The Role of Context Models in Association with Flexible Design Processes – Position Paper

Mirjam Minor, Daniel Schmalen, Andreas Koldehoff*

University of Trier, Department of Business Information Systems II
54286 Trier, Germany
{minor, schmalen}@uni-trier.de

*Silicon Image, GmbH, Garbsener Landstr. 10
30419 Hannover, Germany
andreas.koldehoff@siliconimage.com

Abstract. In this position paper, we discuss the interdependencies of context models and flexible design processes in the chip industry. We illustrate this by an ontology-based context model for digital design processes. Open issues concern the role of context adaptation for a case-based authoring support of the process models as well as the representation and explanation of changes of the context model.

Introduction

Our basic hypothesis for this position paper is that the semantics of the adaptation of design processes can be captured by means of formal context models. In order to discuss this, we associate process models with context models in a one-to-one relationship. During the ongoing processes, modifications of the process models themselves as well as of the context models may occur. The aim of this work is to develop context-aware methods [4] for the authoring support of process modifications. Case-based reasoning is employed to reuse experience with the adaptation of processes. We illustrate our considerations with sample scenarios from the chip design domain. We believe that the results are also applicable for other domains with flexible processes, e.g. for agile software development projects.

Context in Digital Design Projects

Our work on context relies on Dey's definition of *context* [2] as "any information that can be used to characterize the situation of an entity". The entities that we consider in the chip domain are the digital design projects at a certain stage of their life-cycle. They are characterized by sets of context factors. A context factor describes an influence factor that has a significant impact on the design process. For instance, the technological risk is an important context factor as it may happen that previously working algorithms will not run when they are implemented with a new, smaller chip technology or that their performance will not be sufficient. A high value of the technological risk factor may lead to different hardware design processes for the same algorithm than a low value. In order to deal with a high risk, alternative solutions can be implemented in parallel, for instance, or the frequency of reviews can be planned accordingly. The context factors describe either basic conditions for the design process like risk factors or they may cover requirements to the object of the design project like the range of temperature the chip shall be applied for. Please regard that the entity for which we describe a situation by the context model is the process and not the object of the design. So, we think it is legitimate to integrate a subset of the requirements that apply for the final product into the context of the production process. The aim of reasoning about this context is to reuse the adaptations that have been applied to the ongoing processes (see below). We agree with our reviewers that the borderline between context knowledge and application knowledge is fuzzy in our scenario. We consider factors from both the application and the design context. The *application context* concerns the requirements of the customers and the end customers, e.g. the users of a mobile phone, whereas the *design context* concerns the realisation of the design by tools and designers.

Goals of the context model

The goals of acquiring a context model are manifold in chip design: First, the model is used in order to *document factors* that have a significant impact on the design processes. All design processes follow a standard design flow which describes the design steps of a project ordered in a particular control flow. The context model is integrated with the design flow in order to realize the documentation of context factors as follows:

- The context model provides a central repository for important features like the description of pins and registers within a project. Typos in pin descriptions have been one of the main sources of error in past digital design projects.
- Changeable factors like the market development are reviewed regularly during the design flow and adapted to the current situation.
- The development of particular factors can be monitored for the assessment of projects concerning its state, the remaining risk, the estimated time to completion and so on.

At the moment, the context model is acquired manually but there is a certain potential for automating this documentation task in future, e.g. by importing data or by observing factors. As a second goal, the context model is used for *automatic analysis and reasoning*:

- The analysis of risk factors provides support for the risk management.
- The dependency analysis of context features is useful for the change request handling.
- We are planning to apply case-based reasoning for authoring support when design processes have to be adapted to changes [3].

Requirements analysis

The above goals have been figured out during a systematic requirements analysis including interviews and discussions with several digital design experts, our analysis of the chip domain by means of sample projects, and a literature research on context models.

The discussions with the chip designers have shown that the design processes of different types of chips depend on different sets of context factors. For instance, some chips in the automotive area are designed for multiple applications which might not even be known in advance. The application context for the design of such multiple-purpose chips is restricted to general quality and security aspects, i.e. the application context is not important enough to be modelled in detail. On the other hand, chips for the customized area are highly depending on the application context: The design processes are affected by recent market development issues, for instance like the number of pixels for a camera that is en vogue. Changes cause major adaptations of the design itself as well as of the ongoing design process. Thus, the acquisition of application context factors is crucial for the design process of chips from the customized area. The results from the discussion with the experts led us to the decision to develop a *configurable context model*. For each type of chip, the context model can be tailored to the appropriate set of context factors.

Furthermore, our analysis of sample documents from the chip domain has yielded several interdependencies between context factors. For instance, the output pins of a chip segment have a direct impact on the input pins of a successor chip. We have chosen to organize the context in an ontology-based model in order to *describe interdependencies*.

Korpipää et al [5] (cited according to [6]) identified requirements for ontology-based context models that hold also for our approach:

- *Simplicity*: The used expressions and relations should be as simple as possible to simplify the work of applications developers.
- *Flexibility* and *extensibility*: The ontology should support the simple addition of new context elements and relations.
- *Genericity*: The ontology should not be limited to special kind of context atoms but rather support different types of context.
- *Expressiveness*: The ontology should allow to describe as much context states as possible in arbitrary detail.

The requirement of flexibility and extensibility includes the above requirement of configurable contexts that we derived from our analysis of the chip domain.

The Context Ontology

In order to realize the manifold goals and requirements described in Section 2, we developed a top-level ontology for our context model that is depicted in Figure 1. We distinguish two areas for the definition and the application of context factors:

1. *FactorCategory*, *FactorDefinition* and all its descendant classes (*BooleanFD*, *DateFD*, etc.) belong to the pool of context factors. The pool contains the definitions of the factors including the value types and default values.
2. *Project*, *ProjectUnit* and *Factor* with its descendants (*BooleanFactor*, *DateFactor*, etc.) are for the assignment of factors to project units and their unit-specific values.

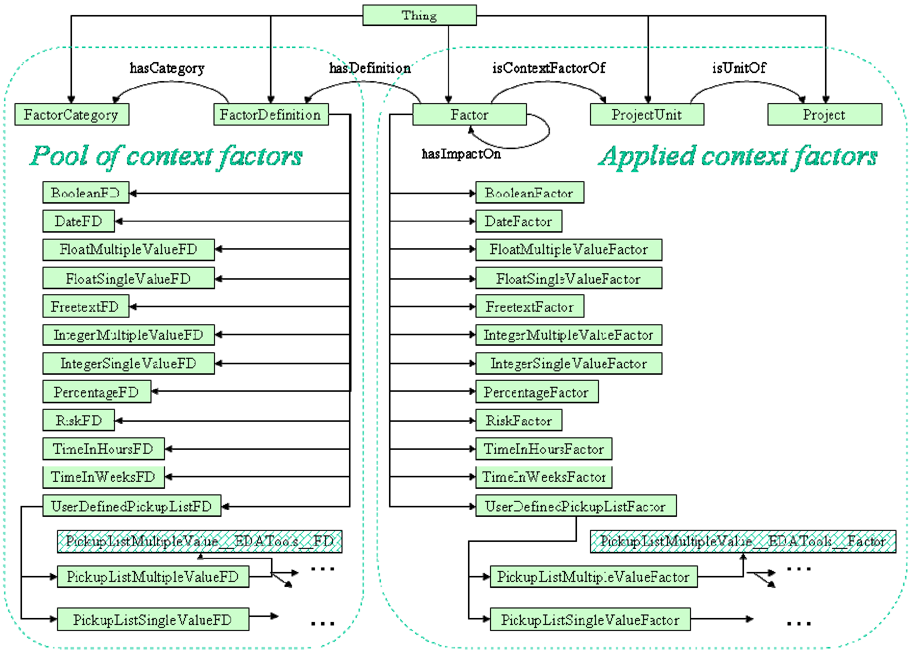

Fig. 1. The top level ontology of the context exchange format.

Assigning a context factor to a project unit means to add it to the context model of the according design process. The interdependencies between applied context factors are captured in the 'hasImpactOn' relation. The top level ontology contains general data type classes that are also applicable for other domains as well as user-defined classes that are dedicated to the chip design domain, for instance, the pickup list for the EDA (electronic design automation) tools. The context models can be specified, updated and exchanged by means of our context acquisition tool based on Protégé-conformant OWL [1]. We hide the complexity of the OWL format from the users. For further information on the context acquisition tool, we refer to the literature [7].

The Role of the Context for the Adaptation of the Processes

The context model that is associated to a process model should be considered by the case-based modelling support mechanism. The case base contains experience with the adaptation of ongoing processes by means of late-modelling and ad hoc changes.

A case consists of two revisions of an ongoing process, namely the process revision before (the problem part) and after the modification (the solution part). The user, i.e. the process planner, can reuse adaptations of processes that have a similar structure and that have been in a similar situation before the modification. The retrieval of useful processes from the case base requires a similarity measure for both, the context model and the control flow structure of the ongoing process. As a straight-forward solution, we have proposed a traditional structural CBR approach for the context model in combination with a new graph-based similarity function for the control flow structure [3]. In the following, we discuss some open issues that might lead to a more sophisticated approach for an integration of context models and flexible process models in the future.

The motivation for this work results from the following observations: The values of the factors within a context model can be modified during the ongoing design processes. Changes of the context model are related to adaptations of the design processes in the following three, alternative ways:

1. The modification of a context factor may **cause** an adaptation of the process.
2. The modification of a context factor may **happen together** with an adaptation of the process.
3. The modification of a context factor may **happen independently from** an adaptation of the process.

For instance, a change request of the customer who has decided to decrease the target area of a chip module during the ongoing process leads to a repetition of the design steps that are affected by the size of the module. In this sample, the modification of the context factor for the target area causes an adaptation of the ongoing process. Alternatively, such a change may go hand in hand with the adaptation of the process when additional features have to be implemented and the planned size is not adequate for the overall functional requirements of the chip module any more. I.e. the adaptation of the context model happens together with the process adaptation. Thirdly, in a very early phase of the chip design process, a modification of the target area may not have an impact on the design process of the module at all. In terms of a case-based support of process authoring, the context factors may belong to the problem part, to the solution part, or to none of both parts of a case. We think that it is not easy to handle these multiple roles of the context factors. The role of a factor modification depends on the particular situation and can presumably not be determined automatically. It seems also hardly possible to explain the different possible roles to the users who specify queries in order to get modelling support.

Furthermore, we assume that the difference between two succeeding values, for instance an increasing risk, is often more crucial for the process adaptation than the absolute value. The representation of this delta for retrieval purposes is an open research issue as well as how it can be decided and communicated to the user whether the absolute values or the deltas of a certain factor should be compared during the retrieval.

References

[1] Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In Staab, S., Studer, R., eds.: *Handbook on Ontologies*. Springer-Verlag, 2004, chapter 4, pp. 67 – 92.

[2] Dey, A. K.: Understanding and Using Context. Personal and Ubiquitous Computing 5/1, Springer-Verlag, 2001, pp. 4 - 7.

[3] M. Minor, A. Tartakovski, R. Bergmann. Representation and Structure-based Similarity Assessment for Agile Workflows. *10th International Conference on Case-Based Reasoning*, accepted for publication.

[4] Kofod-Petersen, A., Mikalsen, M.: *Context: Representation and Reasoning – Representing and Reasoning about Context in a Mobile Environment*. Revue d'Intelligence Artificielle 19, 2005, pp. 479 - 498.

[5] Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H., Malm, E.-J.: Managing context information in mobile devices. IEEE Pervasive Computing 2/3, 2003, pp. 42 – 51.

[6] Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey On Context-Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing, Inderscience Publishers, 2006, (forthcoming).

[7] Minor, M., Schmalen, D., Koldehoff, A., Bergmann, R.: Configurable Contexts for Experience Management. In Gronau, N., ed.: *4th Conference on Professional Knowledge Management - Experiences and Visions*. Vol. 2., Potsdam, Univ. of Potsdam, GITO-Verlag Berlin, 119 – 126, 2007.