Five questions to be clarified before starting to model conceptually

Elmar J. Sinz*,[a]

[a] Fakultät Wirtschaftsinformatik und Angewandte Informatik, University of Bamberg, Germany

Abstract. *Conceptual modeling looks at the concept of a system. An important property of such a concept is that it can be viewed from at least two perspectives: The perspective of the users of the system and the perspective of the implementers. Based on these different perspectives, five sets of questions are identified that should be sufficiently answered before modeling is started. This is not a guarantee for the success of the modeling project, but it is necessary.*

Keywords. conceptual modeling

Communicated by Peter Fettke. Received 2020-11-18. Accepted on 2020-11-30.

Introduction

The term "conceptual modeling" emphasizes that it is about a concept. The system to be modeled is designed as a concept, which refers to the goals pursued with the model and abstracts from the concrete implementation [1] . In other words, alternative solutions for these goals are kept open.

This shows that a conceptual model is similar to an interface, which can be viewed from two different perspectives: the perspective of the user, whose goal is to express his or her requirements, and the perspective of the implementer, whose goal is to transfer these requirements into a solution.

An example from house construction will illustrate this. The conceptual model is the building plan. Both the building owner (user) and the architect (implementer) must understand the building plan in detail. The building owner and the architect communicate using the building plan. It does not matter who creates the building plan. It may be the architect, whereas the building owner only formulates his requirements as a sketch or a list of key points. In the end, the building plan is the basis for the realization of the building project.

Typical for information systems is distinguishing the perspectives of the specialist department (user) and the IT department (implementer). The gap between these perspectives may be particularly wide here, as users and implementers usually have different ways of thinking. In other words, both groups use different metaphors.

The separation between the user's perspective and the implementer's perspective is analogue to the concept of abstract data type which is well known from software engineering. An abstract data type implements external data types and operators (requirements) by a program based on internal data types and operators (Ferstl and Sinz 2013, pp. 327f). The implementation remains hidden for the user or at least the user does not have to know it.

Just as the concept of the abstract data type can be applied to complex and detailed problems, the same applies to conceptual models. Whether it is a comprehensive IT system or a function of an IT component, modeling is always based on the specific perspectives of the user and the

* Corresponding author.
E-mail. elmar.sinz@uni-bamberg.de

Thanks to Dominik Bork (Technical University Vienna) and Felix Härer (University of Fribourg, Switzerland) for reading and commenting on an earlier version of this article.

[1] For different views on conceptual modeling see e.g. (Schlageter and Stucky 2013; Strahringer 2013; Thalheim 2010)

implementer. In doing so, the implementer of one certain level becomes the user of the next lower level.

The separation and interconnection of the perspectives of user and implementer is the focus of this short article. It is carried out on five sets of questions which are assigned along with the components of a modeling project:

1. User scenario
2. Modeling language
3. Modeling tool
4. People
5. Benefit of the model

These questions must be clarified in the run-up to a modeling project. The more carefully this is done and the better the answers are conclusive, the higher the chance for a successful modeling result will be.

1 User scenario

In the very beginning of a modeling project, the user scenario must be defined. Important questions are: Who are the users and who are the implementers in the given modeling project? What goals do both groups pursue with the model? What do users and implementers want to understand, what do they want to communicate?

Two examples should illustrate this: (1) A business process is modeled to be supported by a workflow system. Then the users are the operational managers for the business process, the implementers are the experts for web services and their coordination with BPEL (Web Services Business Process Execution Language (Jordan et al. 2007)). (2) A business process is modeled to simulate its fit to the system of the other business processes. Now the users are the business planners, the implementers the simulation experts.

The goal of (1) is to gain an understanding of the functionality of the business process, both from specialist and IT side. Accordingly, the goal of (2) is to assess the simulation approach from a technical point of view (user) and to estimate its feasibility (implementer). The model should enable users and implementers to communicate about whether their perspectives are compatible.

2 Modeling language

A set of questions arises with selecting the adequate modeling language. Does the modeling language fit the modeling project? Are the terms of the modeling language understandable for users and implementers? Is the metaphor used for the modeling language intuitive for both perspectives? Is the scope of the language appropriate for the modeling project?

If, for example, the structuring of an application landscape into interacting objects is to be modeled from the user's perspective, a data modeling language such as ERM (Entity-Relationship Model (Chen 1976)) would be unsuitable. ERM lacks the ability to model the object behavior. Otherwise, if it is considered to use UML class diagrams (Unified Modeling Language (OMG 2017)) instead, the behavior of the system could be modeled using class methods. However, one would probably have to limit the scope of the language, so that it remains understandable for the users and avoids high learning effort.

The situation is similar for business process modeling with EPC (Event-Driven Process Chains (Nüttgens 2013)) and workflow modeling with BPMN (Business Process Model and Notation (OMG 2014)). Both belong to the class of process modeling languages. While the EPC is tailored to the business level, BPMN supports systems-oriented features like message and sequence flows, different gateways, and the usage of pools.

Another question is important too: Does the modeling language use one view[2] or several views

[2] The terms perspective and view are used in analogy to descriptive geometry. The perspective is defined by the location from which one looks at the system. Different locations allow different perspectives. In contrast, a view focuses on certain system properties. Looking from a given perspective, the union of the associated views reflects the whole system.

(multi-view model)? Multi-view modeling languages allow to focus on certain system properties at a given time. However, it is sometimes strenuous to combine the views to a whole.

For example, SOM (Semantic Object Model (Ferstl and Sinz 1995)) allows to model business processes from a structural and a behavioral view. The two views are derived from a common meta model. They are inherently integrated with each other. Changes in one view are likely to have an immediate effect on the other. On the other side, UML also allows several views on a system through its different diagram types, but the integration is not as strong. In UML, the emphasis is on notation, not on the modeling method.

3 Modeling tool

The general questions of introduction of tools in an organization will not be considered here. These are for example: Will the tool be further developed in the longer term? How will the tool's dissemination develop? Does the tool support several languages?

Instead, the following questions are in the foreground: Is the tool only a drawing tool or does it support exactly the modeling language according to the meta model? What about the syntax of the modeling language? Are syntactic constraints temporarily vulnerable to facilitate modeling?

With multi-view modeling tools (Ferstl et al. 2016), the question arises how the individual views are linked. Is this done using an integrated meta model? Do the effects of changes in one view immediately affect the other views?

4 People

Among the questions of project management or project organization, the following are of particular importance for conceptual modeling projects: Are users and implementers involved sufficiently in the modeling project or is one of the groups underrepresented? Do both groups have basic knowledge of conceptual modeling? Do both groups have sufficient command of the modeling language to communicate using the model? Is there a common understanding of the underlying metaphor? Do they have the skills to use or operate the tool?

5 Benefit of the model

Finally, it must be clarified whether the expected benefit of the model will sufficiently fulfil the goals of the users and the implementers. This goes back to the user scenario. Does the model help to understand and communicate the underlying object and its context for users and implementers? And finally, can the model be further processed by machine as desired?

6 Final remarks

This short article addresses some sets of questions that should be clarified in a modeling project before starting to model conceptually. The questions reflect the experiences of the author. Although the clarification of the questions is not sufficient for the success of a modeling project, it is necessary.

References

Chen P. P.-S. (1976) The entity-relationship model—toward a unified view of data. In: ACM transactions on database systems (TODS) 1(1), pp. 9–36

Ferstl O. K., Sinz E. J. (1995) Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. In: Wirtschaftsinformatik 37(3), pp. 209–220

Ferstl O. K., Sinz E. J. (2013) Grundlagen der Wirtschaftsinformatik. Walter de Gruyter GmbH & Co KG

Ferstl O. K., Sinz E. J., Bork D. (2016) Tool support for the semantic object model. In: Domain-specific conceptual modeling. Springer, pp. 291–310

Jordan D., Evdemon J., Alves A., Arkin A., Askary S., Barreto C., Bloch B., Curbera F., Ford M., Goland Y. et al. (2007) Web services business process execution language version 2.0. In: OASIS standard 11(120), p. 5

Nüttgens M. (2013) EPK. https://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/Geschaftsprozessmodellierung/EPK/index.html/?searchterm=epk (Accessed: 2020-11-10)

OMG (2014) Business Process Model and Notation (BPMN). Version 2.0.2. http://www.omg.org/spec/BPMN/2.0.2 (Accessed: 2020-11-14)

OMG (2017) Unified Modeling Language. Version 2.5.1. https://www.omg.org/spec/UML/2.5.1/PDF (Accessed: 2020-11-10)

Schlageter G., Stucky W. (2013) Datenbanksysteme: Konzepte und Modelle Vol. 37. Springer

Strahringer S. (2013) Konzeptuelle Modellierung von IS https://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/konzeptuelle-modellierung-von-is/index.html?searchterm=konzeptuelle (Accessed: 2020-11-10)

Thalheim B. (2010) Towards a theory of conceptual modelling. In: Journal of Universal Computer Science 16(20), pp. 3102–3137