



7th Educators' Symposium @ MODELS 2011:  
Software Modeling in Education  
(EduSymp2011)

Teaching Modeling—An Initial Classification of Related Issues

Ludwik Kuzniarz, Jürgen Börstler

10 pages

# Teaching Modeling—An Initial Classification of Related Issues

Ludwik Kuzniarz, Jürgen Börstler

{ludwik.kuzniarz,jurgen.borstler}@bth.se

School of Computing

Blekinge Institute of Technology, Karlskrona, Sweden

**Abstract:** Modeling is an important skill needed in both science and engineering domains. In software engineering, in particular, models are ubiquitous artefacts. The development, manipulation and understanding of models is therefore an important learning objective. The paper describes motivations and an outline of a research project aimed at classifying issues related to the teaching and learning of modeling together with some initial research results.

**Keywords:** Modeling, Teaching, Classification

## 1 Introduction

Constructing and using models is an essential element of any constructive human activity. Models are used for understanding the world around us, as well as for creating new things.

Modeling is therefore an important skill needed in all science and engineering domains. Solving complex problems requires knowledge and skills in modeling, to be able to focus on the properties that are most relevant in a specific context without being distracted by details that are irrelevant at a certain level.

In software engineering, in particular, models are ubiquitous artefacts. The development, manipulation and understanding of models is therefore an important learning objective. However, graduating students have problems with designing even small software systems [EMM<sup>+</sup>06, LTZ11] and “underestimate the importance of representing structural groupings and interactions between design parts” [TFB<sup>+</sup>05].

Education related to software engineering, and teaching modeling in particular, is getting increasing attention and understanding in the academic community. The related problems are highlighted and discussed on fora such as software engineering education related symposia and workshops.

In order to better understand the reasons for these problems, as well as to organize future work on improvement of the current situation, we want to provide a classification of issues, problems, and challenges related to the teaching and learning of modeling.

This classification can then be used by the stakeholders involved in education, in two basic ways:

- as an aid for improving our understanding of the problems related to the teaching (and learning) of modeling, and
- as a means for identifying and organizing the current state-of-the-art of the research in this area.

The paper presents the overall idea behind the classification and reports on the first phase of developing such a classification – the elicitation process and its results, and briefly outlines the plans for further research towards a mature and approved classification framework.

## 2 Related Work

There is a large body of work related to a number of aspects relevant for the teaching and learning of modeling. Virtually no work, though, is concerned with categorizations or taxonomies of teaching issues. Various issues regarding the teaching of modeling were discussed and summarized in earlier MoDELS panels (see for example [BFG<sup>+</sup>10]), but the results were not categorized in a systematic way.

In the present paper, we are focusing on modeling in the context of software engineering. We will not go into any details about the literature on modeling in the context of science or engineering teaching in general. However, since the work done in this area is substantial and relevant (e.g., [Gil04, KS11]) we will include it at a later stage.

Diethelm et al. and Schulte and Niere show that UML-like modeling can be successfully introduced in secondary school and helps students grasp basic object-oriented concepts [DGZ05, SN02]. They especially observe that the models the students created helped them to discuss each other's modeling problems. It has also been shown that explicit modeling of scenarios can improve scenario role-play and facilitate “object thinking” [Bör05, BS05].

A general problem of teaching models and modeling is the lack of a commonly accepted definition of model quality. Nelson et al. propose a conceptual modeling quality framework where they integrate product and process views proposed earlier [NPGP11]. This framework consists of four layers representing different aspects of models and modeling (physical, knowledge, learning, and development) and 24 types of quality.

Such a quality framework could be used to categorize model variations. Hadar and Soffer argue that model variations are a sign of vagueness and understanding these variations better will support decision making while modeling [HS06].

Empirical research shows that there are various aspects of models that contribute to their understanding, like decomposition quality [BJ08], diagram layout [SW05], naming [BLMM09], and actual modeling language used [GPS05]. It is therefore important to be aware that there is no “one-size-fits-all” solution to improve model comprehensibility.

Kuzniarz and Staron propose a number of best practices for teaching modeling/UML and particularly emphasize the role of consistency between models and iterative teaching [KS06]. The role of model consistency is also highlighted in StudentUML, a teaching tool for UML diagrams that ensures consistency between class and sequence diagrams [RD07]. Brandsteidl et al. discuss a large scale modeling course with specific emphasis on assessment and propose different types of exercises for assessing theoretical and practical knowledge and skill in UML modeling [BSK09].

A good source for teaching or learning issues are typical problems or errors appearing in student's models or designs. Thomasson et al. found that many students failed to integrate some classes into their solutions, i.e. had problems of connecting the elements of a model to a meaningful “big picture” [TRT06]. Another common problem was attribute location or representation,

leading to classes with low cohesion. In another empirical study, Siau and Loo showed that the problems in learning UML mainly can be grouped into two main categories: (1) problems inherent in UML, like its complexity and semantic issues, and (2) “peripheral” issues, like the learners’ lacking prerequisites or problems with teaching materials [SL06].

There are also numerous modeling/UML tools, but very few of them are suited for educational purposes and even fewer have been specifically designed for teaching.<sup>1</sup> The tools that are typically used for teaching rarely support easy mechanisms for consistency checking and most of them have a very steep learning curve. Modern learning environments support features for collaboratively discussing modeling exercises and solution [BBB<sup>+</sup>10]. Besides reaching more learners, these tools help students understand that there is no single correct solution to a modeling problem.

### 3 Initial Classification

A general observation regarding challenges and issues related to the teaching of modeling, is that some of them share a common denominator, a specific aspect of or a point of view on the teaching of modeling. We therefore propose a two level classification structure with the points of view, or perspectives on the top level, and the specific issues on the bottom level (following the vocabulary used for a similar purpose in [KA11]).

We observed that the perspectives could be characterized by question words and the specific issues could be elicited by formulating questions around those words in the following way:

**Why to teach** – concerning the reasons for introducing the modeling into the teaching curricula,

**What to teach** – concerning the contents of the teaching,

**How to teach** – concerning the ways and means used for teaching.

The *why*, *what* and *how* perspectives were the starting point for an initial elicitation of issues and challenges based on (a) a structured panel discussion on the 8th Nordic Workshop on Model Driven Software Engineering and (b) interviews with some domain experts. The panelists and the approached experts were members of academia, who were involved in teaching different subjects related to software engineering that included elements of modeling. Both groups were asked the same questions, aimed at finding specific issues in the identified perspectives. The actual questions were the following:

1. Why do you think we should teach modeling?
2. What should be taught?
3. How we should teach modeling?

The questions represented the identified perspectives. Based on the answers gathered from the approached experts, specific issues in the perspectives were derived, named and described.

---

<sup>1</sup> Please see <http://www.jeckle.de/umltools.htm>, <http://www.oose.de/service/uml-werkzeuge.html> or <http://www.freebase.com/view/base/umltools> for the most comprehensive lists of such tools.

### 3.1 Basic perspectives

Our initial investigation among targeted teaching experts revealed that in addition to our three initially suggested perspectives, two additional perspectives needed to be considered:

**Where to teach** – concerning the place for education, and

**When to teach** – concerning the time and place of education

These perspectives explicitly address context and issues broader than just academic education.

The outline of the resulting initial classification is shown in Table 1. The perspectives and specific issues in the perspectives are listed and shortly described below. They express specific reasons and motivations behind introducing specific teaching modeling elements into the curriculum.

**Why perspective.** This perspective concerns motivations and arguments for teaching modeling in general as well as for introducing general courses on modeling as such or including elements of modeling into specific courses (see Table 2).

**What perspective.** This perspective addresses the overall scope and the specific contents of a modeling curriculum. In particular specific knowledge and abilities related to proper and effective application of modeling in solving (software) development problems (see Table 3).

Table 1: Initial classification scheme.

PERSPECTIVE	WHY	WHAT	HOW
	<i>Why should modeling be included in a curriculum?</i>	<i>What should be included in modeling curriculum?</i>	<i>How should modeling be taught?</i>
ISSUES	<ul style="list-style-type: none"> <li>● Way of thinking</li> <li>● Problem solving</li> <li>● Successful research</li> <li>● Successful development</li> <li>● Being up-to-date</li> <li>● Being competitive</li> </ul>	<ul style="list-style-type: none"> <li>● Creating models</li> <li>● Using models</li> <li>● Informal models</li> <li>● Formal models</li> <li>● Integration of models</li> <li>● Model transformation</li> <li>● Code generation</li> <li>● Languages</li> <li>● Tools</li> <li>● Best practices</li> <li>● Basic principle</li> <li>● Consistency</li> </ul>	<ul style="list-style-type: none"> <li>● Examples</li> <li>● Exercises</li> <li>● Projects</li> <li>● Industrial practices</li> <li>● Industry lectures</li> <li>● Communication</li> <li>● Presentations</li> <li>● Discussions</li> <li>● Teaching methods</li> </ul>

Table 2: Description of issues in the Why-perspective.

WHY PERSPECTIVE	
Way of thinking	Promoting and inspiring thinking at high abstraction levels, that is to encourage formulating tasks and defining problems considering only relevant details and using appropriate abstractions.
Successful research	Establishing sound background and tools that enable performing successful research (as in the research activity it is essential to precisely formulate the problem using adequate vocabulary and way to describe the problem abstracting from details).
Successful development	Enabling and ensuring successful development of software.
Being up-to-date	Preparing students to be up to date with recent trends and developments in the field of software engineering.
Being competitive	Being competitive in the labor market by being acquainted with a new, or at least additional, emerging software development technology.

**How perspective.** In this perspective various elements related to the actual teaching of modeling are grouped. They consider the teaching process and teaching activities as well as methods and tools used in the teaching process (see Table 4).

### 3.2 Complementary perspectives

The initial investigation among targeted teaching experts revealed that in addition to our three initially suggested perspectives, an additional perspective addressing the the context of the education was identified, and we considered for further investigation and including it in the classification. As a result we got the following structure:

**Context perspective.** This perspective addresses the environment in which modeling is taught. It considers the placement of teaching and learning of modeling from the point of view of related elements “around” (other skills and courses related to software development) in time and space (see Table 5).

Our initial results show that the type of answers to “When” and “Where” seem limited to the following:

- When to teach
  - *Before other software development courses.* This suggest that teaching modeling should be done as an independent course prior to other courses and be a sort of a prerequisite for all other courses.
  - *Inside software development courses.* Modeling should be a part of other software development courses rather than being a separate course.

Table 3: Description of issues in the What-perspective.

WHAT PERSPECTIVE	
Creating models	Skills needed for the creation of models, such as choosing proper perspective and level of abstraction.
Using models	Awareness of where and how models can be used and ability to use them in a proper way.
Informal models	Using sketchy drawings in order to convey basic characteristics of a problem or ideas behind a solution.
Formal models	Developing models that conform to rigorous formal rules and being aware of the formal rules.
Integration of models	Mixing and merging different models prepared by different developers and during different phases within a project.
Model transformation	Generating and building new models from existing models in a prescribed, systematic and automatic way.
Code generation	Generating executable software from models.
Languages	Modeling languages and their proper usage considering the purpose and the development environment.
Tools	Modeling tools and development environments that support modeling.
Best practices	Justified and proved approaches that are recommended to be used in when performing modeling in software development.
Basic principles	Essential, commonly accepted principles and rules to be used and obey in order to do the modeling right.
Consistency	Awareness of specific relationships between models produced and used within development process, and ensuring that those relationships are preserved.

- *When joining a professional team.* Special “equalizing” courses on using modeling in software development for new members joining development teams.
- *At the first possibility.* A software developer should be introduced to modeling as soon as possible.
- Where to teach
  - *At university.* Modeling is thought and included in academic curricula.
  - *In company.* A company organizes specific courses to ensure that new employees are or become familiar with the necessary knowledge and skills related to modeling.
  - *Individual study.* Individuals take responsibly to get acquaintance with modeling to increase their competitiveness.

The above structure constituting the context perspective is rather closed, that is, is limited to two issues with a limited, rather fixed, number of possible instances of each issue. It can be

Table 4: Description of issues in the How-perspective.

HOW PERSPECTIVE	
Examples	Using suitable and effective examples (for the purpose) by the teachers during the teaching process.
Exercises	Encouraging active learning of specific topics by (small) problem solving.
Projects	Carrying out projects of different size in order to practice how modeling is used in a wide context.
Industrial practices	Organizing activities that are reflecting practices used in industry.
Industry lectures	Guest lectures, seminars or presentations given by presenters that are actively applying modeling in the development in the industry.
Communication	Using models as media for communication with students.
Presentations	Encouraging students to use models when presenting their work.
Discussions	Encouraging and stimulating discussions between students.
Specific methods	Using specific, dedicated teaching methods and processes that will encourage and ensure better comprehension and stimulate learning.

Table 5: Description of issues in the Context-perspective.

CONTEXT PERSPECTIVE	
When to teach	Concerns the relation in time between teaching modeling as compared to other skills related to software development. Considers the relation to other courses as well as the point of time.
Where to teach	Considers the location (place in space) or teaching environment where the education takes place.

added to the classification to make it complete, but further usage would be rather limited. And also perhaps not of specific interest when evaluating, positioning and planning of future research, so we decided not to include that perspective in the further research.

## 4 Summary and Future Work

We outlined a research aimed at providing a classification of the issues and challenges related to teaching and learning modeling in the context of software development and presented results from the first step of our research aimed at elicitation of an initial classification.

The reported research was executed as a pilot investigation aimed at elaborating the overall structure of the classification and also preparing and testing means for more elaborative elicitation of the required elements of the classification.

The proposed classification is structured into two levels: perspectives on the high level identifying an aspect, and issues on the low level, identifying specific problems within a given per-

spective.

After analysis of the data from the initial survey, we also found out that several details of the classification, concerning both perspectives and issues, should be further investigated and improved. We plan to do this using available discussion fora related to teaching modeling, and also approaching a bigger number of experts using Internet.

Our final aim is to develop an empirically justified classification scheme based on a survey targeting experts involved in teaching modeling. The updated classification will then be extended with empirical data on relative prioritization using a similar approach as in the classification of consistency problems in model driven software development reported in [KA11].

**Acknowledgements:** We want to thank the participants of the pilot investigation for the classification, especially the panelists on the special session of the 8th Nordic Workshop on Model Driven Engineering

## Bibliography

- [BBB<sup>+</sup>10] J. Börstler, O. F. Bay, M. Baturay, S. Trapp, M. Heintz, S. Weber. embed4Auto: a PLE for software modelling. In *Proceedings of the 15th annual conference on Innovation and technology in computer science education*. P. 322. 2010.
- [BFG<sup>+</sup>10] J. Bezivin, R. France, M. Gogolla, O. Haugen, G. Taentzer, D. Varro. Teaching Modeling: Why, When, What? In *MODELS 2009 Workshops*. Pp. 55–62. Springer, 2010.
- [BLMM09] D. Binkley, D. Lawrie, S. Maex, C. Morrell. Identifier length and limited programmer memory. *Science of Computer Programming* 74(7):430–445, 2009.
- [Bör05] J. Börstler. Improving CRC-card role-play with role-play diagrams. In *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA)*. Pp. 356–364. 2005.
- [BS05] J. Börstler, C. Schulte. Teaching Object Oriented Modelling with CRC Cards and Roleplaying Games. 2005.
- [BSK09] M. Brandsteidl, M. Seidl, G. Kappel. Teaching Models @ BIG: On Efficiently Assessing Modeling Concepts. In *Proceedings of the MoDELS 2009 Educators' Symposium*. 2009.
- [BJ08] A. Burton-Jones. The Effects of Decomposition Quality and Multiple Forms of Information on Novices' Understanding of a Domain from a Conceptual Model. *Journal of the Association for Information Systems* 9(12):748–802, 2008.
- [DGZ05] I. Diethelm, L. Geiger, A. Zündorf. Teaching Modeling with Objects First. In *Proceedings of the 8th World Conference on Computers in Education*. 2005.

- [EMM<sup>+</sup>06] A. Eckerdal, R. McCartney, J. E. Moström, M. Ratcliffe, C. Zander. Can graduating students design software systems? In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*. Pp. 403–407. 2006.
- [Gil04] J. Gilbert. Models and Modelling: Routes to More Authentic Science Education. *International Journal of Science and Mathematics Education* 2(2):115–130, 2004.
- [GPS05] G. Guizzardi, F. Pires, M. van Sinderen. An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages. In *Proceedings MoDELS 2005*. Lecture Notes in Computer Science 3713, pp. 691–705. Springer, 2005.
- [HS06] I. Hadar, P. Soffer. Variations in Conceptual Modeling: Classification and Ontological Analysis. *Journal of the Association for Information Systems* 8:568–592, 2006.
- [KA11] L. Kuzniarz, L. Angelis. Empirical extension of a classification framework for addressing consistency in model based development. *Information and Software Technology* 53(3):214–229, 2011.
- [KS06] L. Kuzniarz, M. Staron. Best Practices for Teaching UML-based Software Development. In *Satellite Events at the MoDELS 2005 Conference*. Pp. 320–332. 2006.
- [KS11] M. Khine, I. Saleh (eds.). *Models and Modeling: Cognitive Tools for Scientific Enquiry*. Models and Modeling in Science Education 6. Springer Verlag, 1 edition, 2011.
- [LTZ11] C. Loftus, L. Thomas, C. Zander. Can graduating students design: revisited. In *Proceedings of the 42nd ACM technical symposium on Computer science education*. Pp. 105–110. 2011.
- [NPGP11] H. J. Nelson, G. Poels, M. Genero, M. Piattini. A Conceptual Modeling Quality Framework. *Software Quality Journal*, pp. 1–28–28, Apr. 2011.
- [RD07] E. Ramollari, D. Dranidis. StudentUML: An Educational Tool Supporting Object-Oriented Analysis and Design. In *Proceedings of the 11th Panhellenic Conf. on Informatics (PCI 2007)*. 2007.
- [SL06] K. Siau, P. Loo. Identifying Difficulties in Learning UML. *Information Systems Management* 23(3):43–51, 2006.
- [SN02] C. Schulte, J. Niere. Thinking in Object Structures: Teaching Modelling in Secondary Schools. In *ECOOP Workshop on Pedagogies and Tools for the Learning of Object-Oriented Concepts*. 2002.
- [SW05] D. Sun, K. Wong. On Evaluating the Layout of UML Class Diagrams for Program Comprehension. In *Proceedings of the 13th International Workshop on Program Comprehension*. Pp. 317–326. 2005.

- [TFB<sup>+</sup>05] J. Tenenberg, S. Fincher, K. Blaha, D. Bouvier, T. Chen, D. Chinn, S. Cooper, A. Eckerdal, H. Johnson, R. McCartney et al. Students Designing Software: A Multi-national, Multi-institutional Study. *Informatics in Education* 4(1):143–162, 2005.
- [TRT06] B. Thomasson, M. Ratcliffe, L. Thomas. Identifying novice difficulties in object oriented design. In *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*. Pp. 28–32. 2006.