An Experimental Investigation of UML Modeling Conventions

Christian F.J. Lange[1], Bart DuBois[2],
Michel R.V. Chaudron[1], and Serge Demeyer[2]

[1] Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
C.F.J.Lange@tue.nl, M.R.V.Chaudron@tue.nl
[2] Lab On REengineering (LORE), University of Antwerp, Belgium
Bart.Dubois@ua.ac.be, Serge.Demeyer@ua.ac.be

Abstract. Modelers tend to exploit the various degrees of freedom provided by the UML. The lack of uniformity and the large amount of defects contained in UML models result in miscommunication between different readers. To prevent these problems we propose modeling conventions, analogue to coding conventions for programming. This work reports on a controlled experiment to explore the effect of modeling conventions on defect density and modeling effort. 106 masters' students participated over a six-weeks period. Our results indicate that decreased defect density is attainable at the cost of increased effort when using modeling conventions, and moreover, that this trade-off is increased if tool-support is provided. Additionally we report observations on the subjects' adherence to and attitude towards modeling conventions. Our observations indicate that efficient integration of convention support in the modeling process, e.g. through training and seamless tool integration, forms a promising direction towards preventing defects.

1 Introduction

The Unified Modeling Language (UML [19]) is used in different phases during software development such as requirements analysis, architecture, detailed design and maintenance. In these phases it serves various purposes such as communication between project stakeholders, prediction of quality properties and test case generation. The UML is designed as a visual multi-purpose language to serve all these needs. It allows to choose from 13 diagram types, it offers powerful extension mechanisms, but it lacks a formal semantics. Due to these characteristics the user has the freedom to choose the language features that fit his purpose of modeling. However, the UML does not provide guidelines on how to use the language features for a specific purpose. For example, there is no guidance that describes when it is useful to use multiplicities or when a class' behavior should be described by a state diagram. As a result, the UML user is confronted with a large degree of freedom.

The UML possesses the risk for quality problems due to its multi-diagram nature, its lack of a formal semantics and the large degree of freedom in using

O. Nierstrasz et al. (Eds.): MoDELS 2006, LNCS 4199, pp. 27–41, 2006.

it. The large degree of freedom and the lack of guidelines results in the fact that the UML is used in several different ways leading to differences in rigor, level of detail, style of modeling and amount of defects. Industrial case studies [16] and surveys give empirical evidence that individuals use the UML in many different ways (even within the same project team) and that the number of defects is large in practice. Moreover, experiments have shown that defects in UML models are often not detected and cause misinterpretations by the reader [15].

The effort for quality assurance is typically distinguished between *prevention* effort and *appraisal* effort [22]. Prevention effort aims at preventing for deviations from quality norms and appraisal effort is associated with evaluating an artifact to identify and correct deviations from these quality norms. There are techniques in software development to detect and correct the deviations from quality norms. Reviews, inspections and automated detection techniques are used in practice to detect weak spots. They are associated with appraisal effort. In programming preventive techniques to assure a uniform style and comprehensibility of the source code are established as coding conventions or coding standards [20]. As an analogy for UML modeling we propose *modeling conventions* to prevent modelers to deviate from quality norms. We define modeling conventions as: ***Conventions to ensure a uniform manner of modeling and to prevent for defects***.

The main purpose of this paper is to explore experimentally the effectiveness of modeling conventions for UML models with respect to prevention of defects.

An additional purpose of this study is to explore subjects' attitude towards modeling conventions and how modeling conventions are used. The observations can be used to improve the future use of modeling conventions.

This paper is structured as follows: Section 2 describes modeling conventions and related work. Section 3 describes the design of the experiment. Section 4 presents and discusses the results. Section 5 discusses the threats to the validity of the experiment and Section 6 discusses conclusions and future work.

2 Modeling Conventions

2.1 Related Work

There is a large variety of coding conventions (also known as guidelines, rules, standards, style) for almost all programming languages. The amount of research addressing coding conventions is rather limited though. Omam and Cook [20] present a taxonomy for coding conventions which is based on an extensive review of existing coding conventions. They identify four main categories of coding conventions: general programming practice, typographic style, control structure style and information style. They found that there are several conflicting coding conventions and that there is only little work on theoretical or empirical validation of coding conventions.

Our review of literature related to modeling conventions for the UML revealed the following categories: design conventions, syntax conventions, diagram conventions and application-domain specific conventions.

Design conventions address the design of the software system in general, i.e. they are not specific for UML. Design conventions such as those by Coad and Yourdon[6] aim at the maintainability of OO-systems. The conventions that include for example high cohesion and low coupling are empirically validated by Briand et al. [5]. The results of their experiment show that these conventions have a beneficial effect on the maintainability of object-oriented systems.

Syntax conventions deal with the correct use of the language. Ambler [3] presents a collection of 308 conventions for the style of UML. His conventions aim at understandability and consistency and address syntactical issues, naming issues, layout issues and the simplicity of design. Object-oriented reading techniques (OORT) are used in inspections to detect defects in software artefacts. OORT's for UML are related to modeling conventions in the sense that the rules they prescribe for UML models can be used in a forward-oriented way during the development of UML models to prevent for defects. Conradi et al. [7] conducted an industrial experiment where OORT's were applied for defect detection (i.e. an appraisal effort). The results show defect detection rates between 68% and 98% in UML models.

Diagram conventions deal with issues related to the visual representation of UML models in diagrams. Purchase et al. [21] present diagram conventions for the layout of UML class diagrams and collaboration diagrams based on experiments. Eichelberger [9] proposes 14 layout conventions for class diagrams aiming at algorithms for automatic layout of class diagrams.

Application-domain specific conventions. A purpose of UML profiles is to support modeling in a particular application domain. Hence, profiles are in fact application-domain specific conventions. Kuzniarz et al. [12] conducted an experiment on the effect of using stereotypes to improve the understandability of UML models. Their results show that stereotypes improve the correctness of understanding UML class diagrams by 25%.

2.2 Model Quality

In this experiment we investigate the effectiveness of modeling conventions on model quality, in particular we are interested in:

- Syntactic quality: The degree to which the model contains flaws.

Here we define *flaws* as: lack of coverage of the model's structural parts by behavioral parts, presence of defects, non-conformamce to commonly accepted design rules, and absence of uniformity in modeling.

Syntactic quality is one of the three notions of model quality according to Lindland's framework for conceptual models [17]. The two other notions according to Lindland are:

- Semantic quality: The degree to which the model correctly represents the problem domain.
- Pragmatic quality: The degree to which the model is correctly understood by its audience.

Evaluation of semantic and pragmatic quality involves participation of several people, and, hence, is an experiment itself. This would be beyond the scope of this experiment. We will investigate the effect of modeling conventions on semantic and pragmatic quality in a follow-up experiment.

2.3 Modeling Conventions in This Experiment

Based on the literature review and the experience from our case studies, we selected a set of modeling conventions. To keep the set of modeling conventions manageable and comprehensible we decided that it should fit on one A4 page. This led to 23 modeling conventions after applying these selection criteria:

- Relevance. The modeling convention should be relevant to improve the quality of the UML model by preventing for frequent defects [16].
- Comprehensibility. The modeling convention should be easy to comprehend (e.g. it relates to well-known model elements).
- Measurability. The effect of the modeling convention should be measurable.
- Didactic value. Applying the modeling convention should improve the subjects' UML modeling skills.

Examples of modeling conventions used in this experiment are given in Table 1. The entire set of modeling conventions can be found in [13]. In this experiment we focus on assessing syntactic quality, but we deliberately don't limit the collection of modeling conventions to syntactic conventions only. As described by Omam and Cook [20] there can be interaction between several conventions. To obtain realistic results it is necessary to use a representative set of modeling conventions. Therefore we chose conventions of all categories presented in Section 2.1.

Table 1. Examples of Modeling Conventions used in this Experiment

ID	Name	Description
4	Homogenity of Accessor Usage	When you specify getters/setters/constructors for a class, specify them for all classes
9	Model Class Interaction	All classes that interact with other classes should be described in a sequence diagram
10	Use Case Instantiation	Each Use Case must be described by at least one Sequence Diagram
14	Specify Message Types	Each message must correspond to a method (operation)
15	No Abstract Leafs	Abstract classes should not be leafs (i.e. child classes should inherit from abstract classes)
19	Low Coupling	Your classes should have low coupling. (The number of relations between each class and other classes should be small)

3 Experiment Design

3.1 Purpose and Hypotheses

We formulate the goal of this experiment according to the Goal-Question-Metric paradigm by Basili et al. [4]:

Analyze modeling conventions for UML
for the purpose of investigating their effectiveness
with respect to model quality and effort
from the perspective of the researcher
in the context of masters students at the TU Eindhoven.

Modeling conventions require model developers to adhere to specific rules. Therefore we expect the quality of models to be better, i.e. there are fewer defects in a model that is created using modeling conventions. When additionally using a tool to check for adherence to the modeling conventions, we expect the model quality to be even better than without tool-support. In other words, we formulate in the null hypothesis that there is no difference between the treatments:

- $H1_0$: There is no difference between the syntactic quality of UML models that are created without modeling conventions, with modeling conventions and with tool-supported modeling conventions.

Adherence to modeling conventions requires special diligence. We expect that this leads to higher effort for modeling. When additionally using the tool, the expected effort is even higher. Therefore we formulate the second hypothesis of this experiment as follows:

- $H2_0$: There is no difference between the effort for modeling UML models that are created without modeling conventions, with modeling conventions and with tool-supported modeling conventions.

3.2 Design

The purpose of this experiment is to investigate the effect of modeling conventions. Therefore the treatment is to apply modeling conventions with and without tool-support during modeling. We define three treatment levels:

NoMC: no modeling conventions. The subjects use no modeling conventions. This is the *control group.*
MC: modeling conventions. The subjects use the modeling conventions that are described in Section 2.3.
MC+T: tool-supported modeling conventions. The subjects use the modeling conventions and the analysis tool to support adherence.

The experimental task was carried out in teams of three subjects. We have randomly assigned subjects to teams and teams to treatments. According to [10] this allows us to assume independence between the treatment groups. Each team performed the task for one treatment level. Hence we have an unrelated between-subjects design with twelve teams for each treatment level.

3.3 Objects and Task

The task of the subjects was to develop a UML model of the architecture of an information system for an insurance company. The required functionality of

the system is described in a document of four pages [13]. The system involves multiple user roles, administration and processing of several data types. The complexity of the required system was chosen such that on the one hand the subjects were challenged but on the other hand there was enough spare time for possible overhead effort due to the experimental treatment. The subjects used the Poseidon [2] UML tool to create the UML models. This tool does not assist in adhering to the modeling conventions and preventing model flaws.

The task of the teams with treatment MC and MC+T was to apply modeling conventions during development of the UML model. The modeling conventions description contains for each convention a unique identifier, a brief descriptive name, a textual description of the convention, and the name of the metric or rule in the analysis tool, that it relates to.

The subjects of treatment MC+T used the SDMetrics [24] UML analysis tool to assure their adherence to the modeling conventions. SDMetrics calculates metrics and performs rule-checking on UML models. We have customized [13] the set of metrics and rules to allow checking adherence to the modeling conventions used in this experiment.

3.4 Subjects

In total 106 MSc students participated in the experiment, which was conducted within the course "Software Architecting" in the fall term of 2005 at the Eindhoven University of Technology (TU/e). All subjects hold a bachelor degree or equivalent. Most students have some experience in using the UML and object oriented programming through university courses and industrial internships. We analyzed the results of the students' self-assessment from the post-test questionnaire and found no statistically significant differences.

The students were motivated to perform well in the task, because it was part of an assignment which was mandatory to pass the course (see Section 4.4).

The students were not familiar with the goal and the underlying research question of the experiment to avoid biased behavior.

3.5 Operation

Prior to the experiment we conducted a pilot run to evaluate and improve the comprehensibility of the experiment materials. The subjects of the pilot experiment did not participate in the actual experiment.

In addition to prior UML knowledge of the students we presented and explained UML during the course before the experiment. The assignment started with an instruction session to explain the task and the tooling to all students. Additionally the subjects were provided with the assignment material [13] including a detailed task description, the description of the insurance company system, and instructions of the tools. The modeling conventions and the SDMetrics tool were only provided to the teams which had to use them. The teams of treatment MC and MC+T were explicitly instructed to apply the treatment regularly and to contact the instructors in case of questions about the treatment. The experiment was executed over a period of six weeks.

3.6 Data Collection

We collected the defect data of the delivered UML models using the SDMetrics, because the majority of the applied modeling conventions is related to rules and metrics that we defined for SDMetrics.

The subjects were provided with an Excel Logbook template to record the time spent during the assignment in a uniform manner. They recorded their time for the three activities related to the development of the UML model: modeling itself, reviewing the model and meetings related to the model.

We used a post-test questionnaire to collect data about the subjects' educational background, experience, how the task was executed and subjects' attitude towards the task. The 17 questions of the questionnaire were distributed through the university's internal survey system.

3.7 Analysis Techniques

For quality and effort we have to analyze number of defects and time in minutes, respectively. These metrics are measured on a ratio scale. We use descriptive statistics to summarize the data. For hypothesis testing we compare the means using a one-way ANOVA test. We have analyzed the data with respect to the assumptions of the ANOVA test and have found no severe violations. The analysis is conducted using the SPSS [1] tool, version 12.0. As this is an exploratory study we reject the null hypothesis at the significance level of 0.10 ($p<0.10$).

The data from the post-test questionnaire, which was designed as a multiple-choice questionnaire, were answers on a five-point Likert-scale. Hence, they are measured on an ordinal scale. We summarize the data by presenting the frequencies as percentages for each answer option and providing additional descriptive statistics where appropriate. The answer distributions between different treatment groups are compared using the χ^2-test [18]. Microsoft Excel was used for this test. We apply the threshold of $p<0.10$ for statistical significance. When comparing three distributions (NoMC, MC and MC+T) a χ^2 value greater than 13.36 implies that $p<0.10$. In cases of comparing only two distributions the threshold is $\chi^2 = 7.78$.

4 Results

4.1 Outlier Analysis

During the duration of the experiment eight subjects dropped out (7.5%). The affected teams were distributed evenly over all treatments, therefore we do not exclude their data. One team in group MC+T completely dropped out, therefore we exclude its data.

To check whether the data is reasonable and to identify invalid data sets we analyze the outliers. Figure 1 shows the boxplots for the size of the obtained models (number of classes, on the left) and the total amount of time needed by the teams to complete the task (on the right). According to Wohlin [23] the

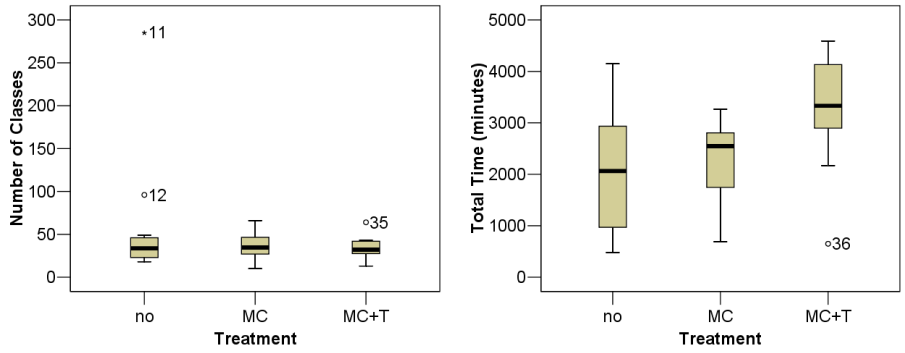

Fig. 1. Boxplots for Number of Classes and Total Time

reasons for an outlier should be analyzed in order to decide whether to include or to exclude the data point in the analysis. We scrutinized the outliers and came to the conclusion that they are not due to a rare event that can never happen again. As these outliers can happen in other situations as well, we decided to include them in the analysis.

4.2 H1: Presence of Defects

Total Number of Defects. We assess the quality of the UML model in terms of number of defects as described in Section 3.2. Figure 2 shows the boxplot for the total number of defects (on the left) and the number of defects normalized by the size of the model (on the right). Table 2 shows the descriptive statistics. The percentages in Table 2 are relative to the treatment level NoMC. The descriptive statistics for the normalized number of defects show that modeling conventions (MC) reduce the mean and the median. Tool-supported modeling conventions (MC+T) result in a larger reduction of defects. However, according to the ANOVA test (see Table 3) the results are not statistically significant and we cannot reject the null hypothesis $H1_0$.

Detailed Results. In addition to the total number of defects which is discussed above, we have conducted a detailed analysis of 19 metrics and rules that are related to the modeling conventions applied in this experiment. For nine of these metrics the results for both MC and MC+T are better than for the control group. An example is the metric *Number of Sequence Diagrams per Use Case* which indicates how well the functionality defined in use cases is specified by the sequence diagrams. Compared to the control group this metric is 30.8% greater for MC and 80.5% greater for MC+T (these results are statistically significant). Three metrics show an improvement for MC+T but a decrease for MC. An example is the metric *Number of Objects*. The metric *Coupling between Objects (CBO)* is the only one that has worse results for both MC and MC+T than for the control group. A possible explanation could be, that the subjects applying

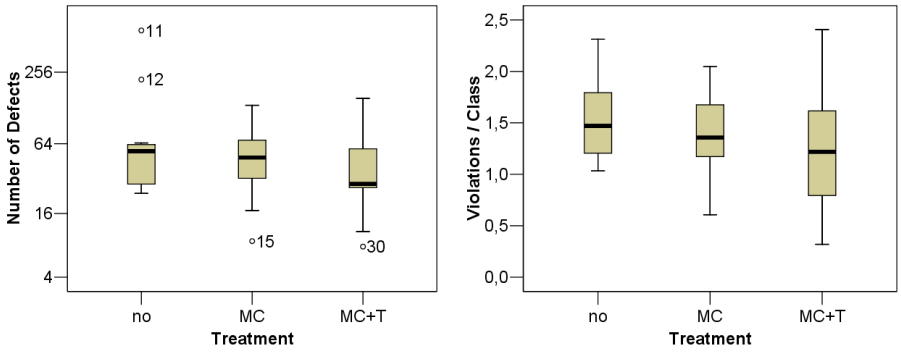

Fig. 2. Boxplots for absolute Number of Defects and Defect Density

modeling conventions model associations between classes more explicitly, resulting in a higher CBO. The results of six metrics are inconclusive because of the small number of occurrences of the rule-violations. Due to space limitations we cannot provide the entire detailed results here. They can be found in [14].

4.3 H2: Effort

We measure the effort to develop the UML model in minutes using logbooks. Table 2 shows the descriptive statistics for modeling, reviewing and team meetings. The columns showing percentages are relative to the treatment level NoMC. The descriptive statistics show that both the mean and the median increase for MC are higher for MC+T. Additionally we performed an ANOVA-test for hypothesis testing. The results of the ANOVA-test are shown in Table 3. The results for the total effort are statistically significant. Hence, we reject the null-hypothesis $H2_0$. However, when we analyze at the level of activities, we see that only the results of modeling are statistically significant.

4.4 Attitude

To fully investigate the usefulness of modeling conventions it is necessary to assess the subject's attitude towards modeling conventions. We investigated the subjects's attitude using the post-test questionnaire. The questions are multiple-choice questions with answers on a Likert scale ranging from 1 (very low agreement) to 5 (very high agreement). The results are summarized in Table 4.

The subjects perceived the difficulty of the task as medium. The difficulty of performing the task with tool-supported modeling conventions is about 10% higher than for MC.

There is a statistically significant difference in the degree to which the subjects enjoyed the task. The mean for control group (NoMC) is almost one point higher than for the other two treatment groups. The lower enjoyment might be caused by the extra effort (see Section 4.3).

Table 2. Descriptive Statistics for Defects and Modeling Effort (in Minutes)

	Treatment	Mean	Perc.	Median	Perc.	StDev	Max	Min
Defects	NoMC	102.42	100.0%	55.5	100.0%	157.280	572	42
(total)	MC	53.67	52.4%	49.0	88.3%	34.102	135	9
	MC+T	46.91	45.8%	29.0	52.3%	40.990	154	8
Defects	NoMC	1.5181	100.0%	1.4720	100.0%	0.3964	2.312	1.032
(normalized)	MC	1.3740	90.5%	1.3564	92.1%	0.4121	2.045	0.607
	MC+T	1.2443	82.0%	1.2195	82.8%	0.6671	2.406	0.320
Effort	NoMC	1069.17	100.0%	910	100.0%	670.22	2125	120
(Modeling)	MC	1157.92	108.3%	982.5	108.0%	718.225	2280	105
	MC+T	1885	176.3%	2010	220.9%	834.554	3130	540
Effort	NoMC	367.5	100.0%	300	100.0%	329.224	1155	0
(Reviewing)	MC	385.83	105.0%	272.5	90.8%	299.4	900	75
	MC+T	524.55	142.7%	600	200.0%	379.727	1250	0
Effort	NoMC	555.42	100.0%	375	100.0%	499.297	1710	0
(Meeting)	MC	720	129.6%	640	170.7%	632.488	1770	0
	MC+T	862.73	155.3%	690	184.0%	839.069	3060	0
Effort	NoMC	1992.08	100.0%	2062.5	100.0%	1187.498	4150	480
(Total)	MC	2245.42	112.7%	2545	123.4%	852.471	3265	690
	MC+T	3272.27	164.3%	3330	161.5%	1151.838	4590	650

The results show that the subjects of all treatment groups slightly indicate that they have confidence in the quality of their models. There is no significant difference between the treatment groups.

The results show that the task and the treatment were well understood and that the subjects were well motivated. This is necessary to be able to draw valid conclusions from the experiment. The χ^2-test did not show significant differences between the treatments groups.

4.5 Adherence to the Treatment

We used the answers to the post-test questionnaire to investigate the subjects' adherence to treatment MC and MC+T. The answers are summarized in Table 5. The table shows the percentages for the points '1' (very low adherence) to '5' (very high adherence). On average both treatment groups adhere better than neutral to the modeling conventions (the mean is greater than 3). The χ^2-test shows that the difference between MC and MC+T is not statistically significant.

The reported average adherence to the analysis tool is below the neutral point (3). We conducted a χ^2-test to find out whether the adherence differs significantly from the adherence to the modeling conventions of the same treatment group. The difference is statistically significant at the 10% significance level.

Furthermore we asked the subjects how they applied the treatment. For both treatment groups that applied modeling conventions, more than 80% of the subjects indicate that they read the modeling conventions several times during the project. The tool was used up to ten times during the project at an average of 3.32 times. The two authors who were instructors of the course report that

Table 3. Results of the ANOVA test for Defects and Effort

		$\sum$ **Squares**	**df**	**Mean Squr.**	**F**	**Sig.**	**Hypothesis**
Defects	Betw. Groups	21570.1	2	10785.09	1.144	.331	$H1_0$
(total)	With. Groups	301708.5	32	9428.39			failed to
	Total	323278.7	34				reject
Defects	Betw. Groups	.432	2	.216	.858	.433	$H1_0$
(normalized)	With. Groups	8.048	32	.251			failed to
	Total	8.479	34				reject
Effort	Betw. Groups	453675.4	2	2268187.708	4.129	.025	rejected
(Modeling)	With. Groups	17580265	32	549383.268			
	Total	22116640	34				
Effort	Betw. Groups	166964.89	2	83482.446	.738	.486	failed to
(Reviewing)	With. Groups	3620239.4	32	113132.481			reject
	Total	3787204.3	34				
Effort	Betw. Groups	544447.47	2	272223.736	.614	.547	failed to
(Meeting)	With. Groups	14183091	32	443221.597			reject
	Total	14727839	34				
Effort	Betw. Groups	10421703	2	5210851.564	4.535	.018	$H2_0$
(Total)	With. Groups	36772764	32	1149148.875			rejected
	Total	47194467	34				

Table 4. Subjects' Attitudes towards the Task

	Treatment	**N**	χ^2	**Mean**	**1**	**2**	**3**	**4**	**5**
Difficulty	NoMC	34	11.860	2.94	0.00%	23.53%	61.76%	11.76%	2.94%
	MC	36		3.00	2.78%	19.44%	52.78%	25.00%	0.00%
	MC+T	33		2.61	6.06%	42.42%	36.36%	15.15%	0.00%
Enjoy	NoMC	34	18.886	3.47	0.00%	14.71%	32.35%	44.12%	8.82%
	MC	36		2.58	16.67%	27.78%	36.11%	19.44%	0.00%
	MC+T	33		2.58	21.21%	21.21%	36.36%	21.21%	0.00%
Confidence	NoMC	34	5.526	3.18	2.94%	17.65%	41.18%	35.29%	2.94%
in Quality	MC	36		3.31	0.00%	11.11%	47.22%	41.67%	0.00%
	MC+T	33		3.24	3.03%	21.21%	27.27%	45.45%	3.03%
Understanding	NoMC	34	4.089	3.18	8.82%	14.71%	35.29%	32.35%	8.82%
Task	MC	36		3.08	2.78%	27.78%	33.33%	30.56%	5.56%
	MC+T	33		2.91	9.09%	27.27%	30.30%	30.30%	3.03%
Motivation	NoMC	34	3.862	3.56	5.88%	8.82%	23.53%	47.06%	14.71%
	MC	36		3.44	5.56%	5.56%	36.11%	44.44%	8.33%
	MC+T	33		3.67	3.03%	3.03%	30.30%	51.52%	12.12%

Table 5. Adherence to the treatment

Adherence to	**Treatment**	**N**	χ^2	**Mean**	**1**	**2**	**3**	**4**	**5**
Modeling	MC	36	5.027	3.638	0.00%	5.56%	33.33%	52.78%	8.33%
Conventions	MC+T	33		3.303	3.03%	6.06%	54.55%	30.30%	6.06%
Analysis Tool	MC+T	33	9.326	2.727	12.12%	27.27%	42.42%	12.12%	6.06%

they received questions about both the modeling conventions and the analysis tool starting from the second week of the experiment.

5 Threats to Validity

Internal Validity. Threats to internal validity can affect the independent variables of an experiment. A possible threat to internal validity is that the treatment groups behave differently because of a confounding factor such as difference in skills, experience or motivation. Our analysis results show no significant differences between the treatment groups for these factors.

A risk is that subjects apply a treatment they should not apply, because they are eager to learn about new technology. We minimized this risk by (i) not telling the subjects the goal of the experiment, (ii) by informing the subjects that their grade is not influenced by the treatment group that they were in, (iii) by making modeling conventions and tool available only to the appropriate teams, and (iv) by informing the subjects that all technology would be made available to all subjects after completion of the task. In the case that subjects would have received a different treatment despite these precautions, it would only decrease the effect between the treatment groups. Hence, in case this happened, the effect would be larger in reality.

External Validity. Threats to external validity reduce the generalizability of the results to industrial practice. As described in Section 3 the experiment is designed to render a realistic situation. Hence, the experimental environment is designed to maximize generalizability (at the cost of statistical significance). We use students as subjects, which might be a threat to external validity. However, all students in this experiment hold a BSc degree in computer science and have relevant experience.

Due to curricular constraints the amount of training and, hence, experience with modeling conventions and the analysis tool is limited. This renders the situation in the introduction phase of the technology. We assume that more experience results in a reduction of extra effort and possibly a larger effect on model quality.

Construct Validity. Construct validity is the degree to which the variables measure the concepts they are to measure. The concept of quality is difficult to measure and it consists of several dimensions[11]. It is not feasible to cover all dimensions in a single experiment. We limit the scope of this experiment to defect containment. Using well-established tooling to measure the defect containment we are confident to measure this dimension of model quality correctly.

Conclusion Validity. Conclusion validity is concerned with the relation between the treatment and the outcome. The statistical analysis of the results is reliable, as we used robust statistical methods.

We minimized possible understanding problems by testing the experiment material in a pilot experiment and improving it according to the observed issues. The course instructors were available to the students for clarification questions.

The results of the post-test questionnaire show that the task was well understood. Hence, we conclude that there were no understanding problems threatening the validity of the reported experiment.

The metrics of the UML models (defects, size...) were collected using an analysis tool and are therefore repeatable and reliable. A possible threat to the conclusion validity is the reliability of the measured time and the data from the post-test questionnaire. For time collection a logbook template was used to assure uniformity. The authors analyzed the data for validity and no obvious problems were found.

6 Conclusions

The UML consists of different diagram types, has no formal semantics and does not provide guidelines on how to use the language features. Inherent to these characteristics is the risk for quality problems such as defects and non-uniform use of the language. In this study we propose modeling conventions as a forward-oriented means to reduce these quality problems. Our literature review shows that existing work focusses on particular categories of conventions for UML modeling and that there is lack of empirical validation of conventions for UML modeling.

Our main contribution is an experiment that provides empirical data about the application of modeling conventions in a realistic environment. Our results show that the defect density in UML models is reduced through the use of modeling conventions. However, the improvement is not statistically significant. Additionally, we provide data about the additional effort needed to apply modeling conventions with and without tool-support. The presented data quantifies the trade-off between improved model quality by using modeling conventions and the cost of extra effort. Additional observations describe the developers' attitude towards modeling conventions and how the modeling conventions were applied within the development teams. We observed that the adherence to modeling conventions, especially for tool-supported modeling conventions, bears potential for improvement. Furthermore the subjects using modeling conventions enjoyed their task less than the subjects who did not use modeling conventions, indicating that the commitment in using modeling conventions can be improved.

Due to the time constraints of the experiment, we provided the subjects with a set of modeling conventions, instead of letting them select the conventions themselves. However, the subjects had no experience whether the modeling conventions were useful for their task, and the subjects received no reward for delivering a better quality model (the typical reward would be less effort during use of the UML models in a later phase). In practice it would be desirable if the developers who must eventually use the conventions participate in establishing the set of modeling conventions. This would increase their knowledge about and trust in the conventions and we expect they would have more commitment in using modeling conventions. We expect that the commitment will also be improved in a practical situation because the models will be used after they have been developed, resulting in rewarding the models' quality. The subjects in this

experiment were not experienced using modeling conventions or the analysis tool. Therefore the experiment resembles the introduction of modeling conventions to a project. We expect that for more experienced developers the quality improvement is larger and the amount of extra effort will be reduced.

The tool-support for adherence to the modeling conventions was given by a stand-alone tool. We expect that integrating adherence checks into UML development tools will decrease the extra effort and result in higher adherence, because of a shorter feedback loop. Egyed's instant consistency checking [8] is a promising technique for short feedback loops.

The observations made in this experiment potentially lead to the following guidelines for applying UML modeling conventions:

- Attention must be paid to control the adherence to the modeling conventions.
- Commitment of the developers increases the adherence to the modeling conventions.
- Modeling conventions should be tailored for a specific purpose of modeling.
- Tool support to enforce adherence to the modeling conventions increases the quality improvement. A short feedback loop is required to minimize the amount of necessary rework.

In future work the effect of adherence and experience on the effectiveness and efficiency of modeling conventions should be investigated in more detail. External replications of the reported experiment should be conducted to further confirm our findings. We focussed at syntactical quality of UML models in this experiment. We are conducting a follow-up experiment where we investigate semantic and pragmatic quality.

References

1. SPSS, version 12.0. http://www.spss.com.
2. Gentleware AG. Poseidon for UML, community edition, version 3.1. http://www.gentleware.com.
3. Scott W. Ambler. *The Elements of UML 2.0 Style*. Cambridge University Press, 2005.
4. Victor R. Basili, G. Caldiera, and H. Dieter Rombach. The goal question metric paradigm. In *Encyclopedia of Software Engineering*, pages 528–532, 1994.
5. Lionel C. Briand, Christian Bunse, and John William Daly. A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. *IEEE Transactions on Software Engineering*, 27(6):513–530, June 2001.
6. Peter Coad and Edward Yourdon. *Object Oriented Design*. Prentice-Hall, first edition, 1991.
7. Reidar Conradi, Parastoo Mohagheghi, Tayyaba Arif, Lars Christian Hedge, Geir Arne Bunde, and Anders Pedersen. Object-oriented reading techniques for inspection of UML models – an industrial experiment. In *Proceedings of the European Conference on Object-Oriented Programming ECOOP'03*, volume 2749 of *LNCS*, pages 483–501. Springer, July 2003.
8. Alexander Egyed. Instant consistency checking for the UML. In *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*, pages 381–390. ACM, May 2006.

9. Holger Eichelberger. Aesthetics of class diagrams. In *Proceedings of the First IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2002)*, pages 23–31. IEEE CS Press, 2002.
10. Norman E. Fenton and Shari Lawrence Pfleeger. *Software Metrics, A Rigorous and Practical Approach*. Thomson Computer Press, second edition, 1996.
11. Barbara Kitchenham and Shari Lawrence Pfleeger. Software quality: The elusive target. *IEEE Software*, 13(1):12–21, Januari 1996.
12. Ludwik Kuzniarz, Miroslaw Staron, and Claes Wohlin. An empirical study on using stereotypes to improve understanding of UML models. In *Proceedings of the 12th IEEE International Workshop on Program Comprehension (IWPC'04)*, pages 14–23. IEEE CS Press, 2004.
13. Christian F. J. Lange. Material of the modeling conventions experiment. http://www.win.tue.nl/~clange.
14. Christian F. J. Lange, , Bart DuBois, Michel R. V. Chaudron, and Serge Demeyer. Experimentally investigating the effectiveness and effort of modeling conventions for the UML. CS-Report 06-14, Technische Universiteit Eindhoven, 2006.
15. Christian F. J. Lange and Michel R. V. Chaudron. Effects of defects in UML models - an experimental investigation. In *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*, pages 401–411. ACM, May 2006.
16. Christian F. J. Lange, Michel R. V. Chaudron, and Johan Muskens. In practice: UML software architecture and design description. *IEEE Software*, 23(2):40–46, March 2006.
17. Odd Ivar Lindland, Guttorm Sindre, and Arne Sølvberg. Understanding quality in conceptual modeling. *IEEE Software*, 11(2):42–49, March 1994.
18. Meerling. *Methoden en technieken van psychologisch onderzoek*, volume 2. Boom, Meppel, The Netherlands, 4th edition, 1989.
19. Object Management Group. *Unified Modeling Language, Adopted Final Specification, Version 2.0*, ptc/03-09-15 edition, December 2003.
20. Paul W. Omam and Curtis R. Cook. A taxonomy for programming style. In *Proceedings of the 18th ACM Computer Science Conference*, pages 244–250, 1990.
21. Helen C. Purchase, Jo-Anne Allder, and David Carrington. Graph layout aesthetics in UML diagrams: User preferences. *Journal of Graph Algoritms and Applications*, 6(3):255–279, 2002.
22. Sandra A. Slaughter, Donald E. Harter, and Mayuram S. Krishnan. Evaluating the cost of software quality. *Communications of the ACM*, 41(8):67–73, August 1998.
23. Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlesson, Björn Regnell, and Anders Wesslen. *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2000.
24. Jürgen Wüst. The software design metrics tool for the UML, version 1.3. http://www.sdmetrics.com.