Empirical Studies into UML in Practice: Pitfalls and Prospects

Michel R.V. Chaudron
Dept. of Computer Science and Engineering
Chalmers | Gothenburg University, Gothenburg, Sweden
chaudron@chalmers.se

***Abstract*— A key open question in the area of software modeling is which costs and benefits it brings to software development and maintenance. For answering this question, better empirical studies into software modeling are needed. In this paper I focus on what I believe are the pitfalls in- and prospects for such types of studies. This paper is an abstract for an invited keynote at the Modeling in Software Engineering (MiSE) workshop at ICSE 2017.**

Keywords: Software Modeling, Software Design, UML, Model-based software development, Empirical Software Engineering.

I. INTRODUCTION

The UML has emerged in the mid 1990's as a standardized notation for modeling software systems. The UML notation caters for describing systems at various levels of abstraction and was meant as a common notation that could be used by a multitude of software modeling methodologies. The fact that modeling is currently done in very many ways has led to a confusion of tongues in academic studies. I argue for the practice of including a definition of scope of 'modeling' in research papers: In this talk we take an broad view on 'modeling' and take its scope to range from sketching box-and-lines with UML-like meaning on a whiteboard, to using UML or its various specializations and extensions as a basis for automated code-generation.

Little evidence exists regarding if and how UML modelling benefits software development. On the one hand there seems to be an increased uptake of model-based software development practices by various companies: some run their businesses solely on the use of model-based software engineering technologies, such as Mendix[1]. Also, at recent MODELS conference, large multinationals like Thales and Ericsson presented a strategy for creating an ecosystem of model-based development tools in open source. Academic studies into industrial use of UML modeling comprise mostly surveys and case studies. Interestingly no consistent picture emerges from these studies. In order to advance the scientific understanding of UML modeling in industrial software development, I propose that we change our focus from 'Is modelling beneficial to software development?' to 'When is software modeling beneficial to software development?' As contribution to answering this question, I will present a selection from the last decade of empirical research in the area of software modeling using UML and point out pitfalls and prospects in this field.

I will structure my talk around the following questions:

(1) What do surveys tell us about the state of UML modeling in practice - both in industry and in open source projects?
(2) What are the difficulties in empirical studies UML modeling in practice, and can we avoid them?
(3) How should we set up empirical studies that help understand in which context UML modelling actually benefits software development?

I provide a brief preview to the discussion of each of these questions:

(ad 1): The surveys performed over the last 15 years give a fairly consistent picture regarding i) which UML diagrams are commonly used, and ii) for what purposes UML modeling is used. One of the main purposes of modeling that is often mentioned is that of communication between members of a development team. For this purpose, models are often made at a high level of abstraction, hence can be sketchy and do not use many of the syntactical elements of the UML. Some surveys also inquire into the scale of adoption and benefits of UML modeling. Responses are based on 'expert opinions' about which we do not know how objective and representative they are. The diversity of findings from these surveys seems to indicate that a subjective bias of the respondents shines through. There is a need for more objective - ideally measurement-based - evidence for costs and benefits of UML modeling.

(ad 2): There are various experiments that study the use of UML in performing software development tasks in controlled environments in university labs. From these we can learn that there are some benefits from particular layout styles and use of modeling constructs at the level of basic tasks. Unfortunately, studying UML modeling in real-life development projects is much more challenging. To start with, we do not know which (dependent) variables to measure for assessing the impact of UML modeling on

[1] www.mendix.com

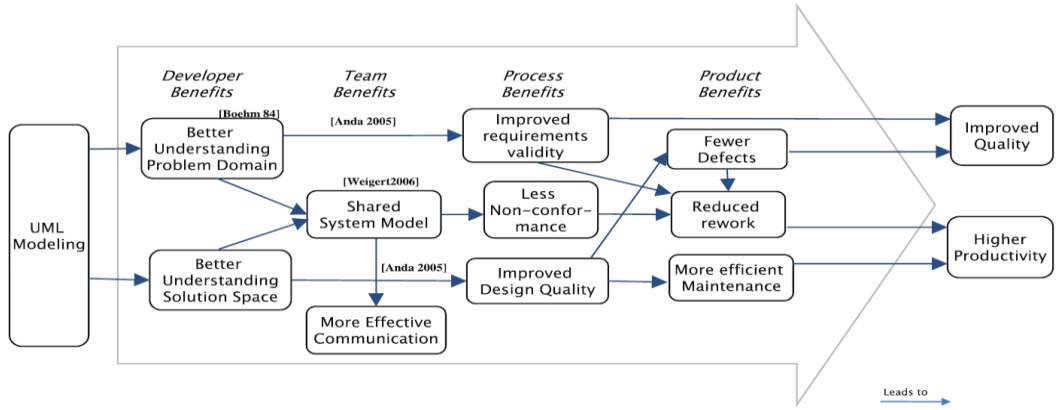

Figure 1: Proposal for theory about the benefits of UML modeling (from [1])

software projects. Examples of candidate variables for this should be derived from some theory, such as show in Fig 1 (from [1]): performance of the project (e.g. in terms of budget and meeting deadlines), quality of the design, quality of the implementation. One complication that hampers studying these types of project-wide measures is the fact that a (UML) design is often an intermediate product on the way to a running system: UML modeling may be used to achieve the final result, but it is neither necessary nor sufficient. Moreover, there are many factors in projects as a whole that i) reduce or even counter benefits that could be reaped, and ii) introduce 'noise' in the measurements of our performance variables. Hence, we need to carefully think how we scope industrial studies into UML. We can reasonably assume that benefits depend on a collection of modeling-related practices that are used in concert. Here we can think of: tool-chains, support in version- and change management, quality assurance for modeling, etc. For future studies, action research may be a suitable research strategy because it allows observing the changes that occur after introducing (or removing) particular modeling practices.

A related difficulty in this field is the synthesis of evidence. This difficulty may be due to the large diversity of the findings presented in the existing literature. One of the ways in which primary studies into the use of UML should improve is in reporting their context [3]. This will help in clarifying the confusion that arises from the many ways in which researchers refer to 'UML modeling' and help in identifying context in which UML modeling does and does not work. An attempt at synthesizing evidence into a theory about benefits of UML modeling is shown in Fig. 1 [1].

(ad 3): Prospects in research in the use of UML modeling:
One research direction is to study modeling more 'in-vivo', i.e. in real-life projects. This will enable researchers to capture types of data about modeling that have so far been neglected, such as e.g. i) costs/effort spent on modelling (by who and when in the project)?, and ii) the communication that happens in a project and the role that UML models play in this. Given the difficulties in accessing industrial projects, another research direction is to study modeling in open source projects. As a resource for such studies we published a list of 12000 projects that include UML models (see [2], [4], http://oss.models-db.com/). Finally we propose to explore new types of modeling tools [5] to better support designing activities and to better integrate models in the entire development lifecycle.

II. Speaker Bio

Michel Chaudron is Full Professor at the Software Engineering division at the joint Department of Computer Science of Chalmers and Gothenburg University in Sweden. Prior to this, he worked at Universities in Leiden and Eindhoven in the Netherlands. He obtained his Ph.D. in the area of formal methods and programming calculi for parallel computing. His research interests are in: software-architecture, design, (UML) modeling, and software composition. He has an interest in empirical software engineering in industrial contexts.

References

[1] Chaudron, Michel R. V., Werner Heijstek, and Ariadi Nugroho. "How effective is UML modeling?: An empirical perspective on costs and benefits (Expert's Voice)." Software and systems modeling 11, no. 4 (2012): 571-580, doi:10.1007/s10270-012-0278-4

[2] Hebig, Regina, Truong Ho Quang, Michel R.V. Chaudron, Gregorio Robles, and Miguel Angel Fernandez. "The quest for open source projects that use UML: mining GitHub." In Proc of the ACM/IEEE 19th Int. Conf on Model Driven Engineering Languages and Systems, pp. 173-183. ACM, 2016, Doi: 10.1145/2976767.2976778

[3] Dyba, Tore. "Contextualizing empirical evidence." IEEE software 30, no. 1 (2013): 81-83.

[4] Truong Ho-Quang, Regina Hebig, Gregorio Robles, Michel R. V. Chaudron and Miguel S. Fernandez, Practices and Perceptions of UML Use in Open Source Projects, ICSE 2017 SEIP-track.

[5] Chaudron, Michel R.V., and Rodi Jolak. "A Vision on a New Generation of Software Design Environments." Human Factors in Modeling workshop, ACM MODELS conference, 2015.