

# An Empirical Assessment of Using Stereotypes to Improve Reading Techniques in Software Inspections

Mirosław Staron, Ludwik Kuzniarz, Christian Thurn

Department of Systems and Software Engineering

School of Engineering

Blekinge Institute of Technology

Box 520, Soft Center

SE-372 25 Ronneby, Sweden

(Miroslaw.Staron, Ludwik.Kuzniarz)@bth.se, Christian.Thurn@ipbolaget.com

## Abstract

*Stereotypes were introduced into the Unified Modeling Language (UML) in order to provide a means of customizing the language for particular needs. The stereotypes can increase the comprehension of UML diagrams and therefore influence reading techniques used for inspections of software artefacts. In this paper we evaluate how the usage of stereotypes in UML designs influences outcomes of three reading techniques used for verification and validation of UML models. The study presented in this paper is done in the context of the UML domain modeling, but the results can be generalized to other kinds of models expressed in UML. The results show that the presence of stereotypes improves the efficiency and effectiveness of the studied methods and shows the magnitude of the improvements. We also investigate which of the reading techniques are the most efficient and effective for analysis of UML designs with stereotypes.*

## 1. Introduction

The original role of stereotypes in software development was to provide a means of secondary classification of objects according to the role the objects played in the system [1, 2]. The primary classification mechanism is regarded to be the object's type – its class – and the inheritance hierarchy of the class. With the introduction of the Unified Modeling Language (UML, [3]) the stereotypes evolved into a full fledged extension mechanism – they became a means of extending the

general purpose UML for specific needs. As it appears in practice [4, 5] the stereotypes are used in both ways in UML designs – as a secondary classification and as a means of language extension. Additionally, certain stereotypes can be used in order to enable automating software development (for example code generation or automatic model transformations - [6, 7]). Other stereotypes are not intended to help in automating software development, but to increase the readability of designs. Our previous experiments [8, 9] showed that a proper introduction of stereotypes can increase the understanding of UML models by 24% - 131% (compared to the equivalent non-stereotyped models). The increased understanding is useful in improving communication within the development team and in verification and validation of UML designs. It is our intention in this paper to evaluate the influence of stereotypes on inspecting software artefacts via a controlled experiment in the context of reading techniques for inspecting software artefacts. The experiment was performed at Blekinge Institute of Technology and was aimed at comparing three reading techniques in terms of the number of faults found by each technique and the time required to inspect the documents. The documents consisted of class diagrams and collaboration diagrams. In order to increase the number of data points in the analysis (while keeping the constant number of subjects) two kinds of collaboration diagrams were used – stereotyped and non-stereotyped. We analyze the data with focus on the efficiency and effectiveness of the techniques. We statistically test hypotheses on the influence on stereotypes on reading techniques. Nevertheless, we do not regard the kind techniques as an independent variable in our study in order to be able to use statistical significance testing despite a limited number of data points in our study.

The outline of the paper is as follows. Section 2 overviews the most important related work in the field. Section 3 introduces the necessary details about stereotypes and reading techniques used in the experiment. The design of the experiment is presented in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

3-WoSQ '05, May 17, 2005, St Louis, Missouri, USA.

Copyright 2005 ACM 1-59593-122-8/05/0005...\$5.00.

section 4, and the results of the experiment are analyzed in section 5. Finally we draw conclusions on the influence of stereotypes on the reading techniques in section 6.

## 2. Related work

The related work to the research presented in this paper is connected to these research fields:

- Investigations of reading techniques for inspecting different artefacts – e.g. [10-13]
- Previous experiments in which we have evaluated the role of stereotypes on the understanding of UML models – i.e. [8, 9]
- Analysis of the notion of stereotypes – i.e. [4, 14-16]
- Cost of using stereotypes in the course of industrial software development – i.e. [6, 7]

Research in all the above areas have influenced the design of this study as we used the experiences and best practices in order to design the experiment.

## 3. Background

Empirical assessment of the influence of stereotypes on reading techniques of UML documents is done in the context of the two experimental elements – UML stereotypes and the reading techniques used in the experiment.

### 3.1. Stereotypes

As defined in the UML specification documents [17], the main idea behind using stereotypes is to introduce new semantics to the existing model elements. The UML definition of stereotypes involves the definitions of other extension mechanisms – tagged values and constraints (c.f. [5, 18]). Stereotypes allow extending the language in a way, which is consistent with the definition of the language and they are useful in automatic model transformations, like for example code generation for a specific purpose (e.g. [19]).

In addition to the above, there is also another way of perceiving stereotypes – the original intention of introducing the notion of stereotypes. The stereotypes can provide a secondary classification of model elements. This concept was initially introduced in [2]. Such stereotypes provide a means of expressing some classification of the stereotyped model elements, adding properties, which cannot be defined for all model elements of the same kind, but only for some. These stereotypes can be classified as *transitive stereotypes* (according to the classification presented in [15]), because they are added to classifiers on the model level, but should also be recognized on the instance level. They are useful as a secondary classification mechanism [1] since they both brand the classifier and its instances with additional meaning.

### 3.2. Reading techniques

Software inspections as described in [20] are used to find errors in software artefacts. In the course of inspections, different reading techniques are used to examine the artefacts and to find errors. In the investigation presented in this paper, we use two specific reading techniques –

checklist-based reading (CBR, [20]) and perspective-based reading (PBR, [10]) and an unstructured reading (further referred to as the ad-hoc technique).

In the context of software inspections, the reading techniques are only a part of the whole process. Usually, the complete process consists of planning, overview, preparation, meeting, rework and follow-up. The details of all steps in the inspection process can be found in [20].

Checklist based reading (CBR) is a reading technique in which the reader is given a checklist with specific kind of faults to look for. The items in the checklist can be expressed as questions or as statements. In particular, the checklist contains items that help in finding logical errors in the inspected documents – errors that cannot be verified in an automatic way by a modeling tool (in the case of UML models).

Perspective based reading (PBR) is a reading technique in which artefacts are examined from certain perspectives. Each perspective is intended to provide a different way of examining the document. Using different perspectives allow focusing on various aspects of the document (for example user's or designer's perspective). One of the assumptions of PBR is that the reader can better identify faults if he/she works in a structured manner. The PBR is a special kind of scenario-based reading techniques [21].

The third kind of reading can be characterized as ad hoc reading. It denotes a technique which provides no guidelines and implies that the readers use their personal experience to find faults. Only a general description of the task is provided as part of the instructions for this reading technique.

As stereotypes are abstractions of certain common properties, they are used to provide additional information for designs – in particular information that could influence the reading techniques.

## 4. Experiment design

The experiment presented in this paper was performed in an academic setting, and it was based on the results of our previous experiments on software comprehension in both academia and industry.

### 4.1. Objects




Objects in the study consisted of several artefacts, which were presented to subjects sequentially (see section 4.6). The set of artefacts consist of:

- 1) UML model – class and collaboration diagrams for stereotyped and non-stereotyped models, and
- 2) Requirements specification – informal text in English describing the domain under consideration.

The UML model describes a domain of radio and television transmissions. The class diagram contains the set of ten classes used to describe different kinds of elements in the domain (e.g. transmission stations, radio receivers, and television broadcast stations). The non-stereotyped class diagram contains seven faults and was examined by subjects before examining the

collaboration diagrams. The stereotyped class diagram contained no errors and was given out to subject together with the stereotyped collaboration diagram.

The collaboration diagrams (which contain 30 objects) describe one of the possible situations – how a news program is broadcasted in a country. There were two kinds of collaboration diagrams – non-stereotyped diagram and stereotyped diagram. The set of stereotypes (and their graphical representation) used in the experiments is presented in Table 1. The diagrams are presented in [22] and are based on diagrams from [23]. Both collaboration diagrams were of equal size and complexity (30 objects). The stereotypes in the stereotyped model were replaced by inheritance and notes (on the class diagrams with the same information that the information in the description of each stereotype) in the non-stereotyped model. In this case we have replaced the secondary classification mechanism with the primary one – inheritance – in such a way that both models express the same information [2, 18]. The size of the diagrams was found to be similar to an average diagram used in practice [24].

Stereotype	Icon	Description
Sender		This stereotype represents a class, which instances send telecommunication signals to instances of other classes, stereotyped Receiver or Transmitter
Receiver		This stereotype represents a class, which instances receive telecommunication signals from instances of other classes, stereotyped Sender or Transmitter.
Transmitter		This stereotype represents a class, which instances are used to relay telecommunication signals.

**Table 1. Stereotypes used in the experiment**

Each of the collaboration diagrams has ten errors seeded. The errors were of the same kind, but they were seeded in different places in the diagram in order to minimize the learning effect. The errors that were introduced into the designs were logic errors, not syntax errors – i.e. all diagrams were syntactically correct UML diagrams, but they did not model the situation correctly. For example, some senders were receiving signals and some receivers were sending signals; some of the receivers were “overloaded”, i.e. received signals from more than two sources. The subjects were familiar all these kind of errors – they were given a lecture on them before the experiment.

Additionally, each set of objects contains the requirement specification document which describes the domain of broadcasting and the situation that is modeled in the collaboration diagrams.

#### 4.2. Subjects

Subjects in the study were students of the last year of the Master of Science in software engineering program at Blekinge Institute of Technology, Sweden. They have experience in working in teams since they followed a set of several project courses – including a course in team software development (in teams of approx. 15 members) Having the project courses as a background was required since our intention was to have subjects whose experience is as close as possible to the industrial – especially in performing different verification and validation tasks. All

of them attended a UML course and they were familiar with stereotypes. They had used UML in their projects previously.

There were eleven subjects in the study. Based on their background the subjects were divided into three groups (blocking with respect to knowledge in inspections and object-orientation was applied) – each group used different reading technique (CBR, PBR, and ad-hoc).

#### 4.3. Instrumentation

The instruments in the study were faults finding questionnaires, in which the faults found in the inspected design were reported. The data gathered for each fault included:

- Time when the fault was found,
- Diagram which contain the fault (class diagram, non-stereotyped collaboration diagram, stereotyped collaboration diagram),
- Description of fault (subject’s description of the fault).

In addition to the fault report forms, the subjects were given guidelines for the appropriate reading techniques. For the CBR, the checklist to be used was given. For PBR, the description of the perspective was given which contained some guidelines on the errors that were to be searched for in the diagrams. In the experiment the designer’s perspective was used. For the ad-hoc method, a short description on the kinds of faults present in the diagrams was given, but no guidelines on how to look for the faults. All subjects were given a description of their task. The subjects were also required to write the time when they started and finished inspecting each diagram.

#### 4.4. Variables

There is one independent variable (factor) in our study – the type of model – with two different factor levels: stereotyped and non-stereotyped model. There are two dependent variables in the study:

- Inspection time (TIME)
- Number of faults found (FAULTS)

The dependent variables are used to calculate the efficiency and effectiveness of the reading technique. The *efficiency* is the ratio of the number of faults found to the inspection time. The *effectiveness* is the ratio of the number of faults found to total number of faults (seeded). The total number of faults in the inspected document was known and constant for both stereotyped and non-stereotyped model.

Although in the analysis presented in [22] there were two independent variables: kind of diagram (class/non-stereotyped collaboration/stereotyped collaboration) and reading technique (CBR/PBR/Ad-hoc) the number of data points did not allow to use appropriate statistical methods (one-way ANOVA) for analysis. Therefore, in this paper, we do not regard different reading techniques as an independent variable.

#### 4.5. Hypotheses and data analysis methods

There are four hypotheses in our study, which are evaluated separately:

$H_{0\text{-effectiveness}}$ : Introducing stereotypes does not influence the effectiveness of finding faults by subjects

$H_{1\text{-effectiveness}}$ : Introducing stereotypes influences the effectiveness of finding faults by subjects

$H_{0\text{-efficiency}}$ : Introducing stereotypes does not influence the efficiency of finding faults by subjects

$H_{1\text{-efficiency}}$ : Introducing stereotypes influences the efficiency of finding faults by subjects

Each variable is intended to be evaluated separately and we use a repeated measures design in order to evaluate the change in the variables for the two treatment levels for each subject.

The obtained data is to be tested whether it is normally distributed with the Shapiro-Wilk test. If it is normally distributed, the parametric paired t-test [25] is used for statistical significance testing. If the data is not normally distributed, then the Wilcoxon test [25] is used. In addition to the statistical significance testing for analyzing whether the stereotypes improve the effectiveness and efficiency of reading techniques, we also analyze the influence of stereotypes on each reading technique in a qualitative way. This analysis allows us to find out which of the assessed reading techniques is improved by the introduction of stereotypes to the largest extent. However, due to the small number of data points for each reading technique (between three and four data points) we use only the descriptive statistics. It is our intention to only identify the possibly “best” reading technique, which we intend to put under further investigations in our further work.

#### 4.6. Operation

The experiment was divided into two parts – preparation and conducting the study. The preparation took place a week before the experiment. The subjects were chosen and they were asked to fill in a short background questionnaire about their knowledge related to the matter of experiment objects. The experiment was conducted during approximately three hours during one day at a lecture room which the subjects were used to have lectures in.

During the experiment, the subjects were given an introduction to their tasks, the kind of errors that they were supposed to search for in the examined diagrams. The diagrams were presented to the subjects in the following order:

1. non-stereotyped class diagram
2. non-stereotyped collaboration diagram
3. stereotyped class and collaboration diagram

Firstly, the class diagram was given for the inspection. When a subject indicated that he/she is finished with the first diagram, a non-stereotyped collaboration was given to the subject. The class diagram was not taken away from the subject, but the subject was not supposed to examine the class diagram (the role of the diagram was to provide a description of the objects in the collaboration diagram). When a subject indicated finishing inspecting the non-stereotyped collaboration diagram, both diagrams and the fault report forms were collected from the subject (non-stereotyped diagrams). Then the subject was given the stereotyped class diagram (with no errors, see section 4.1)

and the stereotyped collaboration diagram (which the subject was to examine).

Since the task of the subjects was to use the different reading techniques (and not the whole inspection process, which requires team work), the subjects were asked not to communicate with each other during the experiment.

#### 4.7. Threats to validity of the study

As all empirical studies this study has threats to its validity. We group and analyze the threats according to the classification presented in [26].

The *external validity* threats relate to the generalizability of the results of the study. In the case of this study the threat is that the results obtained from the student subjects are not generalizable to industry professionals. In order to minimize this threat, the sample of the experiment consisted of students of the last year of the software engineering program. As discussed in section 4.2 the experience of these subjects was close to the experience of newly trained software developers in industry. Another generalizability threat is the kind of UML diagrams and the set of stereotypes used. Both were prepared at the university by experimenters, but it was our intention to retain the high degree of control of the complexity of the models.

A threat to the *construct validity* of the study might be the set of stereotypes used in the study. The set of stereotypes was small and the stereotypes were simple. Nevertheless, the intention was to evaluate how properly defined stereotypes can help in finding faults in design documents. The proper definition of the stereotype in this context relates to the role which is intended for this stereotype (in this case the role was to simplify certain properties of designs – c.f. section 3.1 – and the stereotypes were found in our previous experiments do fulfill this role).

There is also an *internal validity* threat. As all subjects were given the stereotyped collaboration diagram after the non-stereotyped collaboration diagram, the observed effect could be caused by the order of investigation of documents and not because of the introduction of stereotypes. In the previous experiments with similar objects and a similar sample, we have found that the order of presentation of diagrams does not influence the results [8, 9]. It is only after thorough investigation of the results of our previous experiments that we have decided to use this design in this experiment.

A threat to the *conclusion validity* in the study is the small sample size. In order to be able to use statistical methods for significance testing we decided to use only one group and use the repeated measures design (paired experiment design [26]). This provided us with the possibility to measure the improvement after the introduction of stereotypes against each subject’s own performance. Another threat to the conclusion validity is the fact that the statistical significance testing is not used in the analysis of the influence of stereotypes on separate reading techniques. Nevertheless, we present the descriptive statistics in order to explore the

variability in the improvements for each technique and identify the techniques that should be investigated more thoroughly in our future experiments.

## 5. Analysis of experiment results

In this section, we analyze the results in order to verify the hypotheses posed in section 4.5. We also present the analysis of the influence of stereotypes on each reading technique separately which is intended to provide an exploratory investigation of which technique is affected by the introduction of stereotypes to the largest extent.

### 5.1. Efficiency

The summary of the efficiency results of both collaboration diagrams is presented in Figure 1.

The basic descriptive statistics for the efficiency are presented in Table 2. The descriptive statistics suggest that there is an improvement in efficiency of 76% after the introduction of stereotypes.

Factor level	Mean	Std. Deviation	Percentage
Stereotyped	0.59	0.125	176%
Non-stereotyped	0.34	0.095	100%
Differences Stereotyped – non-stereotyped	0.25	0.118	76% = $0.25/0.34*100\%$

Table 2. Descriptive statistics for efficiency

The box-plots presented in Figure 2 summarize the data set in a graphical form.

The Shapiro-Wilk test does not allow rejecting the assumption of the data being normally distributed with significance level of 0.822. The paired t-test allows rejecting the null hypothesis ( $H_{0\text{-efficiency}}$ ) with the significance level of less than 0.0001 – the observed difference is significant. The introduction of stereotypes improves the efficiency of the reading techniques (by 76% as indicated by descriptive statistics).

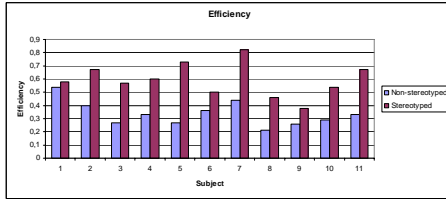


Figure 1. Efficiency of subjects

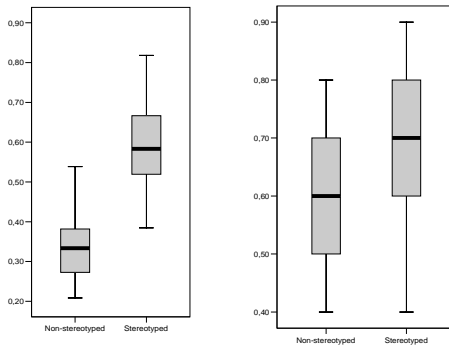


Figure 2. Efficiency (left-hand side) and effectiveness (right-hand side) - box-plots for two factor levels

### 5.2. Effectiveness

The summary of the results for effectiveness of both collaboration diagrams is presented in Figure 3.

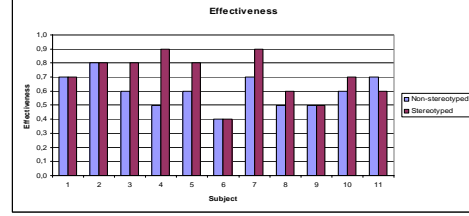


Figure 3. Effectiveness of subjects

The basic descriptive statistics for the effectiveness are presented in Table 3. The descriptive statistics indicate that there is an improvement of 17% in the effectiveness after introducing stereotypes.

Factor level	Mean	Std. Deviation	Percentage
Stereotyped	0.7	0.161	117%
Non-stereotyped	0.6	0.118	100%
Differences Stereotyped – Non-stereotyped	0.1	0.141	17% = $0.1/0.6*100\%$

Table 3. Descriptive statistics for effectiveness

The box-plots in Figure 2 graphically characterize the data. The Shapiro-Wilk test does not allow rejecting the assumption of the data being normally distributed with the significance level of 0.259. The paired t-test allowed rejecting the null hypothesis ( $H_{0\text{-effectiveness}}$ ) with the significance level of 0.041. Using stereotypes improves the effectiveness of the reading techniques (by 17%).

### 5.3. Improvements for each technique

In order to investigate which of the studied reading techniques was affected most by introducing stereotypes, we perform an analysis of the effect of introducing stereotypes for each method. The analysis is done only with descriptive statistics due to the small number of data points for each reading technique (between three and four subjects).

Summary of the number of subjects in each group and the descriptive statistics is presented in Table 4. The improvement is calculated in the same way as in Table 3 (e.g. for PBR improvement =  $(0.61-0.40)/0.40*100\%$ ).

Technique	Number of subjects	Type	Mean	SD	Improvement [%]
PBR	3	N-S	0.40	0.135	53%
		S	0.61	0.055	
CBR	4	N-S	0.35	0.071	89%
		S	0.66	0.141	
Ad-hoc	4	N-S	0.27	0.051	89%
		S	0.51	0.124	

Table 4. Summary of differences in efficiency by reading techniques. S in column Type denotes the stereotyped model and N-S denotes the non-stereotyped model.

The descriptive statistics indicates that all reading techniques have been positively influenced by the introduction of stereotypes. The box-plots in Figure 4 show graphically the differences between the two treatments for different perspectives.

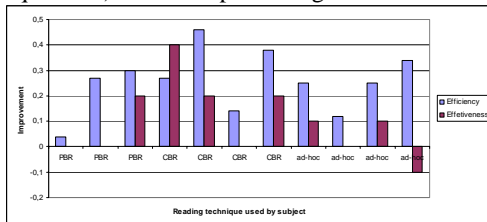
The boxes show different improvements in all reading techniques. The descriptive statistics indicate that the CBR technique is the most efficient technique for investigating UML diagrams which contain stereotypes (the efficiency is 0.66). The techniques also show the largest improvement after introducing stereotypes (89%). The efficiency of PBR is only a little lower than the efficiency of CBR (PBR – 0.61).

Technique	Number of subjects	Type	Mean	SD	Improvement [%]
PBR	3	N-S	0.70	0.100	10%
		S	0.77	0.058	
CBR	4	N-S	0.55	0.129	36%
		S	0.75	0.238	
Ad-hoc	4	N-S	0.58	0.096	3%
		S	0.60	0.082	

**Table 5. Summary of differences in effectiveness by reading techniques.**

Summary of the number of subjects for each technique and the descriptive statistics are presented in Table 5. The calculations of improvements are done in similar way as calculation in Table 4.

The descriptive statistics indicate that the effectiveness was positively influenced after introducing stereotypes into designs, although in some cases the improvements were rather small (3% for the ad-hoc reading). The differences are depicted graphically by plotting the improvements of each subject (and showing the reading technique used) in the bar plot in Figure 4.



**Figure 4. Efficiency & effectiveness grouped by reading techniques**

The CBR technique showed the best results in terms of the largest improvement (36%) after introducing stereotypes. However, the best effectiveness was showed by the PBR technique (0.77) – which indicates that the PBR is the most effective reading technique for investigating stereotyped UML designs. The effectiveness of the CBR technique was not much smaller than the effectiveness of the PBR technique (CBR – 0.75). The performance of one subject was a deterioration in effectiveness (the last subject in Figure 4).

## 6. Conclusions

This paper presents a controlled experiment aimed at evaluating whether stereotypes improve efficiency and effectiveness of reading techniques for software inspections. The results show that the presence of stereotypes improves both aspects of the reading techniques. The efficiency is increased by 76% and the effectiveness by 17%. It was our intention to verify whether the increased comprehension of software artefacts results in improvements of inspection techniques. The results from this experiment seem to support such

hypothesis. In other words, the experiment provides evidence that the presence of stereotypes in UML designs improves the correctness of models after the inspection process since stereotyped models contained less faults (more faults were found in the models during inspections). This conclusion stems also from a set of experiments on the influence of stereotypes on comprehension done on 72 subjects from both academia and industry [8, 9]. An additional outcome of the study is the identification of a reading technique which is potentially the most suitable one for inspecting UML designs with stereotypes. As it appears, CBR is the most efficient technique and PBR is the most effective one. However, more empirical studies are needed to assess the differences between the techniques in the context of stereotyped UML designs on a bigger sample.

The results indicate that introducing stereotypes into design since they cause improvement in the software inspection process and in consequence the quality of models (since less faults remains undiscovered after the inspection process). This in consequence shows that the notion of stereotypes as a means of model simplification mechanisms provides certain benefits. However, the benefits of using the stereotypes should be balanced with the cost of developing the proper stereotypes (finding the proper reusable abstractions, defining stereotypes, introducing them into tool and into designs) should be evaluated. In the case of this experiment, the set of stereotypes was small and the abstractions used are reusable for several domains<sup>1</sup>. The costs of developing and using them are negligible. Nevertheless, it is not always the case that the cost can be neglected as our other studies show – for example [6].

Our further research directions include investigating an influence of the presence of stereotypes on other parts of the inspection process – e.g. re-work or follow up – and which reading technique is the most suitable for inspecting UML designs with stereotypes.

## Acknowledgements

The authors would like to thank the subjects, without whom the study would not be possible.

## References

- [1] Wirfs-Brock R., "Stereotyping: A Technique for Characterizing Objects and Their Interactions", *Object Magazine*, vol. 3, 1993, pp. 50-3.
- [2] Wirfs-Brock R., Wilkerson B., and Wiener L., "Responsibility-Driven Design: Adding to Your Conceptual Toolkit", *ROAD*, vol. 2, 1994, pp. 27-34.
- [3] Object Management Group, "Unified Modeling Language Specification V. 1.5", Object Management Group, 2003, [www.omg.org](http://www.omg.org), last accessed 2003-10-01.
- [4] Atkinson C., Kühne T., and Henderson-Sellers B., "Systematic Stereotype Usage", *Software and Systems Modeling*, vol. 2, 2003, pp. 153-163.
- [5] Gogolla M. and Henderson-Sellers B., "Analysis of UML Stereotypes in the UML Metamodel", In the Proceedings of UML 2002, Dresden, 2002, pp. 84-99.

<sup>1</sup> In our previous experiments [8], the same stereotypes were used for another domain.

- [6] Staron M., Kuzniarz L., and Wallin L., "Factors Determining Effective Realization of MDA in Industry", In the Proceedings of 2nd Nordic Workshop on the Unified Modeling Language, Turku, Finland, 2004, pp. 79-91.
- [7] Staron M., Kuzniarz L., and Wallin L., "A Case Study on Transformation Focused Industrial MDA Realization", In the Proceedings of 3rd UML Workshop in Software Model Engineering, Lisbon, Portugal, 2004.
- [8] Kuzniarz L., Staron M., and Wohlin C., "An Empirical Study on Using Stereotypes to Improve Understanding of UML Models", In the Proceedings of The 12th International Workshop on Program Comprehension, Bari, Italy, 2004, pp. 14-23.
- [9] Staron M., Kuzniarz L., and Wohlin C., "An Industrial Replication of an Empirical Study on Using Stereotypes to Improve Understanding of UML Models", In the Proceedings of Software Engineering Research and Practice in Sweden, Linköping, Sweden, 2004, pp. 53-62.
- [10] Basili V. R., Green S., Laitenberger O., Shull F., Sorumgard S., and Zelkowitz M. V., "The Empirical Investigation of Perspective-Based Reading", *Empirical Software Engineering*, vol. 1, 1996, pp. 133-64.
- [11] Regnell B., Runeson P., and Thelin T., "Are the Perspectives Really Different? Further Experimentation on Scenario-Based Reading of Requirements", *Empirical Software Engineering*, vol. 5, 2000, pp. 331-56.
- [12] Thelin T., Erlansson M., Höst M., and Wohlin C., "Experimentation with Usage-Based Reading", in *Esernet 2001-2003*, vol. 2765, *Lncs*, R. Conradi and A. I. Wang, Eds. Berlin Heidelberg: Springer-Verlag, 2003, pp. 193-207.
- [13] Laitenberger O., Atkinson C., Schlich M., and Emam K. E., "An Experimental Comparison of Reading Techniques for Defect Detection in UML Design Documents", *The Journal of Systems and Software*, vol. 53, 2000, pp. 183-204.
- [14] Atkinson C. and Kühne T., "Rearchitecting the UML Infrastructure", *ACM Transactions on Modeling and Computer Simulation*, vol. 12, 2002, pp. 290-321.
- [15] Atkinson C., Kühne T., and Henderson-Sellers B., "Stereotypical Encounters of the Third Kind", In the Proceedings of The Fifth International Conference on The Unified Modeling Language, Dresden, Germany, 2002, pp. 100-114.
- [16] Berner S., Glinz M., and Joos S., "A Classification of Stereotypes for Object-Oriented Modeling Languages", In the Proceedings of The Second International Conference on the Unified Modeling Language, Fort Collins, CO, USA, 1999, pp. 249-64.
- [17] Object Management Group, "Unified Modeling Language Specification V. 1.5", Object Management Group, 2003, [www.omg.org](http://www.omg.org), last accessed 2004-10-01.
- [18] Kuzniarz L. and Staron M., "On Practical Usage of Stereotypes in UML-Based Software Development", In the Proceedings of Forum on Design and Specification Languages, Marseille, 2002, pp. 262-270.
- [19] Uhl A. and Lichter H., "A UML Variant for Modeling System Searchability", In the Proceedings of Object Oriented Information Systems, Montpellier, 2002, pp. 199-211.
- [20] Fagan M. E., "Design and Code Inspections to Reduce Errors in Program Development", *IBM Systems Journal*, vol. 15, 1976, pp. 182-211.
- [21] Porter A. A., Votta L. G., Jr., and Basili V. R., "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment", *Software Engineering, IEEE Transactions on*, vol. 21, 1995, pp. 563-575.
- [22] Thurn C., "Verification and Validation of Object Oriented Software Design: Guidelines on How to Choose the Best Method", Master Thesis (MSE-2004-05), Ronneby, Sweden, Blekinge Institute of Technology, 2004.
- [23] Staron M., "Experiment on the Role of Stereotypes in UML Based Software Development - Materials", 2003, <http://www.ipd.bth.se/mst/Experiment/index.html>, last accessed 2003-09-03.
- [24] Staron M., Kuzniarz L., and Wallin L., "A Case Study on Industrial MDA Realization - Determinants of Effectiveness", *Nordic Journal of Computing*, vol. 11, 2004, pp. 254-278.
- [25] Walpole R. E., *Probability & Statistics for Engineers & Scientists*, 7th ed. Upper Saddle River, NJ, Prentice Hall, 2002.
- [26] Wohlin C., Runeson P., Höst M., Ohlsson M. C., Regnell B., and Wesslén A., *Experimentation in Software Engineering: An Introduction*. Boston MA, Kluwer Academic Publisher, 2000.