Building Specifications as a Domain-Specific Aspect Language

Max E. Kramer[⋆†]
Karlsruhe Institute of Technology[⋆]
max.kramer@student.kit.edu

Jacques Klein[†]
University of Luxembourg, SnT[†]
jacques.klein@uni.lu

Jim R. H. Steel[‡]
The University of Queensland[‡]
j.steel@uq.edu.au

Abstract

In the construction industry an increasing number of buildings are designed using semantically rich three-dimensional models. In parallel, additional information is specified in a natural-language document called a *building specification*.[1] As not all details are present in the model these specifications have to be interpreted whenever costs are estimated or other analyses are performed. In this paper, we argue that building specifications contain cross-cutting concerns. We also argue that domain experts should be given the ability to formulate building specifications using a domain-specific aspect language so that the corresponding details can automatically be integrated into the model.

The language needs to support a multitude of domain-specific abstractions that are absent in the building metamodel. Therefore we propose to allow the domain experts to extend the language iteratively by defining interpretation patterns [1]. Such a model enriching specification will improve tasks requiring detailed information and will allow for earlier or even concurrent development of the building specification along with the model.

Categories and Subject Descriptors D.2.13 [*Software Engineering*]: Reusable Software—Domain Engineering

Keywords DSML, DSAL, MDE, AOM, Model Weaving

1. Introduction

When new buildings are constructed architects and engineers make use of semantically rich three-dimensional models, called Building Information Models (BIM). These models are capable of capturing a very high level of detail. However, many details are omitted because too much effort is required to add specific information in all the affected places. Instead these omitted details are formulated in a natural language text called building specification. An example sentence of a building specification is: "All windows of the ground floor have to be made of high grade steel."

The consequence of sourcing out these details into a specification text is that the building model does not contain all the information relevant for cost estimation and other analyses. For this reason the specification text has to be reinterpreted whenever these tasks are performed, which may be time-consuming and error-prone.

Some of the current BIM tools allow users to trace and link information between building models and building specifications. Others provide import features for specifications. But they fail in adding the corresponding elements at multiple places and do not support conditional introduction.

The contribution of this paper is twofold: First, we argue that building specifications are domain-specific cross-cutting concerns. Second, we propose to define a Domain-Specific Aspect Language (DSAL) for building specifications that can be used to enrich building models. We are confident that an aspect language will help us to capture the usually cross-cutting concerns expressed in building specifications as easily and precisely as possible. In order to be suited for direct introduction of details into the building model this language will have to refer to model elements and can also be called a Domain-Specific Model Transformation Language (DSMTL). To give domain experts the ability to use domain-specific abstractions that are not present in the metamodel we propose to let them define the semantics of the language using interpretation patterns.

The resulting enriched building models will speed up tasks that need detailed information that is currently only indirectly available in the specification text. Concurrent development of executable building specifications would also give stakeholders the ability to assess the impacts of their decisions at an earlier stage.

The remainder of this paper is structured as follows. Section 2 introduces basic concepts of the involved domain and the techniques used. Section 3 explains that building specifications contain domain aspects whereas Section 4 presents a concept how to automatically integrate these aspects into the building model. Section 5 draws some final conclusions.

[1] Building specifications are very different from software specifications.

DSAL'12, March 27, 2012, Potsdam, Germany.
Copyright © 2012 ACM 978-1-4503-1128-1/12/03. . . $10.00

2. Foundations

We want to give the reader a short introduction to the concepts and techniques on which our work is based.

2.1 Model Driven Engineering

Model Driven Engineering (MDE) is a paradigm in the field of Software Engineering that elevates models to first-class citizens during all phases of systems design. All engineering artefacts are expressed as models, so that model-based techniques can be applied to them. Thus, MDE is mainly about what perspectives and formalisms can be helpful when designing systems. Metamodels are an essential concept of MDE. They describe the structure of models, thus defining what elements can be found and how they can be related.

Model Weaving Aspect-Oriented Modelling (AOM) allows modellers to describe cross-cutting concerns using concepts that are specifically designed for this purpose. AOM techniques typically make use of the same concepts used in Aspect-Oriented Programming (AOP). A *pointcut* describes at which points of a model an aspect should be applied. An *advice* defines what should be done whenever a part of a model matches the description of a pointcut. When an aspect is applied to a base model the first step is to identify the points where a given aspect should be applied, also called *join points*. The second step is to execute the changes described in the advice at these points. This process of incorporating the model elements of the advice into the base model is also called *model weaving*.

Several tools have been proposed for Model Weaving. SmartAdapters [3] is an approach in which the weaving process is specified using a *composition protocol*. GeKo [2] is a generic model weaver sharing some concepts with SmartAdapters. It uses a declarative mapping from pointcut to advice elements instead of an imperative composition protocol.

2.2 Building Information Modelling (BIM)

If a building model contains semantic information in addition to the geometry we speak of Building Information Modelling. Most Computer Aided Design (CAD) tools use proprietary formats for representing and rendering these models but export them using the de-facto standard Industry Foundation Classes (IFC). We will use a framework that bridges the IFC and EMF technological spaces as described by Steel et al. [6] in order to apply MDE techniques to building models. As many stakeholders use partial models of remarkable size and complexity, building models present challenges to the MDE community in terms of scalability and integration.

A common technique to avoid adding the same details at various places in a building model is to define the details in a document called *building specification*. This natural language text can contain very exact information as it has a legal character for contractors. Together with the building model the building specification is used as the main input for various analysis tasks like cost estimation.

3. Building Specifications are Cross-Cutting

The purpose of a building specification is to provide additional details for an existing building model. In many cases these details affect more than a single group of related building elements. Instead, various groups of buildings elements, that occur independent from each other in different contexts, are affected. The additional detail expressed within the building specification, however, is the same for the different contexts. For these reasons such concerns of a building specification text can be seen as cross-cutting concerns of the problem domain. So far there was no consensus on whether domain models contain such domain-specific aspects [4, 7]. Irrespective of this, it is understood that cross-cutting concerns in general are best tackled using aspect-oriented (modelling) languages with explicit support for them.

The example provided in Section 1 presents only one possible type of specification concern. Note that we do not claim that all building specification concerns are cross-cutting concerns or even that they follow the structure of our example. How many concerns of building specifications are cross-cutting and in which way is still an open research question.

4. Executing Domain-Specific Aspects

We argue that the current way of writing building specifications as natural language text is not the best way to deal with cross-cutting building concerns. The building design process is making increasing use of design analyses, including automated processes. But the formal imprecision and lack of machine readability of the building specification means that its information is not considered in these analyses, particularly during early design. A quantity surveyor, for example, calculates the quantities of material and work needed to construct a building [5]. Although he can use the model to retrieve the rough structure he also has to take the details of the specification text into account. Therefore the process of cost estimation, which is vital to construction projects, is complex, time-consuming and requires a lot of skill and experience. If a significant part of the details expressed in building specifications were directly available in the building model this and other analysis tasks could be performed faster and with a higher degree of precision and automation.

Getting rid of natural language texts One possibility to integrate specification details into the model could be to transform existing specification texts into a machine-readable format and to use this format to add the corresponding details. This strategy would involve Natural Language Processing and is probably more complex than needed. Another approach could be to create building specification texts that are used by model experts to add the corresponding details to the model. Both solutions suffer from two problems. First, writing and understanding building specifications requires a lot of domain-knowledge and therefore programs and model experts will have difficulties to accomplish these tasks. Second, like all natural language texts, building specifications can be open to multiple interpretations. This complexity and possible ambiguity makes building specifications unsuitable for these two transformative approaches.

Domain experts increase precision For new building projects it is sufficient to provide domain experts a DSAL. Instead of writing a building specification as a natural language text they use a tool that helps them in writing formal sentences using the DSAL. These sentences represent instructions that specify which details have to be added to which building elements under which conditions. In order to allow the integration of additional information into the model, the DSAL has to overcome the ambiguities of natural language building specifications. As building specifications are characterized by a multitude of domain-specific abstractions that are not present in the building model, the language has to provide its users the ability to use these abstractions.

Both goals can be achieved using an interpretation pattern approach that is similar to the one used by Lugato et al. [1]. The DSAL is defined as a controlled natural language that may parse sentences that cannot be interpreted. Whenever a domain expert writes a specification sentence that can be parsed but cannot be interpreted using an existing interpretation pattern he is asked to create a new pattern as described in the next paragraph. This gives the domain experts the ability to write executable building specifications incrementally while increasing the expressive power of the language they are using. Steel and Drogemuller [5] presented a related but fixed DSMTL for calculating the quantity of material and work that is needed to construct a building.

Semantics using declarative aspects For our proposed approach to be useful the domain experts have to be capable of defining the semantics of their interpretation patterns. To achieve this we propose the use of the generic model weaver GeKo. A central property of GeKo is that pointcuts and advice can be specified using the domain metamodel that is already used for the base model. This means that users do not have to learn a new notation or language but can directly express their aspects in a known language. Weaving is performed based on a user-defined mapping from pointcut elements to advice elements. This simple mapping is very intuitive and can be retrieved automatically by the weaver in all unambiguous situations. A new implementation of GeKo, that is still under development, is available online.[2]

An interpretation pattern for our DSAL is defined by domain experts in two steps. First, they specify the structure of the sentences to be interpreted as an abstract syntax tree pattern. Then, they define which building elements, attributes and relations have to occur in the model when the specification information should be applied (pointcut). Finally, they define which elements, attributes and relations should be present once the specification information has been applied (advice). Only if the weaver could not determine the correspondences between all pointcut and advice elements remaining correspondences have to be added in a last step.

The aspect that is generated to implement our example from Section 1 is presented in Fig. 1. Elements that are part of the pointcut but not part of the advice are marked in gray.

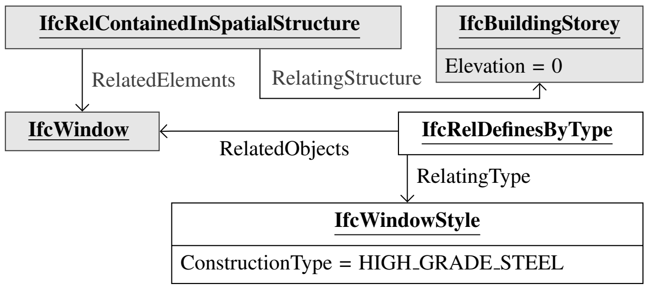

Fig. 1. Example implementation aspect (pointcut in gray).

5. Conclusions

In this paper we argued that some of the concerns expressed in building specification texts are cross-cutting. This is remarkable as it has been strongly discussed whether domain models are aspect free [7] or not [4]. We proposed the definition of a DSAL for building specifications that is iteratively defined by domain experts in terms of interpretation patterns. Moreover, we explained how to use this language to automatically integrate specification concerns into building models using a generic model weaver. We illustrated characteristics of the approach, provided an example and gave an outlook on how construction industry stakeholders could profit from the language and resulting enriched building models.

References

[1] D. Lugato, F. Maraux, Y. L. Traon, V. Normand, H. Dubois, J.-Y. Pierron, J.-P. Gallois, and C. Nebut. Automated functional test case synthesis from thales industrial requirements. In *Proc. of RTAS 2004*, pages 104–111. IEEE, 2004.

[2] B. Morin, J. Klein, O. Barais, and J.-M. Jézéquel. A generic weaver for supporting product lines. In *Proc. of Early Aspects at ICSE'08*, pages 11–18, New York, NY, USA, 2008. ACM.

[3] B. Morin, O. Barais, G. Nain, and J.-M. Jézéquel. Taming Dynamically Adaptive Systems with Models and Aspects. In *Proc. of ICSE'09*, Vancouver, Canada, 2009.

[4] A. Rashid and A. Moreira. Domain models are not aspect free. In *Proc. of MoDELS 2006*, volume 4199 of *LNCS*, pages 155–169. Springer-Verlag, Berlin/Heidelberg, 2006.

[5] J. Steel and R. Drogemuller. Domain-specific model transformation in building quantity take-off. In *Proc. of MoDELS 2011*, LNCS, pages 198–212, Berlin/Heidelberg, 2011.

[6] J. Steel, K. Duddy, and R. Drogemuller. A transformation workbench for building information models. In *Theory and Practice of Model Transformations*, volume 6707 of *LNCS*, pages 93–107. Springer-Verlag, Berlin/Heidelberg, 2011.

[7] F. Steimann. Domain models are aspect free. In *Proc. of MoDELS 2005*, volume 3713 of *LNCS*, pages 171–185. Springer-Verlag, Berlin/Heidelberg, 2005.

[2] code.google.com/a/eclipselabs.org/p/geko-model-weaver