade-offs for decision-making.

Two typical examples are “part count” and “tool count”. Both carry the implicit assumption that a racking system design with fewer parts or requiring fewer tools would improve the installation process, making it simpler, faster and therefore cheaper. While the assumption seems reasonable, the relationship between number of parts and tools with installation efficiency is not so straightforward. Sometimes fewer parts lead to bigger and heavier parts, making the installation process more labor intensive and open to injuries, among other issues. On the other hand, two competing design systems requiring just one tool for their entire installation may take different times to be completed, or demand two different skills levels from installers, with different training costs associated with each. Therefore, there is no guarantee that “less is better”. In this situation, only a deeper understanding of the entire context of the installation process, taken as multidimensional problem, may provide a sound basis for a proper description of requirements.

Such multidimensional problems demand a multidisciplinary approach, where every domain-specific perspective is treated as an opportunity for the development of novel solutions. For that purpose, the research launched a partnership between industry and university, involving professionals, students and faculty members from seven different areas, including Architecture, Mechanical Engineering, Civil Engineering, Electrical Engineering, Business, Physics and Biology. As it may be expected, one particular difficulty was to find a common language to capture the shared understanding of the problem, and to support more effective communication and collaboration among team members with such a diverse domain background. More specifically, the

problem was in providing a more systematic representation to capture the set of requirements that needed to be satisfied, along with the network of functional interactions that emerge among them.

Model-Based Design

To address the problem of a common representation, the research adopted a model-based approach rooted on methods developed in Computer Sciences and Systems Engineering. These methods involve the formalized application of modeling to support the description of design requirements, along with analysis, validation and verification procedures necessary to ensure for the quality of proposed designs (INCOSE, 2007).

In relation to requirements, one of the main advantages of the application of a model-based approach is that it enables the description of system design problems from multiple perspectives, thus providing computational support for multidisciplinary collaboration. In particular, it is a platform on which different functional perspectives can be formally modeled. Formalization in this sense refers not only to a common language that all members can understand and use to communicate, but also to a representation that can be computationally processed to automate certain tasks.

Some of these benefits have been made available by the development of modeling languages, such as the Systems Modeling Language (SysML). SysML provides resources to relate requirements to other aspects of a system design, such as design parameters, behavioral models, and analysis procedures to support validation and verification of design alternatives (Friedenthal et al. 2011).

Methodology

Because of these advantages, SysML has been adopted in the SIMPLE BoS project to model the extensive set of requirements associated with the Balance of Systems domain. Given the particular needs of the project, in relation to the design principles adopted, the research team decided to customize the requirement notation contained by default in SysML. The customization was intended to formally capture the behavioral-centric nature of key functional requirements. This step was critical for the purpose of identifying the real problem behind a requirement formulation, namely, the physical phenomena that need to be put under control for the achievement of the main design goals.

This feature was designed to minimize the mistake often found in requirements specification, where solutions are usually interpreted as requirements. For example, in the case of the installation of PV systems in flat roofs of commercial buildings, a common solution to fix the position of the array to resist wind is the use of ballast. This is because in some situations the penetration of the roof membrane is not allowed. As the use of ballast became a very conventional solution, it started to be

treated as an implicit form of requirement for similar situations. The negative implication of this type of mistake is that designers are not exposed to the root of the problem, meaning the physical phenomena (e.g. wind uplift, drag and vibration) that need to be handled. By not doing so, the solution space gets arbitrarily reduced, hindering efforts for design innovation.

The SIMPLE BoS domain specific language

The standard SysML Requirements Diagram and the non-normative Extended Requirements Diagram notations were customized into a Domain Specific Language (DSL), using the UML's profile mechanism. The new DSL, called SBReq inherits basic semantics from the original SysML requirement diagrams, including resources for hierarchical composition, identification (IDs), description of risk, and traceability relations such as refinement, derivation, satisfaction, and verification among others. However, the semantics of the SysML requirements notation are limited, by lacking a proper classification structure. This limitation was addressed in SBReq by the elaboration of a custom classification structure.

The new classification structure is the core of the SIMPLE BoS requirements model, developed following ontological principles for differentiation of categories. At the top of the hierarchy is the category *SB_Requirement*, which adds two new attributes for the description of requirements: the enumerated type *stakeholder*, comprising of a list of typical stakeholders involved in a PV system design; and the enumerated type *priority*, which ranks the relevance of the requirement, from 1 to 5. The top category *SB_Requirement* has two direct sub-categories: *SB_Business_Requirement* and *SB_Design_Requirement*.

In the context of this project, a *SB_Business_Requirement* typically specifies the top level goal of stakeholders. Many of these goals are expressed in terms of some cost reduction strategy, but others focus on strategies to increase revenue (e.g. by improving customer acquisition through marketability aspects of the racking design). A *SB_Business_Requirement* is related to a *SB_Design_Requirement* with a dependency relation. In this model, it means that the latter depends on the former. If a business requirement changes, then the design requirement needs to be revised and eventually updated.

The distinctive feature of *SB_Design_Requirement* is that it must be specified by a constraint, on either some structural property of the designed artifact (e.g. maximum length, a particular shape, etc.) or on some behavioral property (e.g. a minimum electrical resistance level, a maximum deflection, or some other capability of interest). While the first type of constraint specifies limits on values of static characteristics of the designed artifact, the second one specifies limits on the values of behavioral properties when evaluated under specific circumstances.

When a requirement of type *SB_Design_Requirement* is specified by a behavioral constraint, it is specialized into a functional requirement (i.e. *SB_Functional_Requirement*). This approach is based on the Functional Representation (FR) schema, developed by Chandrasekaran and Josephson (2000). The other specialization of *SB_Design_Requirement* is the category *SB_Aesthetic_Requirement*.

An instance of *SB_Functional_Requirement* can be decomposed into sub-requirements of the same type, thus supporting the method of functional decomposition originally formulated by Pahl and Beitz (1996). This method has been used in the domain of building design to support design-analysis integration by Gielingh (1988), Augenbroe (1992), and Eastman (1992), among others.

More recently, Augenbroe (G. Augenbroe, 2011) has proposed a reformulation of this method, according to which a functional requirement needs to be described in terms of a measurable expression of performance. This means that, in addition of a behavioral constraint, a performance indicator (PI) has to be identified, preferably after negotiation between stakeholders. A functional requirement described in terms of a PI becomes a specialization called a performance requirement. A performance requirement is the lowest level (i.e. the most detailed level) of specification for a requirement, which can be subjected to quantitative analysis or other form of experiment for verification performance.

The SBReq language captures the semantics proposed by Augenbroe's method through the definition of the category *SB_Performance_Requirement*. This category includes a relationship with at least one PI (*SB_Performance_Indicator*), and with at least one performance quantification method (*SB_Performance_Quantification*).

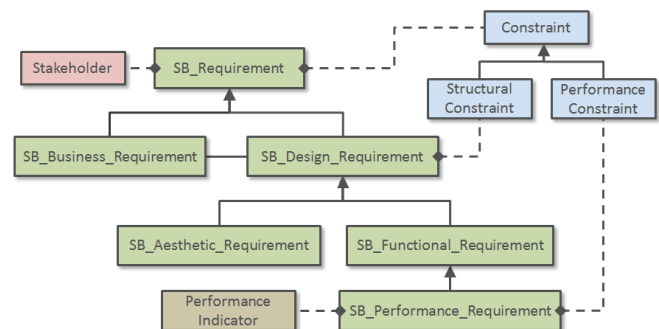


Figure 2: Ontology of requirements used in SBReq (extract).

However, before a functional requirement can get specialized into a performance requirement, it is necessary to validate it. This means that the functional requirement under consideration must rigorously capture the phenomena implied by its description, in such a way that it allows a comprehensive understanding of the problem, possibly from more than one perspective. In this research, phenomena refer to the largest set of behaviors known by experts to be possibly involved in a design problem (Borgo et al. 2009). In this way, the context gets fully characterized, by including not only the set of intended functional behaviors, but also those which are not intended, but that are expected to occur given the interactions with other components of the system.

Since the design principles of functional integration and multi-functionality are predicated upon this contextualization, a descriptive model of the identified set behaviors and their interactions is needed. In SBReq, this is satisfied through the instantiation of a *SB_Behavioral_Description_Model*. The concept of a functional requirement explained in terms of a behavioral model is similar to the concept of function as behavioral abstraction proposed by the Structure-Behavior-Function (SBF) schema (Goel et al. 2009).

In concrete terms, a *SB_Behavioral_Description_Model* is a pointer to a SysML behavior diagram, usually an Activity Diagram or a State Machine Diagram. Generally speaking, these diagrams allow the description of behavior in terms of a set of states, inputs, outputs, and transformation operations that explain how inputs get transformed into outputs. A *SB_Behavioral_Description_Model* can point to other inner behavioral models, each containing sets of participant elements (e.g. physical, agentive and non-agentive entities), either from the design itself or from the broader context.

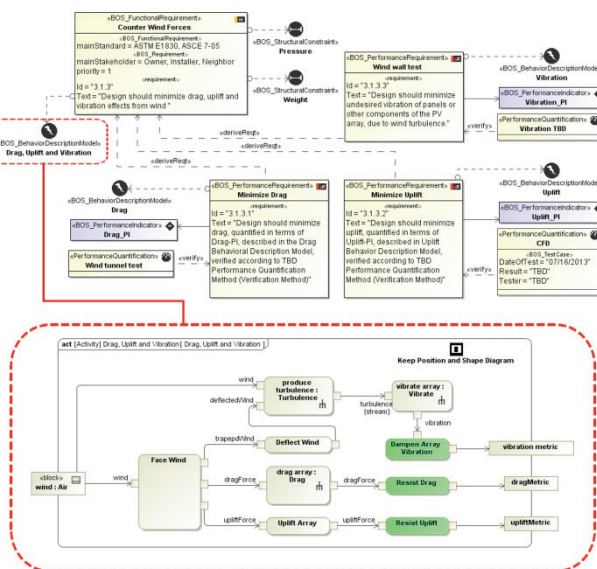


Figure 3: Example of a *SB_Functional_Requirement* to counter wind forces. The pointer *SB_Behavioral_Description_Model* inside small dashed box on top left. Set of relevant behaviors related to wind inside big dashed box, at the bottom.

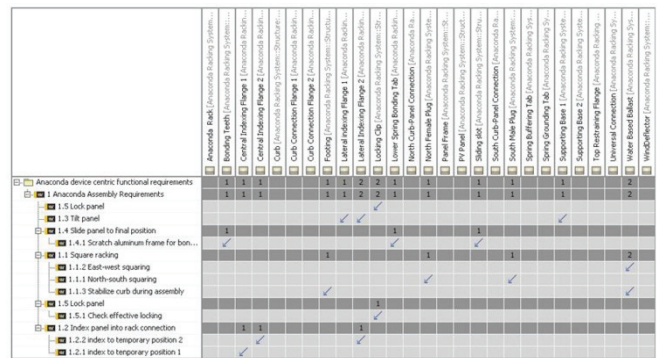


Figure 4: Satisfaction Matrix. Functional integration depicted horizontally. Multifunctional components depicted vertically.

Discussion

In many design disciplines, including Architecture, Industrial Design and Engineering, the description and management of requirements is often problematic. Key requirements may be ill-defined, which hinders the early identification of dependencies and interactions among them. This situation gives rise to “floating requirements” that are difficult to assess and prioritize given the lack of more precise description of relationships with other requirements and context of use.

Particularly problematic is the lack of relationships between so-called functional requirements with high level business goals. These relationships are important to justify any effort and investment in seeking a solution, as well as to establish pragmatic trade-off decision criteria.

Similarly, the absence of more comprehensive description about the phenomena to be controlled by the design limits the scope of design possibilities, inhibiting innovation. For example, it is common to find structural constraints (e.g. legacy solutions) specified as functional requirements. This situation is particularly evident when requirements are derived from prescriptive codes and standards, or conventions of practice. A well characterized functional requirement in turn, opens up the possibility to override structural constraints if a novel design is proven to meet minimum performance levels that capture relevant goals of the stakeholders (e.g. efficiency, productivity, safety, revenue, etc.).

For this to happen, the characterization should be based on model-based description the set of causal behaviors involved in the phenomena. Such model-based approach should enable multiple perspectives of the identified phenomena, in order to support multidisciplinary collaboration. Ultimately, this multidimensional characterization will promote novel design approaches, based on principles of functional integration and multi-functionality.

Conclusions

The paper presented a model-based methodology aimed to support rigorous description of functional requirements. The methodology was developed in the context of the SIMPLE BoS project, motivated by the need to promote new design solutions in the application of photovoltaic panels in buildings. A SysML domain specific language called SBReq was introduced, along with its main motivations and theoretical background.

Future work will address the validation of the methodology and language, in relation with two main aspects. The first deals with issues of usability, consistency, and effectiveness regarding the specific goals of the SIMPLE BoS project. The second aspect refers to the potential generalization and applicability to other design domains, which includes the problem of reusability of models.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy under Award Number DE-EE0005441.001.

Disclaimer

This paper was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

International Council on Systems Engineering (INCOSE), Systems Engineering Vision 2020, Version 2.03, TP-2004-004-02, 2007.

- Augenbroe, G. (1992). Integrated building performance evaluation in the early design stages. *Building and Environment*, 27(2), 149-161.
- Augenbroe, G. (2011). The role of simulation in performance based building. In J. L. M. Hensen & R. Lamberts (Eds.), *Building Performance Simulation for Design and Operation* (pp. 15-36): Spon Press.
- Borgo, S., Carrara, M., Garbacz, P., & Vermaas, P. E. (2009). A formal ontological perspective on the behaviors and functions of technical artifacts. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 23, 3-21.
- Cavieres, A., Al-Haddad, T., Gentry, R., Nagel, K., & Goodman, J. (2013). SIMPLE BoS: Towards a multidisciplinary integration of Photovoltaic Energy in Buildings. 2013 ARCC Spring Research Conference - Architectural Research Centers Consortium, Charlotte, North Carolina.
- Chandrasekaran, B., & Josephson, J. R. (2000). Function in Device Representation. *Engineering with Computers*, 16(3-4), 162-177.
- Eastman, C. M. (1992). A data model analysis of modularity and extensibility in building databases. *Building and Environment*, 27(2), 135-148.
- Friedenthal, S., Moore, A., & Steiner, R. (2011). *A Practical Guide to SysML - The Systems Modeling Language*: Morgan Kaufmann.
- Gielingh, W. (1988). General AEC reference model: ISO TC184/SC4/WG1 document 3.2.2.1
- Goel, A. K., Rugaber, S., & Vattam, S. S. (2009). Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 23, 23-35.
- Goodman, J., Caravati, K., & Kirby, E. (2011). System Life Cycle Evaluation, SLICE SM, for Green Building Technologies. Strengthening the Green Foundation - Research and Policy Directions for Development and Finance, Atlanta.
- Pahl, G., & Beitz, W. (1996). *Engineering design : a systematic approach*. London: Springer.
- Qamar, A., Paredis, C. J. J., Wikander, J., & During, C. (2012). Dependency Modeling and Model Management in Mechatronic Design. *Journal of Computing and Information Science in Engineering*, 12(4), 041009-041009.
- Schamai, W., Helle, P., Fritzson, P., & Paredis, C. J. (2011). Virtual Verification of System Designs against System Requirements. In J. Dingel & A. Solberg (Eds.), *Models in Software Engineering* (Vol. 6627, pp. 75-89): Springer Berlin Heidelberg.