



H. Eichelberger, Y. Eldogan, K. Schmid

A Comprehensive Analysis of UML Tools, their Capabilities and their Compliance

August 2011

Version 2.0

IfI Report Nr. 1/2012

Software Systems Engineering • Institut für Informatik
Universität Hildesheim • Marienburger Platz 22 • D-31134 Hildesheim

Executive Summary

The Unified Modeling Language (UML) has been widely adopted in software engineering practice in industry over the last decade. A large range of different tools have been developed both from industrial providers as well as open source tools. However, due to the complexity of the UML specification it is very difficult for a single tool to support the full range of UML standards faithfully. In practice tools differ significantly in terms of the parts of the UML specification they support.

The decision of a development organization for a specific tool environment is a very significant one, which typically leads to a significant amount of costs (e.g., in terms of training), out of which the licensing costs, though significant, are often only a minor part. Thus, the tool needs to be chosen with very much care. As one aid towards a rational tool selection decision, this report provides the most comprehensive comparison of UML tools currently available.

This study provides the most comprehensive assessment of the current tool support. It combines a detailed information collection effort for each individual tool, a broad range of tools analyzed and also analyzed UML compliance as defined by the OMG. Other tool evaluations often just list the provided types of UML diagrams without discussing the quality of the realization or focus on specific aspects, e.g. the evaluation approach itself or cross-tool compatibility without thoroughly considering the UML specification itself.

The focus of this study is on the availability of the defined UML capabilities. This has been evaluated in a very comprehensive and detailed way for all UML tools that we found currently to be available and which are still supported. A detailed breakdown that relates for each identified tool its capabilities to the diagram features defined by the UML enables a fast analysis regarding the applicability of a tool for a specific development context. This study focuses mainly on the modeling support. Further aspects like usability, model export and interchange and code generation are also addressed, but take a second place in this analysis.

In order to support the selection of tools, the UML already introduced the so-called *UML compliance*. This provides different compliance levels that can be used to categorize the UML capabilities of a tool according with respect to the standard. As part of our analysis, we also characterize the tools in terms of their UML standard compliance as defined by the OMG.

As part of this study, we identified approximately 200 tools claiming UML modeling functionality, out of which 72 were analyzed thoroughly, here as a reevaluation of updated or new tools one year after publishing the first edition in 2009 [12]. The remaining tools were not considered for evaluation due to technical reasons, e.g. the vendor or the tool do not exist anymore, the tool cannot be installed or no maintenance was done for the particular tool since the first version of UML 2 was published. In summary, only the result of 62 tools could be reported, as the licensing terms of four tools prohibited publication and we could not achieve (so far) an agreement with the tool providers. As a basis for the evaluation, the UML specification was decomposed into 540 features, each of which was individually evaluated for each of the tools. In this report we only report aggregated values. The full detail was too comprehensive to provide in written form. Thus, the full details per tool can be found at the corresponding web site.

Keywords: tool capabilities, feature, UML, UML 2.0, UML compliance, UML modeling tools, tool evaluation, compliance level

Disclaimer

The authors have made every effort to ensure the accuracy of the survey data and additional information (e.g. price, availability). However, no guarantee of accuracy, completeness or fitness for a particular purpose can be given. We do not accept any responsibility or liability in regard of the reliance on, or use of, such data and information.

Index

Executive Summary	I
Index III	
List of Figures.....	VI
List of Tables	IX
List of Abbreviations	X
1 Introduction.....	1
2 Related Work	4
3 Methodology	7
3.1 Collecting Tools for the Evaluation.....	7
3.2 Evaluation Criteria.....	7
3.3 Evaluation Procedure.....	10
3.4 Data Aggregation.....	13
3.5 Planning and Conducting the Evaluation.....	13
4 UML Compliance.....	15
4.1 Introduction to UML Compliance	15
4.2 Object Constraint Language	16
4.3 Compliance Level Calculation.....	17
5 Findings per Tool.....	19
5.1 Example Tool Characterization	23
5.2 Apollo by Gentleware.....	26
5.3 Arcstyler by Interactive Objects	28
5.4 ArgoUML by University of California, Irvine	30
5.5 ARIS UML Designer by IDS Scheer.....	32
5.6 Artisan Studio by ARTiSAN	34
5.7 Artiso VisualCASE by VisualCASE	36
5.8 Astade by Astade Team at Tigris.....	38
5.9 Blueprint Software Modeler by @-portunity B.V.	40
5.10 BOUML by BOUML Team	42
5.11 Cacao by Nulab Inc.	44
5.12 Cadifra UML Editor by Adrian & Frank Buehlmann.....	46
5.13 Concept Draw by Computer Systems Odessa	48
5.14 Delphia Object Modeler by Atos Origin.....	50
5.15 Dia by Alexander Larsson	52

5.16	Eclipse UML by Eclipse.org	54
5.17	Enterprise Architect by Sparx Systems.....	56
5.18	Essential Modeler by Jaczone	58
5.19	eUML2 Studio by Soyatec	60
5.20	Fujaba by Universities of Paderborn, Kassel, Siegen and Darmstadt.....	62
5.21	Gaphor by Gaphor Team at Sourceforge	64
5.22	Gliffy by Gliffy, Inc.	66
5.23	Green UML by University of Buffalo.....	68
5.24	Ideogramic UML by Ideogramic	70
5.25	Innovator by MID	72
5.26	Javelin by Step Ahead software	74
5.27	Jude by ChangeVision	76
5.28	MagicDraw by NoMagic	78
5.29	MEGA development by MEGA international	80
5.30	MetaEdit+ by MetaCase.....	82
5.31	Metamill by MetaMill.....	84
5.32	MyEclipseIDE by genuitec	86
5.33	Netbeans by SUN.....	88
5.34	Objecteering/UML Free Edition by Objecteering Software	90
5.35	objectiF by Microtool.....	92
5.36	Omondo UML Plugin for Eclipse by Omondo	94
5.37	OODesigner by Tae Gyun Kim	96
5.38	Open ModelSphere by Grandite.....	98
5.39	OpenAmeos by ScopeSET.....	100
5.40	Papyrus UML by Papyrus UML Team	102
5.41	Poseidon by Gentleware.....	104
5.42	PowerDesigner by Sybase.....	106
5.43	QuickUML by Excel Software	108
5.45	Rational Software Architect for WebSphere by IBM/Rational	110
5.46	Rational System Architect by IBM/Rational	112
5.47	Rational Tau by IBM/Rational.....	114
5.48	Real Time Developer Studio by Pragmadev	116
5.49	Rhapsody by Telelogic.....	118
5.50	Rose by IBM/Rational	120
5.51	Smartdraw by Smartdraw.....	122
5.52	StarUML by Star UML Development Group	124
5.53	System Architect by Telelogic	126
5.54	Tangible Architect by Tangible engineering	128

5.55	Telelogic Tau/Modeler Edition by Telelogic.....	130
5.56	Teuta by University of Vienna.....	132
5.57	Together by Borland.....	134
5.58	Topcased by Topcased.org.....	136
5.59	Umbrello UML Modeler by Umbrello Project Team at Sourceforge.....	138
5.60	UMLAUT by IRISA/CNRS.....	140
5.61	UMLDiagrammer by Pacestar.....	142
5.63	UMLed by Georg Kubitz.....	144
5.64	UMLet by UMLet group (University of Vienna).....	146
5.65	UMLPad by Luigi Bignami.....	148
5.66	UMLStudio by Pragsoft Cooperation.....	150
5.67	Umodel by Altova.....	152
5.68	violet by C. S. Horstmann and A. de Pellegrin.....	154
5.69	Visible Analyst by Visible Systems.....	156
5.70	Visio by Microsoft.....	158
5.71	Visio with UML2 stencils by Pavel Hruby.....	160
5.72	Visual Paradigm for UML by Visual Paradigm.....	162
5.73	Visual Studio Ultimate by Microsoft.....	164
5.74	Visual UML by Visual Object Modelers Inc.....	166
5.75	yED by yWorks.....	168
6	Results.....	170
6.1	Realization of the UML at Large.....	171
6.1.1	Tool Capabilities.....	171
6.1.2	UML Compliance.....	172
6.1.3	Additional Aspects.....	173
6.2	Realization of the UML on Diagram Level.....	174
6.3	Additional Information.....	178
6.4	Comparison to the last study.....	179
7	Summary.....	181
A	Feature Hierarchy.....	183
B	Reference Diagrams.....	192
	References.....	198

List of Figures

Figure 1: Overview of UML and its diagrams	1
Figure 2: Derivation and aggregation of features.....	8
Figure 3: Excerpt from the feature hierarchy created by analyzing the UML specification.	9
Figure 4: Reference Class Diagram as given in the evaluation instructions.	11
Figure 5: Reference Use Case Diagram as given in the evaluation instructions.....	12
Figure 6: Excerpt from the spreadsheet used to collect the feature data of all tools.	13
Figure 7: Integration of compliance levels with tool capabilities.	17
Figure 8: UML Class Diagram in "Apollo".	26
Figure 9: UML Class Diagram in "Arcstyler"	28
Figure 10: UML Use Case Diagram in "ArgoUML"	30
Figure 11: UML Use Case Diagram in "Artisan Studio"	34
Figure 12: UML Class Diagram in "Artiso VisualCASE"	36
Figure 13: UML Class Diagram in "Astade"	38
Figure 14: UML Use Case Diagram in "Blueprint Software Modeler"	40
Figure 15: UML Class Diagram in "BOUML"	42
Figure 16: UML Class Diagram in "Cacoo"	44
Figure 17: UML Class Diagram in "Cadifra UML Editor".....	46
Figure 18: UML Class Diagram in "Concept Draw"	48
Figure 19: UML Class Diagram in "Delphia Object Modeler"	50
Figure 20: UML Use Case Diagram in "Dia"	52
Figure 21: UML State Chart Diagram in "Eclipse UML".....	54
Figure 22: UML Class Diagram in "Enterprise Architect"	56
Figure 23: UML Use Case Diagram in "Essential Modeler"	58
Figure 24: UML Class Diagram in "eUML2 Studio".	60
Figure 25: UML Class Diagram in "Fujaba"	62
Figure 26: UML Use Case Diagram in "Gaphor"	64
Figure 27: UML Use Case Diagram in "Gliffy"	66
Figure 28: UML Class Diagram in "Green UML"	68
Figure 29: UML Class Diagram in "Ideogramic"	70
Figure 30: UML Use Case Diagram in "Innovator"	72
Figure 31: UML Class Diagram in "Javelin"	74
Figure 32: UML Class Diagram in "Jude"	76
Figure 33: UML Class Diagram in "MagicDraw"	78
Figure 34: UML Class Diagram in "MEGA development"	80

Figure 35: UML Class Diagram in "MetaEdit+"	82
Figure 36: UML Class Diagram in "Metamill"	84
Figure 37: UML Class Diagram (incomplete) in "Netbeans"	88
Figure 38: UML Use Case Diagram in "Objectteering/UML Free Edition"	90
Figure 40: UML Class Diagram in "Omondo UML Plugin for Eclipse"	94
Figure 41: UML Class Diagram in "OODesigner"	96
Figure 42: UML Use Case Diagram in "Open ModelSphere"	98
Figure 43: UML Use Case Diagram in "OpenAmeos"	100
Figure 44: UML Class Diagram in "Papyrus UML"	102
Figure 45: UML Class Diagram in "Poseidon"	104
Figure 46: UML Class Diagram in "PowerDesigner"	106
Figure 47: UML Class-Diagram in "QuickUML"	108
Figure 48: UML Use Case Diagram in "Real Time Developer Studio"	116
Figure 49: UML Class Diagram in "Smartdraw"	122
Figure 50: UML Use Case Diagram in "StarUML"	124
Figure 51: UML Class Diagram in "Tangible Architect"	128
Figure 52: UML Class Diagram in "Teuta"	132
Figure 53: UML Class Diagram in "Together"	134
Figure 54: UML Use Case Diagram in "Topcased"	136
Figure 55: UML Class Diagram in "Umbrello UML Modeler"	138
Figure 56: UML Class Diagram in "UMLAUT"	140
Figure 57: UML Class Diagram in "UMLDiagrammer"	142
Figure 58: UML Class Diagram in "UMLed"	144
Figure 59: UML Class Diagram in "UMLet"	146
Figure 60: UML Class Diagram in "UMLPad"	148
Figure 61: UML Use Case Diagram in "UMLStudio"	150
Figure 62: UML Class Diagram in "UModel"	152
Figure 63: UML Use Case Diagram in "violet"	154
Figure 64: UML Class Diagram in "Visio"	158
Figure 65: UML Class Diagram in "Visio with UML2 stencils"	160
Figure 66: UML Use Case Diagram in "Visual Paradigm for UML"	162
Figure 67: UML Class Diagram in "Visual Studio Ultimate"	164
Figure 68: UML Class Diagram in "Visual UML"	166
Figure 69: UML Class Diagram in "yEd"	168
Figure 70: Summary on the reasons of not considering a tool in the evaluation.	170
Figure 71: Overview of the realization of tool capabilities.....	171

Figure 72: Summary of the realized features grouped according to UML compliance.	172
Figure 73: Summary of the assigned compliance levels.	172
Figure 74: Realization of selected UML features.....	174
Figure 75: Realization of UML features for class diagrams.	175
Figure 76: Realization of UML activity diagrams.	175
Figure 77: Realization of UML sequence diagrams.....	176
Figure 78: Realization of UML state machine diagrams.....	177
Figure 80: Summary of the assigned compliance levels.	178
Figure 81: Additional Reference Class Diagram.....	192
Figure 82: Reference Instance Diagram.....	192
Figure 83: Reference Component Diagram.....	193
Figure 84: Reference Deployment Diagram.....	193
Figure 85: Reference Activity Diagram	194
Figure 86: Reference Sequence Diagram.....	194
Figure 87: Reference Communication Diagram.....	195
Figure 88: Reference State Machine Diagram	195
Figure 89: Reference Timing Diagram.....	196
Figure 90: Reference Interaction Overview Diagram	196

List of Tables

Table 1: Relation between UML compliance levels and notation in this report.....	16
Table 2: Summary of all tools evaluated in this study.	22
Table 3: Example feature fulfilment table for a tool.	23
Table 4: Example tool characteristics and compliance level summary.....	24

List of Abbreviations

CASE	Computer Aided Software Engineering
CDDL	Common Development and Distribution License
DTD	Document Type Definition, ancient form of describing the syntactical validity of XML files
EPL	Eclipse Public License
GPL	GNU Public License
LGPL	GNU Lesser Public License
MDA	Model Driven Architecture as invented by the OMG
OCL	Object Constraint Language
OMG	Object Management Group
QVT	Query View Transform
UML	Unified Modeling Language, Trademark of the Object Management Group
UML 1.x	Unified Modeling Language, one of the earlier versions, e.g. UML 1.4
UML 2	Unified Modeling Language, version 2
XMI	XML Metadata Interchange
XML	Extended Markup Language

1 Introduction

The OMG specifications on the Unified Modeling Language (UML) are widely used in software engineering, e.g. for the design and the documentation of software systems [9, 15]. A large number of tools currently support these standards, often claiming the full support (or support of major parts) of the OMG standards.

In the past, UML was often criticized as being too large to be implemented as a whole, too complex to be realized in detail or as being specified weakly [10, 23, 19]. Consequently, if the tool being used does not sufficiently adopt the specification, e.g. when required model elements are not supported or when incompatible formats prevent migration or further processing, problems while selecting, modeling, upgrading or migration may occur. Thus, to gain a realistic impression on the state of the implementation of UML in current tools, information on the concrete realization of individual modeling elements and the (cross-) compatibility in terms of model persistence is needed

Combining several tools to a tool chain, i.e. to exploit the modeling information in the sense of Model-Driven Software Engineering, increasingly receives attention both from the industry and from research. In particular the OMG specifications on UML and the initiative on the Model-Driven Architecture (MDA) vision had a significant impact both on CASE-tool providers and industrial practice [22]. The MDA approach relies on the transfer of models among tools often produced by different vendors and thus requires as much as possible compliance of the various tool implementations with the underlying (UML) specifications.

In this report we analyze the capabilities of current professional modeling tools with respect to their realization of the UML specification. The results have a strong impact on the applicability of model-driven software engineering in practice, in particular along the lines of the Model-Driven Architecture initiative. The study underlying this report is designed to provide detailed information to decision makers in industry and research. Tool comparisons published so far often provide only generic high-level information, e.g. the price of a tool, the supported diagram types or the version of XMI implemented by a tool. Without detailed information, e.g. on the realized model elements, a decision maker is often forced to re-evaluate a selected number of tools to determine the appropriate one. Thus, we decided to derive our evaluation criteria in a systematic process by analyzing the UML specification for required modeling constructs. Relying on the data provided by our study, the decision making process can be shortened and simpli-

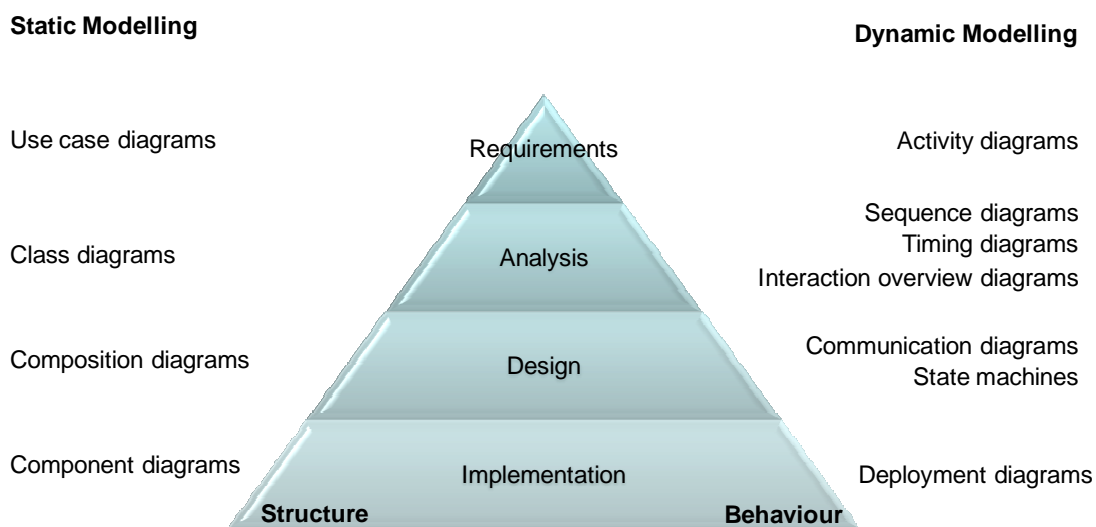


Figure 1: Overview of UML and its diagrams

fied significantly. Similar decisions but naturally from a different perspective are made in research and academia, in particular when the topic is on cutting-edge issues of model-driven engineering and detailed UML support is needed. A preference-driven tool selection process e.g. by weighting individual diagram elements or diagrams for importance is out of the scope for this report.

We focus in this report on the assessment of UML tools with respect to the implementation of modeling concepts as defined in the UML specification. An overview of the diagram types as the basic building blocks of the UML is depicted in Figure 1. We refer to the implementation of the UML in terms of modeling concepts defined by the UML as the *capabilities of a tool*. Using the data on individual modeling concepts describing the capabilities of a tool, we can determine the conformance in terms of UML compliance levels as defined by the OMG. We refer to this type of aggregation of our data from the viewpoint of the OMG as *UML compliance* or as *UML compliance levels*.

In contrast to existing UML tool evaluations, we describe in this report a feature-based evaluation approach capturing the capabilities defined by the OMG specifications. Based on a systematic internet search including well-known tool lists, we scheduled approx. 200 tools for evaluation. Due to technical problems, e.g. because tools are not available anymore, the initial schedule was reduced to 72 UML tools. This detailed evaluation was executed on a sample set of 72 UML tools, covering – in particular – all major professional tools that are widespread in practice. The outcomes of our approach are detailed tool characterizations including individual capabilities in terms of fulfillment degrees and statements on the UML compliance. The aggregated data forms the basis for a comparative analysis of the results, e.g. fulfillment of diagram types over all tools, fulfillment of diagram specific model elements, assigned UML compliance levels, status of the implementation of XML, etc. By aggregating the data, we provide an encompassing overview on the current state of the realization of the UML standards. Another important aspect for research and academia is the consistent application of an evaluation approach based on a specification to a large set of tools.

The remainder of this report is structured as follows: In the next section, we review related work on tool comparisons and research approaches for tool evaluations. In Section 3 we present our evaluation methodology for tool capabilities based on a detailed analysis of the UML specification. Then, in Section 4 we introduce compliance from the viewpoint of the OMG, describe the compliance evaluation schema as defined in the UML specification and combine the UML compliance with our evaluation methodology. Section 5 is devoted to the discussion of the individual findings on each evaluated tool, i.e. we present our results on feature fulfillment and UML compliance and a screen shot for each tool. In Section 6 we discuss the overall results of our study, i.e. the aggregated findings on UML and diagram level. Finally, in Section 7 we draw conclusions and list future work. The appendix lists all criteria we used for collecting data on individual tools.

The detailed findings of all feature groups for all tools not being restricted for publication by the respective vendor can be found on <http://www.sse.uni-hildesheim.de/UMLtools11/>

Reader's guide:

Dependent on your personal interests, the parts of this report may be of different interests for you. This reader's guide is a suggestion for selecting the relevant parts and, thus, gaining the maximum value from this report.

- **Tool vendor:** Tool selection (Section 3.1), evaluation criteria (Section 3.2), criteria aggregation (Section 3.4), UML compliance definition (Section 4.1), calculation of UML compliance levels from the evaluation criteria (Section 4.3), summary of tool findings (introduction to Section 5), description on interpreting individual tool summaries (Section 5.1), individual findings for your tool and competing tools (further subsections in Section 5), results and analysis (Section 6), Appendix A containing the complete list of evaluation criteria.
- **Tool user:** Evaluation criteria (Section 3.2), UML compliance definition (Section 4.1), high-level comparison of used tools (introduction to Section 5), description on interpreting individual tool findings (Section 5.1), individual findings for tools being relevant to you using the data for a preference-based comparison excluding parts of the UML being irrelevant to you (further subsections in Section 5), results and analysis (Section 6).
- **Decision maker:** Evaluation criteria (Section 3.2), UML compliance definition (Section 4.1), summary of tool findings to narrow your search (introduction to Section 5), description on interpreting individual tool findings (Section 5.1), individual findings for tools being relevant to you using the data for a preference-based comparison excluding parts of the UML being irrelevant to you (further subsections in Section 5), results and analysis (Section 6)
- **Researcher:** Related work (Section 2), Evaluation methodology (Section 3), UML compliance definition (Section 4.1), UML compliance level definition and calculation (Section 4), summary of tool findings (introduction to Section 5), description on interpreting individual tool summaries (Section 5.1), results and analysis (Section 6), Appendix A for the complete list of evaluation criteria

For all readers we suggest to read the overall summary in Section 7.

2 Related Work

The normative set of syntactic and semantic rules to be considered when discussing and realizing UML tools is the UML specification, i.e. the current version of the UML 2 specification, i.e. the UML Infrastructure [32], the UML superstructure [33], the XMI specification [31] and the diagram interchange specification [29].

In literature, several evaluations and studies on UML tools are published. In this section we provide guidance to other relevant comparisons of UML tools. This section is structured according to the different perspectives taken by the authors of the studies in literature:

- Evaluations of UML tools for professional use, i.e. data summaries for decision makers in professional environments.
- Evaluation approaches from the research perspective, UML tools were evaluated from different viewpoints, e.g. from the model validation perspective or from the viewpoint of usability or communication efficiency.
- Evaluations of specific aspects, e.g. model exchange formats, in particular XMI.

Decision makers in professional environments can find information on several websites, e.g. the OMG vendor directory listing [34], the Wikipedia page on UML tools [5] or several others like [1, 2, 4]. More detailed lists contain prices of individual tools [3] or provide matrix views on up to 100 different tools and their high-level functionality [16, 35], e.g. supported types of UML diagrams, data formats, target languages for code generation or required operating system platforms. Twelve tools (Artisan Studio, Eclipse UML, Enterprise Architect, Magic Draw, Innovator, Neuland Boardmaker, Poseidon, Rational Systems Developer, Rhapsody, Vision Stencils for UML, Visual Paradigm, Together for Eclipse) are compared in [6]. An extended abstract on the results are given in [39]. In [25], four tools (Rational Rose, Visual UML, Poseidon, XUML) were evaluated according to similar criteria and some shortcomings on each tool are identified.

So far, all listings or studies mentioned above describe UML tools on a rather generic level, i.e. by only enumerating if high level functionality like diagram types are realized. As a result, an in-depth discussion of the compliance with the UML specification is also in the case of commercial studies not provided. Often the information provided is not sufficient for a decision maker, because the realization of individual model elements is not described, and thus, a fine grained decision cannot be made and often own evaluations of selected tools must be conducted. Furthermore, there is a significant difference in the quality of the information provided, i.e. whether data is collected from vendor statements or by directly analyzing a tool implementation as done in our study. In particular, we provide a detailed feature-based analysis of UML modeling elements which is not provided elsewhere.

Research work in this area focuses often specifically on the evaluation approach itself or on very specific topics being evaluated. The main research topics to be discussed in the next paragraphs are: evaluation frameworks for UML tools, metrics-based evaluation, readability and understandability of diagrams as well as communication and coordination aspects.

A general **evaluation framework** is proposed in [18]. The authors consider also economic issues, metrics support or the documentation, and, in particular the usability and the look and feel of the tools. The evaluation framework contains 29 specific criteria for all types of UML 1.3 diagrams. The authors describe in [17] the application of the evaluation framework to concrete tools in the project EvaLUM. So far, seven tools are evaluated (ArgoUML, Describe Enterprise, Elixir CASE, Rational Rose Enterprise, Simply Objects Modeler, Tau UML Suite, Together Control Center). The authors made a restriction for seven tools, be-

cause the complexity of the individual tools caused an enormous effort in applying the evaluation framework. The authors provide results on the concrete compliance of the tools with the UML specification.

A **hierarchical approach** to the evaluation of UML tools is given in [8]. By applying the Logic Scoring of Preference (LSP) method, a list of desired characteristics is constructed, concrete software products are evaluated upon these characteristics and finally the products are rated by criterion functions. The characteristics are determined following a hierarchical decomposition process for requirements derivation. Aside from “complete UML support”, also forward and reverse engineering for different target languages, generation of HTML documentation, model export via XMI, versioning and navigation are mentioned as high-level criteria. Some more detailed criteria are considered in [14], e.g. semantic associations among modeling elements and an object, OCL support, design patterns, customization opportunities, tool extensions and UML extensibility mechanisms. Similar to the evaluations published for decision makers listed above, the evaluation in [8] is on diagram level only, no concrete results on the rating of the tools are presented, and only few criteria are derived from UML itself. Thus, no concrete compliance conclusions with respect to the UML specification were directly drawn. Furthermore, the work in [8, 14] relied on an earlier version of UML 1.x.

In the visualization and diagramming communities, in particular the **understanding of diagrams** is a major research topic. In [38], a set of 14 rules selected according to the laws of perception is used to classify concrete diagrams as well as to evaluate three concrete tools (Wampler, Rose, Together). As a result, Rose and Together considered most of the rules in the automatic layout of simple class diagrams. A more encompassing evaluation on automatic layout of class diagrams is presented in [11, 13] where 42 UML tools are analyzed. The authors report that only few tools are able to model the test diagram and most tools have serious problems in automatic layout. Even if the main topic in our work is not perception or (automatic) layout, we collect qualitative statements on the automatic layout facilities of the evaluated tools and discuss the results in Section 6.

From the viewpoint of **effective communication and coordination** across geographically disparate sites, in [20] several criteria including UML support on diagram level, round trip engineering, model consistency checking, forward and reverse engineering etc. The results were analyzed by calculating a weighted sum of the results per tool. While performing a 25 person days evaluation, Together 5.0, Rose 2001, Embarcadero Describe 2001 were applied to an in-house application consisting of 200 Java classes. According to the evaluation schema, Together reached 95%, Rose 58% and Describe 46% of the considered features.

In contrast to the publications for decision makers discussed above, most of the evaluations in research approaches consider rather small sample sets. This is not only a drawback of these studies from a practical perspective, but also from a research perspective, as it also leads to an insufficient validation of the evaluation frameworks themselves.

Although work on specific compliance aspects can be found in literature, e.g. on **model interchange** using XMI [31]. In [21], the authors describe the results of cross-tool compatibility tests with Eclipse-based tools (Together Architect, Eclipse UML, Rational Software Architect, Magic Draw, Altova UModel). Only two of the tool combinations are able to pass the backward compatibility test on the same tool and only two very specific tool combinations (Together/Eclipse UML/Rational and MagicDraw/UModel) are able to interchange their models. In [36], the authors focus on the differences among open source and commercial tools. 6 combinations out of 9 tools (ArgoUML, Fujaba, Umbrella, Artisan RealTime Studio, Poseidon, Rhapsody, Rose Enterprise, Tau G2, Visio) are able to interchange models, but the authors did not detect compatibility among open source tools and commercial tools. In [37] the authors discuss in particular the question of being locked-in to a tool by using XMI and missing compatible model ex-

change alternatives to other. They test the XMI compatibility for three open source and commercial tools (ArgoUML, Fujaba, Umbrella) and attest that no exchange is possible at all. Thus, after all these results, the authors judge XMI as not being really adopted by the tool vendors as intended and promoted by the OMG. A drawback of the work on analyzing XMI compliance of only modeling tools is that further tools being used in practice like model transformation or code generation frameworks providing tool-specific XMI filters are not considered at all. Thus, a detailed analysis of the modeling facilities provides relevant information beyond a pure analysis of XMI compliance.

In our work, XMI is considered when describing the individual tool characteristics. We refrain from also performing cross-tool XMI compliance tests, as the analysis described already that cross-tool compliance is mostly not possible. To characterize the XMI compliance, we rely on the results of syntactical validity tests of the produced XMI files. Thereby, the files produced by the tools are validated against the XMI DTDs and XMI Schemas provided by the OMG or, in case that no detailed formal description of the grammar is provided, on sample inspection of the files according to rules based on the test suite published by the OMG model interchange working group (MING) [24]. Even if there are some efforts in validating XMI files, currently no functional working tool is available. Particularly, the online XMI validator tool maintained by the National Institute of Standards [26] does not work properly with arbitrary XMI files and seems to focus on the MING test suite.

The proper adoption of a commonly accepted model exchange format would also provide the technical foundation for automated compliance tests. In [7], the authors describe the Java-UML Lightweight Enumerator, a **test suite generator for models**. For a given test-model the violation of OCL wellformedness rules in the implementation of a certain tool is checked. The test generation approach targets the validity of the abstract syntax, its completeness in terms of model elements and the conformance to the semantics (wellformedness rules) as realized by concrete UML tools based on importing the generated models. Currently, it is unclear how many test models are required to cover a concrete UML version and the usability as well as the concrete syntax, i.e. the diagramming language is not taken into account at all.

If compared to existing work, the study we provide in this report has the following benefits: Most tool comparisons offer rather generic information, e.g. the provided types of UML diagrams, or rely on a small tool sample set. Regarding the level of detail, we characterize the abilities of individual tools by collecting information on modeling features defined by the UML specification. By aggregating that data, we obtain detailed tool characterizations on the high-level facilities, e.g. on the fraction of features implemented for a certain diagram. This leads to a more realistic description of the tools. Regarding the sample set, we initially scheduled approx. 200 UML modeling tools for evaluation instead of selecting an arbitrary subset as done in several studies we screened. Due to the fact that many of these tools are not available anymore, we carried out an exhaustive feature-oriented evaluation on the subset of 72 available tools.

The next chapter will discuss how we improved over other available studies.

3 Methodology

We use an empirical approach to evaluate the capabilities of the tools, because for only few tools the implemented meta-model can directly be inspected, e.g. as source code. Thus, we derive a hierarchical feature structure from the UML specification to describe all required modeling features. To assess the realization of the UML syntax, i.e. the UML diagramming language, we collect data on realization of the modeling features by installing and using the concrete tool. In this chapter we focus on the systematic assessment of capabilities, i.e. the implementation of features required by the UML specification. The aggregation of the data gained with the hierarchical structure with respect to UML compliance is discussed in the next chapter.

This chapter is structured as follows: In Section 3.1 we outline how tools were collected for evaluation. In Section 3.2 the evaluation criteria in terms of a hierarchical feature structure as derived from the UML specification. In Section 3.3 we introduce our evaluation procedure. Then, in the following section, we give the data aggregation strategy, i.e. how to aggregate the collected data to gain an overview e.g. of all tools for an individual diagram. Finally, in Section 3.5 we describe the overall course of the evaluation.

3.1 Collecting Tools for the Evaluation

We aimed at conducting an exhaustive evaluation including all tools covering industrial as well as research interests. To gain a list of candidate tools we carried out the steps listed below:

1. We collected sales information (name, vendor, URL) of all tools published in different tool listings or comparisons [1, 2, 16, 3, 34, 35, 4, 5, 6, 39]. Even if it was obvious that several tools were not available anymore or non-modeling tools e.g. diagramming tools were mentioned, we added them to the list.
2. We completed the list by an exhaustive internet search so that tools not added to list in step 1 were also considered.
3. We consolidated the list by removing duplicates.

Following these three steps, initially approx. 200 tools were scheduled for evaluation. As one obvious step in evaluating is to obtain the tool, we expected that tools being not available anymore can be identified easily. To document why a tool is not evaluated, we also noted the reason why a tool is not available for evaluation.

3.2 Evaluation Criteria

The main goal of this study is the evaluation of tool capabilities, i.e. the realization of the UML (2.1) modeling capabilities. As this is strongly related with UML compliance as defined in the UML specification and discussed in Section 4, we will use our data on capabilities to draw conclusions on the UML compliance for each individual tool.

A feature hierarchy containing the model elements and relations of each UML diagram is at the heart of our evaluation approach. This feature hierarchy was derived from the UML superstructure to gain the required modeling features. Figure 2 illustrates the top-down process of deriving features from UML language units and the bottom-up data aggregation process.

1. Starting with the UML language units, i.e. mostly the different diagram types, we dissected the relevant chapters of the UML specification for modeling elements, relations and their properties and map this information to individual features. Also the related standard stereotypes summarized in the annex of the UML superstructure were considered as features.

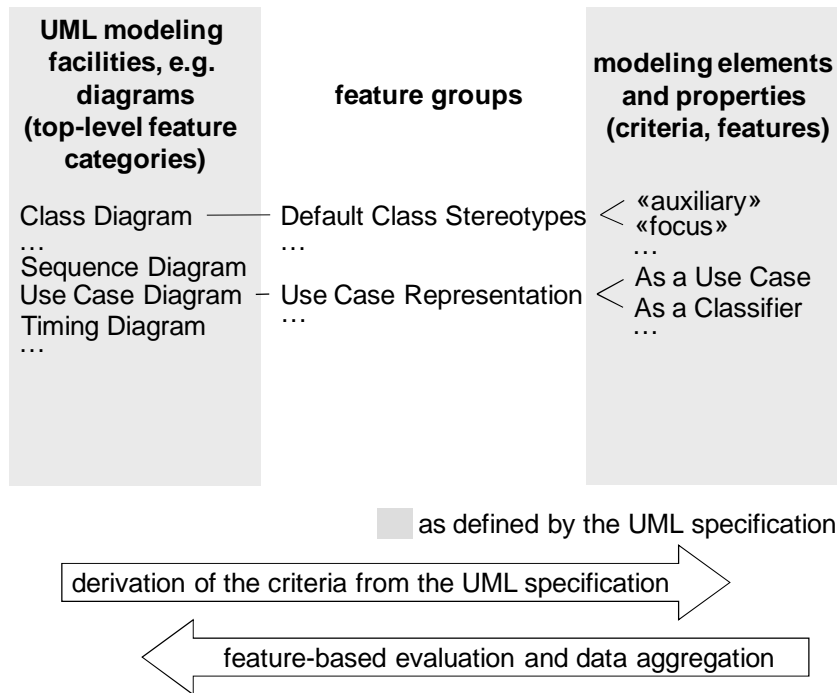


Figure 2: Derivation and aggregation of features.

2. The feature list was then validated against the example diagrams and the particular diagram elements summary given in the UML specification.
3. The initial feature list was reviewed and the features were structured to feature groups, e.g. all features directly related to a use case are grouped together. To simplify the evaluation and the final data analysis, we clustered the 540 features obtained in the last step into 130 feature groups, e.g. all features directly related to the use case modeling element are grouped together. In fact, feature groups are not defined by UML, they were only introduced in our approach to simplify the manual evaluation of the tools and to simplify the presentation of the results.

For the reevaluation we revised the feature hierarchy by further 28 feature groups and 64 features in order to reflect our experience from the last study as well as the most current UML specification (at that point of time version 2.1.1). These new feature groups and features will not be considered in the overall analysis, because they were recorded only for the updated or new tools and, thus, cannot be compared with the results described in [12].

4. Finally, the feature groups are constructed in terms of categories representing the chapters of the UML superstructure, i.e. the main UML language units. Figure 3 depicts an excerpt of the feature hierarchy, in which all top-level categories, the feature groups for use cases and some individual features are shown. The features in the category “model persistence” describe the XMI and DI versions and the result of the structural validation. The entire feature hierarchy used is given in Appendix A.
5. We added further categories not defined by the UML specification to the feature structure in order to collect additional data:
 - Technical Data contains information like vendor name, vendor URL or version of the tool.
 - The category “Traceability” consists of information on links among specific diagrams.

- “Code Generation” collects which target languages are actually offered by the tool implementation, which diagrams are considered in the code generation and how the models influenced the resulting code.

The corresponding hierarchy is shown in Figure 3 on an abstract level. The entire hierarchy is described in Appendix A.

While building the feature hierarchy, we identified three different basic types of features:

1. A *free string feature* is intended for documentation purpose, e.g. the name of a tool or evaluation comments.
2. A *Boolean feature* represents a modeling element or a property of a modeling element, which is required to be realized by a modeling tool. In most cases an evaluator can clearly determine whether a feature is realized and usable in a concrete implementation. In some cases a feature may be not fully functional, e.g. it needs a selection of individual elements which may not be selected in that situation. Such features can be marked as a present but unclear implementation.
3. A *feature group* consists of multiple Boolean features. An example for such a feature group is the use case representation in Figure 3 which consists of two features. Similar to an individual feature, each of the features in the feature group but also the entire feature group can be marked as an unclear implementation.

Following this approach, we obtained 476 individual features assigned to 130 feature groups for the basic feature hierarchy in [12] and 540 features in 158 feature groups for the extended fea-

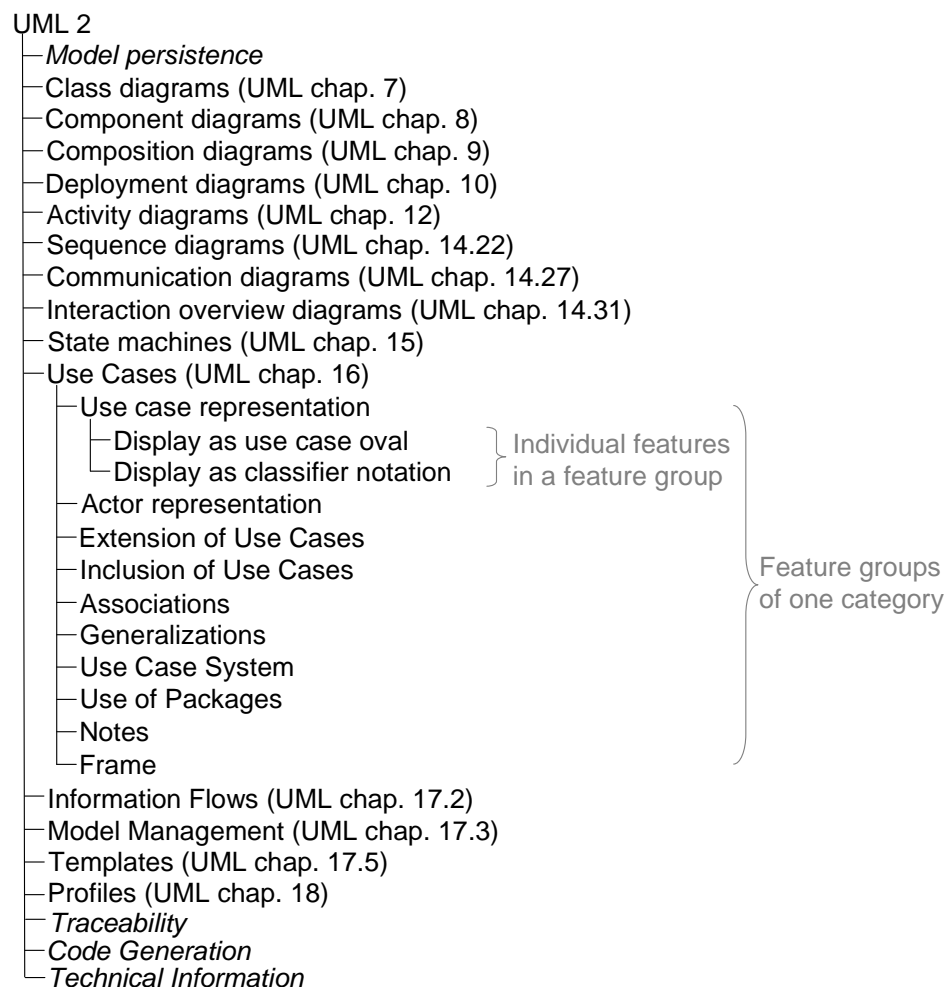


Figure 3: Excerpt from the feature hierarchy created by analyzing the UML specification.

ture hierarchy in this reevaluation. For example, in Figure 3 we grouped all features directly related to the use case modeling elements into one group. The feature groups do not influence the results of the evaluation and are used only to simplify data acquisition and representation.

3.3 Evaluation Procedure

To ensure the objectivity and repeatable evaluation of the tools, we prescribed the course of the evaluation in terms of an evaluation procedure. The procedure described in this Section is an extension of the evaluation in [12] in order to reflect our experiences from the last study and to provide more guideline to the evaluators. Instead of two example reference diagrams, we prescribed reference diagrams for all diagram types defined in the UML. The additional results are collected for the new and changed tools during the evaluation but not considered in the analysis in order to maintain compatibility to our evaluation in [12]. The steps for the evaluation of a particular tool are listed below in sequence:

1. The evaluator obtains the concrete tool implementation from the vendor, i.e. the most current version including as many features as possible (e.g. a so called enterprise version including UML support, model-driven engineering support, code generation, etc.). The evaluator skips the further steps for a tool in case that no version was released by the tool vendor. As a part of this activity, the evaluator performs a registration or a request for an evaluation license. The evaluator registers the reason if the tool is not available at all.
2. The evaluator records the technical data of the tool, e.g. the version number, the exact name of the tool (e.g. if it is a standard or an enterprise version), or the vendor URL. Additionally, the evaluator stores the product description page as found on the internet and registers the price of the tool if available.
3. The evaluator installs the tool into a virtual machine according to the individual installation instructions. As part of the installation he configures the evaluation license, if needed. If any problem occurred while installing the problem so that the evaluation cannot be executed, the evaluator registers the problem. Thereby, the evaluator stores a textual version of the EULA presented by the tool upon installation.
4. The evaluator scans the license for problems on publishing the results of the evaluation and records the result of the analysis in the tool evaluation sheet.
5. The evaluator executes the evaluation. Therefore,
 - a. The evaluator executes the tool and creates a UML model.
 - b. The evaluator tests the required features for each diagram type according to the feature hierarchy. In particular, the evaluator models the following reference diagrams:
 - Class diagram and related Object diagram.
 - Use case diagram
 - Component diagram
 - Deployment diagram
 - Activity diagram
 - Sequence diagram
 - Communication diagram
 - Interaction overview diagram

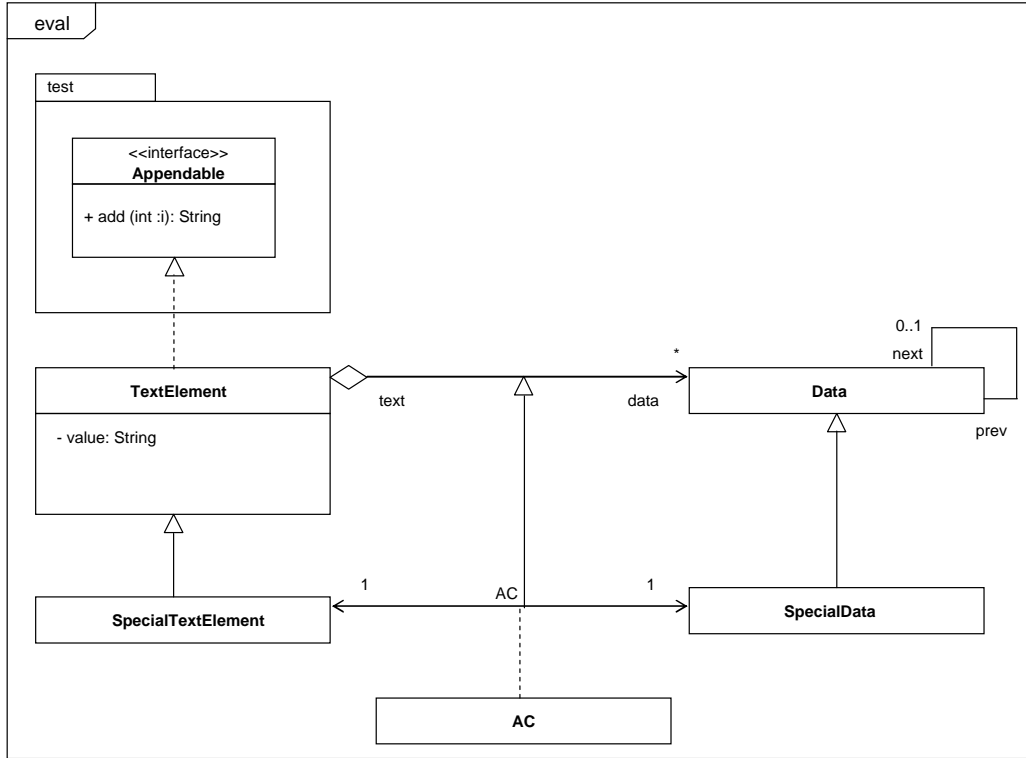


Figure 4: Reference Class Diagram as given in the evaluation instructions. Additionally, the evaluator is requested to add arbitrary comments to classes, relationships and the package.

- State machine diagram
- Timing diagram

The reference examples are intended to collect comparable views on the tools to be presented in the evaluation report and aim at combining most of the features of the diagram type in a meaningful manner. The evaluator is encouraged to add additional comments to classes, components and relationships. The evaluator produces a screenshot for each modeled reference diagrams.

In this section we display only the reference diagrams also used during the evaluation in [12], i.e. the reference class diagram in Figure 4 and the use case diagram in Figure 6. The reference diagrams for the other diagram type defined in the UML are considered as an addition to the evaluation process in [12], particularly to consider references among diagrams and model consistency issues, e.g. whether a class created for the class diagram may be used in diagrams modeled later during the evaluation. The additional diagrams shown in Appendix B are also intended as an additional guideline to the evaluator. In fact, each additional reference diagram implies additional effort during the evaluation. We faced this additional effort, because we evaluated only changed or new versions in this reevaluation.

While performing the evaluation, the evaluator collects information on the feature fulfillment in a spreadsheet prepared according to the feature hierarchy. An excerpt from the spreadsheet is shown in Figure 6.

6. The evaluator collects data on the additional categories “Traceability” and “Code generation”.

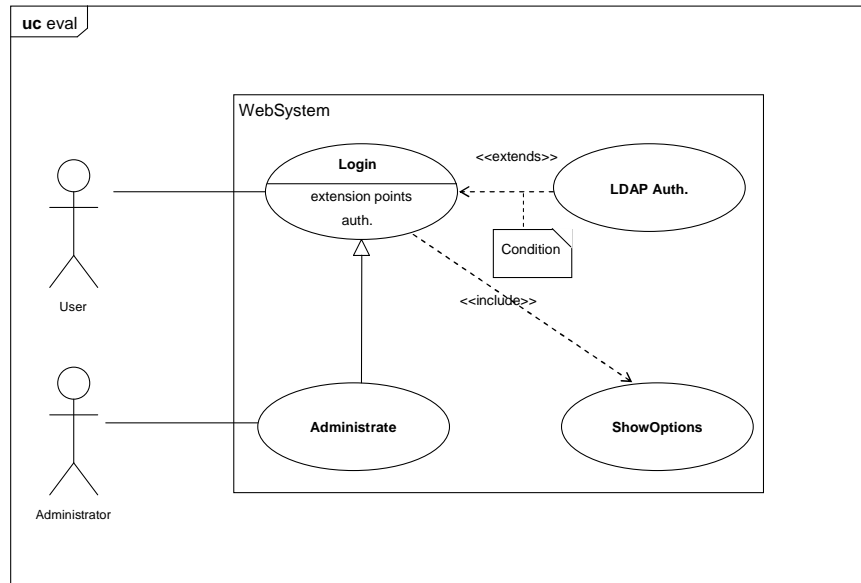


Figure 5: Reference Use Case Diagram as given in the evaluation instructions. Additionally, the evaluator is requested to add arbitrary comments to one actor, one use case and the system.

7. The evaluator saves the model in XMI or DI format if supported by the tool and validates the files by a structural XML validation against the OMG specifications by performing a structural validity test using Altova XMLSpy. The evaluator records the results. In case that an automatic validation cannot be performed because of the of recent grammar specifications for the recent versions of XMI, the evaluator tests the wellformedness of the XML file using Altova XML Sply and performs a sample testing of the written model. Therefore the evaluator searches for the classes, association classes, actors and use cases as well as their relationships specified in the reference diagrams and compares the serialization of the tool with the similar structures in the reference XMI files provided by the OMG MING working group [24]. In some cases, e.g. for association classes, the UML allows alternative form of serializing a model. Then the evaluator consults the UML specification and validates the alternative based on the specification.
8. Additionally, if provided by the tool the evaluator executes the automatic layout mechanisms on all diagrams and records the (personal) impression.
9. The evaluator records personal impressions on the usability (problems selecting elements, small mouse cursor, unintuitive sequences of commands to model certain parts, etc.) and the overall impression of the tool (e.g. general problems, installation problems, helpful support, etc.).

10. Finally, the evaluator validates the collected information by applying a checking tool on the evaluation spreadsheet and stores the spreadsheet along with the files obtained from the tool in the evaluation repository.

3.4 Data Aggregation

To gain an overview, e.g. on the degree of realization for the entire UML or for individual diagrams, the features must be summarized. Aggregations of feature groups, e.g. to determine the fulfillment of a category, are calculated as the average value of the individual features. We apply the following rating schema according to the types of the feature groups:

- Free string features are not considered in the data aggregation.
- For Boolean features 1 was assigned if it was present, 0 if it was absent. In the case of unclarity the entire feature was ranked with 0.5.
- Feature groups are not considered at all.

The result of this data aggregation is called feature fulfillment degree, i.e. the percentage of features being fulfilled. Depending on the subset of features considered for calculating the fulfillment degree, the feature fulfillment of individual UML diagrams (i.e. often chapters of UML) or of the entire UML specification can be expressed.

3.5 Planning and Conducting the Evaluation

To conduct our study, we planned and prepared the evaluation, executed the survey, checked the results for overall consistency and performed an analysis on the collected results. This section describes these steps.

We aimed at conducting an exhaustive evaluation including all tools covering industrial as well as research interests. Thus, we **prepared our study** by listing the relevant tools as described in Section 3.1. In parallel, we extracted the feature hierarchy from the UML specification as described in Section 3.2. Additionally we decided to collect technical information on the tool itself (price, URL, operating system, licensing, Eclipse integration), on data handling (UI concepts, file formats, multi-user access, repository support, versioning mechanisms), the advertised metamodel, the extensibility of the tool, the traceability among diagrams as well as issues on code generation, e.g. whether behavioral code is produced and in which way association kinds are considered while code generation. Additionally, we defined the evaluation procedure.

	A	AU	AV
1	General information		
2	Name	Visual Paradigm for UML	Visual UML
3	vendor	Visual Paradigm	Visual Object Modelers Inc.
4	version	6.1 sp1_20071003	5.26 build 634
143	tabular notation UML Annex E(x)	-	-
144	use cases UML chap. 16		
145	(u)se case, (c)lassifier	u(c)	uc
146	actor (c)lass, (i)con	(c)i	ci
	(e)xtension, extension (p)oint, (c)onstraint in		
147	note	ep	epc
148	include(x)	x	x
149	s(y)stem, (s)tereotype subsystem	y	y
150	(a)ssociation, (m)ultiplicity	a	am
151	generalization (u)secase, (a)ctor	ua	ua
152	in packages or frames(x) {p 593}	x	(x)
	Note (f)ree, (a)ttachable,		
153	(m)ember attachable	f(a)m	fam
154	Frame (u)c	(u)	(u)
155	information flows UML chap. 17.2		
	information (i)tem, (s)tereotype flow,		
156	(b)lack triangle	-	-

Figure 6: Excerpt from the spreadsheet used to collect the feature data of all tools.
The spreadsheet is organized according to the hierarchical feature structure.

Additionally, we realized supporting tools, i.e. for the evaluation results a spreadsheet was prepared containing the evaluation criteria and a special sheet to collect the reasons for omitting tools from the evaluation.

In the second step, we executed the **evaluation** according to the evaluation instructions as described in Section 3.3. The result is a spreadsheet containing all collected data (as shown in Figure 71) and a repository containing of XMI files, XMI validation reports, tool homepage snapshots and screenshots.

Finally, we **analyzed, summarized and reported** the results. The results are discussed in Section 6. For example, we determined the feature fulfillment per diagram type and the concrete UML compliance as discussed in the next Section. We derived several statistics, e.g. on the capabilities in terms of feature fulfillment for all tools. This is described in Section 6.

For each tool we checked the (evaluation) license in order to find out whether the data obtained by the evaluation of the tool may be published. In the case that the evaluator found license statements indicating legal problems in the case of publishing the results, we contacted the vendor for an explicit permission. If a vendor did not permit the publication, this is explicitly stated in the section devoted to the results of the tool.

4 UML Compliance

In this chapter we give a short introduction to the compliance with the UML specification as defined by the OMG. As stated before, UML compliance is an aggregated view on tool capabilities in terms of modeling elements defined by the UML as discussed in the last chapter.

This chapter is structured as follows: In the first section we describe the compliance mechanism as defined in the UML, i.e. the characterization of compliance with the UML in terms of compliance levels, abstract and concrete syntax compliance. In Section 4.2 we discuss the influence of the Object Constraint Language (OCL) on our work. In the last section of this chapter we introduce our approach to determine compliance levels from tool capabilities, i.e. the integration with the feature data collected while conducting the evaluation.

4.1 Introduction to UML Compliance

UML defines two orthogonal dimensions of conformance, the so called **compliance levels** and the **syntax compliance** in both, the UML infrastructure and the UML superstructure. Compliance levels are described in terms of realized modeling constructs and diagrams. The syntax compliance relates to the implementation of the metamodel, the support exchange mechanisms and the realization of the UML notation for the modeling constructs. We discuss the compliance levels and the syntax compliance as defined by the OMG in this section.

The UML infrastructure [32] defines two compliance levels:

- *Level 0* defines entry-level modeling capability based on the UML infrastructure, i.e. pure class-based modeling. It is intended as a low-cost common denominator serving as a basis for interoperability among UML tools.
- *Level LM* extends Level 0 by constructs for the specification of meta models. The class-based modeling language is extended by additional and more detailed information. Furthermore, profiles and model management facilities are required. We consider this level in our evaluation to distinguish among tools providing basic class modeling capabilities and tools also implementing profile and model management features.

The UML superstructure [33] extends the levels in the infrastructure by adding more specific compliance levels.

- *Level 0* equates to the Level 0 defined by the UML infrastructure.
- *Level 1* requires Level 0 and use case, sequence, timing, component, composite structure and activity diagrams as well as full UML class, package and object diagrams.
- *Level 2* adds deployment mechanisms, state machines and profiles to Level 1.
- *Level 3* represents the complete UML, i.e. it extends Level 2 by information flows, templates and model packaging.

The second, orthogonal dimension is the compliance to the UML syntax. It is expressed as

- *Abstract syntax compliance*, i.e. compliance with the UML meta model, the structural relationships and the (well-formedness) constraints. Also the ability for model persistence in XMI format [31] is required.
- *Concrete syntax compliance* requires that the UML notation for the elements and the diagram types is realized as defined in the UML specification.
- *Abstract and concrete syntax compliance* combines both syntax compliance levels mentioned above.

OMG compliance level	Defined in	Notation in this report
Level 0	UML 2 superstructure [33]	L2-0
Level M	UML 2 infrastructure [32]	L2-M
Level 1	UML 2 superstructure [33]	L2-1
Level 2	UML 2 superstructure [33]	L2-2
Level 3	UML 2 superstructure [33]	L2-3
-	UML 1.3 specification [27]	L1.3
-	UML 1.4 specification [28]	L1.4

Table 1: Relation between UML compliance levels and notation in this report.

- *Abstract and concrete syntax compliance with the ability of model and diagram persistence* includes all syntax compliance levels above and requires the implementation of the diagram interchange specification [29].

In the remainder of this report we refer to the compliance levels as L2-0, L2-M, L2-1, L2-2 and L2-3 as shown in Table 1 to also provide a clear naming for tools being compliant to earlier versions of UML [27, 28], i.e. L1.3 and L1.4. To indicate the syntax compliance, we add appropriate abbreviations to the compliance levels. Depending on the realization of XMI, the abbreviations (A) for abstract syntax compliance, (C) for concrete syntax compliance, (AC) for abstract and concrete syntax compliance and (ACP) for abstract and concrete syntax compliance including model and diagram persistence we write e.g. L2-0A.

The compliance levels defined by UML can be integrated into our feature-based evaluation schema in a straight forward manner: As described above, each compliance level requires the implementation of some UML language units, e.g. diagrams. Our evaluation schema is derived from these language units down to features. Thus, the compliance levels can be used as the topmost level for aggregating feature fulfillment degrees as depicted in Figure 7. In fact, the entire UML 2 specification is represented by L2-3.

When determining the concrete compliance level for a tool, in some cases a tool may fulfill multiple levels simultaneously. In fact, L2-M subsumes L2-0 and L2-3 the levels L2-2, L2-1, L2-0 in the given sequence. For the analysis of our findings, we need to assign the maximum compliance level to a concrete tool. Therefore, we assume the following ordinal scale L2-0, L2-M, L1.3, L1.4, L2-1, L2-2, L2-3 so that tools being compliant to earlier versions of UML and only to L2-0 or L2-M are assigned to an appropriate UML 1.x level.

4.2 Object Constraint Language

The Object Constraint Language (OCL) [30] is a model query and constraint specification language aligned with the UML specification. We excluded an OCL assessment from this study, because OCL

- OC is an optional part of the OMG specifications. Even if constraints on the meta model of the UML specification are expressed in OCL, OCL is not a mandatory part of UML. Moreover, as stated in the UML specification, constraints in user specified UML models may be given in OCL but also in terms of a programming language such as Java or in a natural language.

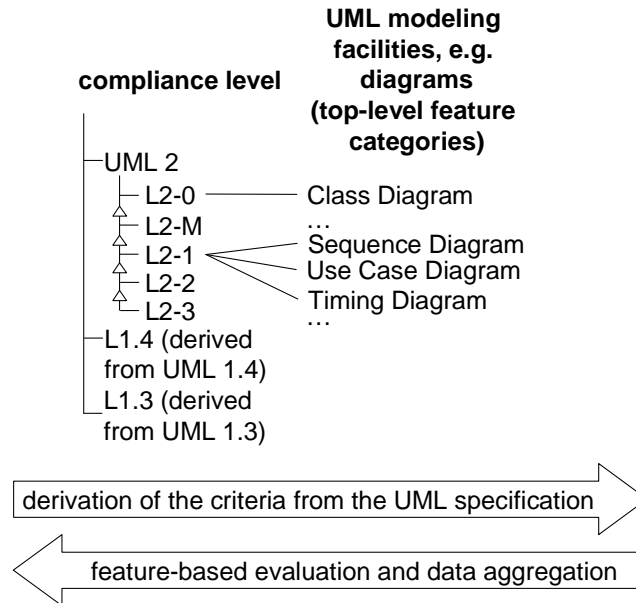


Figure 7: Integration of compliance levels with tool capabilities.

- Evaluating OCL would also require a detailed feature analysis. The OCL specification defines a complex language by describing the grammar of OCL language, the relations to user-specified models and an associated object library. Assessing the OCL implementation of a concrete tool would significantly affect the effort for conducting our tool study.

Thus, we decided to collect only basic information on the OCL support of the modeling tools as additional information.

4.3 Compliance Level Calculation

The collected results on tool capabilities in terms of feature fulfillment are used to determine the UML compliance level per tool. For a given compliance level only a subset of the features in the feature hierarchy is relevant. Thus, we can derive for each compliance level in Section 5.1 a compliance profile, i.e. a projection of the relevant features in the feature hierarchy to automate the compliance level calculation. By using the compliance profiles as a feature selection mechanism, the results of an individual tool can be calculated with respect to the compliance profile. In fact, UML requires the (complete) realization of certain diagram types per compliance level. From this strict viewpoint of UML compliance as specified in the UML no tool in our evaluation would receive a compliance level at all. Thus, in this study we attest for each candidate level with 50%-75% fulfillment a partial compliance, for more than 75% fulfillment an (acceptable) full compliance.

The compliance level calculation is done following a two step algorithm:

- Determine candidate compliance levels: As the compliance levels are defined in terms of the top-level categories in the feature hierarchy, for each of the UML-defined categories (i.e. of except model persistence, technical data or additional data) at minimum one feature must be implemented. By comparing the categories required by the compliance levels and the implemented features, the candidate levels are determined.
- Validate the candidates upon the average feature fulfillment: By using the compliance profiles as a feature selection mechanism, i.e. to ignore all features not activated in the compliance profile, the feature fulfillment of an individual tool can be calculated with respect to the compliance profile. Thus, we can obtain a feature fulfillment degree per

compliance profile and per tool. As described above, we attest for each candidate level with 50%-75% fulfillment a partial compliance, for more than 75% fulfillment an (acceptable) full compliance.

5 Findings per Tool

In this section the individual findings per tool are given. First, we display an overview summary table on all evaluated tools in Table 2. The last column of Table 2 reflects weather the findings on the individual tool were

- Taken over from [12] as no new version was released by the vendor. (-)
- Updated due to the reevaluation. (update)
- Renamed by the vendor, e.g. due to company fusions and updated during the reevaluation. (rename)
- Added as a new tool to this study based on an update search for tools. (new)

Section 5.1, we discuss an example tool characterization. Then, we list for each tool a similar capability and compliance characterization along with a screenshot in individual sections.

Tool name	Vendor	Version	Compliance level	Change
Apollo	Gentleware	3.0	-	update
Arcstyler	Interactive Objects	5.5.414	L2-0C	-
ArgoUML	University of California, Irvine	0.24	partial L2-0C	-
ARIS UML Designer	IDS Scheer	7.0.2.207949	publication not permitted	-
Artisan Studio	ARTiSAN	7.0.20	partial L2-MAC	update
Artiso VisualCASE	VisualCase	2.13.0	partial L2-0C	-
Astade	Astade Team	0.8.3	-	-
Blueprint Software Modeler	@-portunity B.V.	1.4.0	partial L2-MAC	-
BOUML	BOUML	4.22.2	partial L2-0AC	update
Cacoo	Nulab Inc.	June 2010	-	new
Cadifra UML Editor	Adrian & Frank Buehlmann	1.3.1	-	-
Concept Draw	Computer Systems Odessa	8.0.7.3 Professional Evaluation	partial L1.4	update
Delphia Object Modeler	Delphia Object Modeler	3.2.6	-	-
Dia	Alexander Larsson	0.96.1	-	-
Eclipse UML	eclipse.org	0.7.0	partial L2-MAC	-

Tool name	Vendor	Version	Compliance level	Change
Enterprise Architect	Sparx Systems	7.0.817 (corporate)	partial L2-3AC	-
Essential Modeler	Jaczone	2.00.0010 R1	-	-
eUML2 Studio	Soyatec	3.4.0.20091121	partial L2-0AC	update
Fujaba	Uni Paderborn, Kassel, Siegen, Darmstadt	5.0.4 20070622	-	-
Gaphor	http://sourceforge.net	0.12.5	-	-
Gliffy	Gliffy, Inc.	13/01/2009	-	update
Green UML	University of Buffalo	3.5	-	update
Ideogramic UML	Ideogramic	2.3.3	-	-
Innovator	MID	2008 10.0.03 Object eXcellence	partial L2-0C	update
Javelin	Step ahead software	7.1.1.3	-	update
Jude	ChangeVision	5.1b1 community	partial L2-0C	update
MagicDraw	NoMagic	16.0 Enterprise SP1	L1.4	update
MEGA development	MEGA international	2009 SP 1 patch 3.0 721-2496	L2-0C	update
MetaEdit+	MetaCase	4.5	-	-
MetaMill	MetaMill	5 build 860	partial L2-1AC	update
MyEclipseIDE	genuitec	8.0-20091120	publication not permitted	update
Netbeans	www.netbeans.org	5.5.1 with UML Module 1.1.14	partial L2-0C	update
Objectteering/UMLFree Edition	Objectteering Software	6.1.00	partial L2-0C	new
objectiF	Microtool	7.0.133	-	-
Omondo UML Plugin for Eclipse	Omondo	3.3.0.v20070629 2007 free	partial L2-MAC	-
OODesigner	Tae Gyun Kim	0401 2004-01-12	-	-

Tool name	Vendor	Version	Compliance level	Change
Open Modelsphere	Grandite	3.0 Build 904	partial L2-0C	rename
OpenAmeos	ScopeSet	10.1 (Build 26)	partial L1-4	update
Papyrus UML	Papyrus UML	1.6.2	L2-MAC	-
Poseidon	Gentleware	6.0.1	partial L2-0C	-
PowerDesigner	Sybase	12.5.0.2169	partial L1.4	-
QuickUML	Excel Software	3.0.4	-	update
Rational Software Architect	IBM/Rational	7.5.4	publication not permitted	update
Rational System Architect	IBM Rational	11.3.1	publication not permitted	rename
Rational Tau	IBM /Rational	4.3.0.0.13660	publication not permitted	rename
Real Time Developer Studio	Pragmadev	4.0.3 2009-07-20	-	update
Rhapsody	Telelogic	7.1.1.0 Build 893629	publication not permitted	-
Rose	IBM/Rational	7.0	publication not permitted	-
Smartdraw	Smartdraw	2010.07	-	update
StarUML	StarUML Development Group	5.0.2.1570	partial L2-0C	-
SystemArchitect	Telelogic	10.7.16 SP1	publication not permitted	-
Tangible Architect	Tangible engineering	4.0	-	-
Telelogic Tau/Modeler Edition	Telelogic	3.1.1.0.0.3145	publication not permitted	-
Teuta	Institut für Softwarewissenschaft Universität Wien	-	-	update
Together	Borland	2006 R2 8.1.1Build-ID: 4359.1	L1.4	-
Topcased	Topcased.org	3.2.0.v200911301720	L2-0C	update
Umbrello UML Modeler	Umbrello Project Team	2.3.2	partial L2-0C	update
UMLAUT	IRISA/CNRS	beta 1-8	partial L2-0C	-

Tool name	Vendor	Version	Compliance level	Change
UMLDiagrammer	Pacestar	6.20.2040	-	update
UMLed	Georg Kubitz	1.8.4 b1	-	-
UMLet	UMLet group	10.3	-	update
UMLPad	Luigi Bignami	3.2	partial L2-0C	update
UMLStudio	Pragsoft Cooperation	8.2.1	partial L2-0C	-
Umodel	Altova	Professional 2009	partial L2-2AC	update
violet	C. S. Horstmann und A. de Pellegrin	0.21.1(2007)	-	-
Visible Analyst	Visible Systems	7.6.5	publication not permitted	-
Visio	Microsoft	Professional 2007 12.0.4518.1014	partial L2-0C	update
Visio with UML2 stencils	Microsoft/Pavel Hruby	for Visio 2007	partial L2-0C	update
Visual Paradigm for UML	Visual Paradigm	7.1 (Build 20091009)	L2-2AC	update
Visual Studio Ultimate	Microsoft	10.0.30319.1	L2-0C	new
Visual UML	Visual Object Modelers Inc.	5.26 build 634	L1.4	-
Yed	yWorks	3.4.0.2	-	new

Table 2: Summary of all tools evaluated in this study.

5.1 Example Tool Characterization

In this section we discuss the findings for an individual tool as an example. For each tool the technical data, a vendor statement, a price range, information on the user interface concepts, a screenshot and two tables are given.

- Prices are intended for orientation and not as an attempt to capture the pricing model of a vendor. In fact, often it was rather difficult to get price information for a concrete tool. In the case that a tool is free or open source, this is stated as price information.
- The vendor statement listed is an excerpt from the tool homepage describing the tool in a concise way. As the vendor statement is a quote, it is displayed italic font face.
- Along with the technical data we list the user interface concepts of the tool, e.g. whether dialogs, toolbars or views present the functionality to the user.
- For each individual tool we give a short comment on noticeable problems or special features recognized by the evaluator.
- Information on our evaluation results are summarized in two tables, one containing the tool capabilities in terms of feature fulfillment degrees for UML modeling facilities and another table showing the feature data aggregation with respect to UML compliance levels.

We discuss the two summary tables for an example tool in the remainder of this section.

Modeling criteria	Feature fulfillment degree
Class diagram	16.89%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	5.82%
Traceability	0.00%
Code generation	7.00%

Table 3: Example feature fulfillment table for a tool.

Tool characteristics	
Promoted UML version	2.1
XMI version	2.0
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC
L1.3	10.29%
L1.4	10.19%
L2-0	32.91%
L2-M	21.47%
L2-1	7.38%
L2-2	5.97%
L2-3	5.65%
Assigned level	-

Table 4: Example tool characteristics and compliance level summary.

Table 3 summarizes the tool capabilities, i.e. the degrees to which features, assigned to individual (diagram) categories were identified while evaluating the tool. Obviously, the example tool provides only class diagrams implementing 16.89% of the features specified in UML 2 for class diagrams. Also auxiliary UML features like model management, templates and profiles are considered by individual feature groups (and none of these categories was fulfilled by the example tool). Thus, considering the features for all UML diagrams, the overall UML summary is 5.82%. As additional information, the tool does not provide any traceability features and allows only simple code generation on few target languages (7 %). Detailed information on the missing features can be obtained by consulting the tool evaluation spreadsheet, which are available online through our product selection tool.

In the upper part of Table 4, some tool characteristics being relevant for UML compliance are summarized, i.e. the version of the supported UML specification (here UML 2.1), the XMI version¹ as inferred from the exported XMI files (2.0) and whether the test for syntactical validity was successful (no). In the lower part of Table 4, the information to determine the UML compliance level is given. First, the valid compliance levels are listed. For the example tool, Level 0 with abstract syntax and concrete syntax compliance could be reached, but no higher level is possible, because several required diagram types are not supported (as shown in Table 3). Please note that according to OMG documents XMI files according to version 1.x are not appropriate for UML 2 compliance. Below the XMI validity in Table 4, the degrees of feature fulfillment with respect to the given UML compliance profiles are listed. As stated in Section 4.3, we assign an (acceptable) compliance level, if the degree of feature fulfillment of the level is more than 75% and a partial compliance level indicated by braces if the degree is more than 50% but lower than 75%. Thus, according to the overall threshold of 50% or 75%, respectively,

¹ In fact, in the table we omitted an entry for the diagram interchange standard, because only Gentleware Poseidon supports this specification.

neither a UML 2 compliance level nor the compliance to an earlier UML version can be assigned.

In the following sections the individual findings per tool considered in the evaluation are given.

5.2 Apollo by Gentleware

Name: Apollo

Version: 3.0

Vendor: Gentleware

URL: <http://www.gentleware.com>

Price (if available): 57.12 € per year

Vendor statement: *The fully synchronized roundtrip engineering employs the latest releases of UML 2.1 and Java 5 to provide an instant visualization of any existing Java code through UML class diagrams, and likewise propagate changes to the UML model throughout the code.*

User interface: views, direct editing of diagram elements in the diagram, toolbar, menu and literal programming

Comments: The program needs a lot of additional memory and reacts very slow on user input. Changes to a class diagram are not reflected in the generated code but changes to the code changes the class diagram. Class modelling is strongly related to Java code.

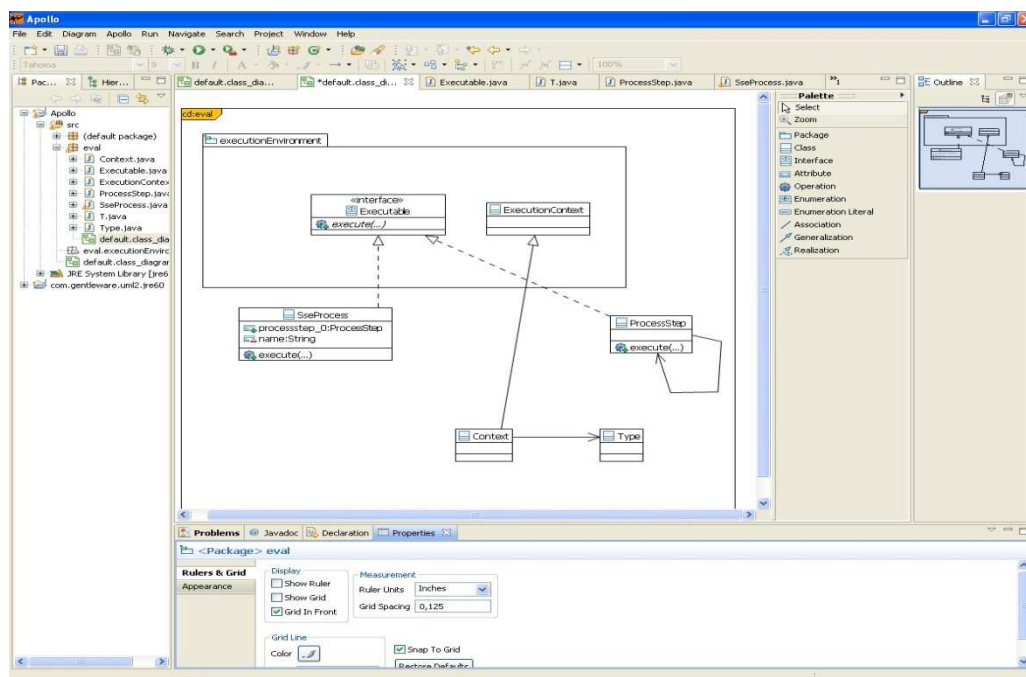


Figure 8: UML Class Diagram in "Apollo".

Due to technical problems as stated above the reference diagram was not completed.

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	17.88%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	5.73%
Traceability	0.00%
Code generation	7.00%

Tool characteristics	
Promoted UML version	2.1
XMI version	2.0
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC
L1.3	10.92%
L1.4	10.86%
L2-0	32.91%
L2-M	21.99%
L2-1	7.50%
L2-2	6.07%
L2-3	5.74%
Assigned level	-

5.3 Arcstyler by Interactive Objects

Name: Arcstyler

Version: 5.5.414

Vendor: Interactive Objects

URL: <http://www.arcstyler.com/>

Price (if available): unknown

Vendor statement:

- *Increased productivity - significantly reduce development time due to visual modelling and comprehensive code generation*
- *Higher quality - enforcement of clear architectures, up-to-date system documentation and constant validation and testing*
- *Enhanced maintainability - easy incorporation of changes reduces time and cost*

User interface: dialogs and toolbar

Comments: This program is based on MagicDraw 9.5. The code generation leads to incomprehensible problems and the diagram information is corrupted while reloading.

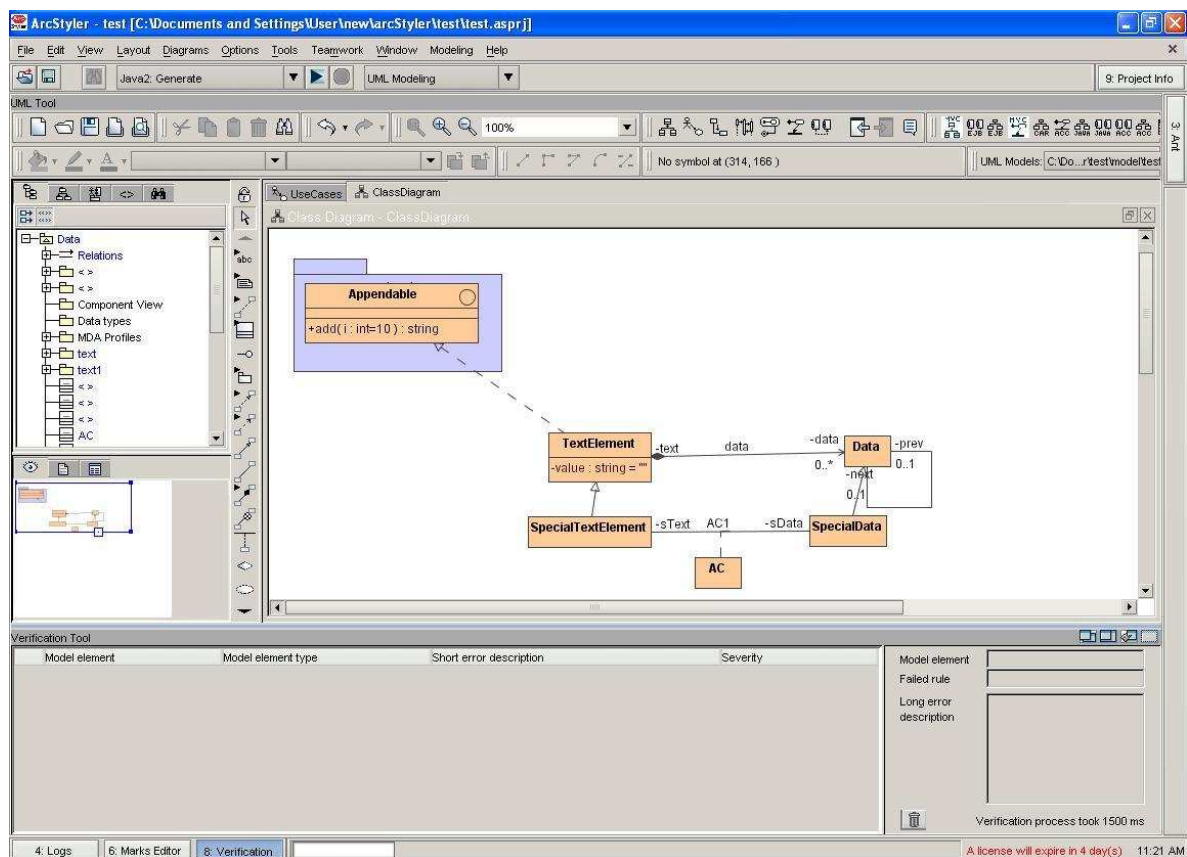


Figure 9: UML Class Diagram in "Arcstyler"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	56.95%
Component diagram	0.00%
Composition diagram	40.00%
Deployment diagram	62.12%
Activity diagram	24.64%
Sequence diagram	35.56%
Communication diagram	61.54%
Interaction overview diagram	50.00%
Timing diagram	0.00%
State machine diagram	55.95%
Use case diagram	84.21%
Information flows	0.00%
Model management	66.67%
Templates	58.33%
Profiles	40.00%
UML summary	45.54%
Traceability	40.00%
Code generation	20.00%

Tool characteristics	
Promoted UML version	1.4
XMI version	1.0
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	73.32%
L1.4	72.44%
L2-0	77.22%
L2-M	65.97%
L2-1	43.19%
L2-2	45.73%
L2-3	45.64%
Assigned level	L2-0C

5.4 ArgoUML by University of California, Irvine

Name: ArgoUML

Version: 0.24

Vendor: University of California, Irvine

URL: <http://www.argouml.org/>

Price (if available): Open Source (BSD license)

Vendor statement: *ArgoUML uses GEF, the UCI Graph Editing Framework to edit UML diagrams. The following diagram types are supported: Class diagram, statechart diagram, activity diagram, use case diagram, collaboration diagram, deployment diagram (includes object and component diagram in one), sequence diagram.*

User interface: views, direct editing of diagram elements in the diagram, menu and toolbar

Comments: The implementation is the code basis of the tool Poseidon. The delete operations are much faster than in Poseidon and several implementation differences between Poseidon and this program can be detected, thus, the detail features of both programs are different. There were several problems in the internationalization of the German user interface.

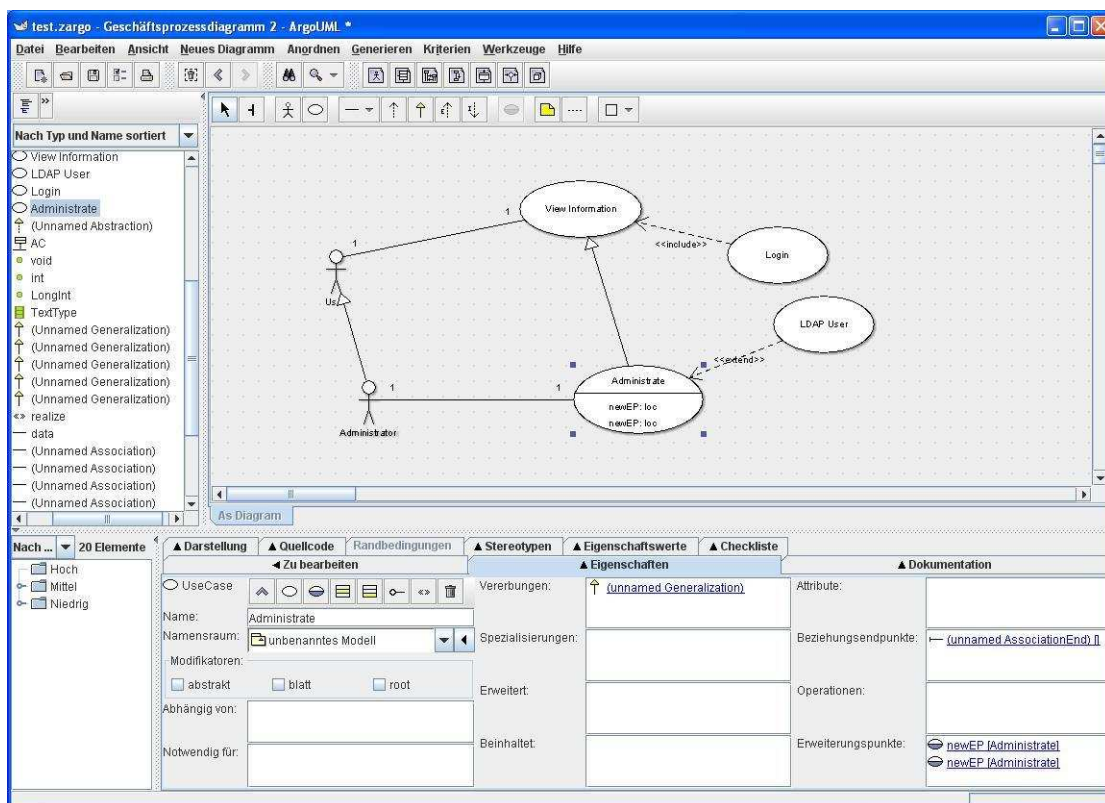


Figure 10: UML Use Case Diagram in "ArgoUML"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	61.59%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	39.39%
Activity diagram	19.57%
Sequence diagram	15.56%
Communication diagram	30.77%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	45.24%
Use case diagram	60.53%
Information flows	0.00%
Model management	33.33%
Templates	0.00%
Profiles	10.00%
UML summary	34.93%
Traceability	0.00%
Code generation	19.00%

Tool characteristics	
Promoted UML version	1.4
XMI version	1.2
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	62.61%
L1.4	62.42%
L2-0	72.15%
L2-M	62.30%
L2-1	35.97%
L2-2	36.52%
L2-3	35.00%
Assigned level	partial L2-0C

5.5 ARIS UML Designer by IDS Scheer

Name: ARIS UML Designer

Version: 7.0.2.207949

Vendor: IDS Scheer

URL: http://www.ids-scheer.de/de/Software/ARIS_Software/ARIS_UML_Designer

Price (if available): unknown

Vendor statement: *ARIS UML Designer is the first tool to speak the languages of business processes and software development. Because it supports the entire software development process, there is no need for extra steps outside this process. With ARIS UML Designer, all process modelers and UML modelers work with an integrated software product. Users access process model data and UML content via a Web browser, thereby enabling processing and change management within a multi-user environment.*

The vendor of this tool prohibited the publication of findings on this tool, because they scheduled a completely revised implementation for 2010.

This page is left intentionally blank.

5.6 Artisan Studio by ARTiSAN

Name: Artisan Studio

Version: 7.0.20

Vendor: ARTiSAN

URL: <http://www.artisansw.com/>

Price (if available): 2495\$

Vendor statement: *Atego's flagship modeling tool suite, Artisan Studio, provides complete support for OMG UPDM, SysML and UML in a single, integrated toolset.*

Artisan Studio® Architect Enterprise Edition provides all the features you need to model architectural frameworks running directly on the multi-user repository giving unparalleled levels of performance, whatever the model size, from one to hundreds of users.

User interface: views, direct editing of diagram elements in the diagram, menu and toolbar

Comments: For creating a model, the program needs an explicit import of a UML profile. The user guidance is difficult. The program offers additional runtime settings on model elements e.g. timings.

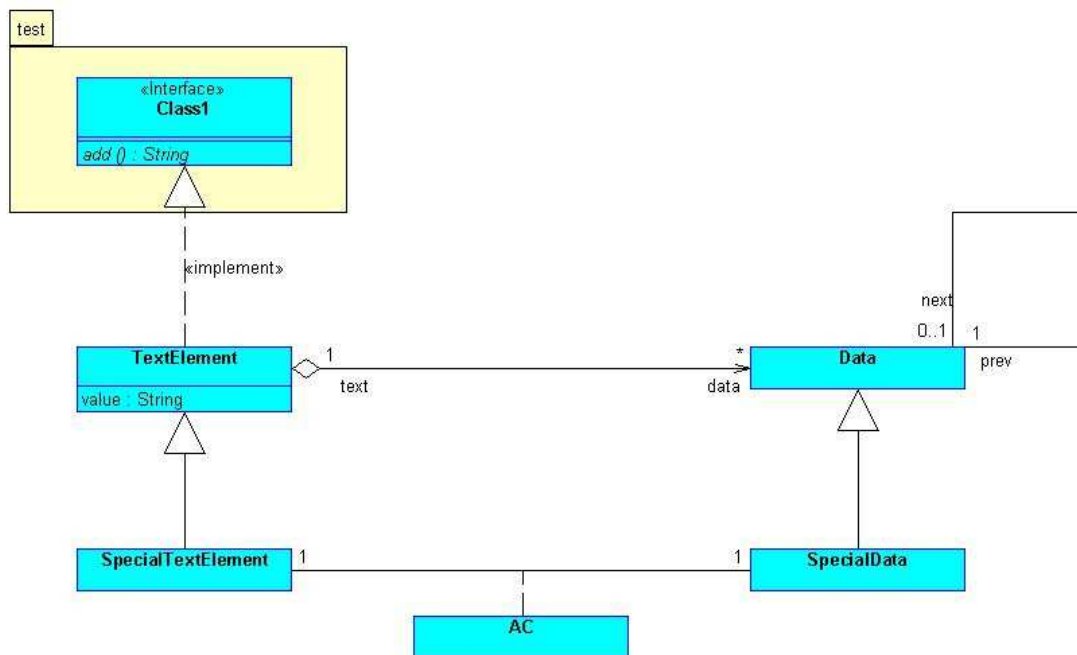


Figure 11: UML Use Case Diagram in "Artisan Studio"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	67.88%
Component diagram	0.00%
Composition diagram	70.00%
Deployment diagram	0.00%
Activity diagram	29.71%
Sequence diagram	15.56%
Communication diagram	65.38%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	55.95%
Use case diagram	63.16%
Information flows	28.57%
Model management	100.00%
Templates	50.00%
Profiles	85.00%
UML summary	43.21%
Traceability	40.00%
Code generation	25.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	2.1
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC, L2-MAC
L1.3	66.81%
L1.4	66.81%
L2-0	83.54%
L2-M	74.35%
L2-1	43.75%
L2-2	42.58%
L2-3	43.30%
Assigned level	partial L2-MAC

5.7 Artiso VisualCASE by VisualCASE

Name: Artiso VisualCASE

Version: 2.13.0

Vendor: VisualCASE

URL: <http://www.visualcase.com>

Price (if available): 2495 \$

Vendor statement: *Support for all eight UML diagrams: Use Case, Class, Sequence, Collaboration, State, Activity, Component and Deployment.*

User interface: dialogs, direct editing of diagram elements in the diagram and toolbar

Comments: -

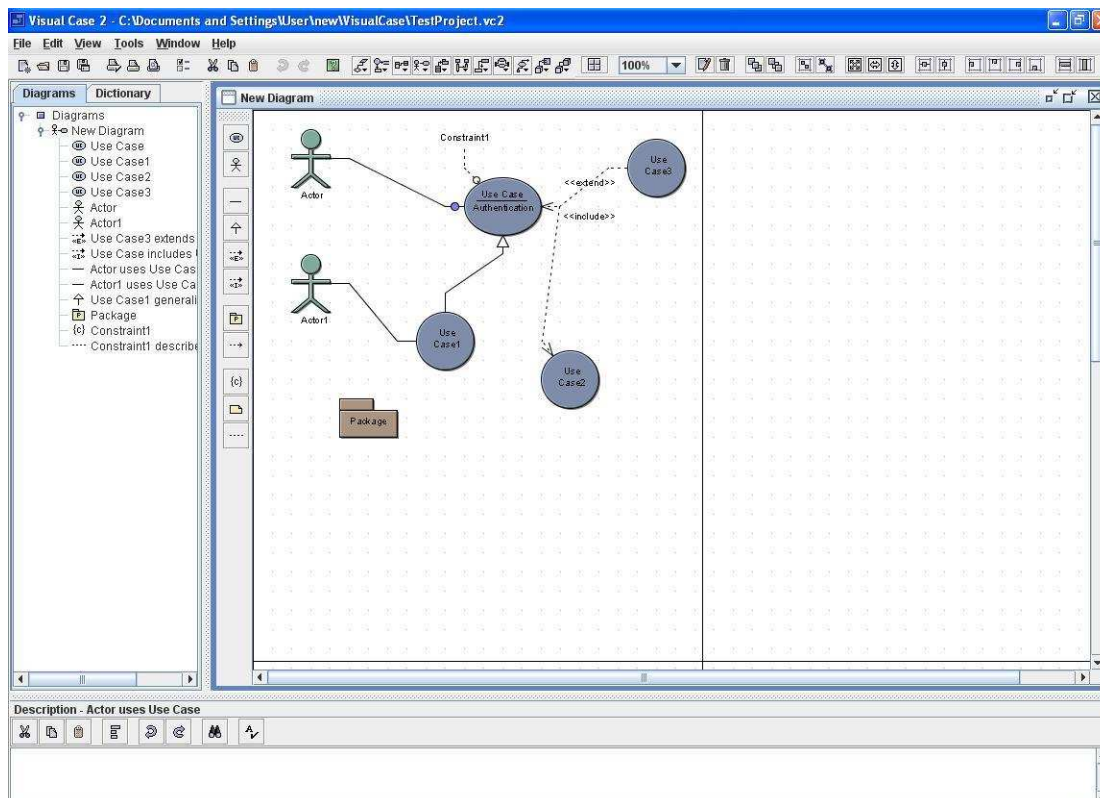


Figure 12: UML Class Diagram in "Artiso VisualCASE"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	35.43%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	15.15%
Activity diagram	15.22%
Sequence diagram	23.33%
Communication diagram	73.08%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	25.00%
Use case diagram	44.74%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	22.93%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	43.49%
L1.4	42.80%
L2-0	55.70%
L2-M	43.98%
L2-1	25.69%
L2-2	24.27%
L2-3	22.98%
Assigned level	partial L2-0C

5.8 Astade by Astade Team at Tigris

Name: Astade

Version: 0.8.3

Vendor: Astade Team

URL: <http://astade.tigris.org/>

Price (if available): Open Source (GPL)

Vendor statement: *The goal of this project is to develop a UML tool where you can develop at model level and automatically generate source code for C++. The program shall be a complete front-end (GUI) for software development (in C++) which can be customized for any compiler (also the GNU compiler will be the first one).*

User interface: dialogs and menu

Comments: The program realizes an unusual usability concept by mainly providing context menus. The sequence diagram editor can be controlled only by textual commands and the code generation requires several non-trivial configuration options.

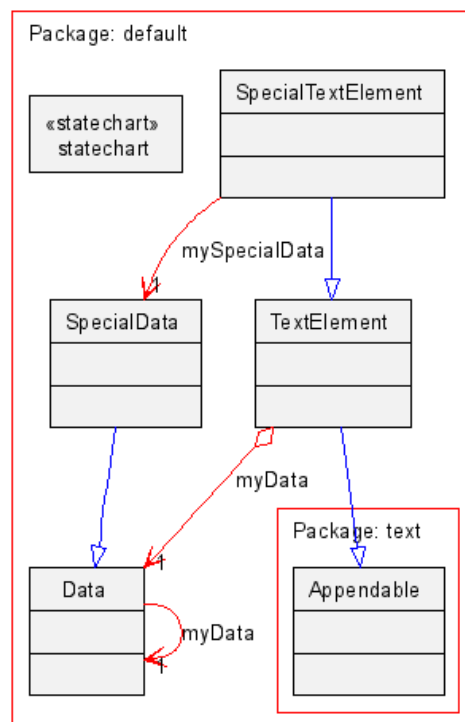


Figure 13: UML Class Diagram in "Astade"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	21.19%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	15.56%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	15.48%
Use case diagram	36.84%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	11.15%
Traceability	0.00%
Code generation	14.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	21.22%
L1.4	21.09%
L2-0	31.65%
L2-M	26.18%
L2-1	13.06%
L2-2	12.02%
L2-3	11.38%
Assigned level	-

5.9 Blueprint Software Modeler by @-portunity B.V.

Name: Blueprint Software Modeler

Version: 1.4.0

Vendor: @-portunity B.V.

URL: <http://www.atportunity.com/>

Price (if available): unknown

Vendor statement:

- *UML2.1: Modeling supporting the most important UML diagrams implemented according the latest OMG UML2.1 specification*
- *OCL2.0: Editor supporting defining constraints including syntax checking, syntax coloring and completion proposals*
- *Meta Modeling and Profiles allowing easy creation of Meta models and generating their corresponding Profiles (with Stereotypes) and Libraries*

User interface: views, menu and toolbar

Comments: The evaluator noticed several usability problems. The program realizes good consistency functionality for objects and links. While working with the program it produces several internal errors and often needs a restart of the host computer. Blueprint software modeler allows the creation of user-specific metamodels.

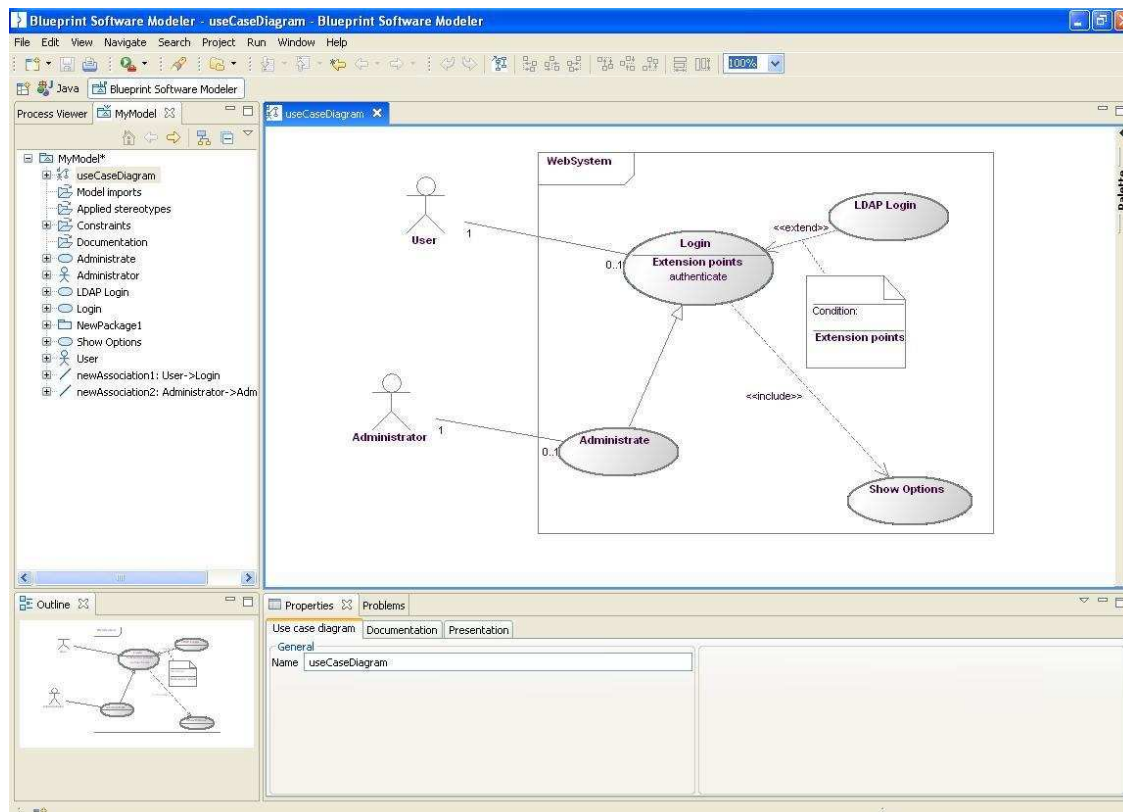


Figure 14: UML Use Case Diagram in "Blueprint Software Modeler"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	72.19%
Component diagram	58.82%
Composition diagram	70.00%
Deployment diagram	0.00%
Activity diagram	60.87%
Sequence diagram	40.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	52.38%
Use case diagram	76.32%
Information flows	0.00%
Model management	100.00%
Templates	0.00%
Profiles	50.00%
UML summary	49.68%
Traceability	70.00%
Code generation	11.00%

Tool characteristics	
Promoted UML version	2.1
XMI version	2.0
XMI valid	Partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC, L2-MAC
L1.3	68.28%
L1.4	68.48%
L2-0	84.81%
L2-M	81.68%
L2-1	55.83%
L2-2	51.24%
L2-3	49.79%
Assigned level	L2-MAC

5.10 BOUML by BOUML Team

Name: BOUML

Version: 4.22.2

Vendor: BOUML Team

URL: <http://bouml.free.fr/>

Price (if available): Open Source (GPL)

Vendor statement: *BOUML is a free UML 2 tool box allowing you to specify and generate code in C++, Java, Idl, Php and Python.*

User interface: dialogs, views, direct editing of diagram elements in the diagram, menu and toolbar

Comments: The authors had some good ideas to improve the usability. The behaviour of deleting elements is sometimes confusing: While running the program, the deletion of model elements is not made persistent and not made until the containing project is closed by the user. The program cares much about implicit consistency among model elements. The evaluator did not succeed in running the code generator.

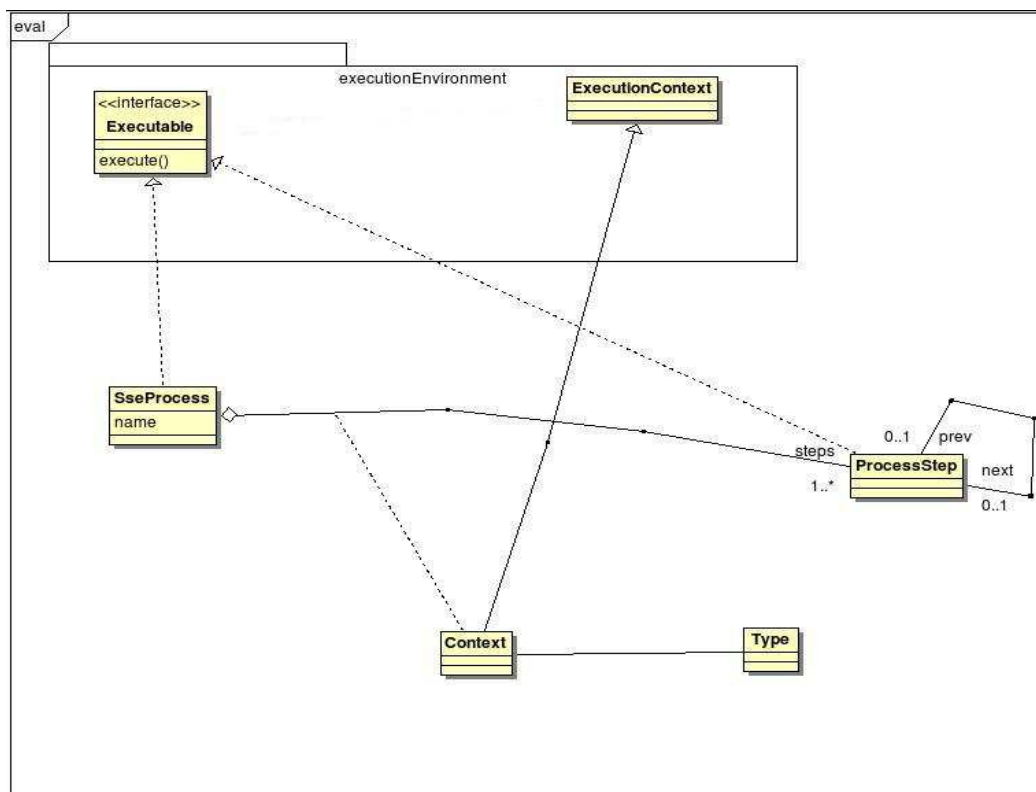


Figure 15: UML Class Diagram in "BOUML"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	59.93%
Component diagram	76.47%
Composition diagram	0.00%
Deployment diagram	25.76%
Activity diagram	47.83%
Sequence diagram	64.44%
Communication diagram	61.54%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	76.19%
Use case diagram	76.32%
Information flows	0.00%
Model management	50.00%
Templates	0.00%
Profiles	0.00%
UML summary	49.26%
Traceability	0.00%
Code generation	21.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	1.2/2.1
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC
L1.3	66.81%
L1.4	67.01%
L2-0	64.56%
L2-M	62.30%
L2-1	52.22%
L2-2	51.35%
L2-3	49.26%
Assigned level	partial L2-0AC

5.11 Cacoo by Nulab Inc.

Name: Cacoo

Version: June 2010

Vendor: Nulab Inc.

URL: <http://www.cacoo.com>

Price (if available): 0\$

Vendor statement: *Cacoo is a user friendly online drawing tool that allows you to create a variety of diagrams such as site maps, wire frames, UML and network charts.*

User interface: dialogs, direct editing of diagram elements in the diagram, menu, toolbar and drag & drop menu

Comments: Cacoo is an online drawing tool.

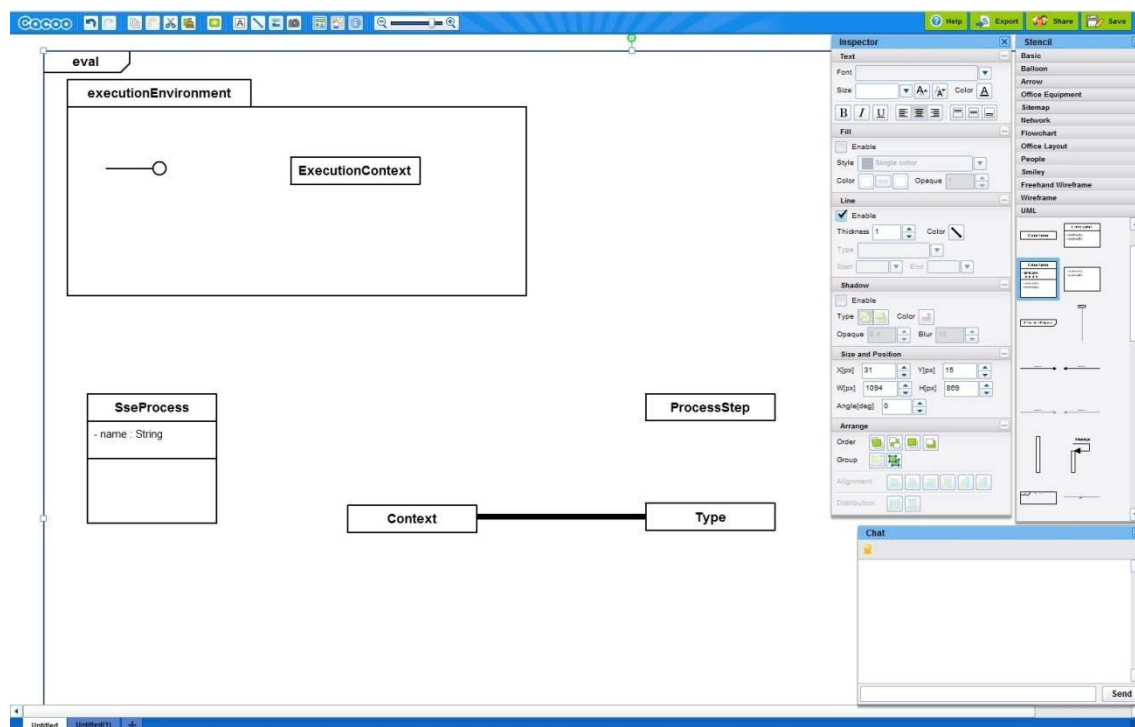


Figure 16: UML Class Diagram in "Cacoo"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	11.59%
Component diagram	52.94%
Composition diagram	50.00%
Deployment diagram	21.21%
Activity diagram	26.09%
Sequence diagram	26.67%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	45.24%
Use case diagram	42.11%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	20.28%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	27.94%
L1.4	28.18%
L2-0	25.32%
L2-M	13.09%
L2-1	19.31%
L2-2	21.46%
L2-3	20.32%
Assigned level	-

5.12 Cadifra UML Editor by Adrian & Frank Buehlmann

Name: Cadifra UML Editor

Version: 1.3.1

Vendor: Adrian & Frank Buehlmann

URL: <http://www.cadifra.com/>

Price (if available): 30\$

Vendor statement: *Intelligent handling of connectors (but not too "intelligent"...). If you move for example a class, all attached associations will be adjusted as needed. But you still keep control of everything. You can always position any segment of a connector exactly where you want it. Cadifra UML Editor supports drawing of tree-like connectors of any complexity. High precision graphics: lines and arrows end exactly on boxes. No need to resort to a generic drawing tool without first-hand built-in UML knowledge. Perfect looking UML diagrams are a must to "sell" your ideas. It just doesn't look professional if you show printouts with arrow heads piercing boxes to a skeptic customer, colleague or manager.*

User interface: dialogs and menu

Comments: Class members are only available as text. The usability can be improved because only context menus are available.

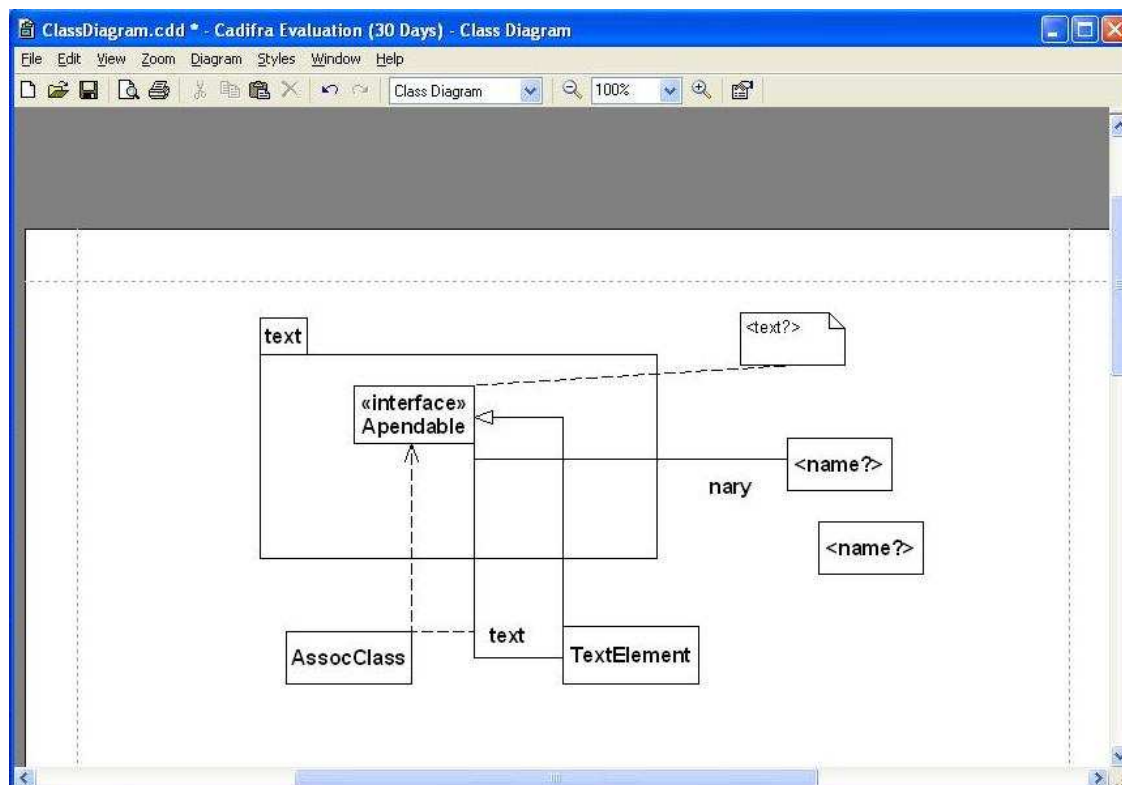


Figure 17: UML Class Diagram in "Cadifra UML Editor"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	15.23%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	12.22%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	16.67%
Use case diagram	50.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	9.55%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	18.70%
L1.4	18.58%
L2-0	27.85%
L2-M	16.75%
L2-1	10.83%
L2-2	10.34%
L2-3	9.79%
Assigned level	-

5.13 Concept Draw by Computer Systems Odessa

Name: Concept Draw

Version: 8.0.7.3 Professional Evaluation

Vendor: Computer Systems Odessa

URL: <http://www.conceptdraw.com/en/>

Price (if available): 159,90 - 199 €

Vendor statement: *ConceptDraw PRO is a powerful business and technical diagramming software. Design professional-looking graphics, diagrams, flowcharts, floor plans and much more in minutes with ConceptDraw PRO. Maintain business processes performance by clear visual documentation. Effectively present and communicate information in a clear and vivid way with ConceptDraw PRO.*

User interface: dialogs, direct editing of diagram elements in the diagram, drag & drop menu, toolbar and menu

Comments: Nesting of elements is not considered at all. The documentation does not provide a section on UML. Many diagrams and diagram specific elements cannot be found in the menus or must be combined from several diagram menus. Sometimes the user guidance is not intuitive.

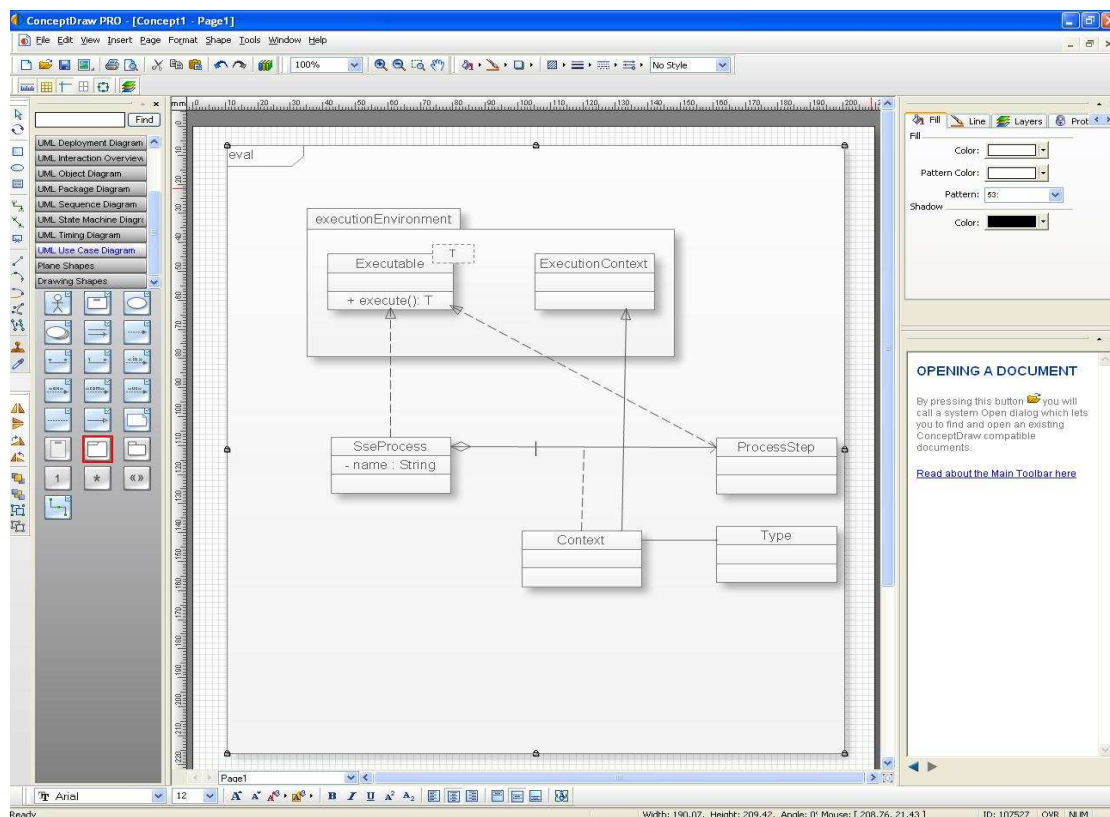


Figure 18: UML Class Diagram in "Concept Draw"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	28.15%
Component diagram	47.06%
Composition diagram	75.00%
Deployment diagram	40.91%
Activity diagram	36.96%
Sequence diagram	28.89%
Communication diagram	46.15%
Interaction overview diagram	21.88%
Timing diagram	39.47%
State machine diagram	63.10%
Use case diagram	68.42%
Information flows	0.00%
Model management	0.00%
Templates	16.67%
Profiles	10.00%
UML summary	35.99%
Traceability	10.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0C
L1.3	50.63%
L1.4	50.52%
L2-0	34.18%
L2-M	31.94%
L2-1	35.42%
L2-2	37.87%
L2-3	36.28%
Assigned level	partial L 1.4

5.14 Delphia Object Modeler by Atos Origin

Name: Delphia Object Modeler

Version: 3.2.6

Vendor: ATOS Origin

URL: <http://www.si.fr.atosorigin.com/dom/english/index.html>

Price (if available): unknown

Vendor statement: DOM (Delphia Object Modeler) is an object modeling and code generation case tool which makes implementation of model-oriented development easier. Team development is made easy by the division in components (package naming and versionning) and the use of team repositories.

User interface: dialogs and toolbar

Comments: The user guidance of the tool is complicated. Furthermore, the program does not check for inconsistencies in models. Sometimes the evaluator had to restart the program in order to open the diagram editor.

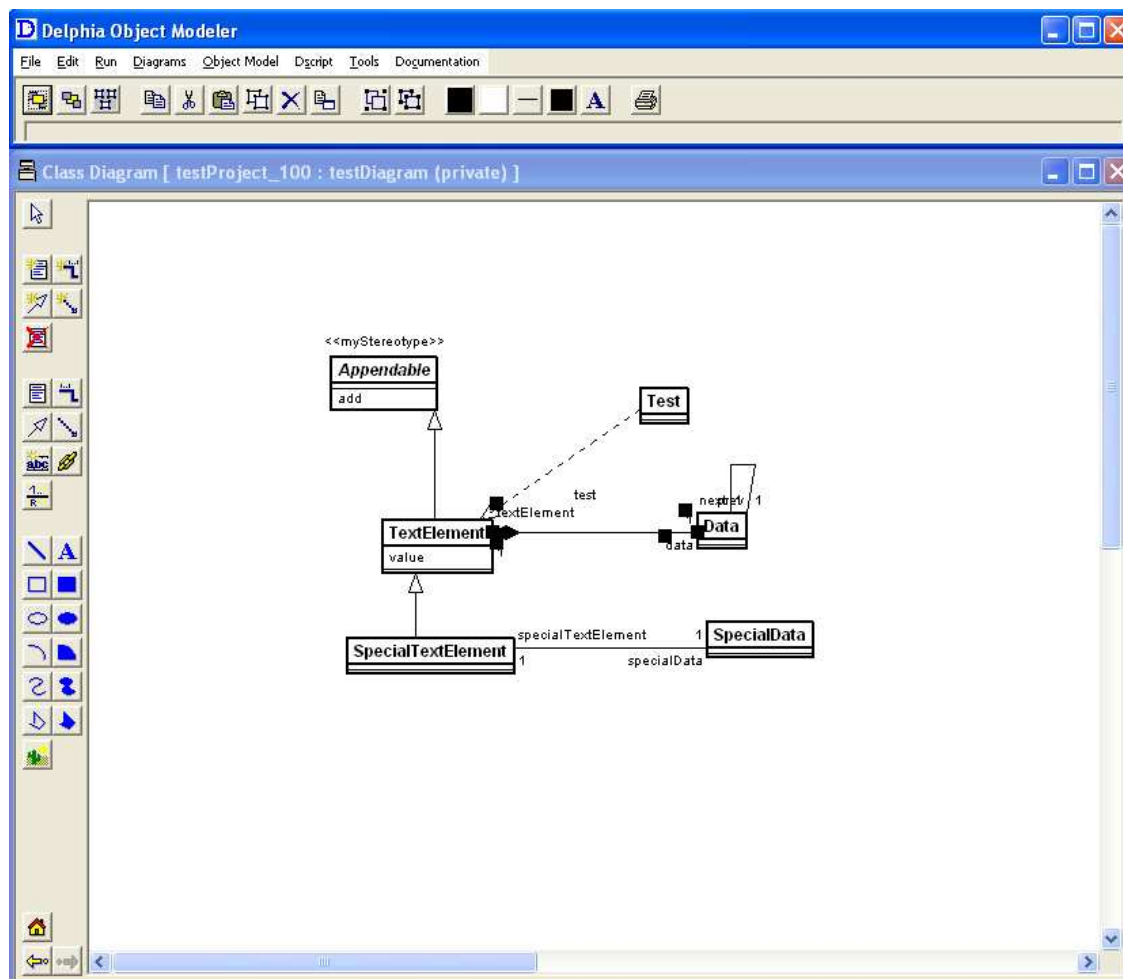


Figure 19: UML Class Diagram in "Delphia Object Modeler"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	18.97%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	22.62%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	8.49%
Traceability	20.00%
Code generation	18.00%

Tool characteristics	
Promoted UML version	-
XMI version	1.0
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	15.34%
L1.4	15.24%
L2-0	35.44%
L2-M	28.80%
L2-1	8.47%
L2-2	8.99%
L2-3	8.51%
Assigned level	-

5.15 Dia by Alexander Larsson

Name: Dia

Version: 0.96.1

Vendor: Alexander Larsson

URL: <http://live.gnome.org/Dia>

Price (if available): Open Source (GPL)

Vendor statement: *Dia is roughly inspired by the commercial Windows program 'Visio', though more geared towards informal diagrams for casual use. It can be used to draw many different kinds of diagrams. It currently has special objects to help draw entity relationship diagrams, UML diagrams, flowcharts, network diagrams, and many other diagrams. It is also possible to add support for new shapes by writing simple XML files, using a subset of SVG to draw the shape.*

User interface: dialogs, direct editing of diagram elements in the diagram, menu, toolbar and drag & drop menu

Comments: -

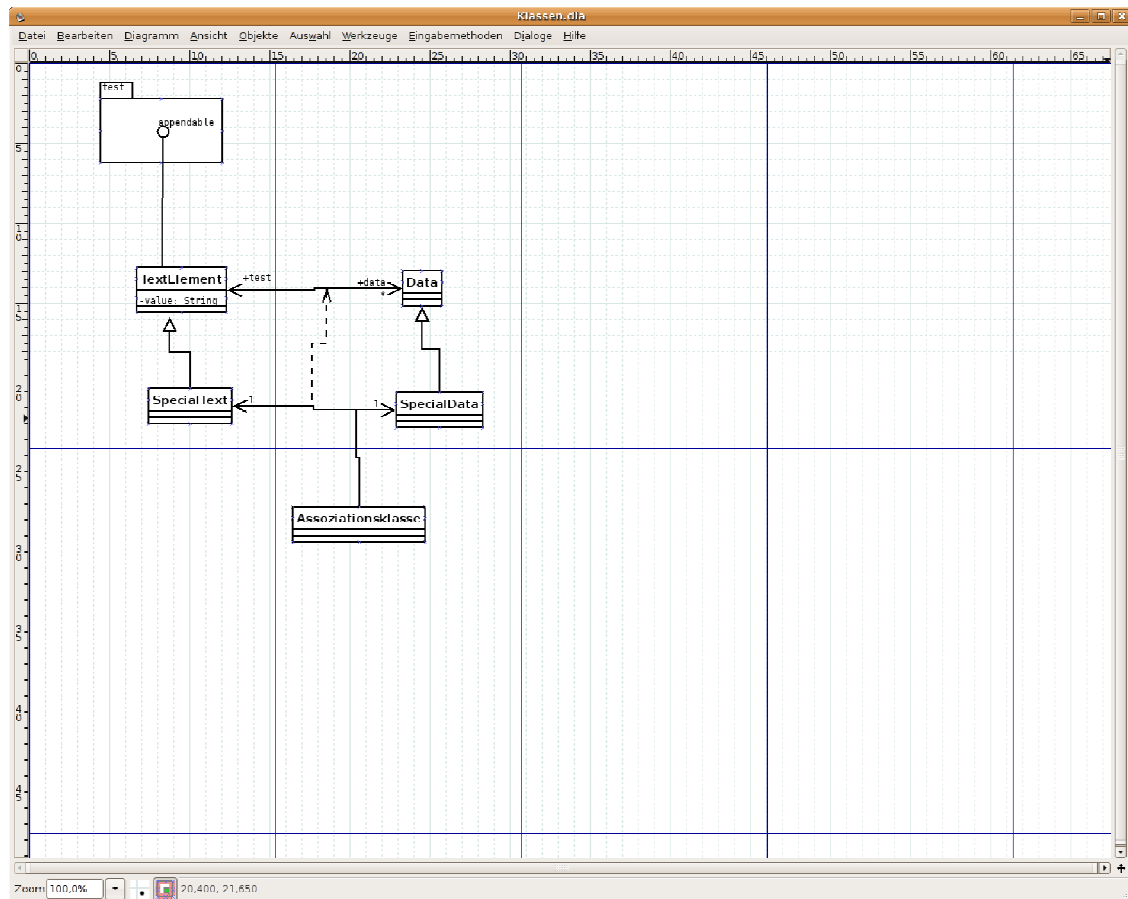


Figure 20: UML Use Case Diagram in "Dia"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	35.76%
Component diagram	41.18%
Composition diagram	0.00%
Deployment diagram	24.24%
Activity diagram	18.84%
Sequence diagram	22.22%
Communication diagram	0.00%
Interaction overview diagram	62.50%
Timing diagram	0.00%
State machine diagram	45.24%
Use case diagram	52.63%
Information flows	0.00%
Model management	0.00%
Templates	16.67%
Profiles	0.00%
UML summary	28.24%
Traceability	0.00%
Code generation	14.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
1.3	45.38%
1.4	45.51%
L2-0	45.57%
L2-M	41.36%
L2-1	29.17%
L2-2	29.66%
L2-3	28.51%
Assigned level	-

5.16 Eclipse UML by Eclipse.org

Name: Eclipse UML

Version: 0.7.0

Vendor: eclipse.org

URL: <http://www.eclipse.org>

Price (if available): Open source (EPL)

Vendor statement: *UML2 is an EMF-based implementation of the Unified Modeling Language (UMLTM) 2.x OMG metamodel for the Eclipse platform. The objectives of the UML2 component are to provide*

- *a useable implementation of the UML metamodel to support the development of modeling tools*
- *a common XMI schema to facilitate interchange of semantic models*
- *test cases as a means of validating the specification*
- *validation rules as a means of defining and enforcing levels of compliance*

User interface: views and menu and toolbars

Comments: Some meta model properties are not available in the editor. Sometimes the program shows strange behaviour and the program is unstable (hang-up, crash), in particular when editing state machines.

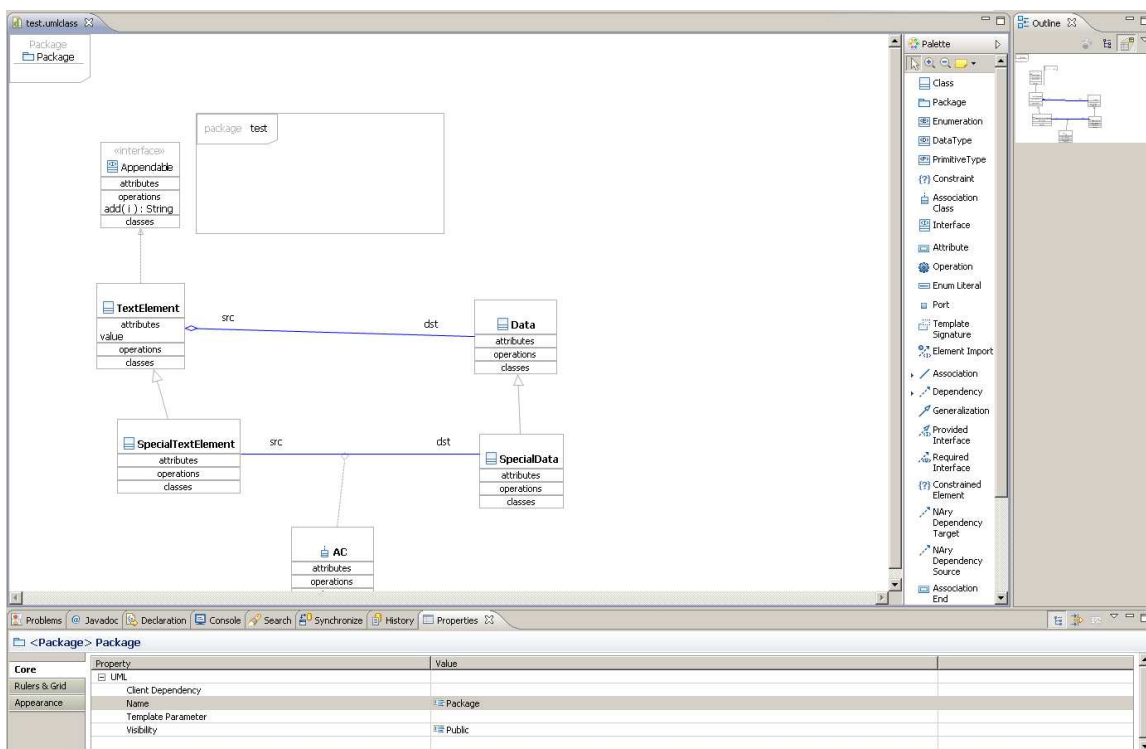


Figure 21: UML State Chart Diagram in "Eclipse UML"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	50.66%
Component diagram	41.18%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	44.93%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	42.86%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	50.00%
Profiles	50.00%
UML summary	30.57%
Traceability	10.00%
Code generation	2.00%

Tool characteristics	
Promoted UML version	2.1
XMI version	2.1
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC, L2-MAC
L1.3	39.29%
L1.4	39.25%
L2-0	54.43%
L2-M	59.16%
L2-1	31.94%
L2-2	31.01%
L2-3	30.64%
Assigned level	partial L2-MAC

5.17 Enterprise Architect by Sparx Systems

Name: Enterprise Architect

Version: 7.0.817 (corporate)

Vendor: Sparx Systems

URL: <http://www.sparxsystems.com.au/ea.htm>

Price (if available): 199 \$

Vendor statement: *Enterprise Architect combines the power of the latest UML 2.1 specification with a high performance, intuitive interface, to bring advanced modeling to the desktop, and to the complete development and implementation team. With a great feature set and unsurpassed value for money, EA can outfit your whole team, including analysts, testers, project managers, quality control staff, deployment team and more, for a fraction of the cost of some competing products.*

User interface: dialogs, views, direct editing of diagram elements in the diagram, menu and toolbars

Comments: The user interface tries to hide the variety of individual features per modeling element and leads to time consuming user operations. Partly the menu options are inconsistent. The tool allows at maximum one stereotype per model element (UML allows multiple).

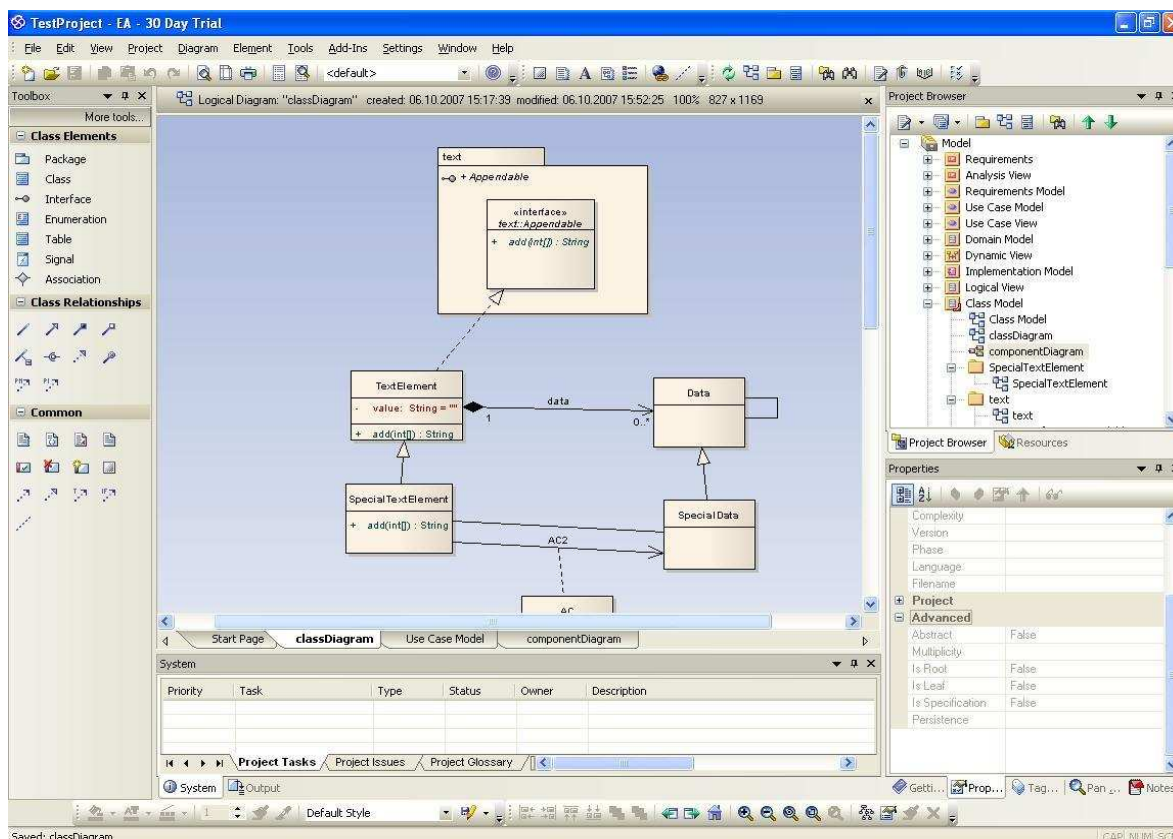


Figure 22: UML Class Diagram in "Enterprise Architect"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	68.87%
Component diagram	58.82%
Composition diagram	80.00%
Deployment diagram	60.61%
Activity diagram	78.99%
Sequence diagram	86.67%
Communication diagram	84.62%
Interaction overview diagram	56.25%
Timing diagram	71.05%
State machine diagram	67.68%
Use case diagram	84.21%
Information flows	92.86%
Model management	66.67%
Templates	37.50%
Profiles	75.00%
UML summary	71.44%
Traceability	90.00%
Code generation	31.00%

Tool characteristics	
Promoted UML version	latest UML 2.1 specification
XMI version	1.3/2.1
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0AC, L2-MAC, L2-1AC, L2-2AC, L2-3AC
L1.3	81.30%
L1.4	81.00%
L2-0	81.01%
L2-M	80.10%
L2-1	73.75%
L2-2	72.25%
L2-3	71.60%
Assigned level	<i>L1.4</i> / partial L2-3AC

5.18 Essential Modeler by Jaczone

Name: Essential Modeler

Version: 2.00.0010 R1

Vendor: Jaczone

URL: <http://www.ivarjacobson.com/products.cfm>

Price (if available): unknown

Vendor statement: *Essential Modeler offers the following:*

- *Support for the essential elements of UML2 for use-case modeling and class modeling*
- *Full featured diagram editor*
- *Navigable links between diagrams*
- *Links to external documents*
- *Smart diagram layout*
- *Copy/Paste of diagrams to other applications such as Microsoft Word*

User interface: views and toolbars

Comments: The program realizes a small subset of the UML modeling elements. The usability can be significantly improved, because many editing operations must be done using the model tree and not the modeling elements in the diagram.

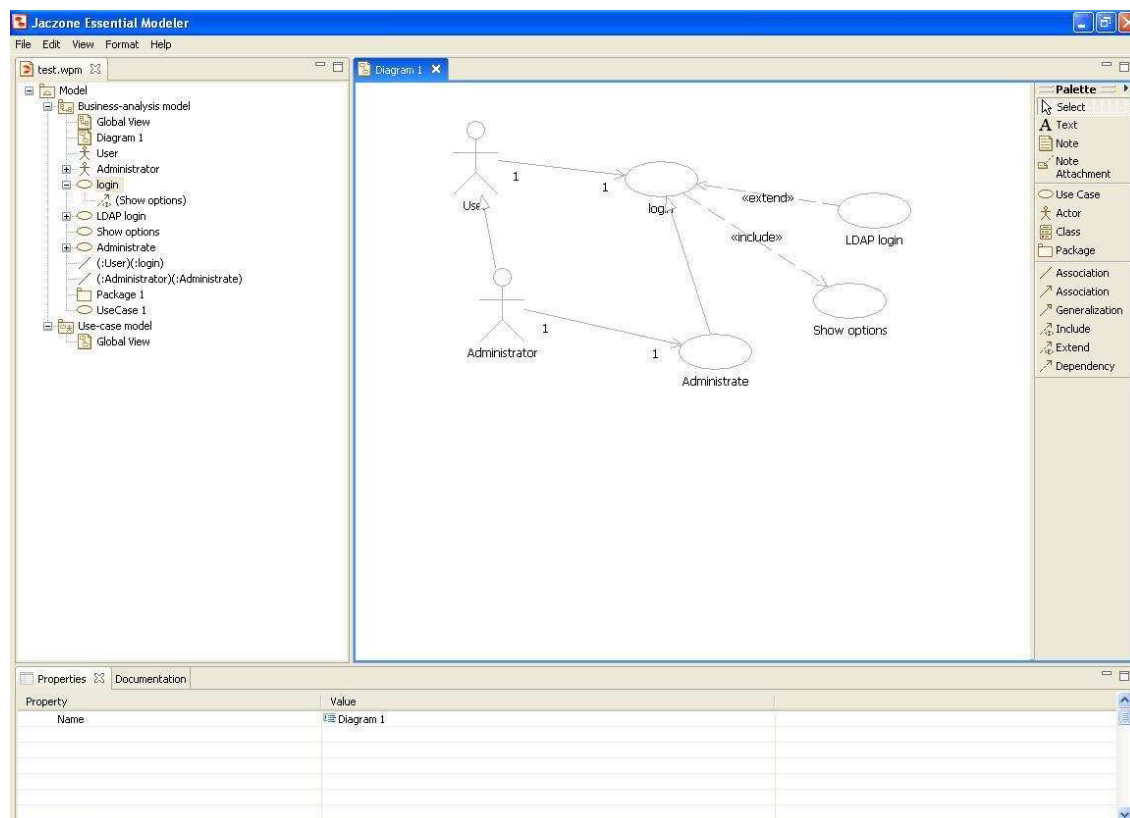


Figure 23: UML Use Case Diagram in "Essential Modeler"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	29.47%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	52.63%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	11.68%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	essential elements of UML2
XMI version	2.0
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC
1.3	21.01%
1.4	20.88%
L2-0	49.37%
L2-M	40.84%
L2-1	15.14%
L2-2	12.25%
L2-3	11.60%
Assigned level	-

5.19 eUML2 Studio by Soyatec

Name: eUML2 Studio

Version: 3.4.0.20091121

Vendor: Soyatec

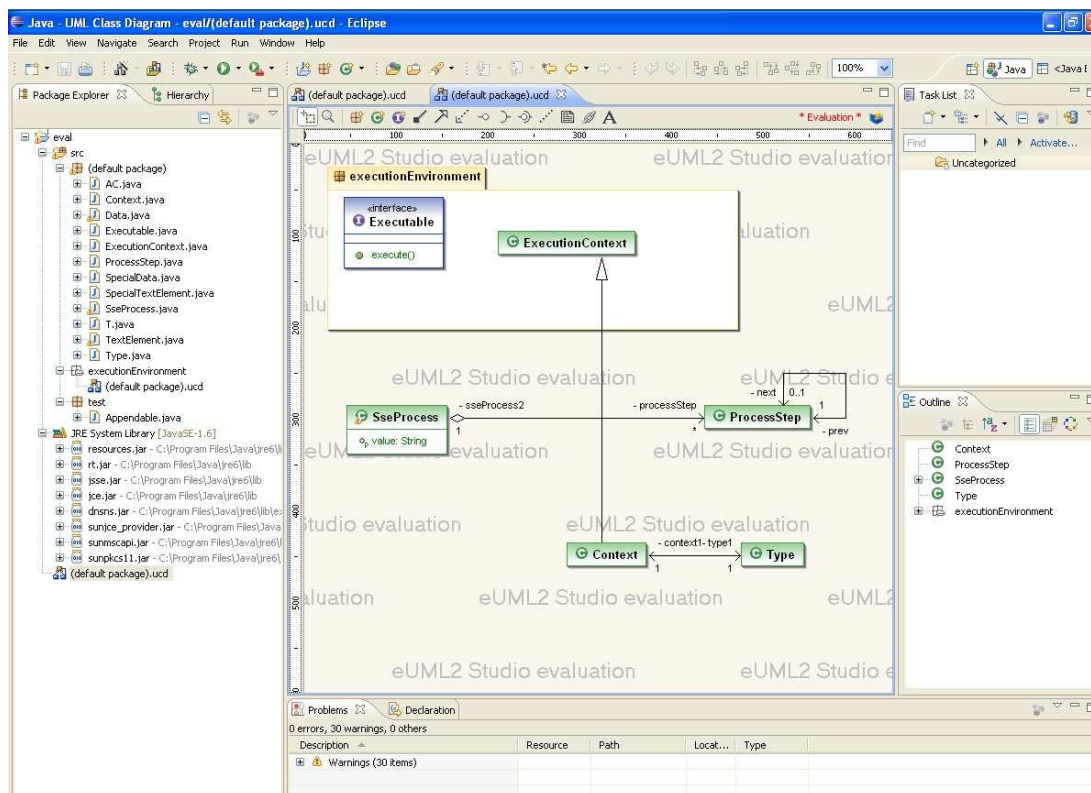
URL: <http://www.soyatec.com>

Price (if available): 0€ - 1600€

Vendor statement: *eUML2 is built on top of the UML2 framework of Eclipse as the UML metamodel, which is in fact the best open source implementation of the latest UML2.1 specification. Particularly, this version supports the OMG XMI storage format, which allows the model exchange with other UML metamodels.*

User interface: dialogs, views, menus, toolbar and direct editing of diagram elements in the diagram

Comments: Appears to be very similar to Omondo.



Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	48.68%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	35.56%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	19.11%
Traceability	10.00%
Code generation	12.00%

Tool characteristics	
Promoted UML version	2.1
XMI version	2.0
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC
L1.3	32.56%
L1.4	32.78%
L2-0	60.76%
L2-M	50.79%
L2-1	25.14%
L2-2	20.34%
L2-3	19.26%
Assigned level	partial L2-0AC

5.20 Fujaba by Universities of Paderborn, Kassel, Siegen and Darmstadt

Name: Fujaba

Version: 5.0.4 20070622

Vendor: Uni Paderborn, Kassel, Siegen, Darmstadt

URL: <http://wwwcs.uni-paderborn.de/cs/fujaba/>

Price (if available): Open Source (LGPL)

Vendor statement: *Fujaba Tool Suite combines UML class diagrams and UML behaviour diagrams (Story Diagrams) to a powerful, easy to use, yet formal system design and specification language. Furthermore the Fujaba Tool Suite supports the generation of Java sourcecode out of the whole design which results in an executable prototype. Moreover the way back is provided, too (to some extend so far, not for productive use), so that Java sourcecode can be parsed and represented within UML.*

User interface: dialogs, menus and toolbars

Comments: The program is difficult to use, because many unnecessary clicks are needed to edit a single element.

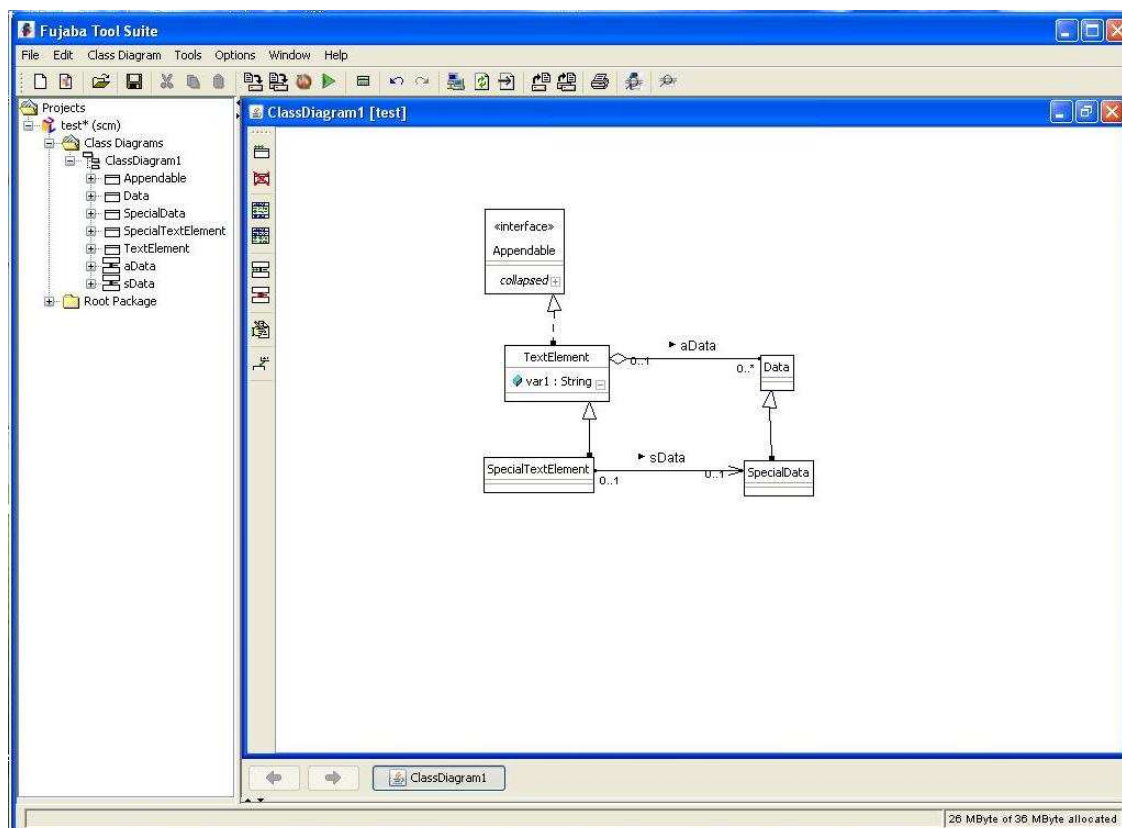


Figure 25: UML Class Diagram in "Fujaba"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	31.79%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	6.52%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	27.38%
Use case diagram	21.05%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	14.44%
Traceability	10.00%
Code generation	21.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	27.10%
L1.4	27.14%
L2-0	49.37%
L2-M	38.22%
L2-1	15.69%
L2-2	15.28%
L2-3	14.47%
Assigned level	-

5.21 Gaphor by Gaphor Team at Sourceforge

Name: Gaphor

Version: 0.12.5

Vendor: Gaphor Team

URL: <http://gaphor.devjavu.com/>

Price (if available): Open Source (GPL)

Vendor statement: *Gaphor is an easy to use modelling environment. This means that you are able to create nice UML diagrams for documentation and to assist you with design decisions. Gaphor will help you create your applications. Gaphor has an UML 2.0 compliant data model.*

User interface: direct editing of diagram elements in the diagram, menu, toolbar and drag & drop

Comments: In particular, the usability could be improved and unnecessary mouse commands could be avoided.

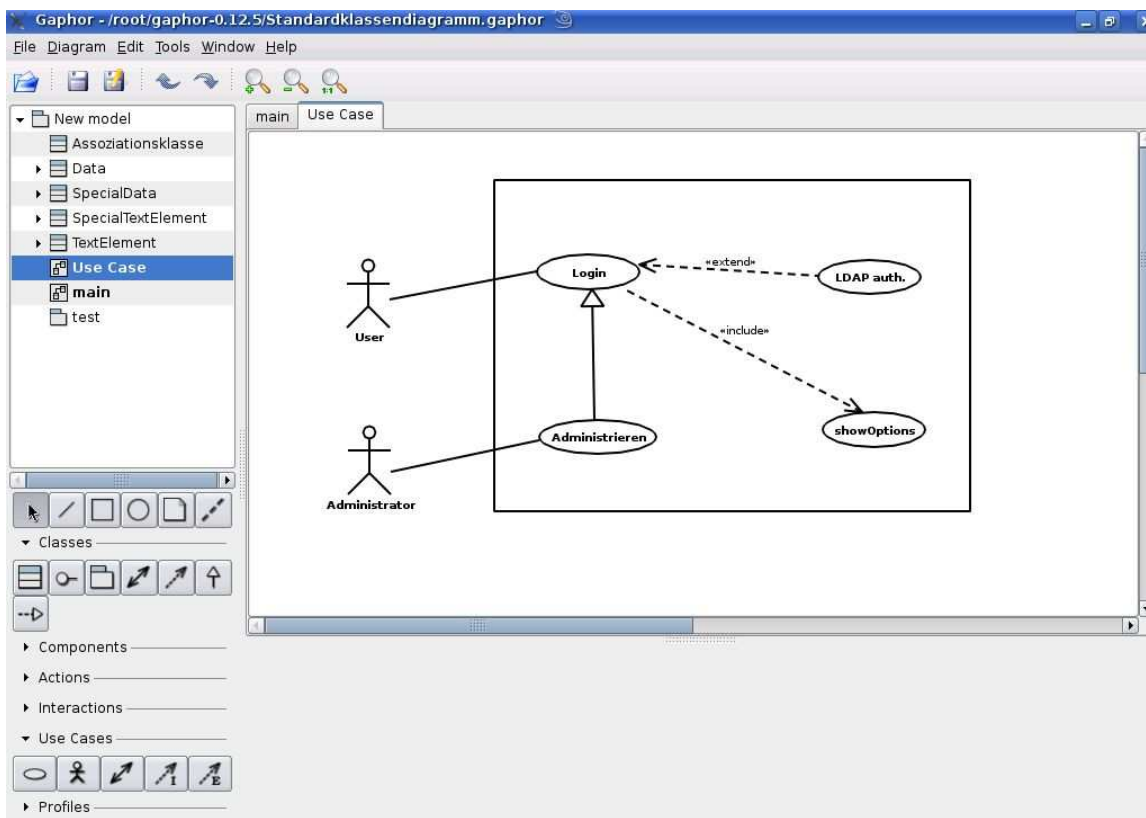


Figure 26: UML Use Case Diagram in "Gaphor"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	34.44%
Component diagram	29.41%
Composition diagram	40.00%
Deployment diagram	0.00%
Activity diagram	19.57%
Sequence diagram	20.00%
Communication diagram	53.85%
Interaction overview diagram	62.50%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	52.63%
Information flows	0.00%
Model management	16.67%
Templates	0.00%
Profiles	50.00%
UML summary	24.52%
Traceability	0.00%
Code generation	4.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	1.2
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	37.61%
L1.4	37.37%
L2-0	39.24%
L2-M	40.84%
L2-1	30.42%
L2-2	25.73%
L2-3	24.57%
Assigned level	-

5.22 Gliffy by Gliffy, Inc.

Name: Gliffy

Version: 13/01/2009

Vendor: Gliffy, Inc.

URL: <http://www.gliffy.com/online.shtml>

Price (if available): online tool, not available as download, 5\$/month

Vendor statement: *With Gliffy online diagram software, you can easily create professional-quality flowcharts, diagrams, floor plans, technical drawings, and more.*

Our online diagram editor makes it easier than ever to create great looking drawings.

User interface: views, direct editing of diagram elements in the diagram, menu, toolbar and drag & drop

Comments: The program is only available in as online version. It offers very few UML elements, which are not grouped according to subjects or diagrams in the toolbar. The evaluator noted as a usability issue that a specific button needs to be pushed to (re)activate the cursor.

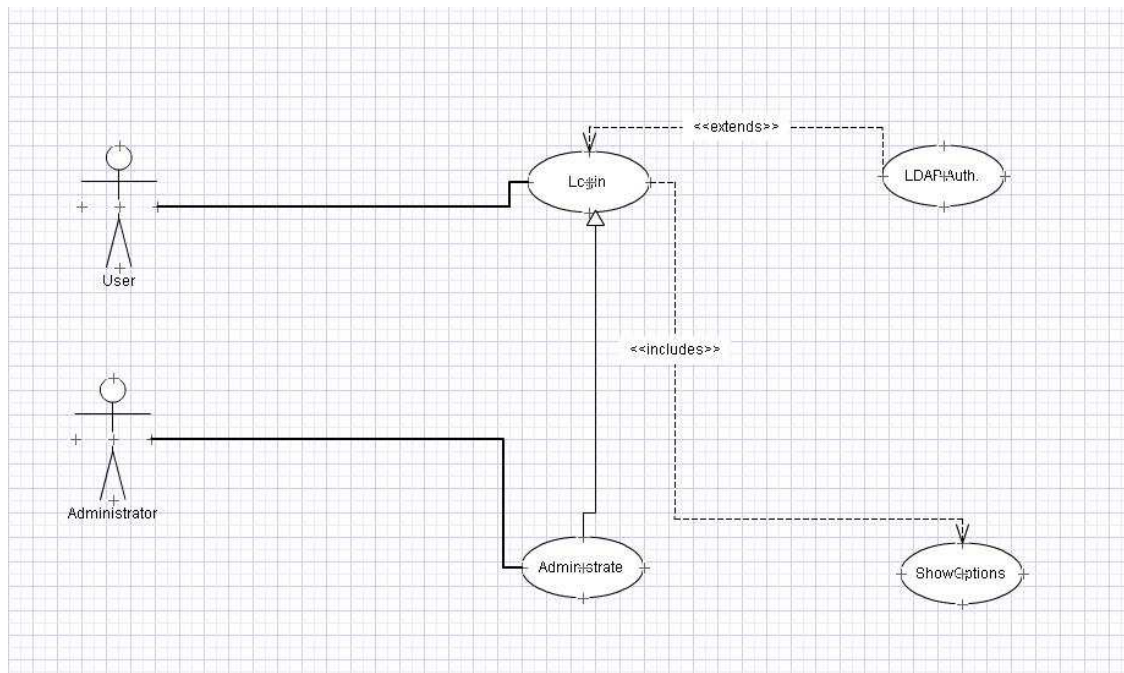


Figure 27: UML Use Case Diagram in "Gliffy"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	22.52%
Component diagram	23.53%
Composition diagram	0.00%
Deployment diagram	30.30%
Activity diagram	0.00%
Sequence diagram	28.89%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	55.26%
Information flows	28.57%
Model management	16.67%
Templates	0.00%
Profiles	0.00%
UML summary	15.82%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	27.10%
L1.4	26.93%
L2-0	43.04%
L2-M	30.37%
L2-1	17.08%
L2-2	16.07%
L2-3	15.85%
Assigned level	-

5.23 Green UML by University of Buffalo

Name: Green UML

Version: 3.5

Vendor: University of Buffalo

URL: <http://green.sourceforge.net>

Price (if available): Open Source (EPL)

Vendor statement: *Green is a LIVE round-tripping editor, meaning that it supports both software engineering and reverse engineering. You can use green to create a UML class diagram from code, or to generate code by drawing a class diagram.*

User interface: dialogs, menu, toolbar and direct editing of diagram elements in the diagram

Comments: Detail changes like visibilities can be performed only by changing the generated code and executing a manual refresh. Attributes in class diagrams appear in the wrong compartment.

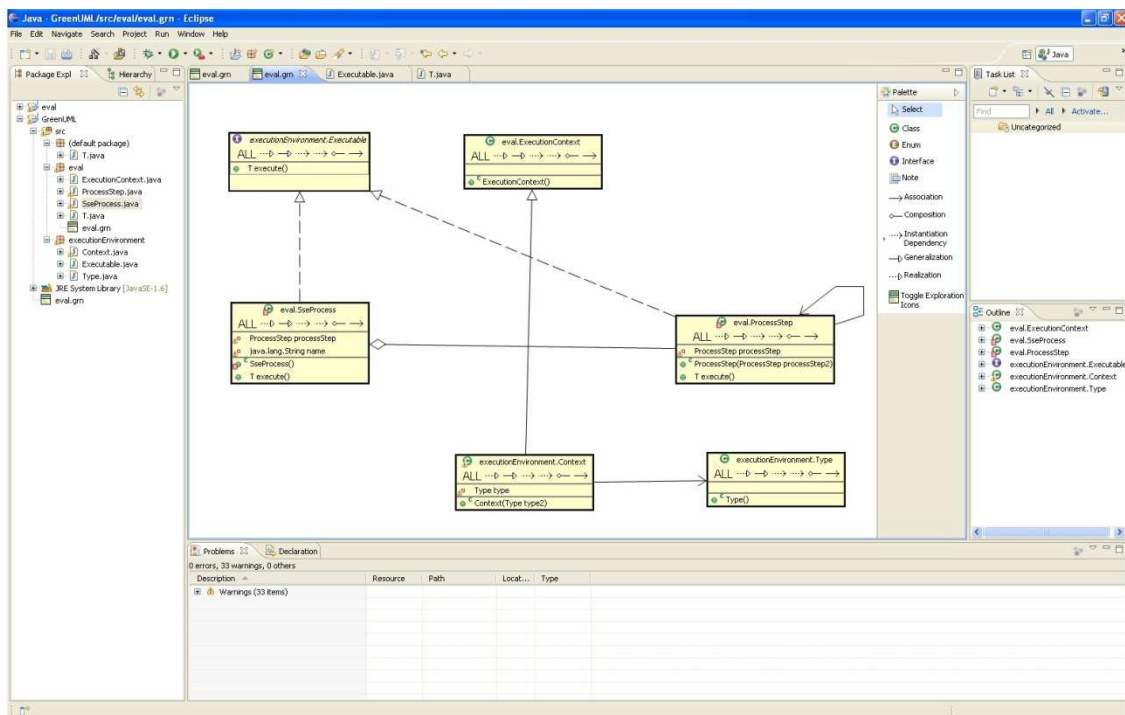


Figure 28: UML Class Diagram in "Green UML".

Due to class modeling problems, the reference diagram was not completed.

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	26.82%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	8.60%
Traceability	0.00%
Code generation	8.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	15.76%
L1.4	15.66%
L2-0	45.57%
L2-M	35.08%
L2-1	11.53%
L2-2	9.33%
L2-3	8.83%
Assigned level	-

5.24 Ideogramic UML by Ideogramic

Name: Ideogramic UML

Version: 2.3.3

Vendor: Ideogramic

URL: <http://www.ideogramic.com/products/tour.html>

Price (if available): 5195\$

Vendor statement: *Ideogramic UML™ is a powerful but lightweight tool for creating UML diagrams. Unlike general drawing programs, Ideogramic UML™ has been designed specifically for UML diagramming, and thus offers an interaction that is much more intelligent and much faster. And unlike heavyweight CASE tools with bloated, hard-to-learn interfaces Ideogramic UML™ offers just the features that you need resulting in a tool that stimulates creativity instead of hindering it.*

User interface: toolbars, direct editing of diagram elements in the diagram and gesture control

Comments: Difficult to use because of a small cursor.

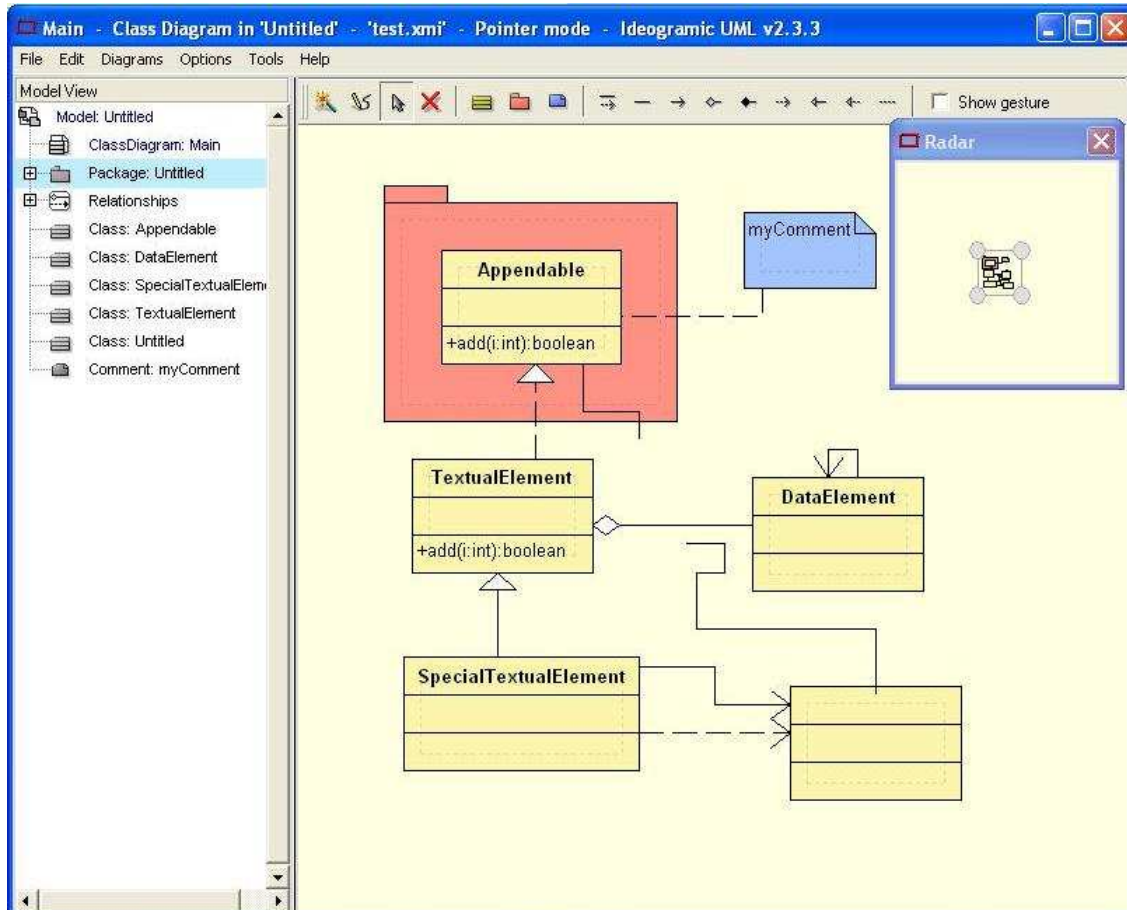


Figure 29: UML Class Diagram in "Ideogramic"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	9.27%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	15.56%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	15.48%
Use case diagram	44.74%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	7.75%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	1.0
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	14.71%
L1.4	14.61%
L2-0	18.99%
L2-M	13.09%
L2-1	8.61%
L2-2	8.43%
L2-3	7.98%
Assigned level	-

5.25 Innovator by MID

Name: Innovator

Version: 2008 10.0.03 Object eXcellence

Vendor: MID

URL: <http://www.mid.de/products/innovator.php3>

Price (if available): unknown

Vendor statement: *As well as business process modeling, object-oriented and structured software analysis, Innovator also supports object-oriented design and data modeling. Innovator is especially suited for model-driven software development as it enables domain-specific language extensions using UML 2 profiles.*

User interface: dialogs, toolbar, direct editing of diagram elements in the diagram and menu

Comments: Innovator is realized according to the client-server model and uses a role based permission model. The user must open many windows to access the intended parts of the model. The usability in particular when creating relationships can be improved. Several mechanisms try to ensure the consistency of the models. The concept of the program requires stepwise refinement of the models to implementation diagrams, which offer only a reduced set of UML elements. The program provides explicit support functions for development according to V-Model XT. The code generation is based on Open Architecture Ware (www.openarchitectureware.org).

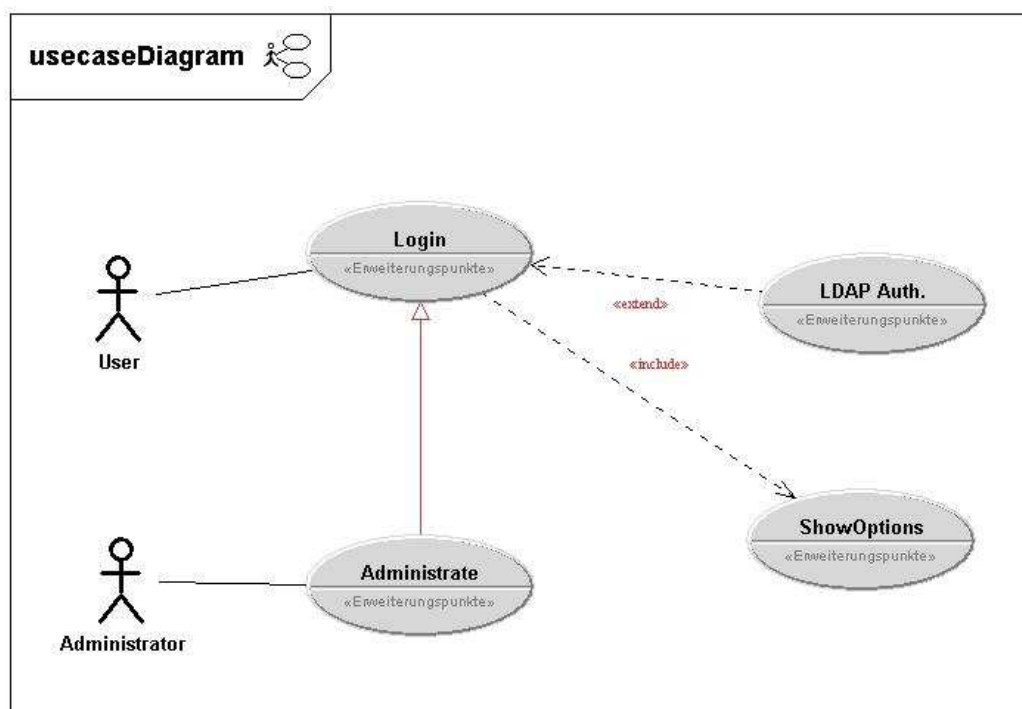


Figure 30: UML Use Case Diagram in “Innovator“

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	54.97%
Component diagram	26.47%
Composition diagram	70.00%
Deployment diagram	42.42%
Activity diagram	67.39%
Sequence diagram	67.78%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	69.05%
Use case diagram	71.05%
Information flows	0.00%
Model management	66.67%
Templates	0.00%
Profiles	20.00%
UML summary	49.68%
Traceability	30.00%
Code generation	19.00%

Tool characteristics	
Promoted UML version	1.4/2.1
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	64.71%
L1.4	64.51%
L2-0	70.89%
L2-M	66.49%
L2-1	51.39%
L2-2	51.69%
L2-3	49.79%
Assigned level	partial L2-0C

5.26 Javelin by Step Ahead software

Name: Javelin

Version: 7.1.1.3

Vendor: Step Ahead software

URL: <http://www.stepahead.com.au/products/javelin/javelin.htm>

Price (if available): unknown

Vendor statement: *Javelin™ has been the world's most intuitive, easy to learn, lightweight, visual modeler/coder tools for Java™ since 1996 because it makes developing object oriented code so natural, easy and efficient. It takes care of all the underlying java source files and presents you with an uncluttered, intuitive visual representation of your model and code where you can concentrate on designing and coding classes and the relationships between them visually instead of battling through a sea of text files and navigating package directories. Javelin uses UML notation for its class diagrams but that does not mean that Javelin is a UML tool or that you have to know UML to use the tool - which makes it most unlike typical heavyweight UML tools.*

User interface: dialogs and toolbars

Comments: The tool implements its own variant of the UML notation. The usability could be improved. In particular the handling of relations should be improved: One must grab the starting element with the mouse in the upper part of the "class/interface", drag it to the ending element and select the type of relationship being created in a dialog window.

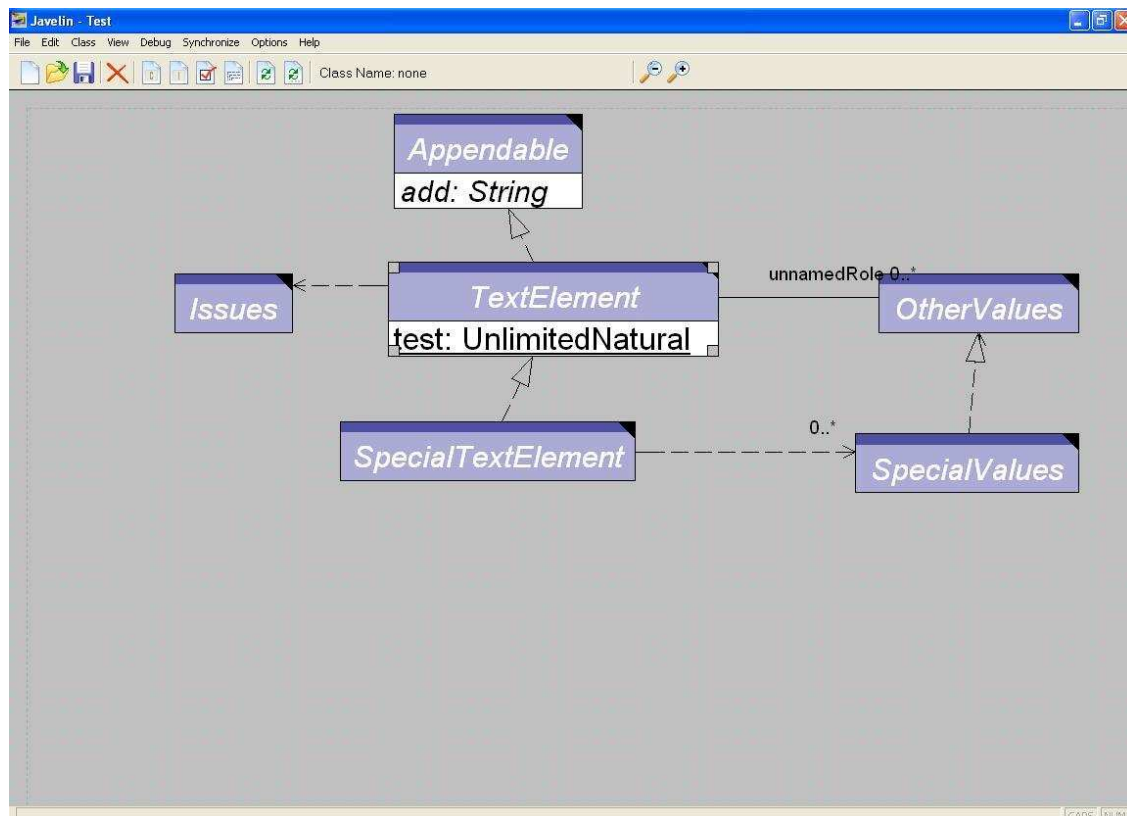


Figure 31: UML Class Diagram in "Javelin"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	18.87%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	6.05%
Traceability	0.00%
Code generation	9.00%

Tool characteristics	
Promoted UML version	UML-like notation
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	11.13%
L1.4	11.06%
L2-0	25.32%
L2-M	24.08%
L2-1	8.19%
L2-2	6.63%
L2-3	6.28%
Assigned level	-

5.27 Jude by ChangeVision

Name: Jude

Version: 5.1b1 community

Vendor: ChangeVision

URL: <http://jude.change-vision.com/jude-web/index.html>

Price (if available): 206\$

Vendor statement: *JUDE/Professional is the System Design Tool lets you draw UML, ER, Flowchart, CRUD, Data Flow Diagram and Mind Map. It has enriched features, such as input-output and diagram creation guidance. It is suitable for business use, large-sized models, and document creation.*

User interface: views, menu and toolbar

Comments: -

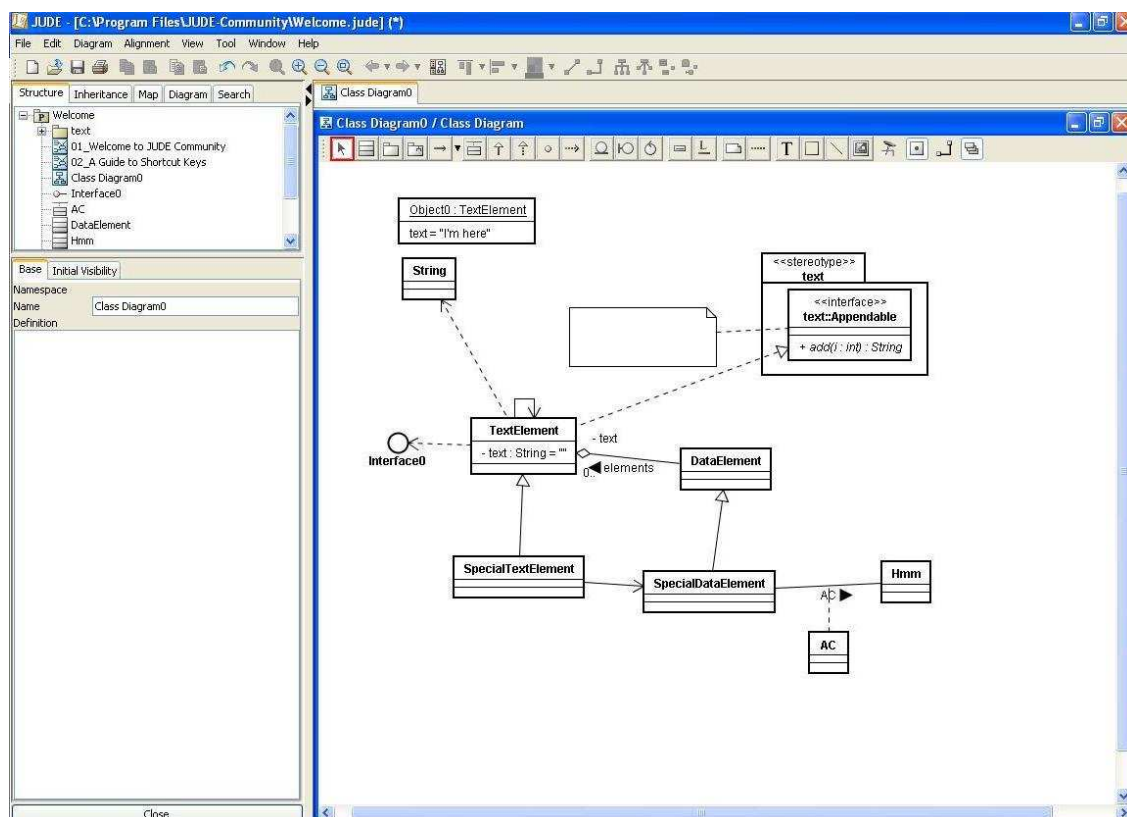


Figure 32: UML Class Diagram in "Jude"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	45.03%
Component diagram	44.12%
Composition diagram	0.00%
Deployment diagram	30.30%
Activity diagram	26.09%
Sequence diagram	40.00%
Communication diagram	69.23%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	48.81%
Use case diagram	68.42%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	34.82%
Traceability	20.00%
Code generation	10.00%

Tool characteristics	
Promoted UML version	1.x
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	60.92%
L1.4	60.96%
L2-0	59.49%
L2-M	50.79%
L2-1	37.08%
L2-2	36.85%
L2-3	34.89%
Assigned level	partial L2-0C

5.28 MagicDraw by NoMagic

Name: MagicDraw

Version: 16.0 Enterprise SP1

Vendor: NoMagic

URL: <http://www.magicdraw.com>

Price (if available): 125€ - 1355€

Vendor statement: *MagicDraw is an award-winning business process, architecture, software and system modeling tool with teamwork support. Designed for Business Analysts, Software Analysts, Programmers, QA Engineers, and Documentation Writers, this dynamic and versatile development tool facilitates analysis and design of Object Oriented (OO) systems and databases. It provides the industry's best code engineering mechanism (with full round-trip support for Java, C++, C#, CL (MSIL) and CORBA IDL programming languages), as well as database schema modeling, DDL generation and reverse engineering facilities.*

User interface: dialogs, views, direct editing of diagram elements in the diagram and toolbar

Comments: Provides several views on the user interface to adjust the available functionality. The user may add class properties or operations by clicking on a special symbol at the border of the visible class area. This leads to the creation of an appropriate model element having a default name which must be edited afterwards. Thereby, sometimes unnecessary elements are created and remain in the model. The evaluator noticed that the usability could be improved by directly opening a specification dialog or a similar mechanism. Some inconsistencies to the documentation have been detected.

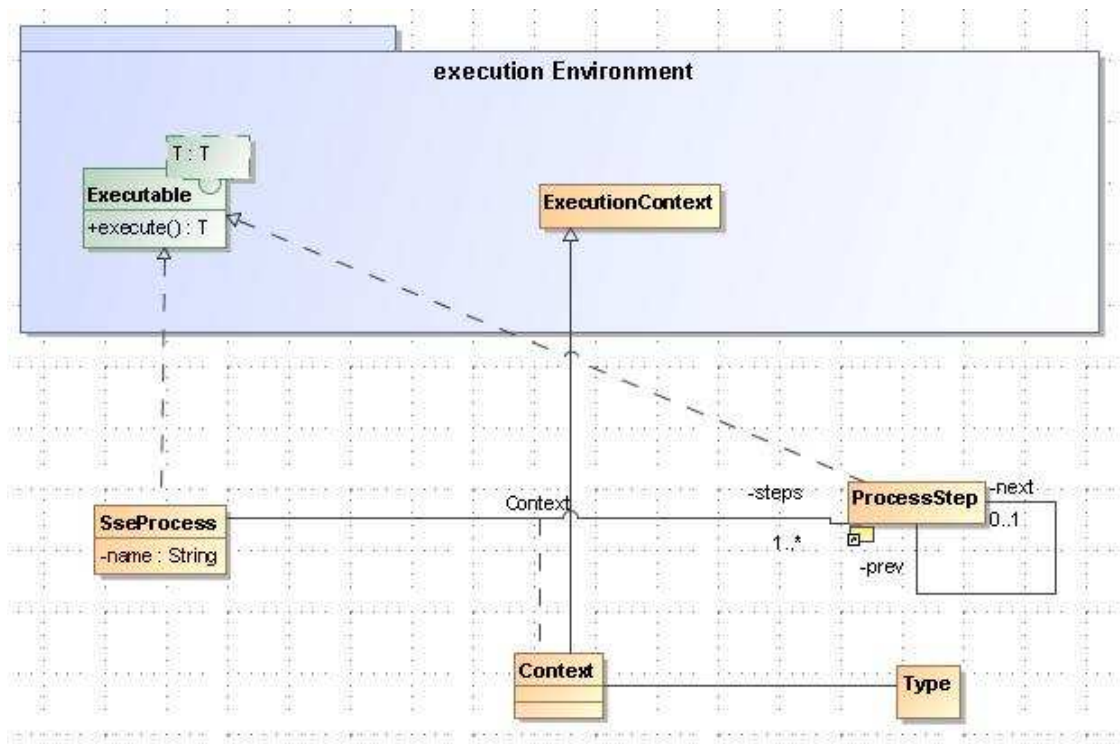


Figure 33: UML Class Diagram in "MagicDraw"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	88.41%
Component diagram	55.88%
Composition diagram	100.00%
Deployment diagram	96.97%
Activity diagram	82.61%
Sequence diagram	71.11%
Communication diagram	84.62%
Interaction overview diagram	56.25%
Timing diagram	0.00%
State machine diagram	80.95%
Use case diagram	92.11%
Information flows	0.00%
Model management	100.00%
Templates	100.00%
Profiles	80.00%
UML summary	78.98%
Traceability	90.00%
Code generation	23.00%

Tool characteristics	
Promoted UML version	2.0%
XMI version	2.0%
XMI valid	va
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0AC, L2-M
L1.3	92.6%
L1.4	93.3%
L2-0	97.4%
L2-M	94.2%
L2-1	77.9%
L2-2	79.6%
L2-3	79.2%
Assigned level	L

5.29 MEGA development by MEGA international

Name: MEGA development

Version: 2009 SP 1 patch 3.0 721-2496

Vendor: MEGA international

URL: <http://www.mega.com/us/products/megasuite/development.htm>

Price (if available): unknown

Vendor statement: *The MEGA Suite provides repository-based modeling tools to support projects ranging from process analysis to risk and control mapping to application analysis and design.*

User interface: dialogs, toolbar, direct editing of diagram elements in the diagram, and menu

Comments: The tool allows at maximum one stereotype per model element (the UML allows multiple). Property values for applied stereotypes are not supported (except for explicit meta model changes). In some cases the evaluation was hindered, because some relevant elements like packages or notes must be made available by the user in a configuration setting so that they appear in the tool palettes for all diagram types. The program allows the simulation of flowcharts and business process diagrams. The program tries to ensure consistency by providing model checks, but some of the offered checks appeared blurry to the evaluator. The edge routing in the diagram editor does not properly avoid overlays of edges.

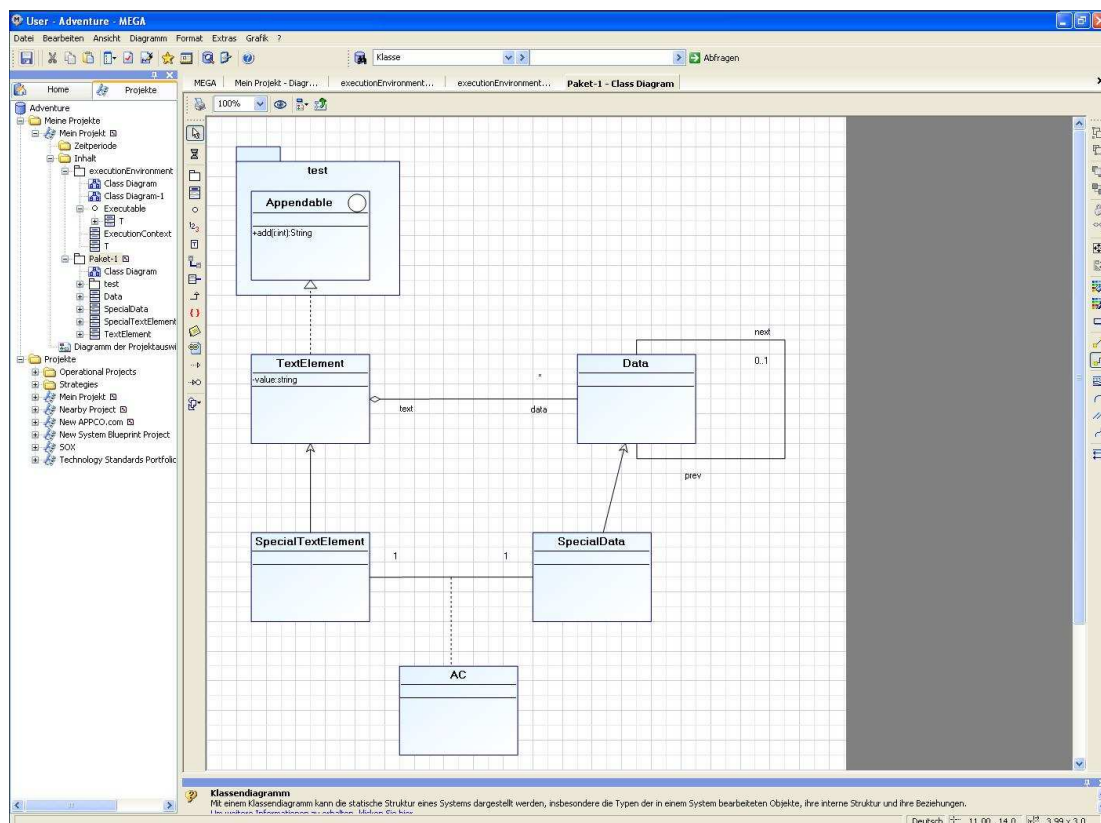


Figure 34: UML Class Diagram in "MEGA development"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	61.26%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	20.29%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	41.67%
Use case diagram	65.79%
Information flows	0.00%
Model management	16.67%
Templates	0.00%
Profiles	0.00%
UML summary	29.30%
Traceability	40.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	49.58%
L1.4	49.48%
L2-0	77.22%
L2-M	63.35%
L2-1	33.06%
L2-2	30.69%
L2-3	29.26%
Assigned level	L2-0C

5.30 MetaEdit+ by MetaCase

Name: MetaEdit+

Version: 4.5

Vendor: MetaCase

URL: <http://www.metacase.com/>

Price (if available): 4500\$

Vendor statement: *MetaEdit+ offers full modeling tool support for your language. Your whole team can immediately start to edit designs as graphical diagrams, as matrices or as tables, switching between views according to your needs. You can browse designs with filters, apply components, link your models to other designs, and check your models with various pre-defined or user-defined reports.*

User interface: dialogs, menus, toolbars and drag & drop

Comments: The tool allows the creation of user-defined meta models and is often used to model domain specific languages and, thus, does not really focus on the realization of UML.

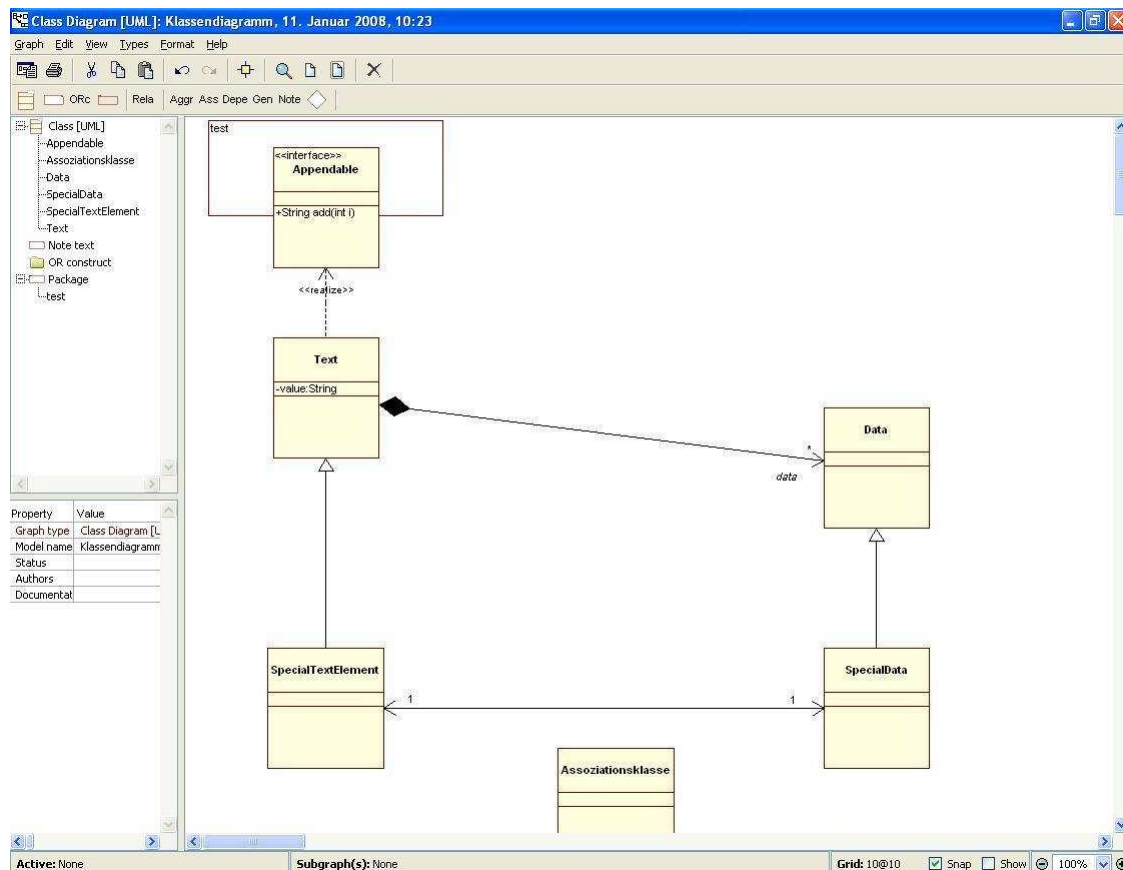


Figure 35: UML Class Diagram in "MetaEdit+"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	53.64%
Component diagram	82.35%
Composition diagram	70.00%
Deployment diagram	33.33%
Activity diagram	17.39%
Sequence diagram	33.33%
Communication diagram	69.23%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	42.86%
Use case diagram	73.68%
Information flows	0.00%
Model management	33.33%
Templates	50.00%
Profiles	0.00%
UML summary	40.23%
Traceability	0.00%
Code generation	22.00%

Tool characteristics	
Promoted UML version	-
XMI version	1.0
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	59.87%
L1.4	59.08%
L2-0	44.30%
L2-M	46.60%
L2-1	42.36%
L2-2	40.79%
L2-3	40.32%
Assigned level	-

5.31 Metamill by MetaMill

Name: MetaMill

Version: 5 build 860

Vendor: MetaMill

URL: <http://www.metamill.com/product.html>

Price (if available): 140\$

Vendor statement: *Metamill is a UML(tm) modeling software targeted for software engineering teams and individuals designing software intensive systems using UML as a modeling language.*

Free evaluation version is available, i.e. you can try it before purchasing it. With Metamill you can capture business requirements using use case diagrams, design software architecture using package diagrams and design components using class diagrams, object diagrams, composite structure diagrams and component diagrams. For dynamic modeling you can use sequence diagrams, statemachine, communication, activity and timing diagrams. UML 2.1 is supported since Metamill version 5.0.

User interface: dialogs, views, menu, tools and direct editing of diagram elements in the diagram

Comments: The tool allows at most one stereotype per model element (UML allows multiple stereotypes). Some stereotypes are not properly assigned to their metamodel elements as given in the annex of the UML. The primitive types as specified in the UML are not defined.

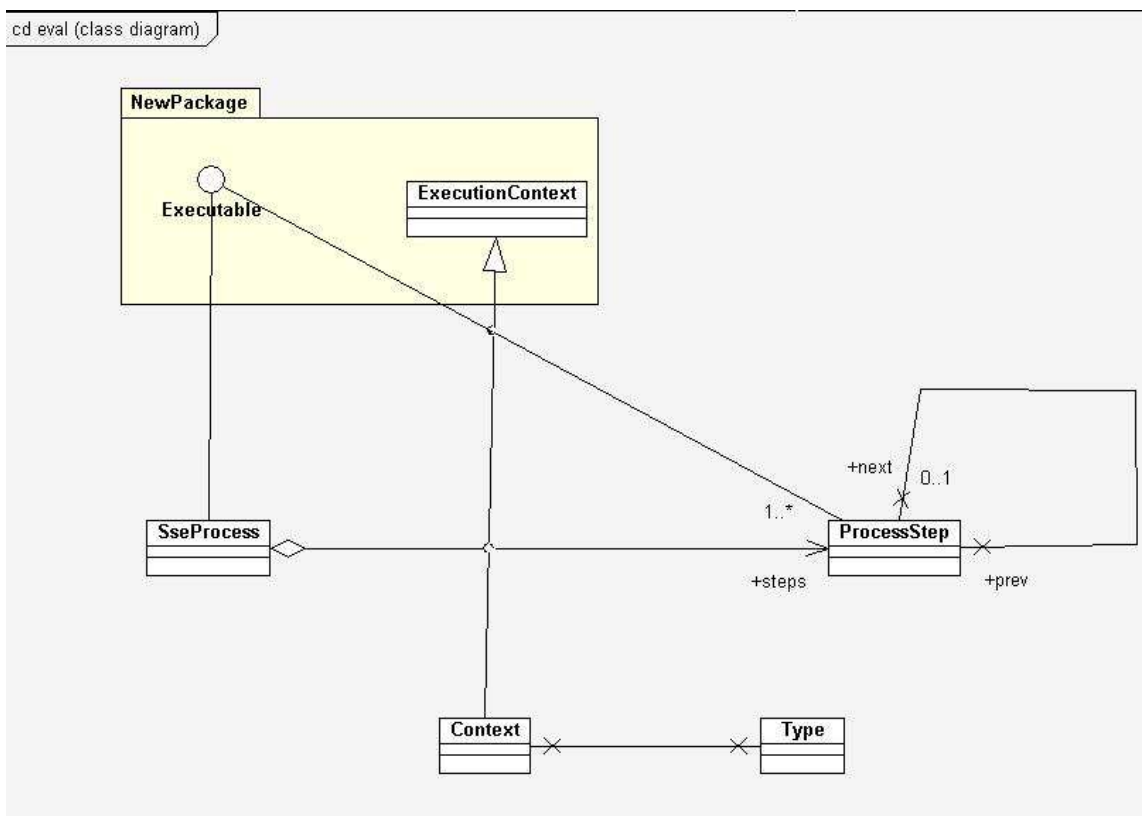


Figure 36: UML Class Diagram in "Metamill"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	56.95%
Component diagram	67.65%
Composition diagram	90.00%
Deployment diagram	77.27%
Activity diagram	40.58%
Sequence diagram	72.22%
Communication diagram	46.15%
Interaction overview diagram	50.00%
Timing diagram	47.37%
State machine diagram	52.38%
Use case diagram	73.68%
Information flows	0.00%
Model management	66.67%
Templates	0.00%
Profiles	0.00%
UML summary	54.35%
Traceability	0.00%
Code generation	22.00%

Tool characteristics	
Promoted UML version	2.1
XMI version	1.1, 1.2, 1.3, 2.0 and 2.1
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC, L2-1AC
L1.3	70.59%
L1.4	71.19%
L2-0	75.95%
L2-M	63.87%
L2-1	56.67%
L2-2	56.52%
L2-3	54.36%
Assigned level	partial L2-1AC

5.32 MyEclipseIDE by genuitec

Name: MyEclipseIDE

Version: 8.0 - 20091120

Vendor: genuitec

URL: <http://www.myeclipseide.com/>

Price (if available): 31.75\$ - 158.95\$

Vendor statement: *MyEclipse is the most comprehensive Java EE / J2EE IDE for the open source Eclipse platform, period. MyEclipse incorporates today's most innovative open-standard technologies to provide a development environment for J2EE WEB, XML, UML and databases and a wide array of application server connectors to streamline development, deployment, testing and portability.*

The vendor of this tool did not respond to a publication request in terms of an explicit opt-in.

This page is left intentionally blank.

5.33 Netbeans by SUN

Name: Netbeans

Version: 5.5.1 with UML Module 1.1.14

Vendor: SUN

URL: <http://www.netbeans.org>

Price (if available): Open Source (CDDL, GPL)

Vendor statement: *A free, open-source Integrated Development Environment for software developers. You get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java language, C/C++, and Ruby.*

User interface: dialogs, views, direct editing of diagram elements in the diagram and toolbar

Comments: The program is unstable in its behavior (hang-up, crash). The evaluator detected several usability problems, e.g. it is complicated to specify multiplicities. The inline diagram editor is very handy and provides good tooltip help. The arrangement of the diagram elements is not stored properly, so after closing the program, the evaluator was able to reread the model but the layout of the diagrams was not restored.

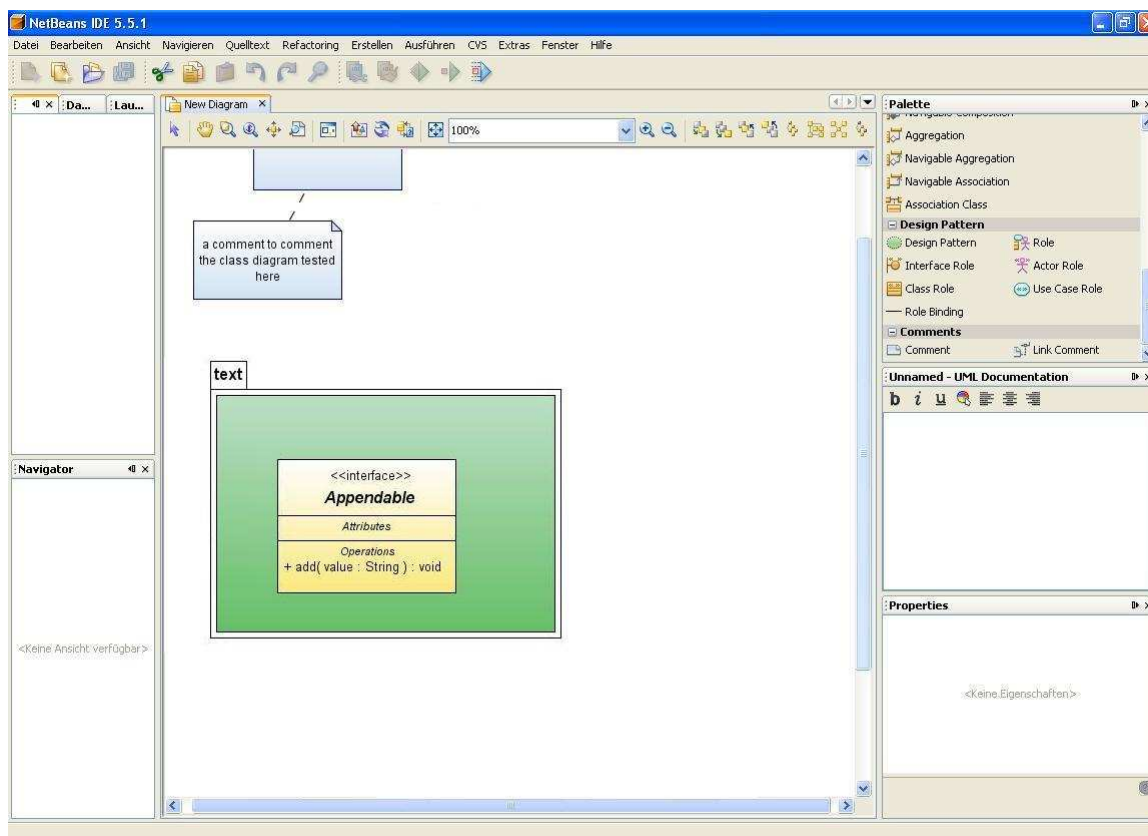


Figure 37: UML Class Diagram (incomplete) in "Netbeans"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	49.34%
Component diagram	32.35%
Composition diagram	45.00%
Deployment diagram	34.85%
Activity diagram	24.64%
Sequence diagram	40.00%
Communication diagram	19.23%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	50.00%
Use case diagram	52.63%
Information flows	0.00%
Model management	0.00%
Templates	29.17%
Profiles	0.00%
UML summary	35.67%
Traceability	10.00%
Code generation	2.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	53.78%
L1.4	53.86%
L2-0	74.68%
L2-M	55.50%
L2-1	36.67%
L2-2	36.97%
L2-3	35.74%
Assigned level	partial L2-0C

5.34 Objecteering/UML Free Edition by Objecteering Software

Name: Objecteering/UMLFree Edition

Version: 6.1.00

Vendor: Objecteering Software

URL: <http://www.objecteering.com/>

Price (if available): 0\$

Vendor statement: *Objecteering UML Free Edition is the freely downloadable, free-of-charge edition of the new Objecteering range (UML2 modeling, XMI import, documentation generation).*

User interface: dialogs, views, direct editing of diagram elements in the diagram, toolbar

Comments: There are several differences to the commercial product:

- The size of the model is limited to 50 elements.
- Groupwork services and the model versioning (diff/merge) as well as configuration management features are not supported.
- Code generation for CORBA IDL, SQL DDL and Fortrain are not available in the free version.

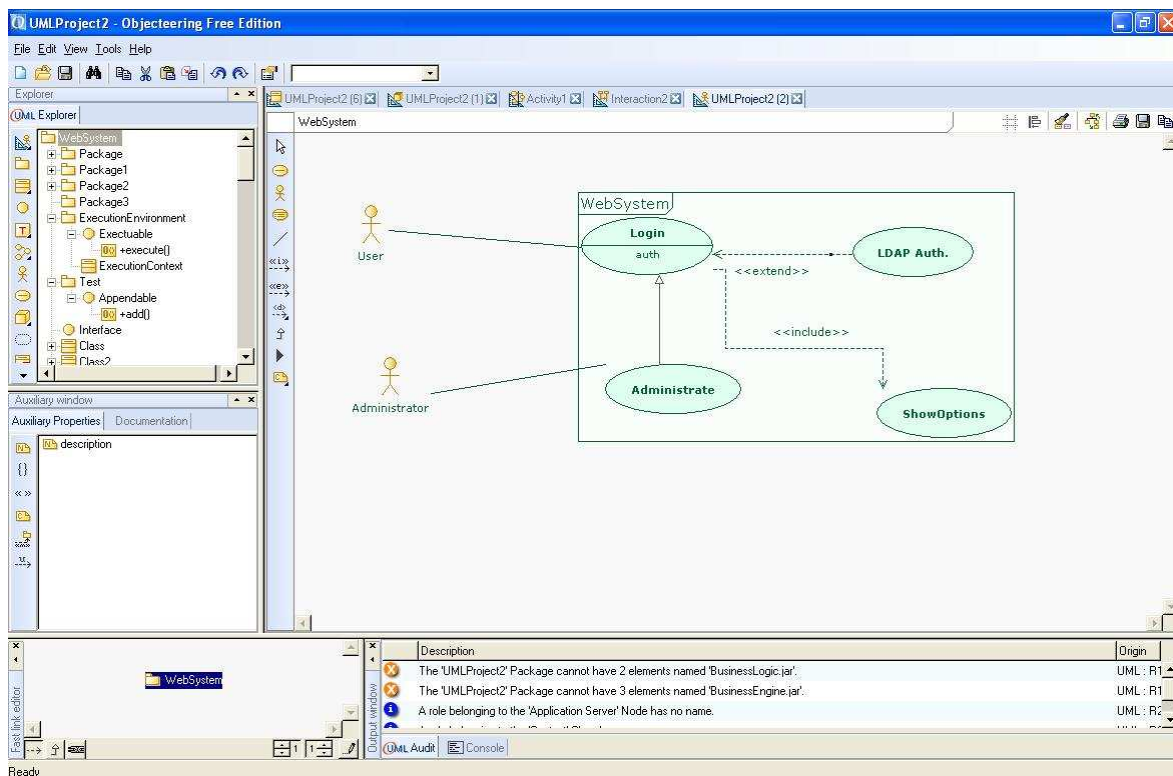


Figure 38: UML Use Case Diagram in “Objecteering/UML Free Edition”

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	62.25%
Component diagram	64.71%
Composition diagram	65.00%
Deployment diagram	45.45%
Activity diagram	43.48%
Sequence diagram	55.56%
Communication diagram	53.85%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	48.81%
Use case diagram	76.32%
Information flows	71.43%
Model management	83.33%
Templates	50.00%
Profiles	0.00%
UML summary	50.85%
Traceability	40.00%
Code generation	2.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	73.32%
L1.4	73.07%
L2-0	68.35%
L2-M	64.92%
L2-1	52.22%
L2-2	50.22%
L2-3	50.96%
Assigned level	partial L2-0C

5.35 objectiF by Microtool

Name: objectiF

Version: 7.0.133

Vendor: Microtool

URL: <http://www.microtool.de>

Price (if available): 2500 \$

Vendor statement: *In objectiF you will find everything you need for efficient development. This includes development of enterprise, SOA and web applications as well as client-server applications and embedded software.*

User interface: dialogs and toolbar

Comments: The UML notation is not well supported. Also the usability of the user interface and the diagram editor can be improved by considering accepted user interface conventions. The program opens a lot of different windows. The evaluator noted that MDA transformations from platform independent models to platform specific models are explicitly supported.

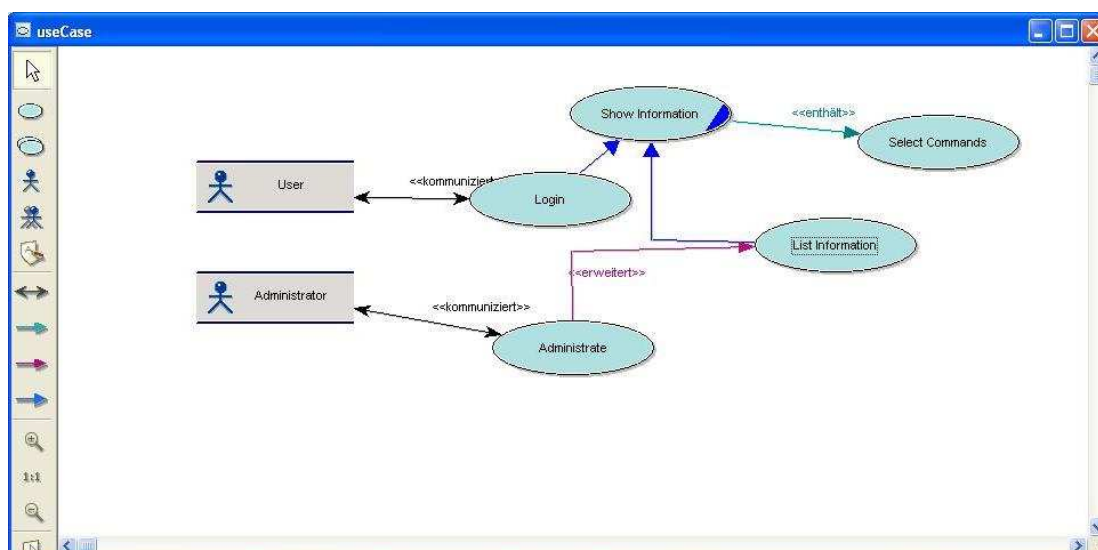


Figure 39: UML Use Case Diagram in “objectiF”

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	35.43%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	18.84%
Sequence diagram	20.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	52.38%
Use case diagram	31.58%
Information flows	0.00%
Model management	0.00%
Templates	16.67%
Profiles	10.00%
UML summary	22.61%
Traceability	30.00%
Code generation	21.00%

Tool characteristics	
Promoted UML version	pragmatic realization
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	40.55%
L1.4	40.29%
L2-0	48.10%
L2-M	43.46%
L2-1	22.64%
L2-2	23.48%
L2-3	22.66%
Assigned level	-

5.36 Omondo UML Plugin for Eclipse by Omondo

Name: Omondo UML Plugin for Eclipse

Version: 3.3.0.v20070629 2007 free

Vendor: Omondo

URL: <http://www.omondo.com>, <http://www.omondo.de>

Price (if available): unknown

Vendor statement: *EclipseUML Studio Edition is an advanced UML solution for Java and Jee modelers and developers.*

User interface: dialogs, menu and toolbar

Comments: The tool and its modeling options are dedicated for Java code generation, but the evaluator recorded code synchronization and generation problems, e.g. for compositions. The program is unstable (hang-up, crash). It does not support multiple stereotypes (as required by the UML). The elements in the toolbar are not properly categorized so that model elements are mixed among different diagram types. The tool introduces a new diagram type, the robustness diagram.

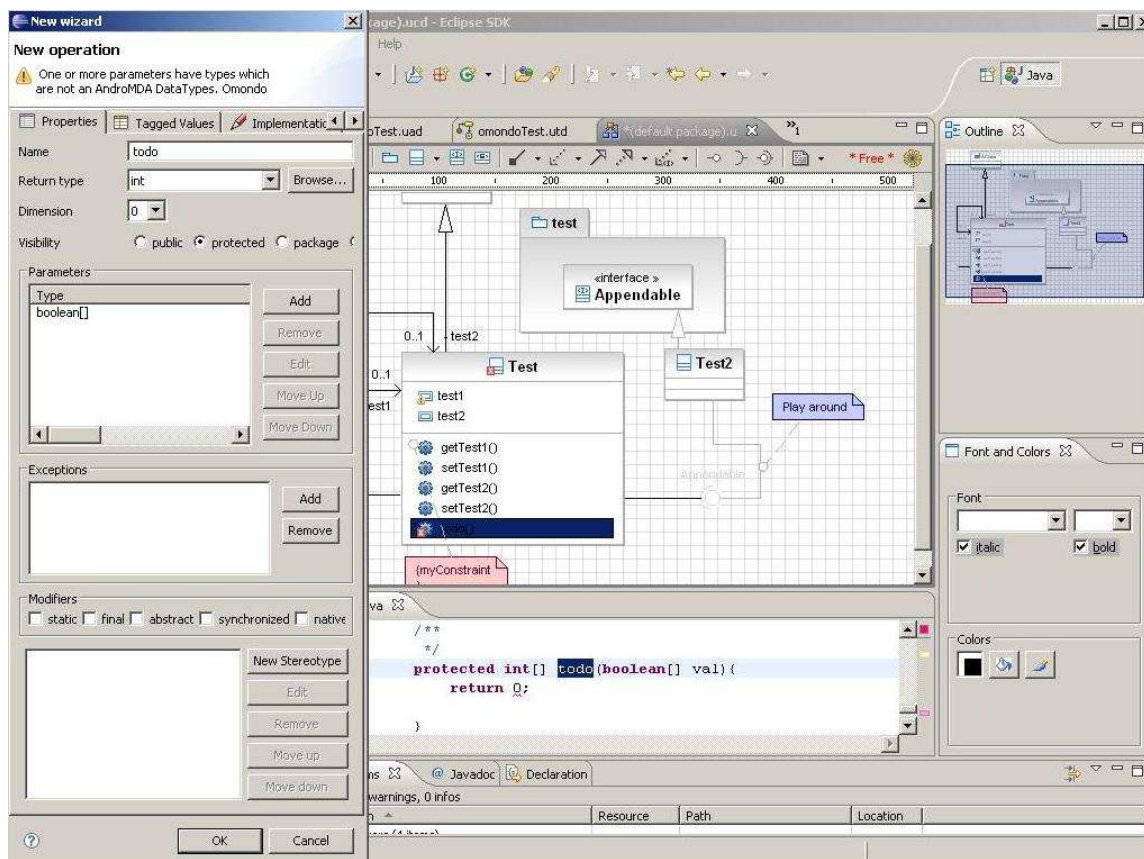


Figure 40: UML Class Diagram in "Omondo UML Plugin for Eclipse"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	55.63%
Component diagram	61.76%
Composition diagram	5.00%
Deployment diagram	31.82%
Activity diagram	31.88%
Sequence diagram	42.22%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	44.05%
Use case diagram	78.95%
Information flows	0.00%
Model management	0.00%
Templates	8.33%
Profiles	30.00%
UML summary	39.17%
Traceability	20.00%
Code generation	12.00%

Tool characteristics	
Promoted UML version	2.1
XMI version	2.1
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC, L2-MAC
L1.3	61.34%
L1.4	61.80%
L2-0	68.35%
L2-M	54.45%
L2-1	41.94%
L2-2	41.12%
L2-3	39.15%
Assigned level	partial L2-MAC

5.37 OODesigner by Tae Gyun Kim

Name: OODesigner

Version: 0401 2004-01-12

Vendor: Tae Gyun Kim

URL: <http://munjong.pufs.ac.kr/ktg/ood.htm>

Price (if available): free, no license

Vendor statement: *This tool is for supporting UML. Functionality: Class Diagram, Use Case Diagram..., C++/Java Code Generation, OLE container & server*

User interface: dialogs, direct editing of diagram elements in the diagram and toolbar

Comments: The usability can be improved significantly. Too many options are not available from the context menu. Editors or dialogs for changing the properties of model elements are missing. Often it is not possible to edit a model element directly after inserting it. The program is not able to read its own data files after storing them, i.e. the evaluator was not able to store and reread the evaluation model.

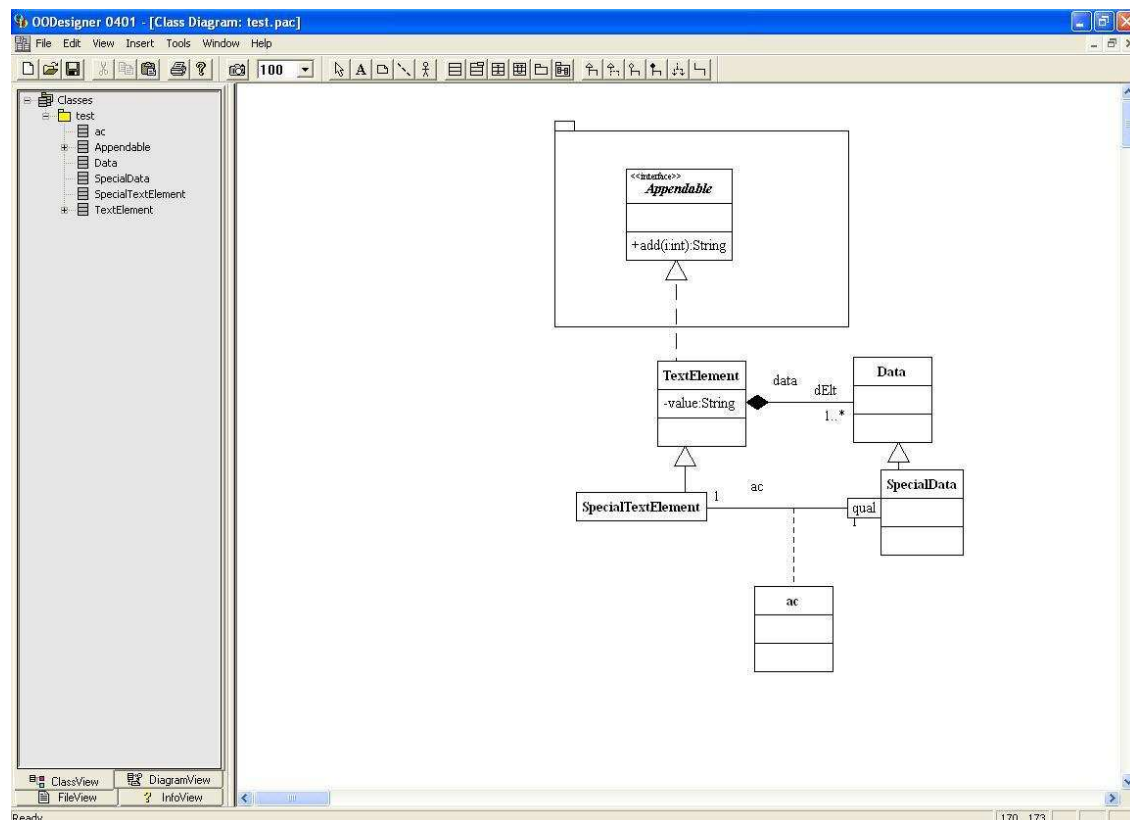


Figure 41: UML Class Diagram in "OODesigner"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	31.79%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	18.18%
Activity diagram	11.59%
Sequence diagram	22.22%
Communication diagram	19.23%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	19.05%
Use case diagram	42.11%
Information flows	0.00%
Model management	0.00%
Templates	4.17%
Profiles	0.00%
UML summary	19.32%
Traceability	0.00%
Code generation	12.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	35.71%
L1.4	35.07%
L2-0	44.30%
L2-M	36.65%
L2-1	21.25%
L2-2	20.34%
L2-3	19.36%
Assigned level	-

5.38 Open ModelSphere by Grandite

Name: Open ModelSphere

Version: 3.0 Build 904

Vendor: Grandite

URL: <http://www.modelsphere.com/modelsphere.html>

Price (if available): 0\$

Vendor statement: *Open ModelSphere - Grandite's Free Business Data and Process Modeling, Software Re-engineering and UML Tools Open a Large Area of Opportunities for Your Enterprise:*

- Support of multiple strategic and operational IT projects, e.g. business process re-engineering, mergers & acquisition, outsourcing, reorganization, evaluation of software packages, computer-aided software engineering (CASE), documentation of databases and applications, database migration
- Benefits in all phases of software development projects, e.g. analysis, business process modeling, conceptual data modeling, logical data modeling, design, physical data / database modeling, database / code generation, database / code reverse engineering

User interface: dialogs, views, direct editing of diagram elements in the diagram and toolbar

Comments: ModelSphere realizes only a subset of UML 1. The evaluator was not able to store the complete model created due to restrictions of the evaluation license.

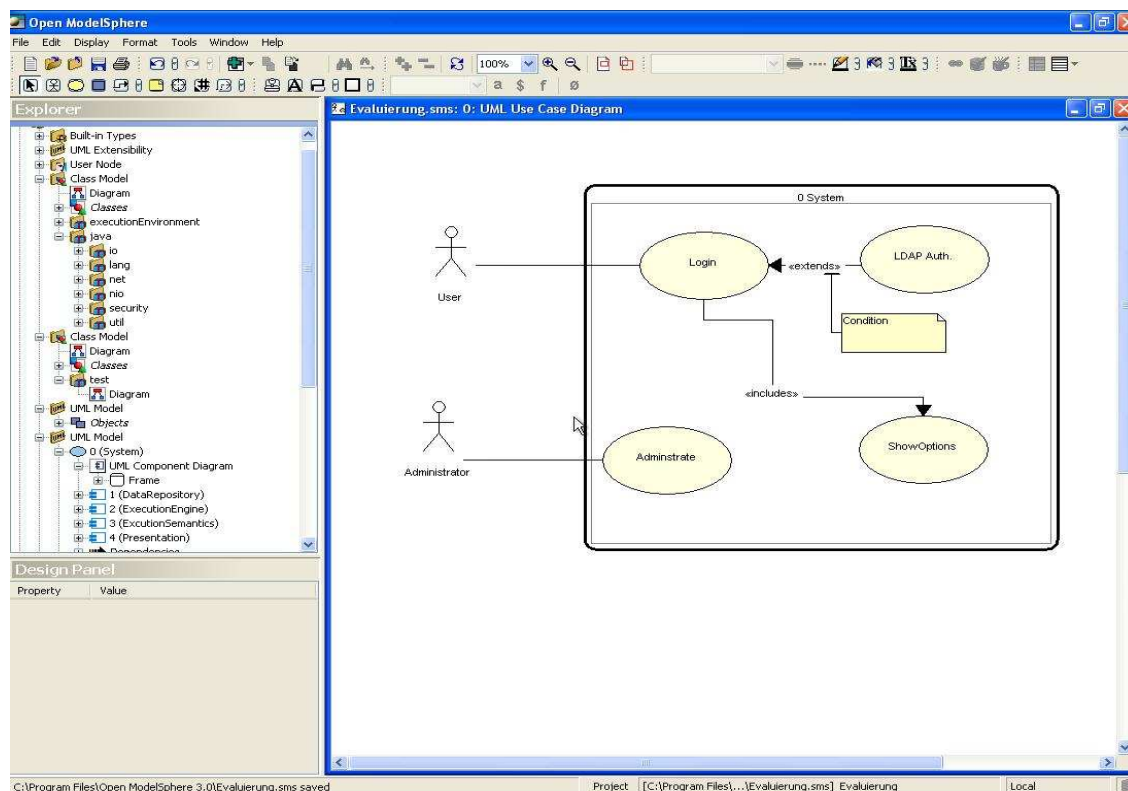


Figure 42: UML Use Case Diagram in "Open ModelSphere"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	46.36%
Component diagram	29.41%
Composition diagram	0.00%
Deployment diagram	22.73%
Activity diagram	20.29%
Sequence diagram	15.56%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	23.81%
Use case diagram	44.74%
Information flows	0.00%
Model management	50.00%
Templates	0.00%
Profiles	0.00%
UML summary	26.65%
Traceability	30.00%
Code generation	10.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	47.90%
L1.4	47.39%
L2-0	64.56%
L2-M	51.31%
L2-1	29.03%
L2-2	27.42%
L2-3	26.60%
Assigned level	partial L2-0C

5.39 OpenAmeos by ScopeSET

Name: OpenAmeos

Version: 10.1 (Build 26)

Vendor: ScopeSET

URL: <http://www.openameos.org/download>

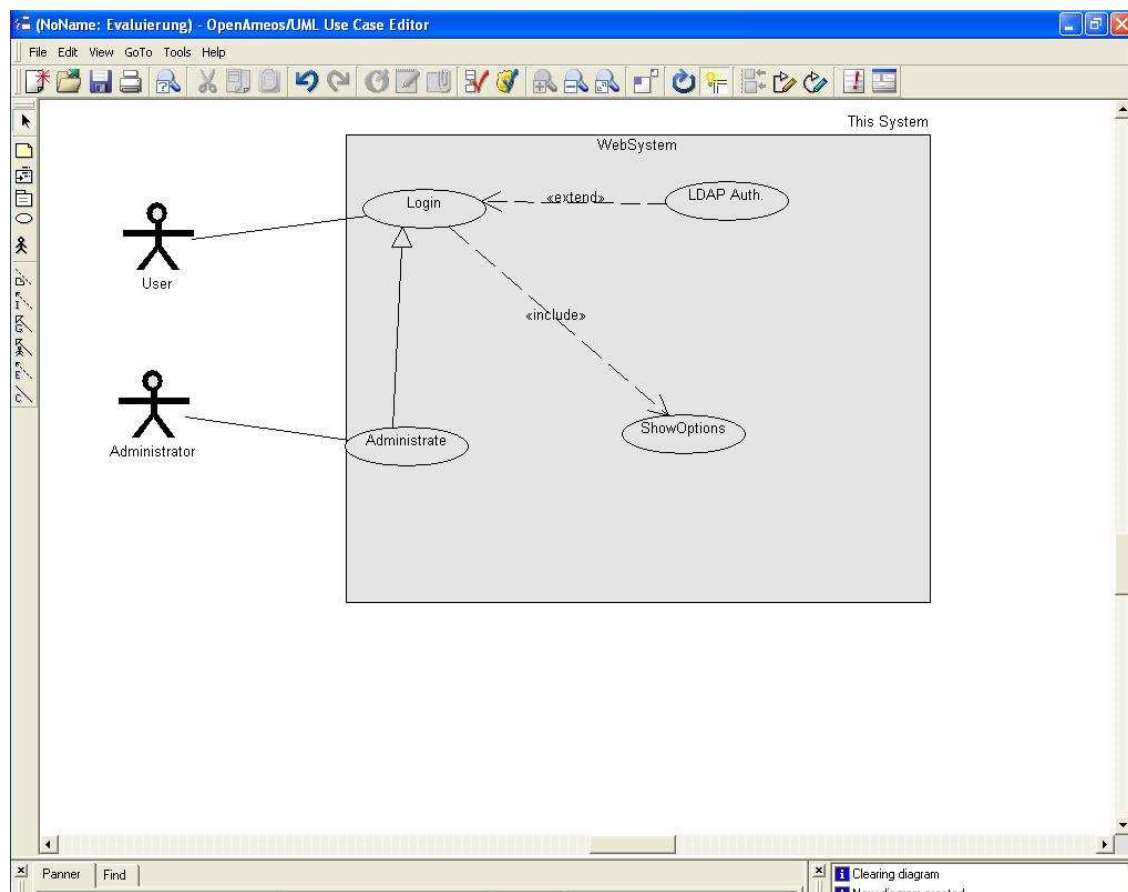
Price (if available): 0\$

Vendor statement: *OpenAmeos is the multi-platform/multi-user UML® Modeling Environment with unique support for UML 2.0 Profiles, MDA® based Model Transformation and support for color to visualize semantics.*

OpenAmeos supports all UML1.x diagram types and is presently in the process of moving towards UML2. MDA based code generation templates are available for Java, C++, C, Ada95, C#. Additional templates and profiles support platforms such as RavenScar or the Java High-Integrity-Platform (HIP).

User interface: dialogs, views, direct editing of diagram elements for some diagram elements, menu, toolbar and drag and drop menu

Comments: The usability can be improved, e.g. to simplify the access to the properties tab of a model element. Due to problems deleting elements in the example database, the evaluator was not able to assess the code synchronization or the repository management.



Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	48.68%
Component diagram	32.35%
Composition diagram	80.00%
Deployment diagram	27.27%
Activity diagram	22.46%
Sequence diagram	36.67%
Communication diagram	38.46%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	63.10%
Use case diagram	60.53%
Information flows	0.00%
Model management	0.00%
Templates	29.17%
Profiles	60.00%
UML summary	38.32%
Traceability	0.00%
Code generation	23.00%

Tool characteristics	
Promoted UML version	UML 2.0 profile support
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0C
L1.3	59.66%
L1.4	59.71%
L2-0	49.37%
L2-M	50.79%
L2-1	37.64%
L2-2	39.78%
L2-3	38.40%
Assigned level	partial L1.4

5.40 Papyrus UML by Papyrus UML Team

Name: Papyrus UML

Version: 1.6.2

Vendor: Papyrus UML Team

URL: <http://www.papyrus-uml.org>

Price (if available): Open Source (EPL)

Vendor statement: *Papyrus is a dedicated tool for modelling within UML2. This open source tool is based on the Eclipse environment.*

User interface: view, menu and toolbar

Comments: The tool was not able to handle more than one diagram per model so that for each diagram type to be tested an own model had to be created. The deletion operation sometimes removes too many elements. Use cases are strictly related to classes. Adjustment of the dimensions of the elements did not work properly. To enter parameter of an operation the user needs knowledge on the UML syntax of operations. The usability of several functionalities can be improved, because the user cannot intuitively run them without consulting the online videos.

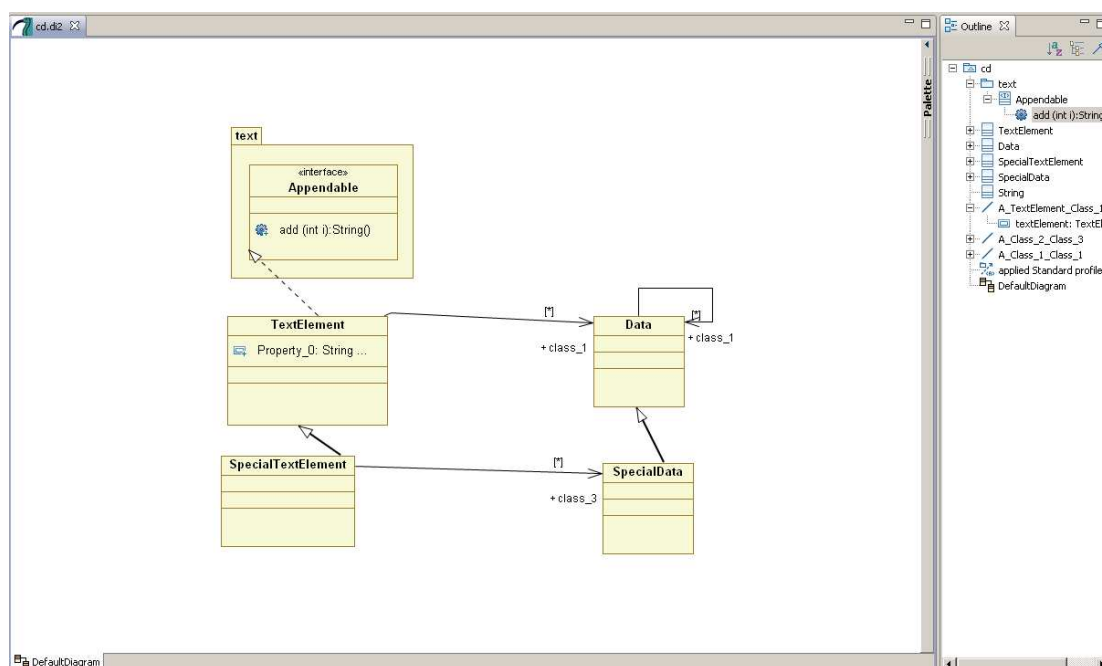


Figure 44: UML Class Diagram in “Papyrus UML“

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	65.56%
Component diagram	58.82%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	83.33%
Templates	83.33%
Profiles	70.00%
UML summary	27.92%
Traceability	20.00%
Code generation	13.00%

Tool characteristics	
Promoted UML version	2
XMI version	2.1
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0AC, L2-MAC
L1.3	37.61%
L1.4	37.79%
L2-0	87.34%
L2-M	85.34%
L2-1	30.28%
L2-2	26.07%
L2-3	27.87%
Assigned level	L2-MAC

5.41 Poseidon by Gentleware

Name: Poseidon

Version: 6.0.1

Vendor: Gentleware

URL: <http://www.gentleware.com>

Price (if available): 249\$-1549\$

Vendor statement: *Don't lose the advantage of clarity and simplicity to an overly complicated tool - the Poseidon for UML software line lets you get down to work without entanglements in your development environment. Powerful features such as round trip engineering and documentation generation have been intelligently implemented without the overhead common to so many other UML tools on the market today.*

User interface: views, direct editing of diagram elements in the diagram, menu and toolbar

Comments: Sometimes the tool adds superfluous elements while inserting a model element. The program does not provide a function to reuse previously inserted elements (in different diagrams). Toolbars do not always properly map to the current diagram in the editor. The usability of the stereotypes selection could be improved.

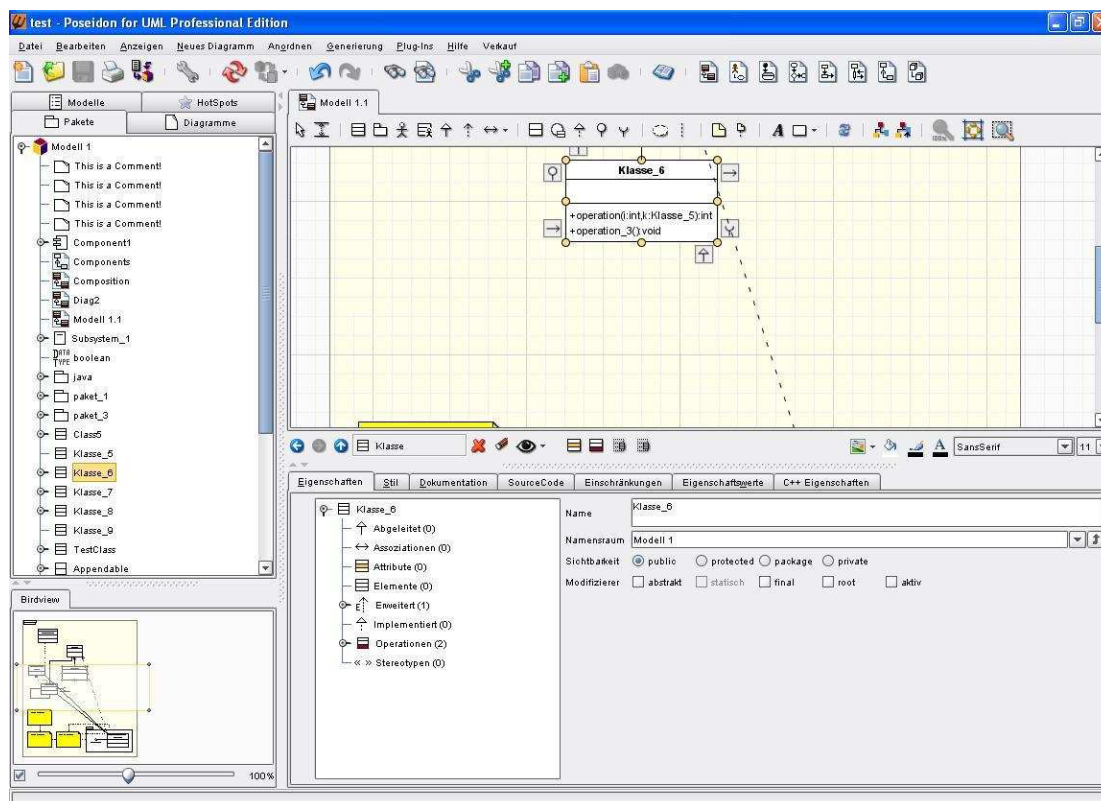


Figure 45: UML Class Diagram in "Poseidon"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	65.23%
Component diagram	73.53%
Composition diagram	60.00%
Deployment diagram	71.21%
Activity diagram	50.00%
Sequence diagram	80.00%
Communication diagram	34.62%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	67.86%
Use case diagram	81.58%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	55.20%
Traceability	30.00%
Code generation	28.00%

Tool characteristics	
Promoted UML version	All 9 UML diagrams
XMI version	1.2
XMI valid	Partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	75.42%
L1.4	75.78%
L2-0	70.89%
L2-M	61.78%
L2-1	57.78%
L2-2	58.43%
L2-3	55.32%
Assigned level	partial L2-0C

5.42 PowerDesigner by Sybase

Name: PowerDesigner

Version: 12.5.0.2169

Vendor: Sybase

URL: <http://www.sybase.com/products/powerdesigner/>

Price (if available): 5990\$

Vendor statement: *Sybase PowerDesigner 12.5, a model-driven approach to aligning business and IT, is an enterprise modeling and design solution that helps you implement effective enterprise architecture and brings powerful analysis and design techniques to your development lifecycle. PowerDesigner uniquely combines several standard data modeling techniques (UML, Business Process Modeling and market-leading data modeling) together with leading development platforms such as .NET, WorkSpace, PowerBuilder, Java™, Eclipse, etc., to bring business analysis and formal database design solutions to the traditional software development lifecycle. And it works with over 60 RDBMS.*

User interface: dialogs, menu and toolbar

Comments: The UML implementation appears to be an addition of UML 2 elements to an UML 1 modeling tool. Often options and functionality is (unnecessarily) hidden. The tool provides a very detailed transformation editor and an impact analysis.

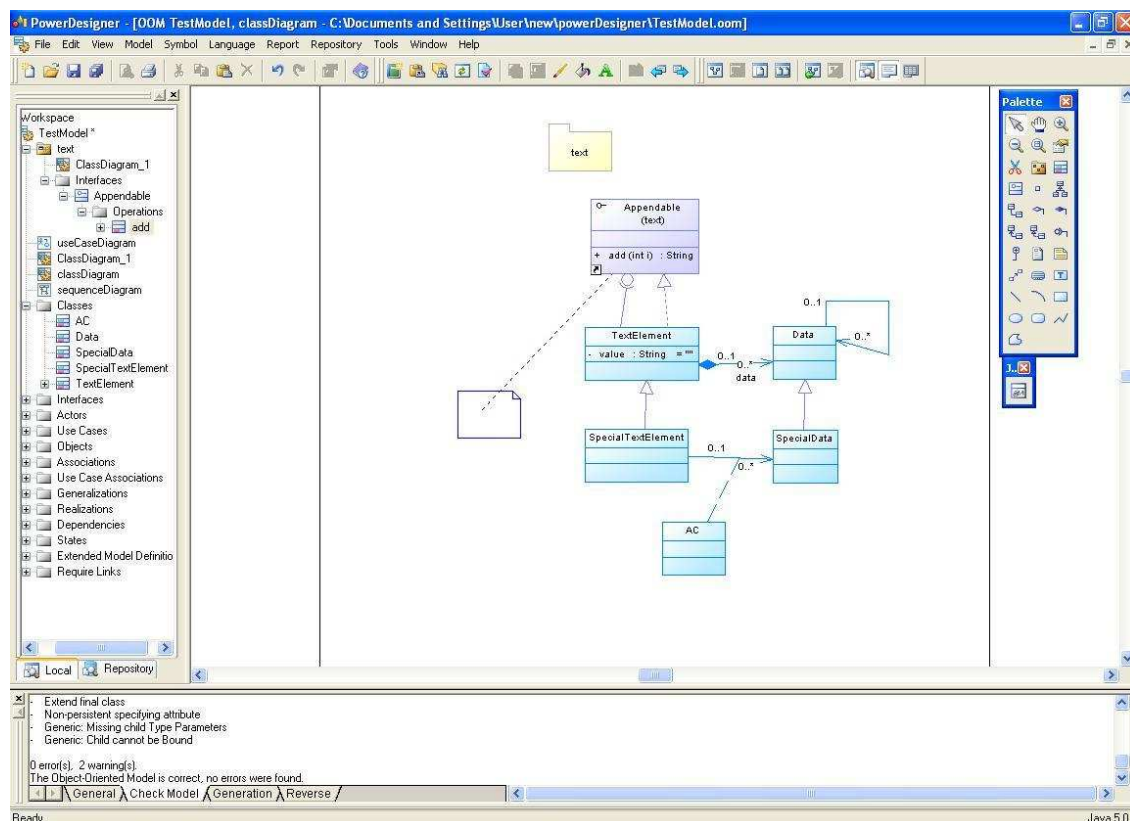


Figure 46: UML Class Diagram in "PowerDesigner"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	49.34%
Component diagram	52.94%
Composition diagram	40.00%
Deployment diagram	30.30%
Activity diagram	28.99%
Sequence diagram	73.33%
Communication diagram	61.54%
Interaction overview diagram	50.00%
Timing diagram	0.00%
State machine diagram	35.71%
Use case diagram	50.00%
Information flows	0.00%
Model management	0.00%
Templates	25.00%
Profiles	10.00%
UML summary	41.51%
Traceability	70.00%
Code generation	27.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	1.0, 1.1
XMI valid	Partial
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0C
L1.3	58.40%
L1.4	58.66%
L2-0	65.82%
L2-M	57.07%
L2-1	46.25%
L2-2	43.26%
L2-3	41.60%
Assigned level	partial L1.4, partial L2-0C

5.43 QuickUML by Excel Software

Name: QuickUML

Version: 3.0.4

Vendor: Excel Software

URL: <http://www.excelsoftware.com/quickumlwin.html>

Price (if available): 495\$

Vendor statement: *Design object-oriented software with a highly integrated, core set of UML models. Your entire project is presented through a multi-panel window showing use cases, class models, object models, dictionary and code. Save your project as a platform neutral XML formatted text file. Edit and share projects from any Windows or Mac OS X computer.*

User interface: dialogs, menu and toolbar

Comments: The tool supports at most one stereotype per element. The usability could be improved significantly. Partly, arbitrary edges, which do not realize any semantics of UML relations, can be inserted into diagrams. The tool requires the user to follow a prescribed engineering process from use cases to class diagrams. Use Case Diagrams are not supported, but use cases are available in an element list.

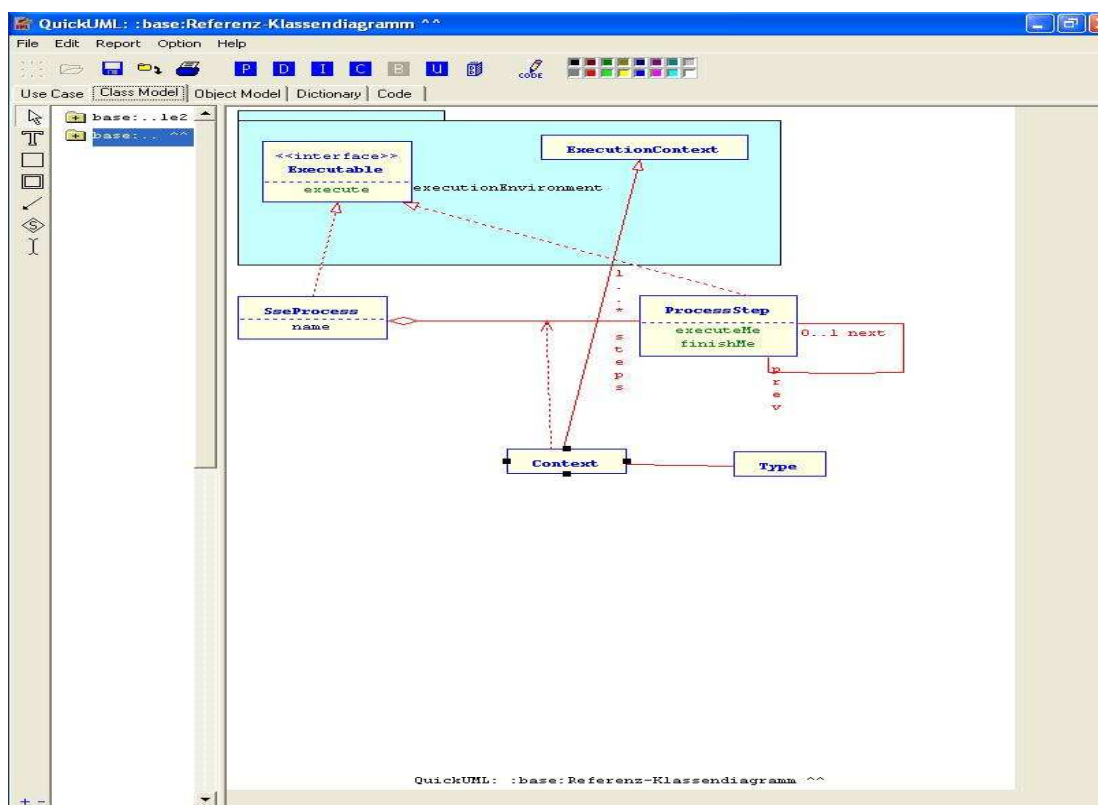


Figure 47: UML Class-Diagram in "QuickUML"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	34.11%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	11.11%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	2.63%
Information flows	0.00%
Model management	16.67%
Templates	0.00%
Profiles	0.00%
UML summary	12.31%
Traceability	10.00%
Code generation	14.00%

Tool characteristics	
Promoted UML version	core set of UML
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	24.37%
L1.4	24.22%
L2-0	35.44%
L2-M	36.65%
L2-1	15.83%
L2-2	12.81%
L2-3	12.34%
Assigned level	-

5.45 Rational Software Architect for WebSphere by IBM/Rational

Name: Rational Software Architect for WebSphere

Version: 7.5.4

Vendor: IBM/Rational

URL: <http://www.ibm.com/software/awdtools/swarchitect/websphere/>

Price (if available): 6310\$

Vendor statement:

- *Powerful modeling and graphical editing across a variety of domains (UML™ 2, BPMN 2, Java/JEE, WSDL, XSD, SCA, SoaML, DoDAF 2 and more). Includes the complete IBM® Rational® Application Developer product for an integrated design and development experience.*
- *Manage projects and risks more effectively: leverage the powers of abstraction, visualization, and traceability to analyze impacts of proposed changes*
- *Quickly create custom factory solutions based upon your own domain examples, using exemplar-driven authoring tools. Use domain specific languages to represent your unique problem and solution domains. Automate development with Patterns Based Engineering.*

The vendor of this tool prohibited the publication of findings on this tool.

This page is left intentionally blank.

5.46 Rational System Architect by IBM/Rational

Name: Rational System Architect

Version: 11.3.1

Vendor: IBM/Rational

URL: http://www.ibm.com/software/products/de/de/ratisystarch/111.ibm.com/ecatalog/Detail.wss?locale=de_DE&synkey=T102496E45339Y47

Price (if available): 3700\$

Vendor statement: *Telelogic System Architect® enables you to build a Business and Enterprise Architecture— a fully integrated collection of models and documents across five keys domains: Strategy, Business, Information, Systems and Technology. Telelogic System Architect's comprehensive solution provides a shared workspace for all team members to understand how to improve the company's architecture and overall business.*

The vendor of this tool prohibited the publication of findings on this tool.

This page is left intentionally blank.

5.47 Rational Tau by IBM/Rational

Name: Rational Tau

Version: 4.3.0.0.13660

Vendor: IBM/Rational

URL: <http://www.ibm.com/developerworks/rational/products/tau/>

Price (if available): 2310\$

Vendor statement: *IBM Rational Tau's iterative requirements-based approach, comprehensive error-checking and automated simulation increases developer productivity from initial requirements to final implementation. IBM Rational Tau supports the latest industry standards for visual systems and software development, including Unified Modeling Language (UML 2.1), Systems Modeling Language (SysML 1.0), Model Driven Architecture (MDA), Department of Defense Architecture Framework (DoDAF 1.5), and Service Oriented Architecture (SOA)*

The vendor of this tool prohibited the publication of findings on this tool.

This page is left intentionally blank.

5.48 Real Time Developer Studio by Pragmadev

Name: Real Time Developer Studio

Version: 4.0.3 2009-07-20

Vendor: Pragmadev

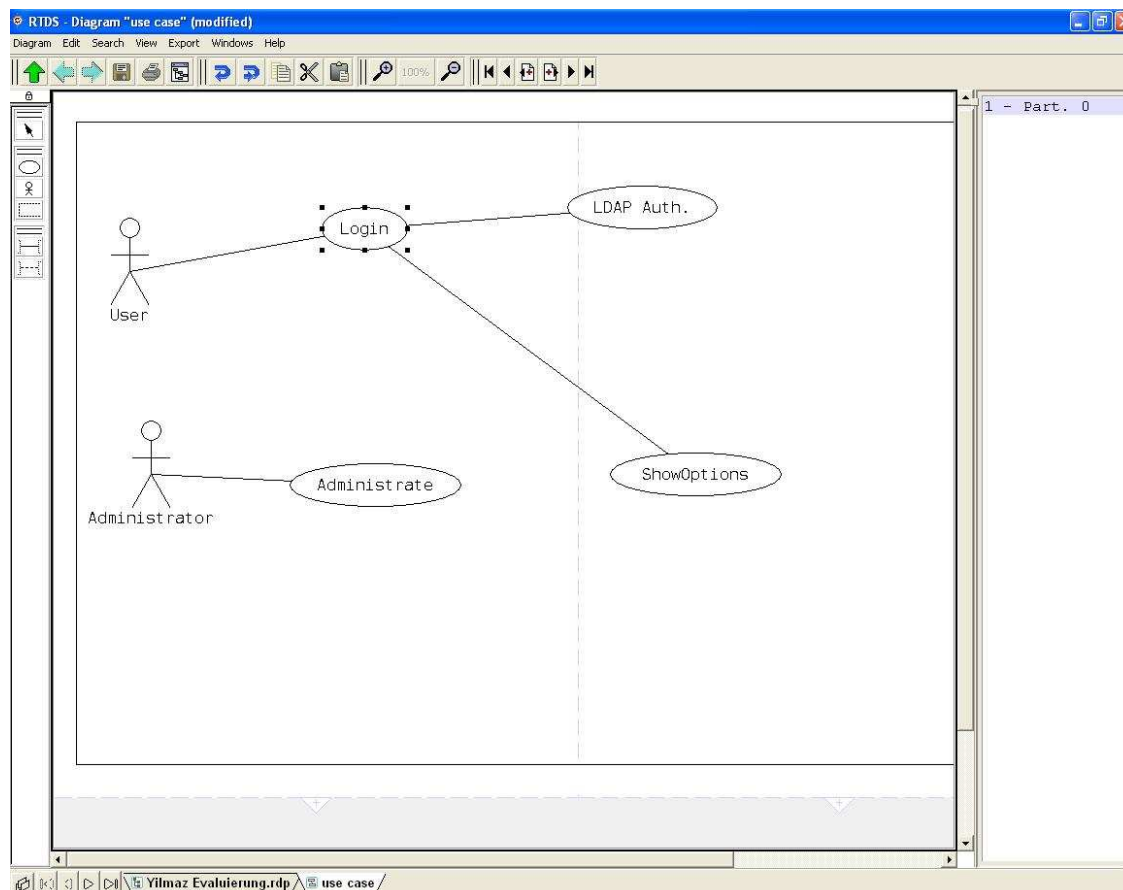
URL: <http://www.pragmadev.com/product/product.html>

Price (if available): unknown

Vendor statement: *Real Time Developer Studio is a modeling tool for real time and embedded software. It is suited for any communicating systems or any application running on top of a Real Time Operating System. It covers requirements, specification, prototyping, design, debug on target, and testing.*

User interface: dialogs, toolbar, toolbar and direct editing of diagram elements in the diagram

Comments: This tool is dedicated to real time development with SDL-RD and SDL Z.100. Only few UML modeling elements and diagrams are supported.



Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	28.81%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	21.21%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	23.68%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	11.68%
Traceability	0.00%
Code generation	11.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	22.90%
L1.4	22.76%
L2-0	45.57%
L2-M	35.08%
L2-1	13.33%
L2-2	12.36%
L2-3	11.70%
Assigned level	-

5.49 Rhapsody by Telelogic

Name: Rhapsody

Version: 7.1.1.0 Build 893629

Vendor: Telelogic

URL: <http://modeling.telelogic.com/products/rhapsody>

Price (if available): unknown

Vendor statement: *Rhapsody's core modeling languages are UML 2.1, latest version of SysML, and DoDAF. Going beyond these standard modeling languages, Rhapsody allows users to extend the modeling environment into Domain Specific Language (DSL) capabilities, a powerful technology differentiator for the Rhapsody family of modeling tools. This means systems engineers and software developers can create their own unique diagrams and diagram elements which may be relevant to the design domain, but outside the UML. Another differentiator from other tools on the market is that Rhapsody makes full use of Profiles, and Rhapsody DSL profiles truly extends the freedom to create a customizable environment, working within the current UML 2.1 /latest version of SysML/ and DoDAF framework to meet the exact design requirements for a project.*

The vendor of this tool prohibited the publication of findings on this tool.

This page is left intentionally blank.

5.50 Rose by IBM/Rational

Name: Rose

Version: 7.0

Vendor: IBM/Rational

URL: <http://www-01.ibm.com/software/awdtools/developer/rose/enterprise/index.html>

Price (if available): 4,640.00\$

Vendor statement: *The IBM Rational® Rose® product family lets you design software solutions using the Unified Modeling Language (UML). Architects, analysts, software and database designers, and real-time/embedded systems developers can all use the Rational Rose family of products to produce visual models of software architectures, databases, application requirements and reusable assets, as well as to formulate management-level communications.*

The vendor of this tool prohibited the publication of findings on this tool.

This page is left intentionally blank.

5.51 Smartdraw by Smartdraw

Name: Smartdraw

Version: 2010.07

Vendor: Smartdraw

URL: <http://www.smartdraw.com/>

Price (if available): 169,5\$-197\$

Vendor statement: *Just as the word processor makes it possible for anyone to create beautifully formatted written documentation, the visual processor makes it possible for anyone to create presentation-quality visuals just as easily.*

Before the visual processor, visuals had to be created manually with complex graphics software. Even for experts, producing a visual like a flowchart was time consuming and the results were often not presentation-quality. SmartDraw automates the creation of visuals to such a degree that anyone can do it, and get great results in minutes.

User interface: direct editing of diagram elements in the diagram, dialogs, menu and toolbars (palette)

Comments: Symbols related to one diagram are spread over multiple palettes. The basis installation is not ready for drawing UML diagrams. Thus, it needs to download individual parts from the vendor server. Nesting of model elements is only supported via the detour of grouping elements. Scaling of some elements is too large upon insertion into a diagram so that manual scaling is needed frequently.

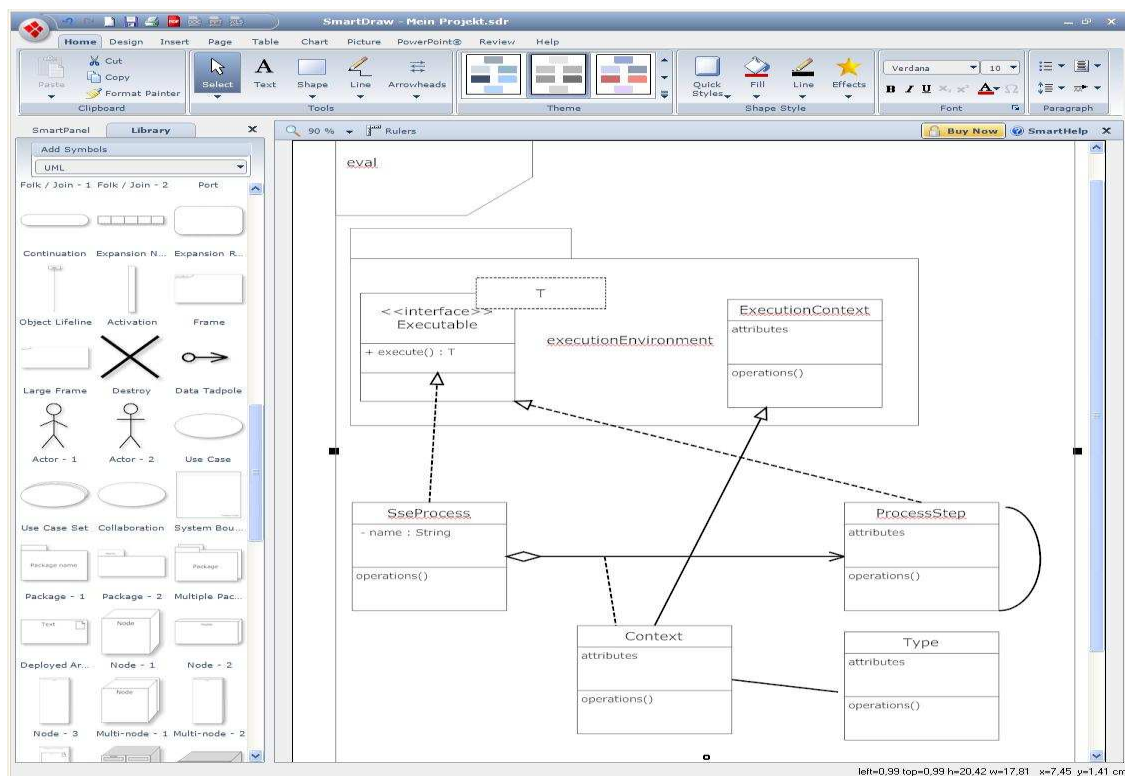


Figure 49: UML Class Diagram in "Smartdraw"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	31.79%
Component diagram	55.88%
Composition diagram	65.00%
Deployment diagram	46.97%
Activity diagram	38.41%
Sequence diagram	25.56%
Communication diagram	42.31%
Interaction overview diagram	59.38%
Timing diagram	0.00%
State machine diagram	72.62%
Use case diagram	68.42%
Information flows	7.14%
Model management	0.00%
Templates	8.33%
Profiles	0.00%
UML summary	37.69%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	50.42%
L1.4	50.52%
L2-0	37.97%
L2-M	35.08%
L2-1	36.11%
L2-2	39.55%
L2-3	37.77%
Assigned level	-

5.52 StarUML by Star UML Development Group

Name: StarUML

Version: 5.0.2.1570

Vendor: StarUML Development Group

URL: <http://staruml.sourceforge.net/en/index.php>

Price (if available): Open Source (GPL)

Vendor statement: *UML is continuously expanding standard managed by OMG (Object Management Group). Recently, UML 2.0 is released and StarUML support UML 2.0 and will support latest UML standard.*

User interface: dialogs, views, direct editing of diagram elements in the diagram, toolbar, drag & drop

Comments: -

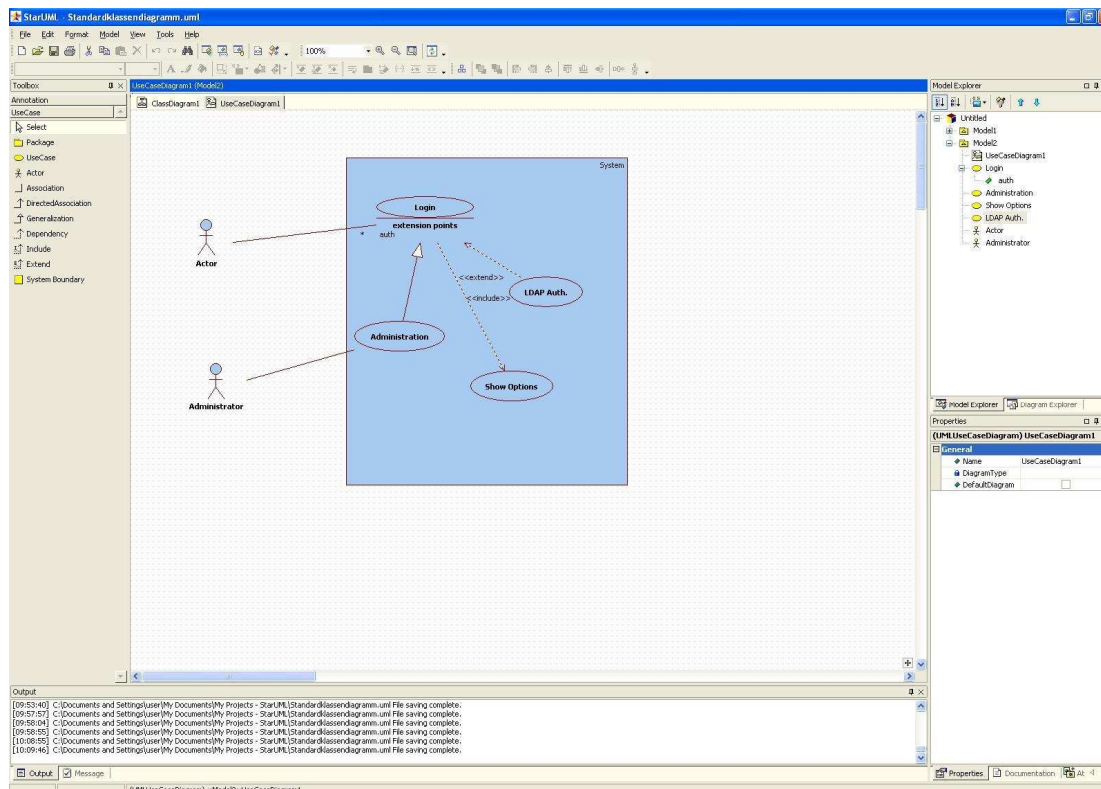


Figure 50: UML Use Case Diagram in “StarUML”

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	61.59%
Component diagram	41.18%
Composition diagram	85.00%
Deployment diagram	39.39%
Activity diagram	27.54%
Sequence diagram	73.33%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	50.00%
Use case diagram	78.95%
Information flows	0.00%
Model management	83.33%
Templates	41.67%
Profiles	0.00%
UML summary	46.71%
Traceability	0.00%
Code generation	18.00%

Tool characteristics	
Promoted UML version	1.3
XMI version	1.1
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	69.33%
L1.4	69.73%
L2-0	67.09%
L2-M	60.21%
L2-1	48.89%
L2-2	47.19%
L2-3	46.81%
Assigned level	partial L2-0C

5.53 System Architect by Telelogic

Name: System Architect

Version: 10.7.16 SP 1

Vendor: Telelogic

URL: <http://www.telelogic.com/products/systemarchitect/index.cfm>

Price (if available): unknown

Vendor statement: *Telelogic System Architect® enables you to build a Business and Enterprise Architecture— a fully integrated collection of models and documents across five keys domains: Strategy, Business, Information, Systems and Technology. Telelogic System Architect's comprehensive solution provides a shared workspace for all team members to understand how to improve the company's architecture and overall business.*

The vendor of this tool prohibited the publication of findings on this tool.

This page is left intentionally blank.

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	17.22%
Component diagram	29.41%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	20.29%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	28.57%
Use case diagram	63.16%
Information flows	0.00%
Model management	33.33%
Templates	0.00%
Profiles	0.00%
UML summary	15.07%
Traceability	0.00%
Code generation	8.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	26.47%
L1.4	26.30%
L2-0	25.32%
L2-M	17.80%
L2-1	16.39%
L2-2	15.96%
L2-3	15.53%
Assigned level	-

5.55 Telelogic Tau/Modeler Edition by Telelogic

Name: Telelogic Tau/Modeler Edition

Version: 3.1.1.0.0.3145

Vendor: Telelogic

URL: <http://modelingcommunity.telelogic.com/modeler-download.aspx?p=tau87>

Price (if available): unknown

Vendor statement: *Introducing Telelogic Modeler™, a free UML 2.1-based software design tool. Designed to remove acquisition cost barriers by a key author and supporter of UML, this free modeling product allows users to leverage the benefits of improved communication by using a standard graphical language to specify, visualize and document systems and software designs. Offered at no charge, Modeler allows users to increase their productivity and shorten design cycles.*

The vendor of this tool prohibited the publication of findings on this tool.

This page is left intentionally blank.

5.56 Teuta by University of Vienna

Name: Teuta

Version: -

Vendor: Institut für Softwarewissenschaft Universität Wien

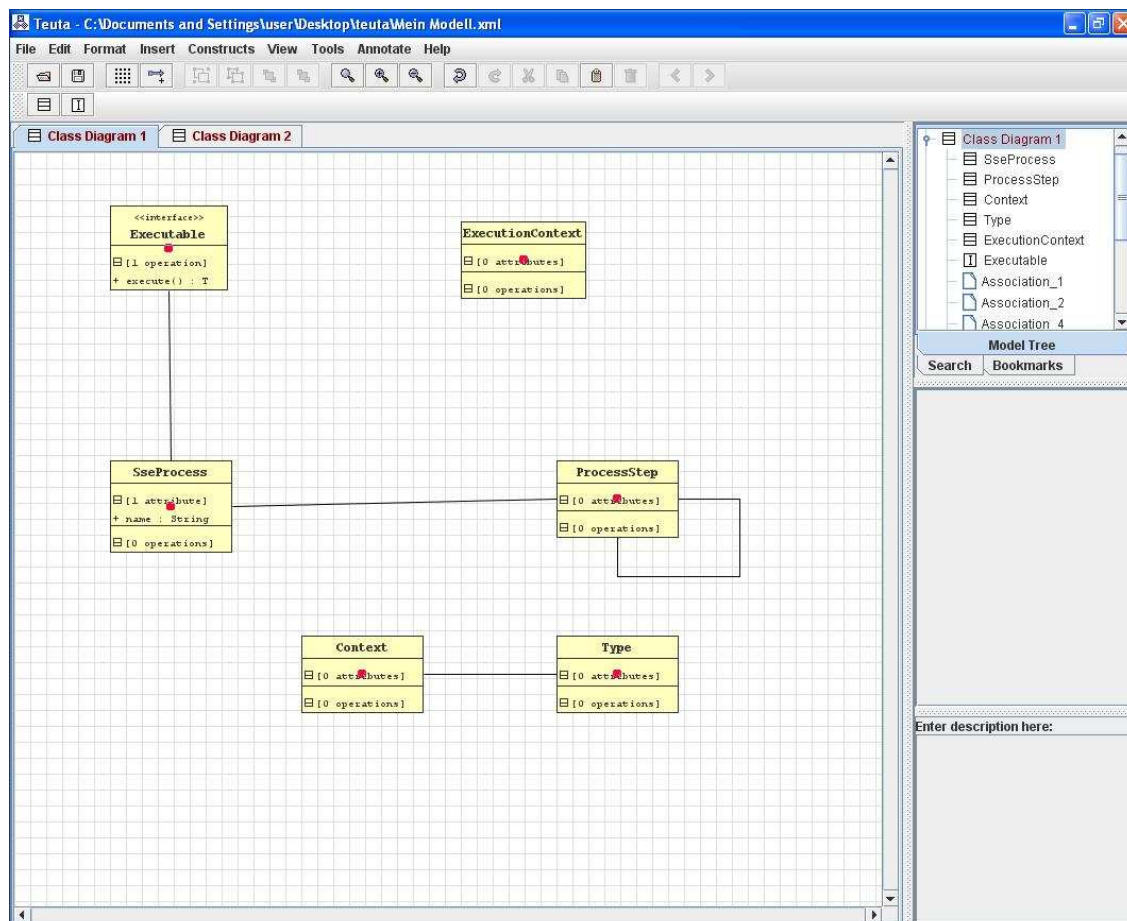
URL: <http://www.par.univie.ac.at/project/prophet/node4.html>

Price (if available): free, no license

Vendor statement: *Teuta is a graphical editor for the UML-based modeling of Distributed and parallel applications, and Grid workflow applications. At present Teuta supports following UML diagrams, Activity, Collaboration, Deployment, Class.*

User interface: dialogs, menu and toolbar

Comments: The tool appears to be an UML based editor supporting a subset of UML 1.x. The tool is unstable, because several exceptions raised while conducting the evaluation and, thus, the evaluator needed to restart the tool. There is not a specific component diagram, but component elements are available in the deployment diagram.



Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	9.60%
Component diagram	5.88%
Composition diagram	20.00%
Deployment diagram	10.61%
Activity diagram	18.84%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	33.33%
Templates	0.00%
Profiles	0.00%
UML summary	7.64%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	13.66%
L1.4	13.57%
L2-0	18.99%
L2-M	13.09%
L2-1	8.75%
L2-2	7.87%
L2-3	7.87%
Assigned level	-

5.57 Together by Borland

Name: Together

Version: 2006 R2 8.1.1Build-ID: 4359.1

Vendor: Borland

URL: <http://www.borland.com/together/index.html>

Price (if available): 3495\$

Vendor statement: *Create UML 2 and business process models (BPMN™) to generate and import business process execution languages with Web Services definitions (BPEL4WS). Boost productivity through Model Driven Architecture® (MDA®) features, including OMG's Query View Transformation (QVT) used in model-to-model transformations and support for OCL 2.0 with syntax highlighting, validation, code sense, debugging and expression evaluation*

User interface: views, direct editing of diagram elements in the diagram, menu and toolbar

Comments: The program realizes some functionality to support consistency among model elements.

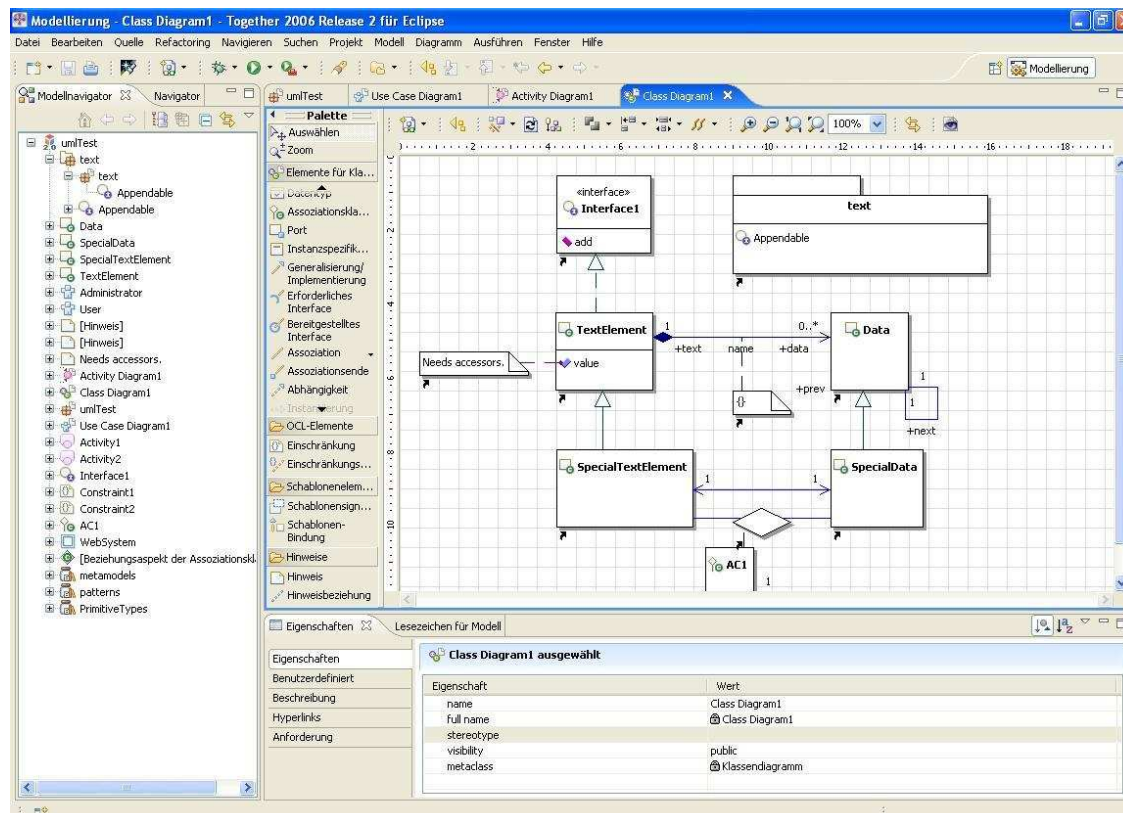


Figure 53: UML Class Diagram in "Together"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	77.48%
Component diagram	94.12%
Composition diagram	90.00%
Deployment diagram	74.24%
Activity diagram	42.03%
Sequence diagram	73.33%
Communication diagram	69.23%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	63.10%
Use case diagram	78.95%
Information flows	0.00%
Model management	0.00%
Templates	58.33%
Profiles	65.00%
UML summary	62.21%
Traceability	40.00%
Code generation	21.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	2.0
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0AC, L2-MAC
L1.3	81.09%
L1.4	81.42%
L2-0	83.54%
L2-M	84.29%
L2-1	63.33%
L2-2	64.16%
L2-3	62.23%
Assigned level	L1.4 / L2-MAC

5.58 Topcased by Topcased.org

Name: Topcased

Version: 3.2.0.v200911301720

Vendor: Topcased.org

URL: <http://www.topcased.org>

Price (if available): Open Source (EPL)

Vendor statement: *Topcased promotes model-driven engineering and formal methods as key technologies.*

User interface: dialogs, views, direct editing of diagram elements in the diagram, menu, toolbar and drag & drop

Comments: The usability of the program can be improved, e.g. often the mouse selection function had to be reactivated.

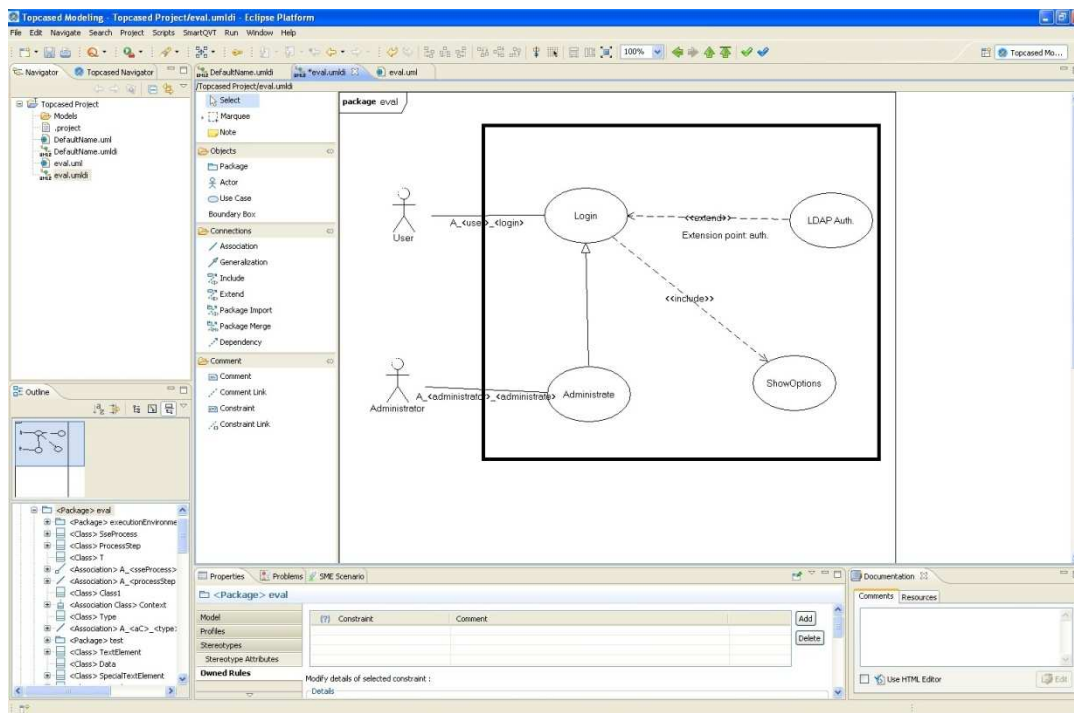


Figure 54: UML Use Case Diagram in "Topcased"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	67.22%
Component diagram	67.65%
Composition diagram	90.00%
Deployment diagram	54.55%
Activity diagram	68.84%
Sequence diagram	60.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	55.95%
Use case diagram	76.32%
Information flows	0.00%
Model management	50.00%
Templates	8.33%
Profiles	40.00%
UML summary	55.31%
Traceability	10.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	69.75%
L1.4	69.94%
L2-0	79.75%
L2-M	73.82%
L2-1	58.33%
L2-2	57.42%
L2-3	55.21%
Assigned level	L2-0C

5.59 Umbrello UML Modeler by Umbrello Project Team at Sourceforge

Name: Umbrello UML Modeler

Version: 2.3.2

Vendor: Umbrello Project Team

URL: <http://uml.sourceforge.net/>

Price (if available): Open Source (GPL)

Vendor statement: *Umbrello UML Modeller is a Unified Modelling Language diagram programme for KDE. UML allows you to create diagrams of software and other systems in a standard format.*

User interface: dialogs, views, menu, toolbar and drag & drop

Comments: The program is unstable (hang-up, crash). The evaluated version crashed several times. The usability could be improved, e.g. the drag & drop menu does not keep the selected element so for each new model element the user must reselect the type of model element to be inserted.

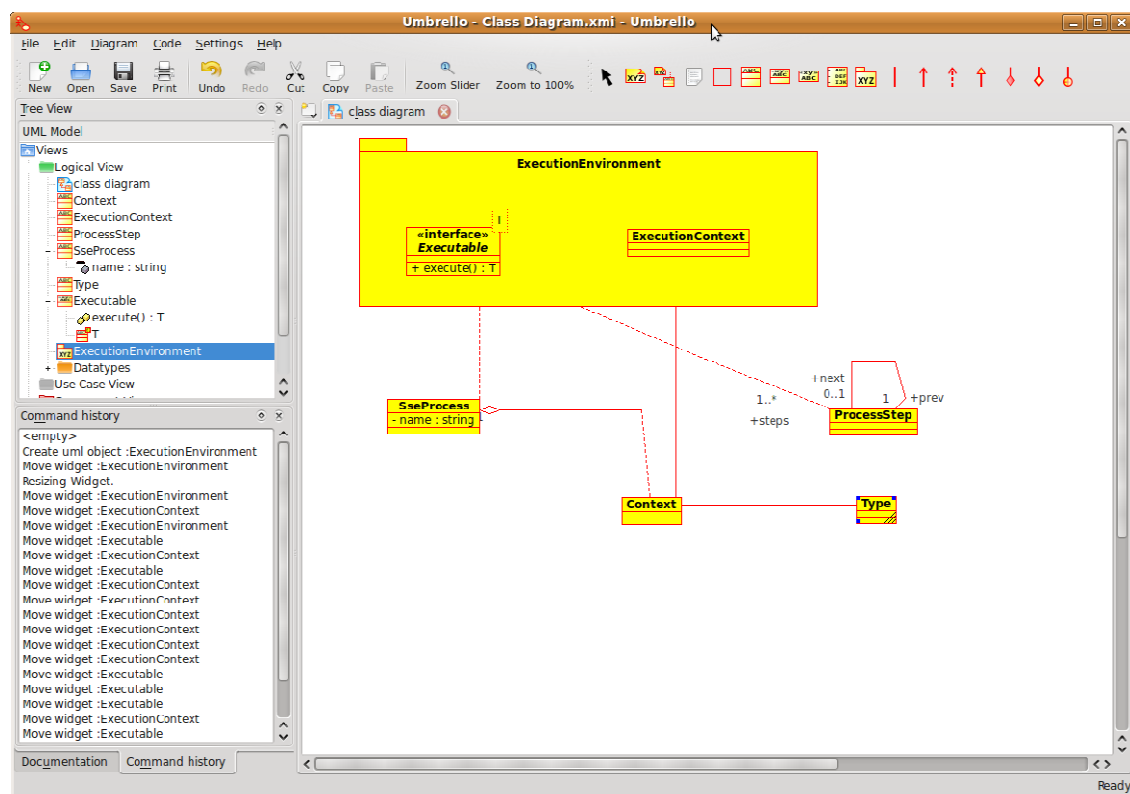


Figure 55: UML Class Diagram in "Umbrello UML Modeler"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	48.34%
Component diagram	23.53%
Composition diagram	0.00%
Deployment diagram	30.30%
Activity diagram	30.43%
Sequence diagram	40.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	40.48%
Use case diagram	50.00%
Information flows	0.00%
Model management	8.33%
Templates	41.67%
Profiles	0.00%
UML summary	33.65%
Traceability	0.00%
Code generation	20.00%

Tool characteristics	
Promoted UML version	-
XMI version	1.2
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	53.15%
L1.4	52.82%
L2-0	67.09%
L2-M	52.88%
L2-1	35.00%
L2-2	34.38%
L2-3	33.72%
Assigned level	partial L2-0C

5.60 UMLAUT by IRISA/CNRS

Name: UMLAUT

Version: beta 1-8

Vendor: IRISA/CNRS

URL: <http://www.irisa.fr/pampa/UMLAUT>

Price (if available): free, no license

Vendor statement: *The central component of UMLAUT is the implementation in Eiffel of the UML meta-model. It allows UML models to be represented in an AST style object structure. A model can be built either directly through a Graphical User Interface, or imported from other tools, e.g. Rational Rose or Objectteering, or even reverse-engineered from Eiffel or Java source code.*

User interface: dialogs, menu, toolbar and drag & drop

Comments: The program is unstable (hang-up, crash). The usability can significantly be improved, e.g. the mechanism to create relationships needs a specific sequence of selecting the elements to be connected.

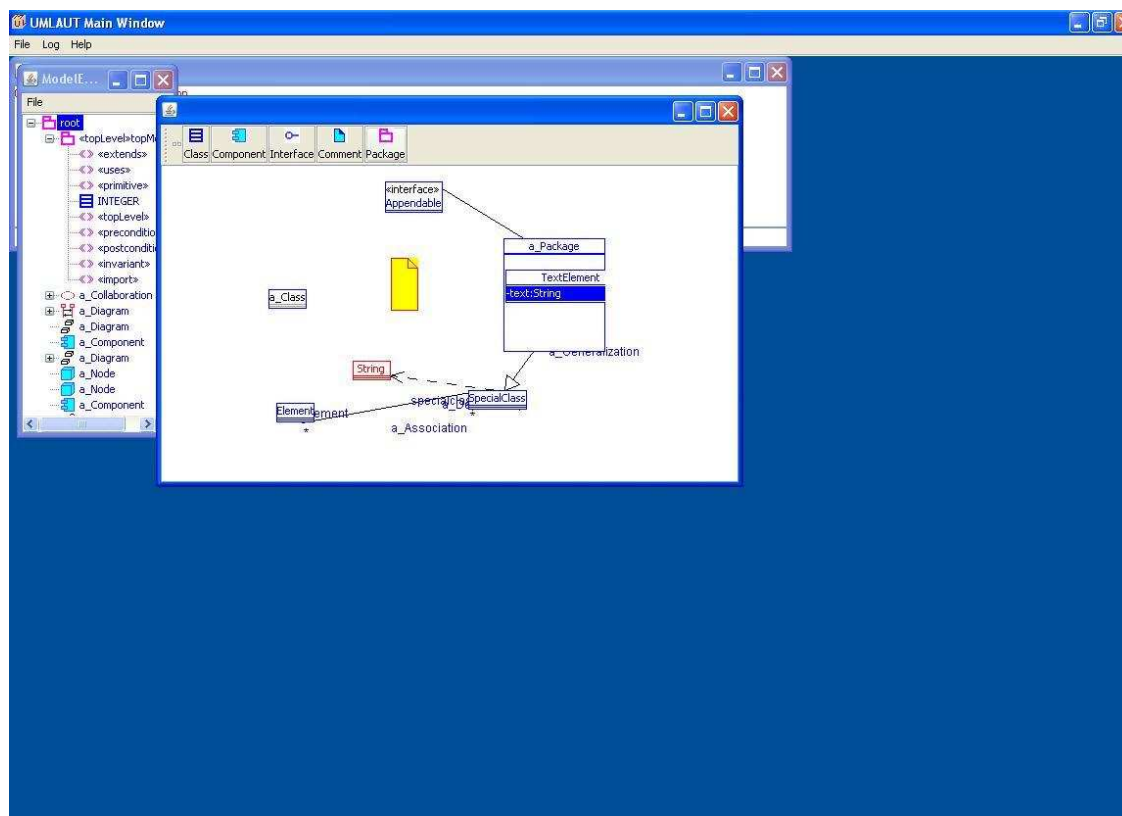


Figure 56: UML Class Diagram in "UMLAUT"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	37.42%
Component diagram	5.88%
Composition diagram	0.00%
Deployment diagram	6.06%
Activity diagram	0.00%
Sequence diagram	4.44%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	42.11%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	14.76%
Traceability	0.00%
Code generation	5.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	26.68%
L1.4	26.51%
L2-0	51.90%
L2-M	46.60%
L2-1	18.75%
L2-2	15.62%
L2-3	14.79%
Assigned level	partial L2-0C

5.61 UMLDiagrammer by Pacestar

Name: UMLDiagrammer

Version: 6.20.2040

Vendor: Pacestar

URL: <http://www.pacestar.com/uml/index.html>

Price (if available): 69\$-239\$

Vendor statement: *Pacestar UML Diagrammer helps you generate UML 2.0 diagrams quickly and easily. Create activity diagrams, class and object diagrams, communication diagrams, use case diagrams, sequence diagrams, state charts, package diagrams, and component diagrams, deployment diagrams, composite structure diagrams, interaction overview diagrams, and even traditional flowcharts.*

User interface: dialogs, toolbar, drag & drop, direct editing of diagram elements in the diagram and menu

Comments: The tool implements an intelligent scaling mechanism for the diagrams. Nesting of elements is only supported via grouping and ungrouping.

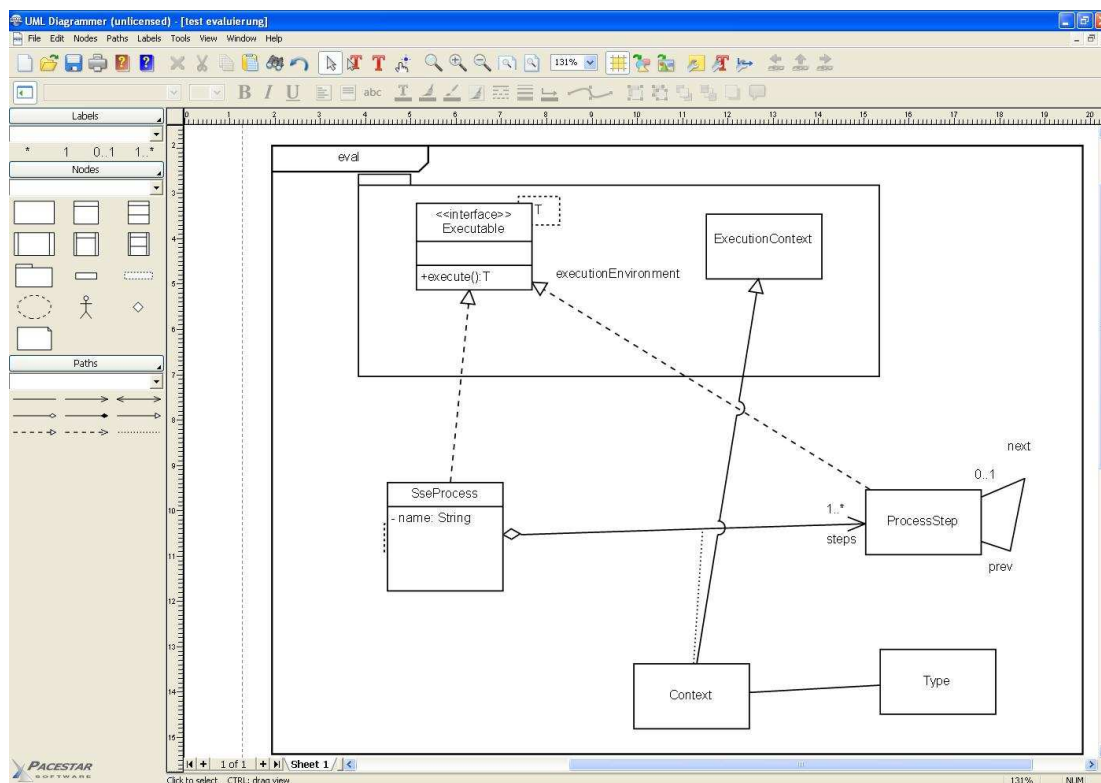


Figure 57: UML Class Diagram in "UMLDiagrammer"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	23.84%
Component diagram	55.88%
Composition diagram	75.00%
Deployment diagram	39.39%
Activity diagram	36.96%
Sequence diagram	34.44%
Communication diagram	34.62%
Interaction overview diagram	62.50%
Timing diagram	0.00%
State machine diagram	54.76%
Use case diagram	71.05%
Information flows	0.00%
Model management	0.00%
Templates	29.17%
Profiles	0.00%
UML summary	34.29%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	2.0
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	45.59%
L1.4	45.51%
L2-0	25.32%
L2-M	25.13%
L2-1	34.17%
L2-2	35.73%
L2-3	34.57%
Assigned level	-

5.63 UMLed by Georg Kubitz

Name: UMLed

Version: 1.8.4 b1

Vendor: Georg Kubitz

URL: <http://www.kubitz-online.de/UMLed/index.html>

Price (if available): unknown

Vendor statement (available in German only):

- *UMLed unterstützt die schulische Arbeit wesentlich durch eine gut gelungene Verbindung zwischen UML und Delphi- bzw. Java-Quelldateien. (Import, Bearbeitung und Export). Es befreit damit Schüler und Lehrer von vielen unnötigen Routinearbeiten, vorausgesetzt, sie arbeiten objektorientiert. Damit macht UMLed objektorientiertes Arbeiten in der Schule noch einfacher!*
- *Klassendiagramme, Beziehungsdiagramme und Botschaftsdiagramme können einfach interaktiv am Bildschirm entworfen werden.*
- *UMLed unterstützt Reverse Engineering: Klassendiagramme können aus fertigen Delphi-Units bzw. Java-Quelltexten importiert werden.*

User interface: dialog, view and menu

Comments: Supports a subset of UML 1 model elements.

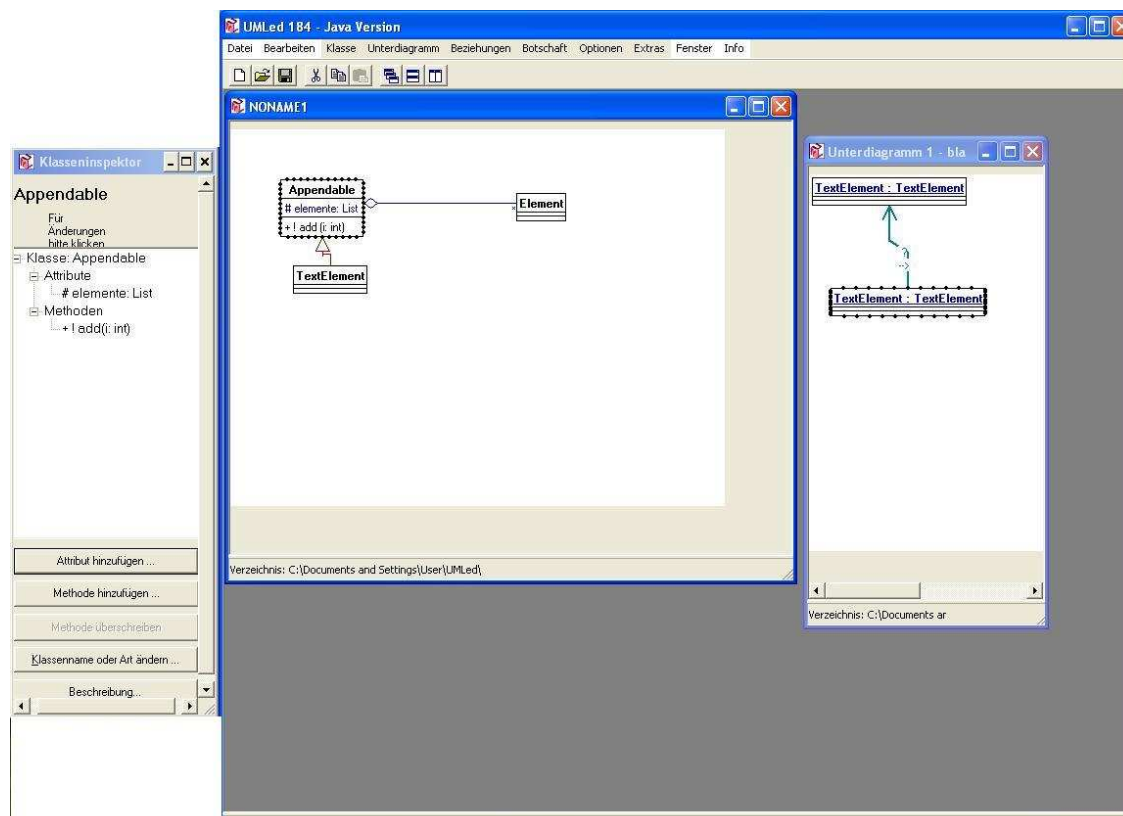


Figure 58: UML Class Diagram in "UMLed"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	18.87%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	0.00%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	6.05%
Traceability	0.00%
Code generation	10.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	12.39%
L1.4	12.32%
L2-0	34.18%
L2-M	25.65%
L2-1	8.19%
L2-2	6.63%
L2-3	6.28%
Assigned level	-

5.64 UMLet by UMLet group (University of Vienna)

Name: UMLet

Version: 10.3

Vendor: UMLet group (University of Vienna)

URL: <http://www.umlet.com>

Price (if available): Open Source (GPL)

Vendor statement: *UMLet is an open-source UML tool with a simple user interface: **draw** UML diagrams fast, **export** diagrams to eps, pdf, jpg, svg, and clipboard, **share** diagrams using Eclipse, and **create** new, custom UML elements.*

UMLet runs stand-alone or as Eclipse plug-in on Windows, OS X and Linux.

User interface: toolbar, drag & drop and “literal” programming

Comments: The program is difficult to use (for novices).

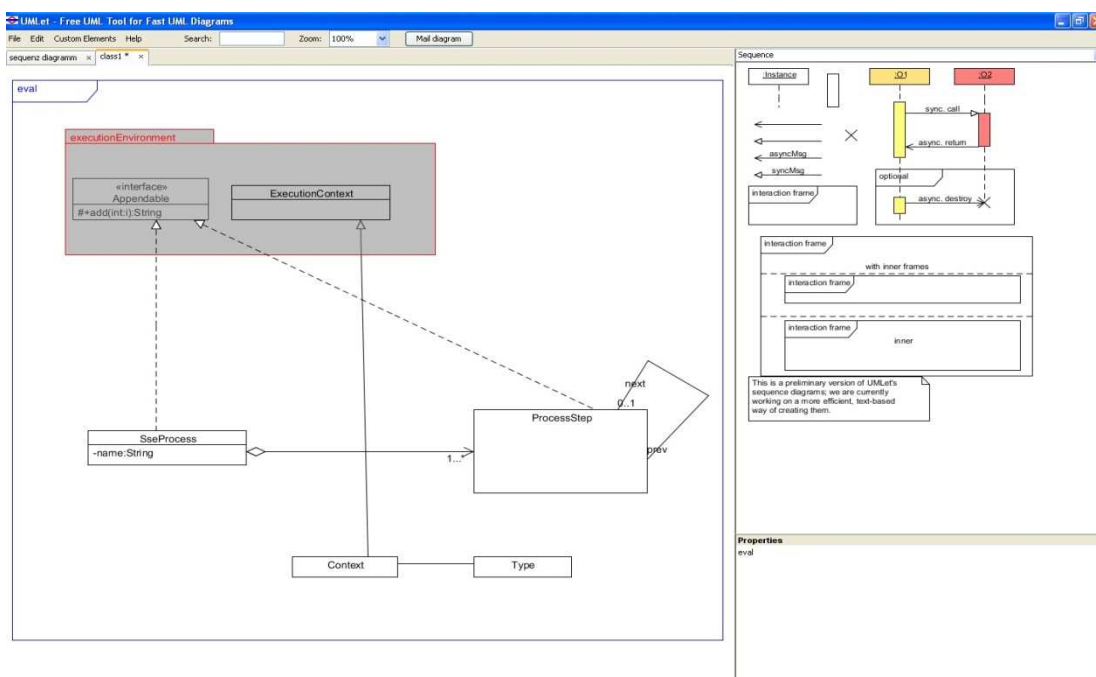


Figure 59: UML Class Diagram in "UMLet"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	38.41%
Component diagram	58.82%
Composition diagram	95.00%
Deployment diagram	37.88%
Activity diagram	34.78%
Sequence diagram	40.00%
Communication diagram	46.15%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	51.19%
Use case diagram	76.32%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	36.94%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	53.15%
L1.4	53.03%
L2-0	43.04%
L2-M	37.70%
L2-1	38.89%
L2-2	39.10%
L2-3	37.02%
Assigned level	-

5.65 UMLPad by Luigi Bignami

Name: UMLPad

Version: 3.2

Vendor: Luigi Bignami

URL: <http://web.tiscali.it/ggbhome/umlpad/umlpad.htm>

Price (if available): Open Source (GPL)

Vendor statement: *UML Pad is a CASE tool for UML diagrams design. It supports Use Case, Class, Sequence, State and Activity diagrams. It allows printing and exporting of the diagram image. For Class diagrams it's also possible to export the documentation in html format. The project has been realized with support of the wxWidgets class library. Current version is 3.2. Now with Use Case diagrams.*

User interface: dialogs, toolbar and menu

Comments: The usability can be improved, in particular the number of (sub-) dialogs to add an operation and its parameters can be reduced.

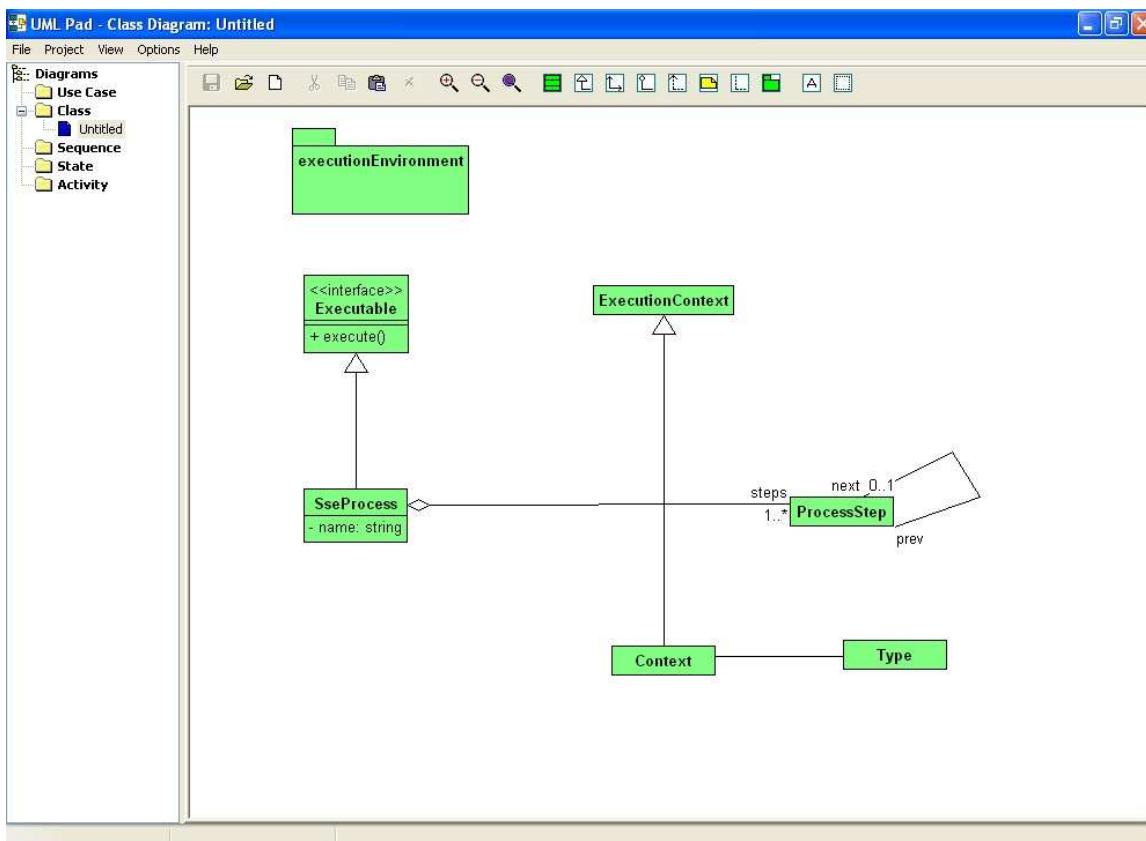


Figure 60: UML Class Diagram in "UMLPad"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	39.07%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	16.67%
Sequence diagram	17.78%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	26.19%
Use case diagram	55.26%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	21.23%
Traceability	0.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	40.97%
L1.4	40.71%
L2-0	56.96%
L2-M	47.64%
L2-1	25.00%
L2-2	22.70%
L2-3	21.49%
Assigned level	partial L2-0C

5.66 UMLStudio by Pragsoft Cooperation

Name: UMLStudio

Version: 8.2.1

Vendor: Pragsoft Cooperation

URL: <http://www.pragsoft.com/>

Price (if available): unknown

Vendor statement: *UMLStudio provides you with pre-packaged, industry standard notations such as UML, Booch, and Data Flow. Use these as provided or customize them to suit your own needs. UMLStudio 8.0 provides extensive support for the UML 2.1 standard. UMLStudio does not force you into using predetermined notations. As a software professional you know only too well that the needs of every organization/project are different. Good communications often requires designing new notations. Create your own in a matter of minutes, or customize the ones that come with UMLStudio.*

User interface: dialogs and toolbar

Comments: The freeware version used for evaluation is restricted to 5 “master elements”, where all stereotypes are automatically set to “restricted”. The toolbar contains modeling elements mixed across all supported diagrams. All relations seem to be implemented by associations. The program is unstable (hang-up, crash). The model elements are named in an usual way and not according to the UML terminology.

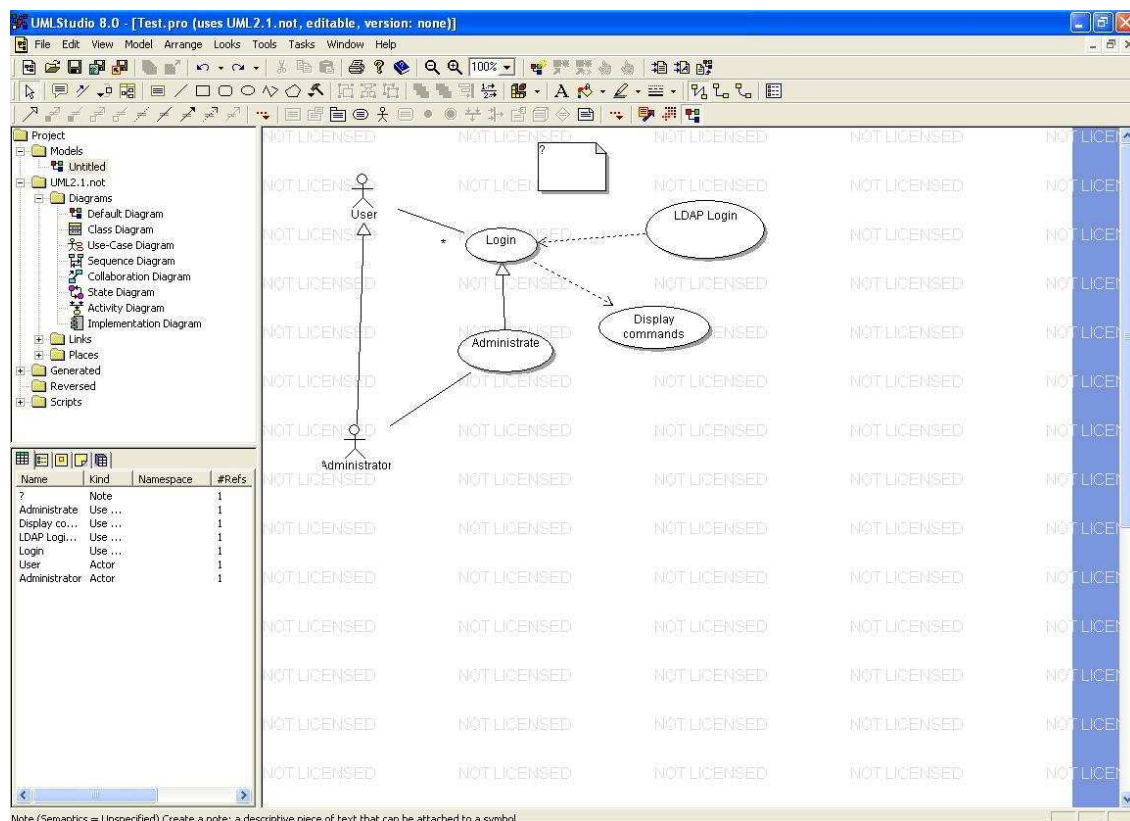


Figure 61: UML Use Case Diagram in "UMLStudio"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	39.40%
Component diagram	47.06%
Composition diagram	0.00%
Deployment diagram	30.30%
Activity diagram	23.91%
Sequence diagram	12.22%
Communication diagram	34.62%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	28.57%
Use case diagram	47.37%
Information flows	0.00%
Model management	0.00%
Templates	4.17%
Profiles	0.00%
UML summary	26.65%
Traceability	70.00%
Code generation	23.00%

Tool characteristics	
Promoted UML version	2.1
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	43.70%
L1.4	43.63%
L2-0	65.82%
L2-M	51.31%
L2-1	28.61%
L2-2	28.09%
L2-3	26.70%
Assigned level	partial L2-0C

5.67 Umodel by Altova

Name: Umodel

Version: Professional 2009

Vendor: Altova

URL: <http://www.altova.com/>

Price (if available): 123,75 €

Vendor statement: *Visually design application models in UML and generate Java, C#, or Visual Basic .NET code and project documentation. Or, reverse engineer existing programs into UML 2 diagrams, then fine tune your designs and complete the round trip by regenerating code. UModel is the UML tool that makes visual software design practical for any project. It is the simple, cost-effective way to draw on UML.*

User interface: dialogs, views, direct editing of diagram elements in the diagram and toolbar

Comments: Stereotypes must first be applied to the root or default package. The consistency check mechanism does not find inconsistencies in activity diagrams. The tool needs about 10 minutes to generate the documentation for the simple evaluation model on an Intel Dual Core 1.86 GHz with 2GB main memory. Classes must be assigned to components for source code generation, because components store the target location in the file system. The evaluator was not able to attach stereotypes properly to the diagram elements.

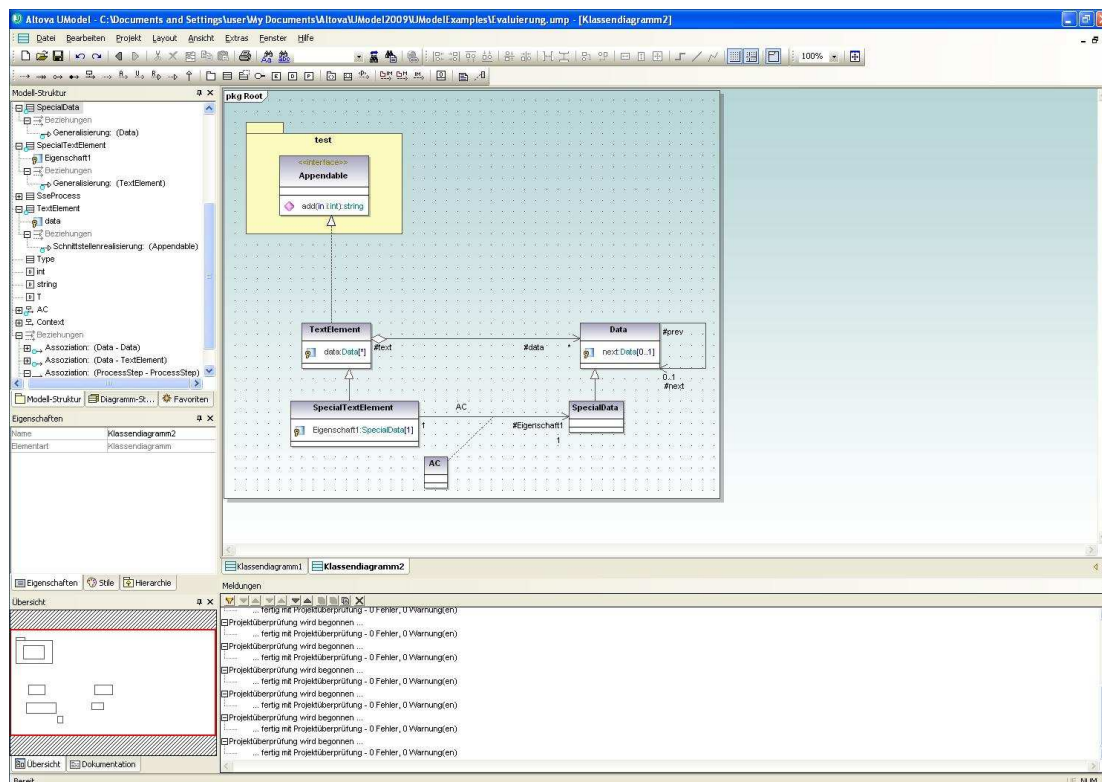


Figure 62: UML Class Diagram in "UModel"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	63.91%
Component diagram	44.12%
Composition diagram	95.00%
Deployment diagram	54.55%
Activity diagram	62.32%
Sequence diagram	71.11%
Communication diagram	69.23%
Interaction overview diagram	75.00%
Timing diagram	84.21%
State machine diagram	69.05%
Use case diagram	71.05%
Information flows	0.00%
Model management	50.00%
Templates	54.17%
Profiles	25.00%
UML summary	63.38%
Traceability	95.00%
Code generation	20.00%

Tool characteristics	
Promoted UML version	2.2
XMI version	2.1
XMI valid	valid
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0AC, L2-MAC, L2-1AC, L2-2AC
L1.3	75.21%
L1.4	74.53%
L2-0	78.48%
L2-M	65.45%
L2-1	66.67%
L2-2	65.06%
L2-3	63.62%
Assigned level	partial L2-2AC

5.68 violet by C. S. Horstmann and A. de Pellegrin

Name: violet

Version: 0.21.1(2007)

Vendor: C. S. Horstmann and A. de Pellegrin

URL: <http://horstmann.com/violet/>

Price (if available): Open Source (GPL)

Vendor statement: *Violet is a UML editor with these benefits:*

- *It is very easy to learn and use*
- *It draws nice-looking class, sequence, state, object and use-case diagrams*
- *It is completely free (includes source, distributed under the GNU General Public License)*
- *It is cross-platform*

Violet is intended for students, teachers, and authors who need to produce simple UML diagrams quickly. It is not intended as an industrial strength tool.

User interface: dialogs and toolbar

Comments: Class members are only available as text, not as individual operations or properties. The program is unstable (hang-up, crash). A connector to a note can only be used as a not connected element, i.e. it cannot be connected to relationships (as shown in Figure 63), model elements such as a class or contained model elements such as a class in a package.

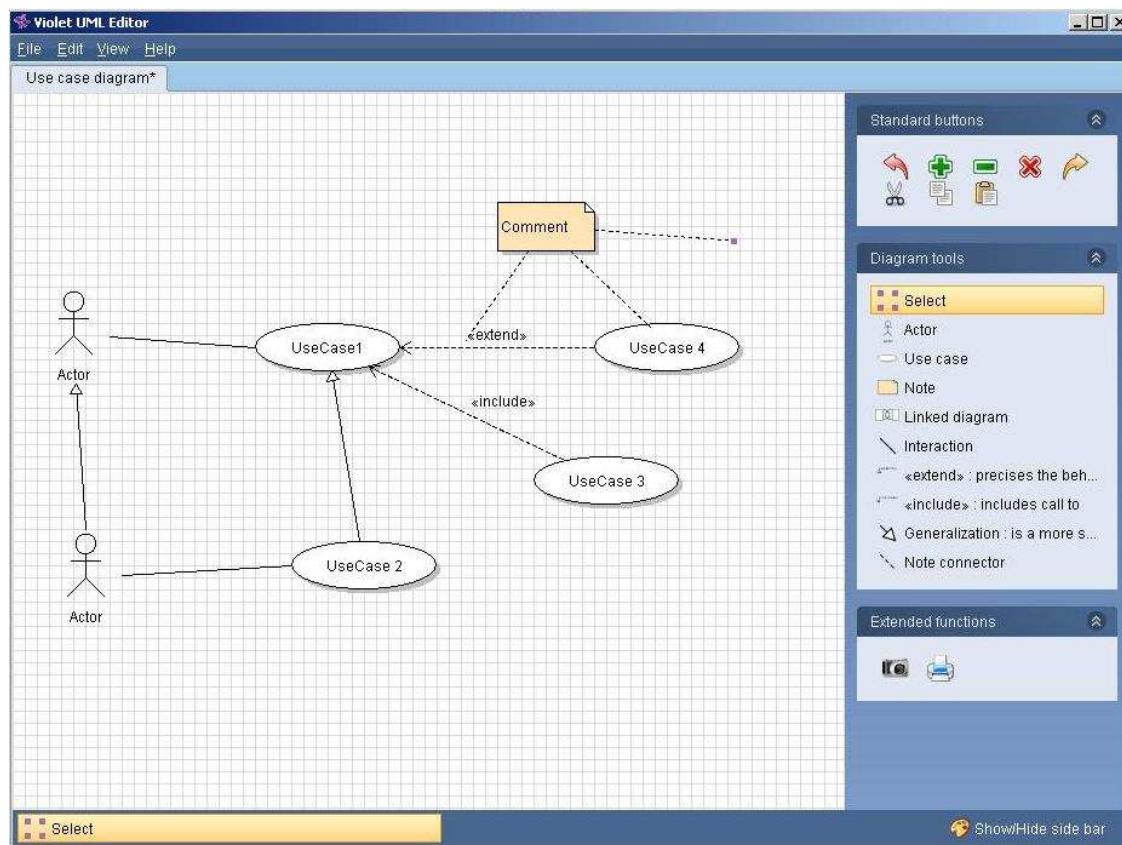


Figure 63: UML Use Case Diagram in "violet"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	13.58%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	15.22%
Sequence diagram	18.89%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	13.10%
Use case diagram	47.37%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	11.46%
Traceability	10.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	22.06%
L1.4	21.92%
L2-0	26.58%
L2-M	18.32%
L2-1	13.75%
L2-2	12.36%
L2-3	11.70%
Assigned level	-

5.69 Visible Analyst by Visible Systems

Name: Visible Analyst

Version: 7.6.5

Vendor: Visible Systems

URL: <http://www.visible.com>

Price (if available): unknown

Vendor statement: *Visible Analyst is the only integrated application development tool that supports all of the most widely used analysis and design techniques:*

- *UML: class, use case, sequence, collaboration, and action diagrams*
- *Data Models: supports conceptual, logical and physical database design; use any of the popular diagramming notations*
- *Process Models: decomposition charts, structure charts, and data flow diagrams*

The vendor of this tool did not respond to a publication request in terms of an explicit opt-in.

This page is left intentionally blank.

5.70 Visio by Microsoft

Name: Visio

Version: Professional 2007 12.0.4518.1014

Vendor: Microsoft

URL: <http://www.microsoft.com/office/visio>

Price (if available): 149.95\$ - 679.95\$

Vendor statement: *Microsoft Visio 2010 advanced diagramming tools help you simplify complexity with dynamic, data-driven visuals and new ways to share on the Web in real time. Whether you're creating an organizational chart, a network diagram, or a business process, the new tools and more intuitive interface in Visio 2010 make it easier to bring your diagrams to life.*

User interface: dialogs, direct editing of diagram elements in the diagram, drag & drop, menu and toolbar

Comments: While being a drawing tool, it supports many semantic elements of the UML. Unfortunately, the semantics of relations are not properly supported.

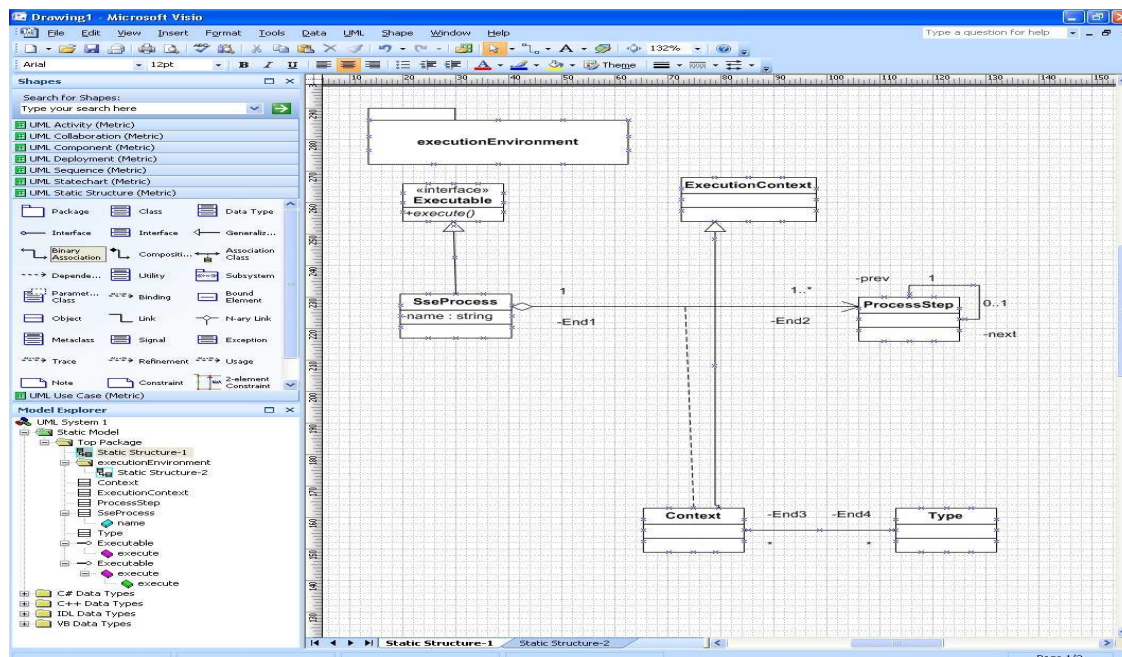


Figure 64: UML Class Diagram in "Visio"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	62.25%
Component diagram	23.53%
Composition diagram	40.00%
Deployment diagram	50.00%
Activity diagram	23.91%
Sequence diagram	33.33%
Communication diagram	53.85%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	44.05%
Use case diagram	73.68%
Information flows	0.00%
Model management	66.67%
Templates	50.00%
Profiles	0.00%
UML summary	42.36%
Traceability	40.00%
Code generation	8.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	71.64%
L1.4	70.77%
L2-0	73.42%
L2-M	60.73%
L2-1	42.92%
L2-2	42.58%
L2-3	42.45%
Assigned level	partial L2-0C

5.71 Visio with UML2 stencils by Pavel Hruby

Name: Visio Stencil and Template for UML 2.0

Version: for Visio 2007

Vendor: Pavel Hruby

URL: <http://www.softwarestencils.com/uml/index.html>

Price (if available): - (needs Visio, see Section 5.71)

Vendor statement: *The UML stencil for Microsoft Visio supports complete UML 2.2, i.e. UML use case diagram, class diagram, package diagram, object diagram, composite structure diagram, interaction diagram, sequence diagram, communication diagram, interaction overview diagram, activity diagram, state machine diagram, component diagram, deployment diagram, profile diagram, timing diagram, and all symbols of the UML 2.2, specified in OMG UML Superstructure Specification, formal/2009-02-02, as well as all previous UML versions, UML 2.1, UML 2.0, UML 1.5, UML 1.4, UML 1.3 and UML 1.1.*

User interface: dialogs, direct editing of diagram elements in the diagram, drag & drop, menu and toolbar

Comments: Does not add additional semantics or consider the semantics provided by Visio. The elements introduced by the UML2 stencils combined with the default UML symbols of Visio. The extension provides only two palettes for (all) new symbols and does not properly group the new elements.

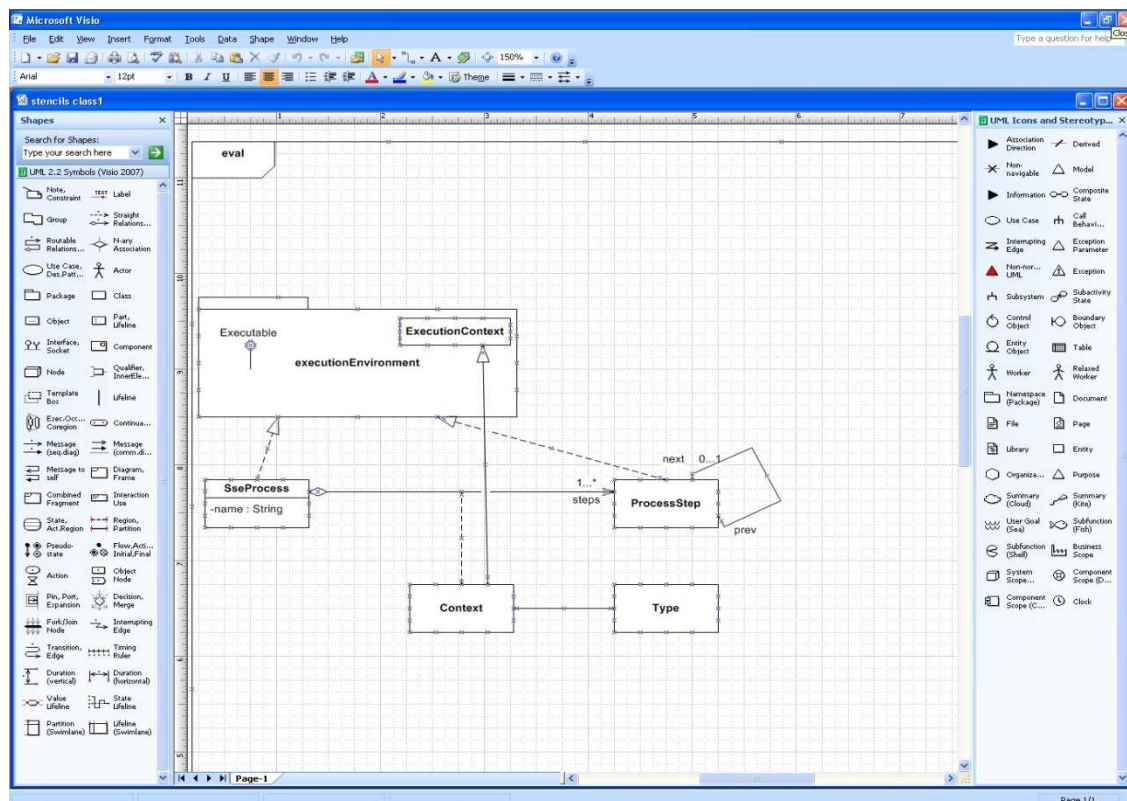


Figure 65: UML Class Diagram in “Visio with UML2 stencils”

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	52.32%
Component diagram	58.82%
Composition diagram	50.00%
Deployment diagram	40.91%
Activity diagram	45.65%
Sequence diagram	33.33%
Communication diagram	69.23%
Interaction overview diagram	68.75%
Timing diagram	71.05%
State machine diagram	66.67%
Use case diagram	76.32%
Information flows	14.29%
Model management	8.33%
Templates	50.00%
Profiles	0.00%
UML summary	50.42%
Traceability	40.00%
Code generation	2.00%

Tool characteristics	
Promoted UML version	1.1,1.3,1.4,1.5,2.0,2.1,2.2
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	64.92%
L1.4	64.93%
L2-0	64.56%
L2-M	57.07%
L2-1	52.36%
L2-2	51.69%
L2-3	50.53%
Assigned level	partial L2-0C

5.72 Visual Paradigm for UML by Visual Paradigm

Name: Visual Paradigm for UML

Version: 7.1 (Build 20091009)

Vendor: Visual Paradigm

URL: <http://www.visual-paradigm.com/>, <http://www.visual-paradigm.eu>

Price (if available): 70,5\$ - 1,678.5\$

Vendor statement: Visual Paradigm for UML is a UML modeler supports all UML 2.x diagrams, SysML requirement diagram and Entity Relationship Diagram (ERD) for software development team to perform system analysis and design.

User interface: dialogs, views, direct editing of diagram elements in the diagram, toolbar and gesture detection

Comments: The program supports creating diagrams by implementing a good positioning help. Sometimes, relationships in class diagrams appear as being connected but are not properly connected among the related model elements. Some dialogs are not scaled properly in size so not all options are visible, in particular when editing associations. The dialogs do not allow the user to edit association ends. The tool offers a good implementation of sequence diagrams (e.g. combined fragments) and provides many refinement options for state machine elements. The realization of timing diagrams appears to be work in progress.

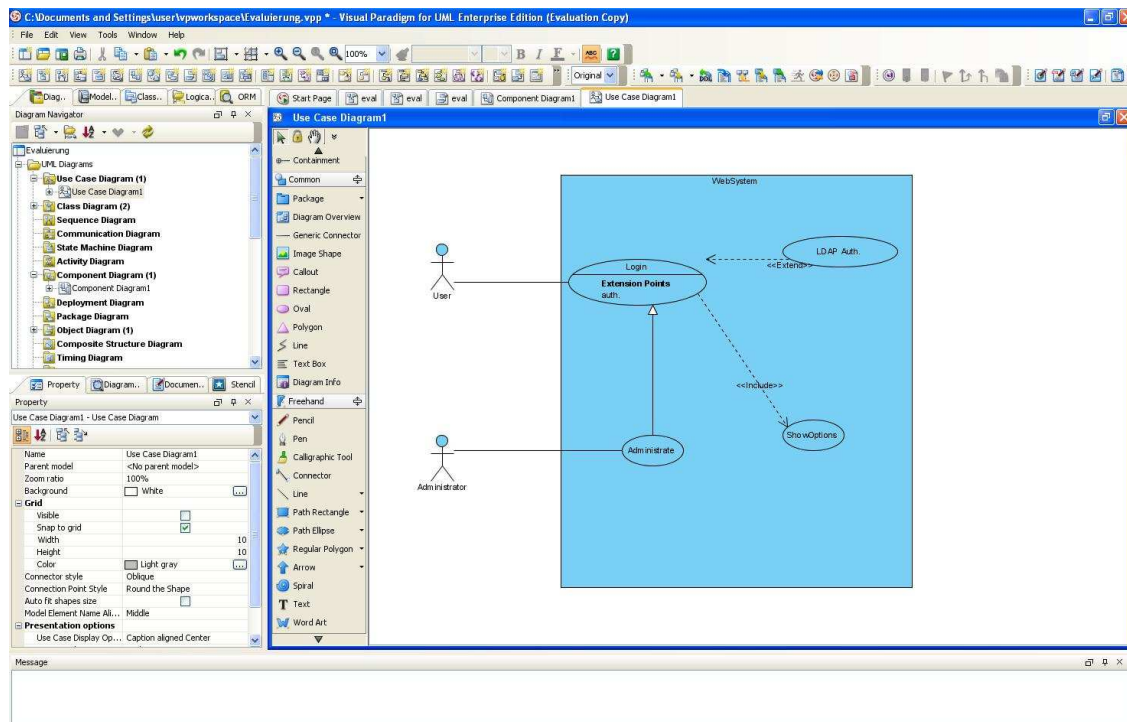


Figure 66: UML Use Case Diagram in "Visual Paradigm for UML"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	79.47%
Component diagram	97.06%
Composition diagram	95.00%
Deployment diagram	66.67%
Activity diagram	60.14%
Sequence diagram	84.44%
Communication diagram	96.15%
Interaction overview diagram	71.88%
Timing diagram	55.26%
State machine diagram	77.38%
Use case diagram	73.68%
Information flows	0.00%
Model management	100.00%
Templates	79.17%
Profiles	50.00%
UML summary	74.10%
Traceability	80.00%
Code generation	40.00%

Tool characteristics	
Promoted UML version	
XMI version	
XMI valid	
Compliance Level	Feature fulfillment
Valid range	L2-0
L1.3	
L1.4	
L2-0	
L2-M	
L2-1	
L2-2	
L2-3	
Assigned level	

5.73 Visual Studio Ultimate by Microsoft

Name: Visual Studio Ultimate

Version: 10.0.30319.1

Vendor: Microsoft

URL: <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/ultimate>

Price (if available): 11899\$

Vendor statement: *Microsoft Visual Studio 2010 Ultimate is the comprehensive suite of application lifecycle management tools for teams to ensure quality results, from design to deployment. Whether you're creating new solutions or enhancing existing applications, Visual Studio 2010 Ultimate lets you bring your vision to life targeting an increasing number of platforms and technologies—including cloud and parallel computing.*

User interface: views, direct editing of diagram elements in the diagram, menu and toolbar

Comments: The tool is very well integrated into the Visual Studio Suite. A free feature pack must be installed to import XMI-files and to generate code.

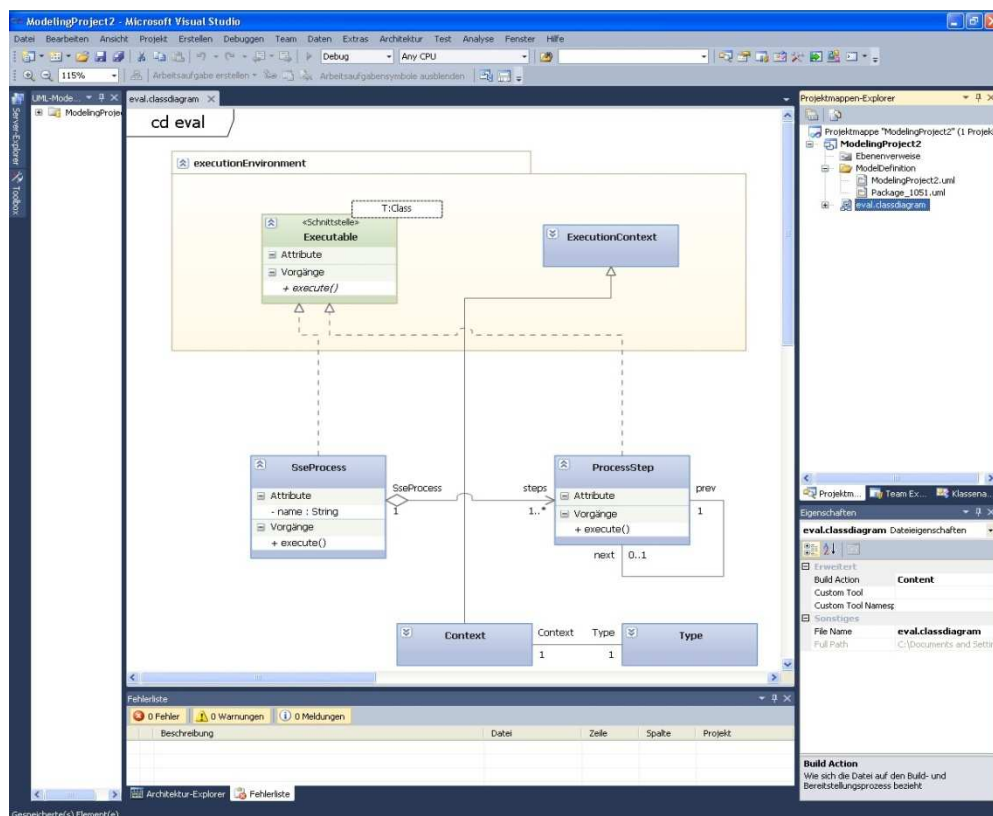


Figure 67: UML Class Diagram in "Visual Studio Ultimate"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	62.25%
Component diagram	88.24%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	39.13%
Sequence diagram	51.11%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	73.68%
Information flows	0.00%
Model management	83.33%
Templates	50.00%
Profiles	0.00%
UML summary	39.07%
Traceability	0.00%
Code generation	12.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	52.10%
L1.4	52.61%
L2-0	79.75%
L2-M	64.40%
L2-1	48.06%
L2-2	38.88%
L2-3	39.15%
Assigned level	L2-0C

5.74 Visual UML by Visual Object Modelers Inc.

Name: Visual UML

Version: 5.26 build 634

Vendor: Visual Object Modelers Inc.

URL: <http://www.visualuml.com/>

Price (if available): 495\$-995\$

Vendor statement: *Visual UML (VUML) is an affordable, easy-to-use yet powerful and full-featured highly UML-compliant object-modeling tool that provides complete and comprehensive support for all ten of the diagram types defined in the OMG 1.3 & 1.4 UML (Unified Modeling Language) specifications: Class, Object, Package, Use Case, Collaboration, Component, Deployment, Activity, State and Sequence diagrams. Plus, Robustness diagrams. Plus, the following diagram types for UML 2.0: Activity, State Machine, Communication, Interaction Overview.*

User interface: dialogs, menu and toolbar

Comments: The evaluator noticed problems with nested objects, in particular when changing the model elements tree. At most one stereotype can be applied to a model element (instead of multiple as specified by the UML).

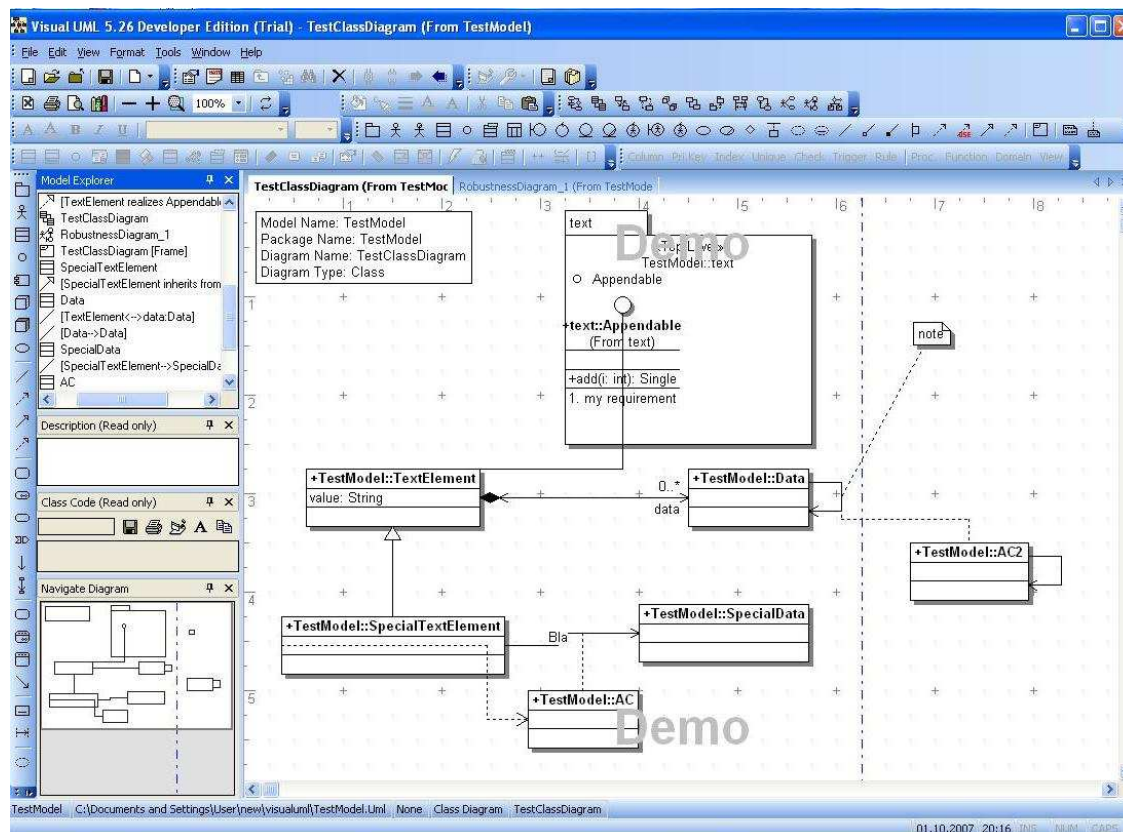


Figure 68: UML Class Diagram in "Visual UML"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	66.89%
Component diagram	38.24%
Composition diagram	40.00%
Deployment diagram	37.88%
Activity diagram	53.62%
Sequence diagram	76.67%
Communication diagram	80.77%
Interaction overview diagram	71.88%
Timing diagram	0.00%
State machine diagram	69.05%
Use case diagram	89.47%
Information flows	0.00%
Model management	0.00%
Templates	45.83%
Profiles	30.00%
UML summary	57.86%
Traceability	30.00%
Code generation	28.00%

Tool characteristics	
Promoted UML version	1.x/2.0
XMI version	1.0/1.1
XMI valid	partial
Compliance Level	Feature fulfillment degree
Valid range	L1.3, L1.4, L2-0C
L1.3	76.89%
L1.4	76.83%
L2-0	74.68%
L2-M	63.87%
L2-1	61.81%
L2-2	60.00%
L2-3	57.98%
Assigned level	L1.4 / partial L2-0C

5.75 yED by yWorks

Name: yEd

Version: 3.4.0.2

Vendor: yWorks

URL: http://www.yworks.com/en/products_yed_about.html

Price (if available): 0\$

Vendor statement: *yEd is a powerful diagram editor that can be used to quickly and effectively generate high-quality drawings of diagrams. Create your diagrams manually or import your external data for analysis and auto-magically arrange even large data sets by just pressing a button.*

User interface: dialogs, menu and toolbar

Comments: -

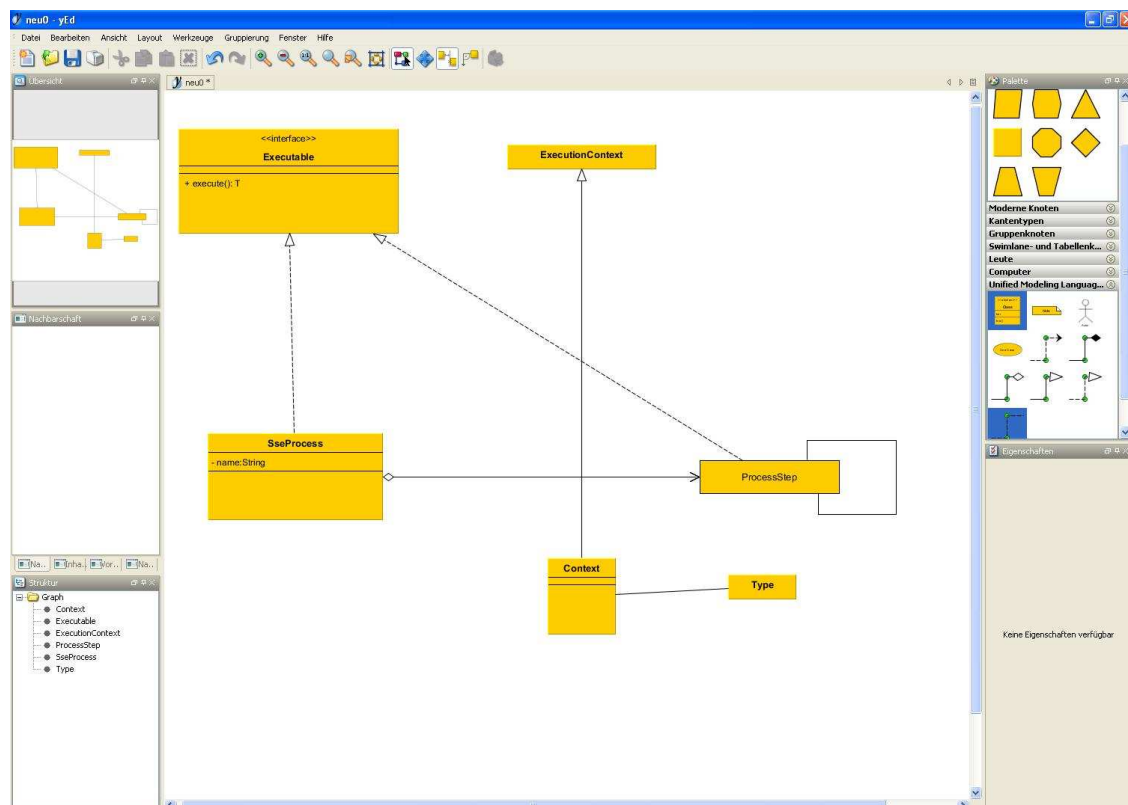


Figure 69: UML Class Diagram in "yEd"

Individual evaluation results:

Modeling criteria	Feature fulfillment degree
Class diagram	20.20%
Component diagram	0.00%
Composition diagram	0.00%
Deployment diagram	0.00%
Activity diagram	0.00%
Sequence diagram	0.00%
Communication diagram	0.00%
Interaction overview diagram	0.00%
Timing diagram	0.00%
State machine diagram	0.00%
Use case diagram	60.53%
Information flows	0.00%
Model management	0.00%
Templates	0.00%
Profiles	0.00%
UML summary	8.92%
Traceability	10.00%
Code generation	0.00%

Tool characteristics	
Promoted UML version	-
XMI version	-
XMI valid	-
Compliance Level	Feature fulfillment degree
Valid range	L2-0C
L1.3	16.39%
L1.4	16.28%
L2-0	21.52%
L2-M	20.42%
L2-1	11.94%
L2-2	9.66%
L2-3	9.15%
Assigned level	-

6 Results

In this section we summarize the results of this study. In contrast to the individual results per tool given in the previous chapter, we discuss here the aggregated results for all tools from different perspectives. In Section 6.1 we discuss the results by considering UML as a whole not distinguishing individual diagram types. In Section 6.2 we focus on the feature fulfillment for individual diagrams. Finally, in Section 6.3 we discuss the findings derived from the additional non-UML categories, i.e. the information collected on traceability and code generation.

As described in Section 3.5, we consulted tool listings to find the initial set of tools to be evaluated. In Figure 70 we display a summary on the reasons why tools are not considered in our evaluation. Several tools mentioned in tool lists are not available, because the product does not exist anymore or the vendor homepage is replaced by a link farm pointing to arbitrary (often unrelated) information. For several tools problems occurred while evaluating the tool, i.e. the evaluator recorded exceptions or that no basic UML functionality is available. More individual problems were not characterized further, e.g. that technical problems could not be solved by the support or only code reverse engineering functionality to UML diagrams is offered. The remaining 71 tools being evaluated consist of 11 pure drawing tools, e.g. Microsoft Visio and 60 UML modeling tools, 10 of 58 tools are implemented as an extension of Eclipse.

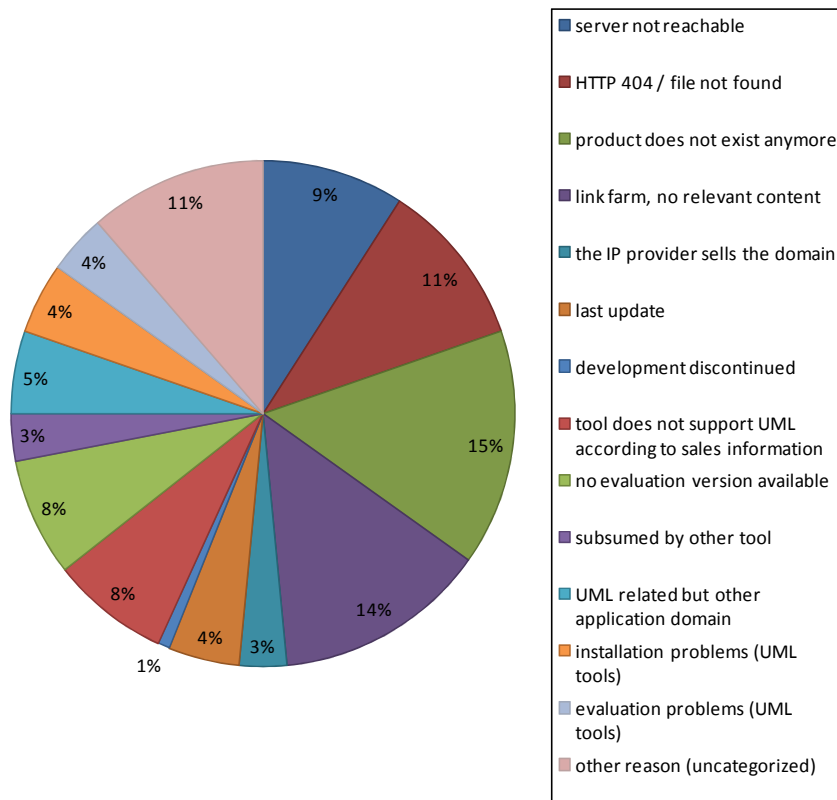


Figure 70: Summary on the reasons of not considering a tool in the evaluation.

6.1 Realization of the UML at Large

In this section, we discuss the aggregated results on level of the entire UML, i.e. we describe how the UML specification as a whole is realized by the evaluated tools. First, we have a look on the overall UML feature realization, then we provide a summary of the assigned compliance levels. Finally, we discuss additional aspects like model exchange, OCL, model consistency mechanisms and automatic layout, i.e. also a summary on additional data collected while conducting the evaluation but not explicitly stated for each individual tool.

6.1.1 Tool Capabilities

Figure 71 displays a high-level overview of the tool capabilities, i.e. of all collected UML features regardless of UML compliance levels. The additional categories traceability and code generation are not considered, but the ability of producing valid XMI is included. Thus, considering all recorded UML 2 features, the chart in Figure 71 shows that 29% of the tools provide a rather limited realization of UML, most of the tools provide a partial realization of UML and 7% of the tools appear to be the best implementations (range of 60-79%). No tool reaches a feature fulfillment degree of 80-100%.

As summarized in Table 2, only SparxSystem Enterprise Architect implements all UML categories, but in fact several detail features are missing. The other tools rated at a high fulfillment degree also miss several categories, e.g. Magic Draw do not provide an user visible realization of timing diagrams or information flows, UModel or Visual Paradigm do not provide information flows. On several categories like composition diagrams or model management many tools realize 100% fulfillment degree or at least 95%.

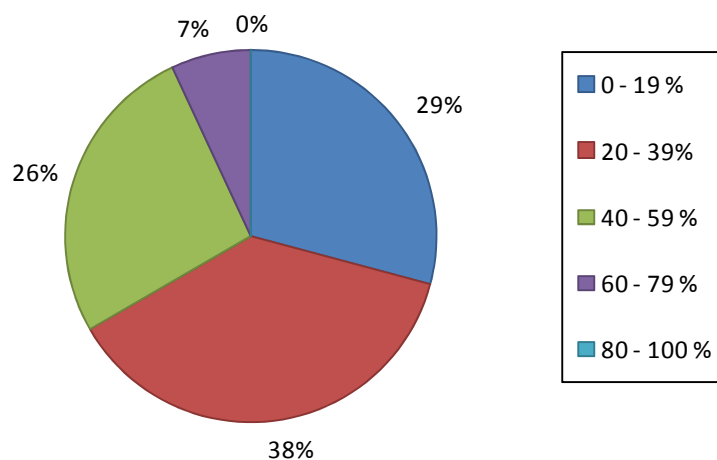


Figure 71: Overview of the realization of tool capabilities.

Data is presented including XMI validation regardless of compatibility levels. The pie slices depict feature fulfillment degrees.

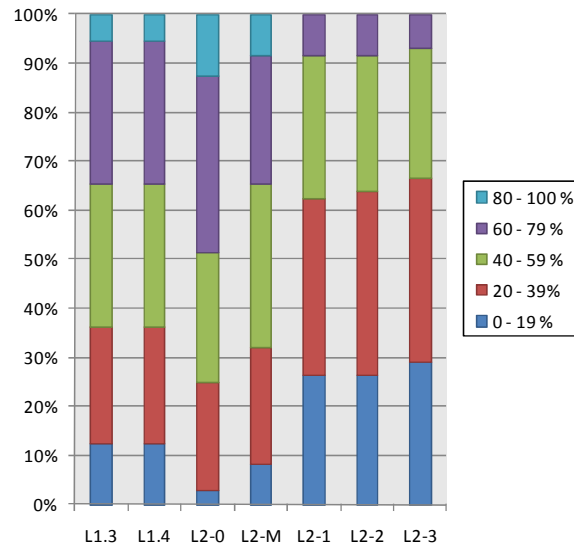


Figure 72: Summary of the realized features grouped according to UML compliance.

The legend points to the degree of feature realization due to which the UML compliance levels were assigned. The chart displays which degree is realized by how many tools.

6.1.2 UML Compliance

As outlined in Section 4.3, we determined the feature fulfillment degree for each tool and related the results to the compliance levels using compliance profiles. Figure 72 depicts how many features required by a certain compliance profile representing a compliance level are realized by how many tools. In fact, this summary shows only the aggregated feature realizations but not whether the diagram types required by a compliance level are met. This is considered in the next paragraph below. 5.6% of all tools reach nearly full compliance to L1.3 and L1.4m 12.5% to L2-0 and 8.33% to L2-M. No tool was able to enter the full compliance range for the advanced levels L2-1, L2-2 or L2-3, but for 8.33% we can attest an acceptable compliance to level L2-1, L2-2 and L2-3.

Figure 73 depicts the summary on the assigned UML compliance levels as described in Section 3.4. No tool reaches acceptable compliance with full level L2-2 or L2-3, some tools

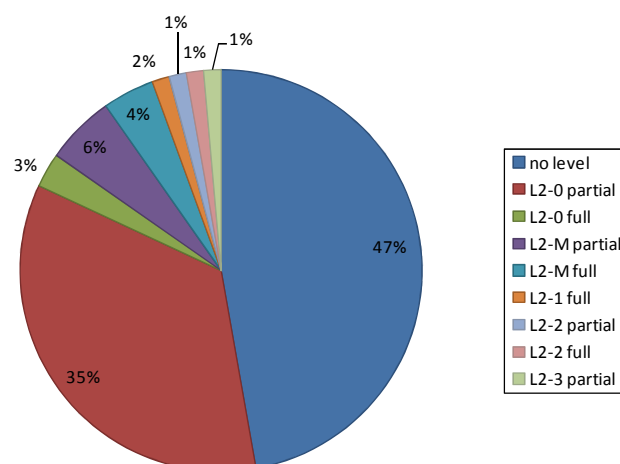


Figure 73: Summary of the assigned compliance levels.

The pie slices visualize the tools being assigned to the same compliance level.

(SparxSystems Enterprise Architect, Altova UModel professional, Visual Paradigm for UML, Metamill) reach partial compliance with L2-2, L2-3 or even full compliance with L2-1 or L2-2. NoMagic MagicDraw, the tool for which we detected the highest degree of feature fulfillment (79%), we can assign only the UML compliance level L2-0/L2-MAC due to some missing diagram types. In summary, to 37.5% of the tools we assigned partial or complete level L2-0, but to 93% of these tools we can only assign level L2-0C, i.e. to level 0 with concrete syntax compliance, because no appropriate XMI version is realized. In fact, 47% of the tools are not assigned to any compliance level due to an incomplete implementation of the basic level requirements defined by UML.

6.1.3 Additional Aspects

Regarding model exchange, 7% (five tools) are able to pass the XMI validity test, for 25% we detected problems in the XMI serialization and the remaining 68% do not implement XMI at all. Compared with the results in [21, 36, 37], some of the tools in our version now provide more current implementations of XMI. However, in our evaluation the analyzed versions of Fujaba, Rational Rose and Microsoft Visio had no XMI implementation at all.

Considering the additional data collected while evaluating the tools, only 23% of the tools implement a model consistency mechanism, e.g. checking for OCL validity constraints etc. Independent from that statement, we are able to detect OCL support for 23% of the tools (regardless of compliance to the OCL specification and the realized OCL version as stated in Section 4.2). 25.6% of the tools offer a (textual) mechanism for comparing models and identifying model differences, 10.8% rely on an external tool or team server for versioning of models.

Even if the main focus of this study is not on automatic layout of diagrams, we also collected additional information on the results produced by automatic layout mechanisms. In contrast to edge routing facilities implemented in most of the diagram editors, here the term automatic layout refers to the ability of calculating the layout, i.e. the spatial arrangement of the diagram elements for a diagram. 41.6% of the tools provide automatic layout facilities. For 70% of these tools the results appear to be crowded, i.e. no appealing placement of the elements was identified by the evaluator. In these cases, usually neither containment nor avoidance of overlaps among node-like elements like classes or use cases is considered. For 16.6% we can attest an acceptable automatic layout for the reference class diagram, i.e. problems as described above did not occur but the layout could be improved significantly. We can attest for only 2 tools i.e. 13.3% a good automated layout (e.g. MagicDraw and VisualParadigm). As inferred from the layout results, it seems that mostly hierarchical layout algorithms or force directed methods (spring embedder) are implemented. Visual Paradigm is the only tool which provides layout facilities for all diagram types. In summary, this result is less disappointing than our specific evaluations for UML class diagram layout in [11, 13], where we analyzed 42 UML tools.

6.2 Realization of the UML on Diagram Level

In this section we discuss the feature fulfillment on the level of individual UML diagrams. First, we give an overview on all modeling facilities, i.e. all UML language units. Then we discuss details on individual diagrams, namely class diagrams, activity diagrams, sequence diagrams and state machine diagrams, the diagram types which are identified in [9] as the most frequently used ones.

An overview of the degree of feature realization grouped according to the top-level categories of the feature hierarchy, i.e. the main diagram types and auxiliary UML facilities, is shown in Figure 74. The entire chart displays the percentage of tools realizing the degrees of feature fulfillment according to the top-level categories. Thus, class diagrams and use case diagrams are widely implemented with acceptable results. This result also fits to the results of a current survey on the usage of UML diagrams [9], where class diagrams and use case diagrams were identified as the most used UML diagrams. Several of the remaining diagrams of UML 1.x, i.e. component deployment, activity, sequence and communication (formerly collaboration) diagrams are on average or moderate realization level. In particular, activity and component diagrams were significantly changed in UML 2 but, however, it is notable that also diagrams known from UML 1.x, e.g. communication diagrams are on moderate or lower level of realization. In fact, only few tools implement the new UML 2 facilities, i.e. interaction overview or timing diagrams, information flows or profiles. Thus, only few tools support modeling and application of UML profiles as intended for lightweight extensions of the UML. Surprisingly, templates, which are known as concept also in the earlier versions of UML, are not supported at all by 60% of the tools.

In Figure 75, the realization of selected UML features for class diagrams is depicted. The results are categorized according to the feature groups as recorded in the feature hierarchy given in Section 3.2. The entire chart depicts the percentage of tools realizing the shown fulfillment levels. Obviously, association types, i.e. the discrimination according to association, aggregation or composition, interface realizations and the different visibilities (public, private, protected, package) are implemented properly by most tools (more than 30/40% of the tools). Several missing features are detected for instance specifications, i.e. object diagrams, association classes (including further relations to association classes), higher associations like ternary associations as well as for the presence of diagram frames as introduced in UML 2.0.

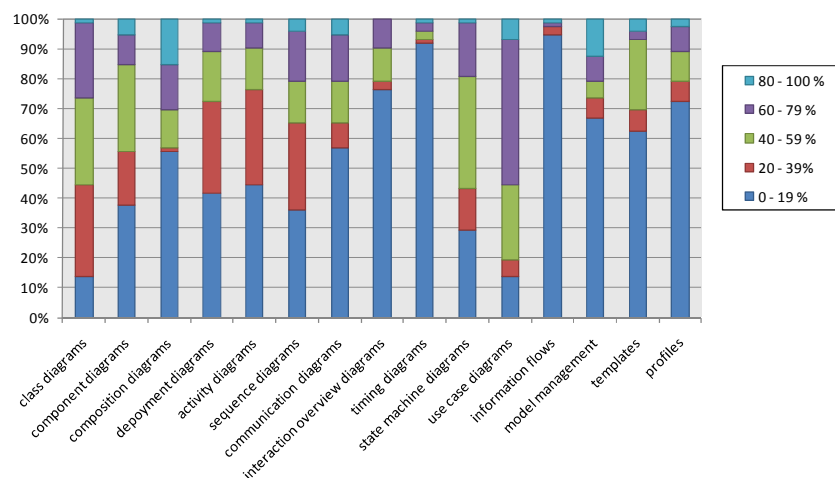


Figure 74: Realization of selected UML features.

The data is displayed according to the top-level categories as derived for the feature hierarchy, i.e. according to the main sections of the UML superstructure specification.

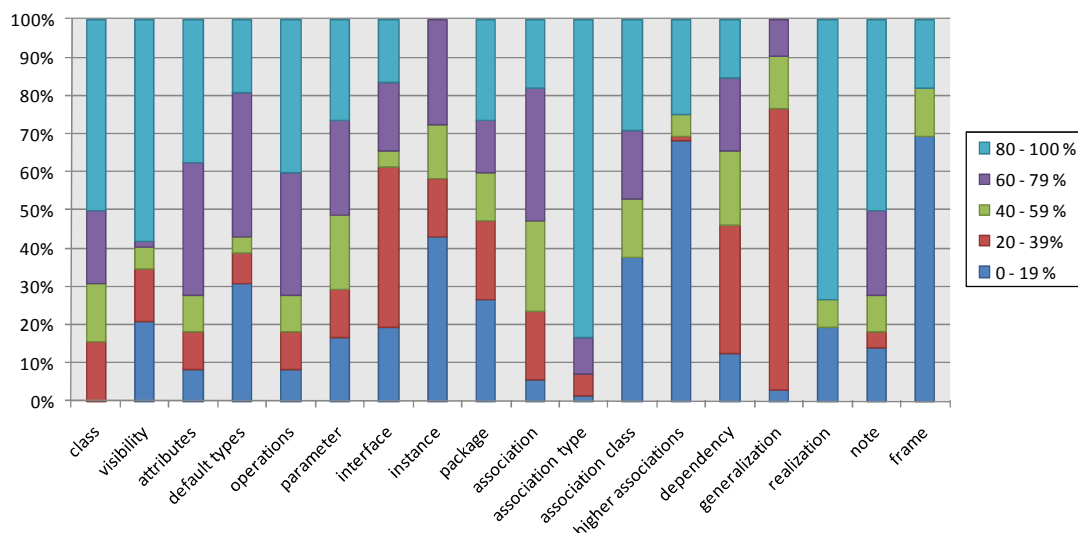


Figure 75: Realization of UML features for class diagrams.
The data is categorized according to selected feature groups.

Figure 76 shows the realization of selected feature groups for activity diagrams. The entire diagram depicts the percentage of tools realizing certain degrees of feature fulfillment of selected feature groups in the feature hierarchy. At first glance, the state of realization as shown in Figure 76 appears to be low. In fact, many tools only realize UML 1.x features for state machines. This can be found in Figure 76, e.g. by discussing the results for the feature groups “action” and “action types”. To give a brief insight, the feature group “action” contains pins and parameter sets, the feature group “action types” the 18 action types like “broadcast signal”. The entire feature hierarchy can be found in Appendix A. Thus, several features introduced in UML 2 are not properly realized by many tools. In summary, less than 10% of the tools are able to provide acceptable support for UML 2 activity diagrams.

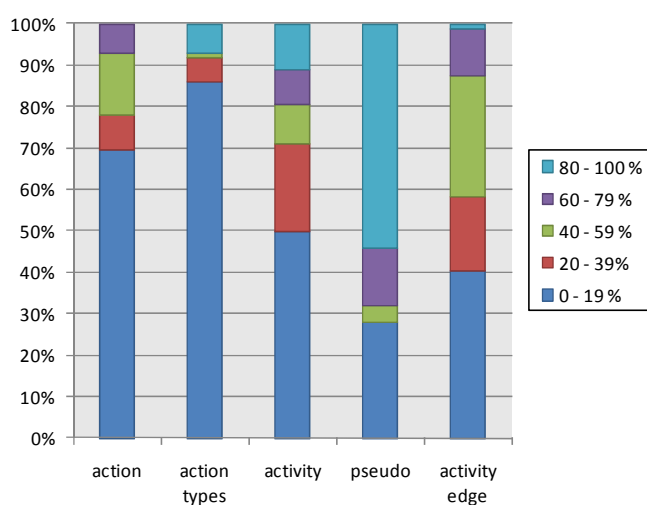


Figure 76: Realization of UML activity diagrams.
The data is grouped according to selected feature groups.

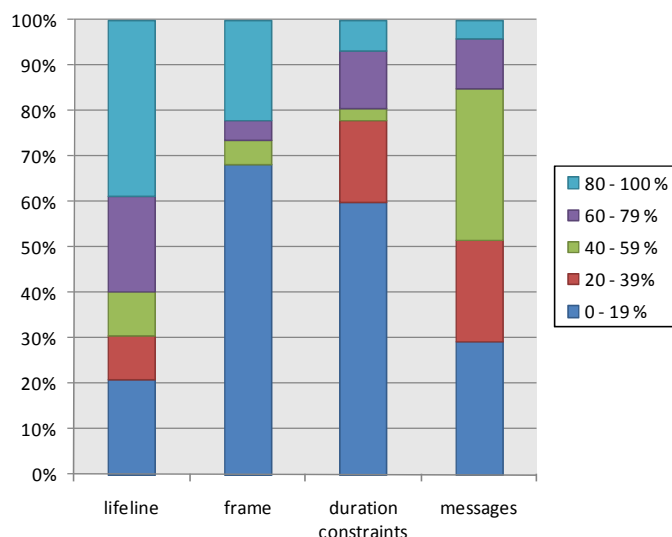


Figure 77: Realization of UML sequence diagrams.

The data is grouped according to selected diagram elements, relations and features.

Similar to the discussion on activity diagrams for Figure 76, the realization of selected feature groups for sequence diagrams is shown in Figure 77. The entire diagram depicts the percentage of tools realizing certain degrees of feature fulfillment of selected feature groups in the feature hierarchy. Also for sequence diagrams, most tools are on UML 1.x level. This can easily be seen in Figure 77 by considering the realization of frames. In fact, UML 2 added several kinds of frames for sequence diagrams, e.g. to specify loops and alternatives for messages. More than 65% of the tools do not realize the various types of frames and only 25% of the tools realize more than 60% of the features specified for frames. Features capturing the duration of messages, i.e. constraints on the execution time of messages are not realized by many tools, even if similar constructs were present in UML 1.x. Also the types of messages, like create, destroy or lost and found are not implemented by many tools. The basic features for sequence diagrams, i.e. the lifeline, nested messages etc. are implemented by most tools.

At a first glance, regarding Figure 78 the support of modeling elements for state machines appears to be more comprehensive. The diagram summarizes the percentage of tools realizing certain degrees of feature fulfillment of selected feature groups. The realization of features related to simple states, state transitions and pseudo states (start, final, fork, join, etc.) is on a good level considering all tools. The degree of fulfillment for protocol state machines shows that more than 95% of the vendors do not adopt this specialized version of state machines introduced in the UML 2 specification.

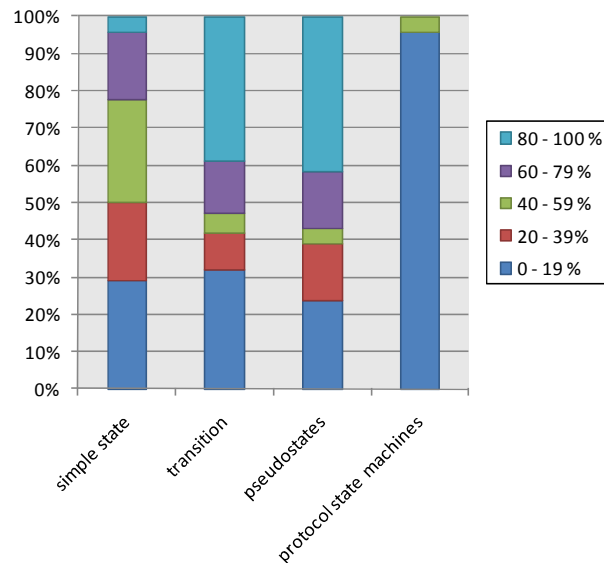


Figure 78: Realization of UML state machine diagrams.
The data is grouped according to selected feature groups.

Figure 79 depicts the realization of features for UML use case diagrams according to some selected feature groups, i.e. the diagram summarizes the percentage of tools realizing certain degrees of feature fulfillment of the selected feature groups. As discussed also for Figure 74, the average realization of use case diagrams appears to be good. The feature groups “use case” and “actor” contain the display option on showing the element as graphical figure (oval or stick man) or in classifier notation. Thus, only few tools provide this choice and many realize the graphical figure only. The typical relations among use cases including the appropriate stereotypes are implemented by most tools (more than 30%).

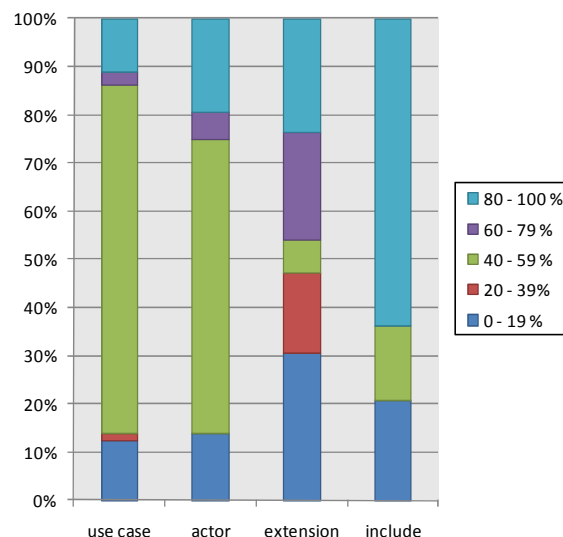


Figure 79: Realization of UML use case diagrams.
The data is categorized according to selected feature groups.

6.3 Additional Information

We collected also additional information to gain an impression on capabilities like traceability among diagrams and basic aspects of code generation for model-based and model-driven software engineering.

In this study, for traceability we are interested in relations among diagram elements and diagrams as well as diagrams to diagrams. A common case for links between diagram elements and diagrams is that UML a state machine can be used to describe the behavior of a certain classifier, e.g. a class, and thus, the user should be able to assign the state machine to the classifier. Similar links may occur from sequence diagrams to referenced sequence diagrams (via reference or call frames), etc. 50% of all tools do not implement any links between modeling elements, 12.5% implement one type of links, usually from a classifier to an assigned state machine, the remaining 37.5% multiple types of links. Links among diagrams are specified in UML 1.x as hyperlinks. 30.6% of the tools (still) implement arbitrary links among diagrams.

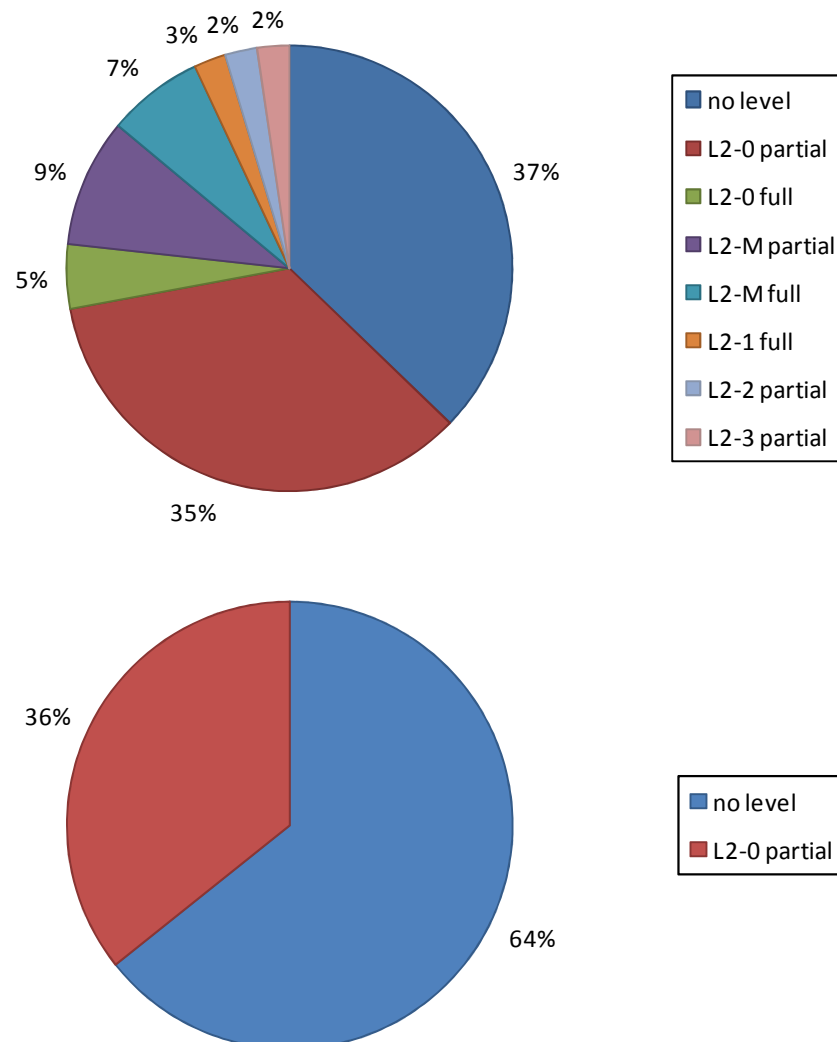


Figure 80: Summary of the assigned compliance levels.

The pie slices visualize the tools being assigned to the same compliance level. The upper chart displays the compliance levels assigned to commercial tools, the lower chart the assignment to open source or free tools.

For code generation we were interested in multiple aspects. We discuss some selected aspects and findings below:

- Which diagrams are considered by the tool for code generation? 59.7% of the tools consider class diagrams to generate code, 9.7% of all tools produce behavioral code from state machines or from other diagram types such as component diagrams (5.6%), sequence diagrams (4.1%) or activity diagrams (4.1%). Surprisingly, only two tools, Telelogic Rhapsody and Artisan Studio (Uno) allow the user to simulate and visually debug the code generated for state charts.
- Which target languages are generated? 58.3% of the tools provide code generation functionality for Java, 41.6% for C++, 27.8% for C# and 12.5% for VB. Other target languages include Ada, Python and Eiffel.
- Do associations have influence on the generated code, e.g. the navigation direction (bi-directional navigation) or the type of associations, i.e. as aggregation or composition? According to our findings in 37.5% the type of the association or the navigation direction have an influence on the generated code, e.g. by special data types or appropriate accessor methods.
- Can custom transformations be specified by the user? 16.6% allow the customization of the code generation by templates and only one tool (Together) allows the specification of QVT (Query View Transform) as specified by the OMG.

As a summary, most of the produced code is organized as code templates to be filled in by the user. Only few tools consider further diagrams to produce also behavioral code.

6.4 Comparison to the last study

On the one side, the results presented in this study can be seen as a standalone snapshot of the market for UML tools in the second part of 2010. On the other side, this study can be understood in the context of its history, i.e. as a reevaluation of the tools in [12]. In this section, we compare the results discussed in this report with the original data presented in [12].

For 30 (46%) of the 64 tools evaluated in [12] the respective vendor provided a relevant update. The average difference of the feature fulfillment degree for these tools within in 1 year is 3.27%. We detected an increase of the fulfillment degree for 25 of these tools particularly for four tools with the highest increase, namely Objectteering (18%), UMLet (15%), Innovator (10.7%) and Metamill (8.1%). However, we recorded the same fulfillment level for four tools and for further four tools a loss in functionality, e.g. due to disabling UML 1.x support and thereby skipping existing diagrams or due to a more detailed view on the XMI compliance.

For 14 out of these 33 tools we detected a change in their individual compliance level. 9 tools increased their compliance level, e.g. Metamill (from partial L2-0C to L2-1AC) or Metaedit (from no level to partial L2-0C). Neither Objectteering nor Innovator mentioned above based on their large increase in feature fulfillment managed to improve their compliance level. As discussed as a conclusion in [12], this contradicting assessment occurs due to the coarse grained definition of the UML compliance level and the large differences between L2-1, L2-2 and L2-3.

From a tool perspective, only some tools managed an increase of their compliance level or their feature fulfillment degree. From an overall perspective, the assignment of compliance levels (as shown in Figure 73) was changed significantly, i.e. in this evaluation we did not assign L1.3 or L1.4 anymore and full Level L2-1 was assigned. This change can also be found in Figure 80. Here, the assignment of feature fulfillment degrees to commercial tools is more diverse in this evaluation while free and open tools are now either assigned to L2-0 partial or to no level (L2.M

is missing). On more detailed level, e.g. on diagram level as discussed in the last sections, the analysis from [12] is still valid with a slightly trend in increased feature fulfillment.

7 Summary

UML models are widely used in software engineering. If a UML modeling tool is chosen which does not properly adopt the specification, serious problems may occur so that the user may be seriously restricted in applying UML. This affects the typical application areas of UML models, i.e. design and documentation of software systems. In fact, this is also a drawback for model-driven software engineering approaches relying on models according to the UML specification. As a remedy, detailed capability and compliance evaluations can provide an in-depth description of the tools, may support tool selection and migration and give an overview on the current state of implementation.

In this report, we presented a feature-based evaluation approach to determine the capabilities of UML tools in terms of feature fulfillment and to derive UML compliance levels from the collected feature data. The approach relies on an encompassing feature hierarchy extracted from the UML specification to capture data on tool capabilities. To determine the UML compliance in terms of compliance levels as defined by the OMG we used the feature data in combination with special compliance profiles for given UML versions. We evaluated 72 out of an initial set of more than 200 tools, covering all major UML tools used in industrial practice today. The remaining tools were not available for evaluation, did not exist anymore, or were not maintained for a longer time, etc.

Regarding the tool capabilities, 7% of the tools evaluated in this study are on a feature fulfillment degree of 60-79%, no tool realizes more than 80% and the majority of tools is below 60% feature fulfillment. New features or language units introduced with UML 2 are often disregarded, e.g. timing diagrams or extended modeling elements for activity or sequence diagrams.

Regarding UML compliance as derived from the tool capabilities data, three tools reach acceptable compliance levels (level 2 or level 3) and more than 47% of all evaluated tools do not implement the UML specification sufficiently, neither with respect to modeling capabilities nor with respect to UML compliance.

The reevaluation of changed tools since our last evaluation indicates minor improvements of the UML implementation of the analyzed tools. Four tools managed to increase their feature fulfillment degree by more than 6%, three of them by more than 10%. Furthermore, nine tools were able to increase their compliance level and we did not assign levels for UML 1.x anymore.

The detailed findings of all feature groups for all tools not being restricted for publication by the respective vendor can be found on <http://www.sse.uni-hildesheim.de/UMLtools11/>

Evaluating a large set of complex tools is an activity which imposes much effort. This was true for the initial study [12] in which we aimed at a comprehensive and detailed characterization of the tool market and it is particularly true for this reevaluation for which we increased the level of collected details during evaluation, e.g. by additional reference diagrams. In fact, continuing this work in terms of regular reevaluations is not possible for us due to resource restrictions. Furthermore, our aim of providing a comprehensive overview is delimited by the publication policies of some tool vendors. The number of tools for which are not allowed to publish detailed data increased and, particularly for this reevaluation, impacts the coverage of the UML tool market. For the future, we plan to consider tools for a (re)evaluation only on explicit request of the respective tool vendor (a license which permits publication is prerequisite) or a specific analysis for tool users on request.

Acknowledgements

We thank all the vendors for providing their tools according to their terms of evaluation licenses. For future evaluations we hope that the companies will support us again with download versions and evaluation licenses.

We also thank our students for developing additional tool support.

A Feature Hierarchy

This appendix describes the hierarchy of features as applied while collecting data for the tool evaluation. First we show the features used to derive the UML compliance level. The features are enumerated in tables so that the table header denotes the modeling facility, each feature group is given as a table row and the individual features are listed along with the feature group. Please note, that additional information like the tabular notation defined in the UML appendix for some diagrams are also collected but not considered in compliance level calculation and therefore printed in italic font face. At the end of this appendix we list the additional features used to collect information on traceability and code generation.

General Information
<i>Name</i>
<i>Vendor</i>
<i>Version</i>
<i>Price if available</i>
<i>URL</i>
<i>Eclipse plugin, integration, standalone plugin, other tool</i>
<i>license open, community, eval, academic, commercial, free</i>
<i>Operating System windows, mac, linux, online</i>
Data Handling
<i>XMI</i>
<i>DI</i>
<i>XML validation xmi, di</i>
<i>Eclipse version (plugins/RCP only)</i>
<i>relies on eclipse EMF, GEF, UML2, OCL, JET</i>
<i>UI concept: dialogs, views, direct edit in diagram, menu, toolbar, drag&drop menu, literal programming, gestures</i>
<i>undo</i>
<i>(other) file formats</i>
<i>multi-user</i>
<i>repository</i>
<i>specific versioning mechanism (not [repository] files by SVN/CVS)</i>
Metamodel

<i>promoted UML version</i>
<i>consistency check, configurable</i>
<i>ocl support</i>
Extensibility
<i>plugins</i>
<i>API</i>
UML Class Diagrams (see UML chapter 7)
<i>Class name, stereotypes, properties, constraints, fqcn of owner</i>
<i>Visibility +, -, ~, #, a für {abstract}</i>
<i>default stereotypes <<auxiliary>>, <<focus>>, <<implementation class>>, <<metaclass>>, <<type>>, <<utility>>, <<realization>>, <<specification>></i>
<i>multiple stereotypes at one element supported</i>
<i>default properties/constraints {leaf}</i>
<i>classes of same [fqcn] in one diagram prohibited, selection of existing classes allowed</i>
<i>attributes, +, -, ~, #, stereotypes, properties, constraints, array type, static, default value, UML syntax</i>
<i>default types {UML 17.5} boolean, integer, string, unlimitedNatural</i>
<i>default properties/constraints {readOnly}, {union}, {subsets}, {redefines}, {ordered}, {unique}, {nonunique}, {isderived}</i>
<i>operations, +, -, ~, #, stereotypes, properties, constraints, body, static, abstract, return type</i>
<i>default properties/constraints {redefines}, {query}, {ordered}, {unique}</i>
<i>parameter:, direction, type, name, array type, stereotype, constraint, property, default value, uml syntax return->op</i>
<i>free compartments</i>
<i>Interface stereotype, lolly, required</i>
<i>instance Specification, {Object} link, attribute values</i>
<i>instance specification: anonymous instance</i>
<i>Package, +, -, #, ~, stereotype, properties, constraints, nesting of elements, [anchor for containment]</i>
<i>default stereotypes <<framework>>, <<modelLibrary>></i>

<i>Association unidirectional, bidirectional / undirected, forbidden, role, multiplicities, association name, qualifiers, properties, constraints, lined constraint {dependency}, generalization, reading direction, stereotype, owning point, reflective</i>
<i>association type: association, composition, aggregation</i>
<i>Association visibility: +, -, ~, #</i>
<i>default properties/constraints {subsets},{redefines},{union}, {ordered}, {bag}, {sequence}/{seq}, {isabstract}, {isleaf}, {isderived}</i>
<i>association class, name semantics related to association, at reflective association, further relations: generalization, association, reflective association, dependency</i>
<i>higher association: rhomb ternary, arbitrary</i>
<i>higher associations: association class</i>
<i>Dependency unidirectional, usage, merge, import, access</i>
<i>merge semantics supported</i>
<i>default stereotypes <<call>>, <<derive>>, <<instantiate>>, <<responsibility>>, <<send>>, <<trace>>, <<substitute>>, <<abstraction>>, <<use>>, <<refine>></i>
<i>generalization, discriminator, powertype [notation], generalization set</i>
<i>generalization [set] constraints: complete, disjoint, incomplete, overlapping, package generalization is prohibited</i>
<i>default stereotypes realization</i>
<i>Note free, attachable, member attachable</i>
<i>Note: multi member attachable</i>
<i>frame</i>
<i>frame headings [<kind> <name> [<p>arameter]</i>
<i>data type declaration <<datatype>>, <<enumeration>></i>
Component Diagrams (see UML chapter 8)
<i>component stereotype, symbol, nesting of parts</i>
<i>interface provided, required, port, complex port, assembly connector</i>
<i>dependency assembly connector, multiplicity, behavior, behavior notation, port type, port name, interface name</i>
<i>Note free, attachable, member attachable</i>

<i>Frame cmp</i>
<i>frame headings [<kind> <name> [<p>arameter]</i>
<i>default stereotypes <<entity>>, <<implement>>, <<process>>, <<service>>, <<subsystem></i>
<i>multiple stereotypes at one element supported</i>
Composition Diagrams (see UML chapter 9)
<i>Part</i>
<i>port</i>
<i>collaboration</i>
<i>collaboration use</i>
<i>connection connector, role binding</i>
<i>Note free, attachable, member attachable</i>
<i>Frame</i>
<i>frame headings [<kind> <name> [<p>arameter]</i>
Deployment diagrams (see UML chapter 10)
<i>artifact, instance, stereotype, symbol, properties</i>
<i>default stereotypes for artifacts: <<document>>, <<executable>>, <<file>>, <<library>>, <<script>>, <<source>></i>
<i>multiple stereotypes at one element supported</i>
<i>dependencies, deploy, manifest</i>
<i>node type, instance, nesting, textual deployment, execution environment</i>
<i>deployment descriptor stereotype, instance notation, attachable, attributes, values in instance</i>
<i>communication link: direction, multiplicities</i>
<i>generalization</i>
<i>semantics instance/class level</i>
<i>Note free, attachable, member attachable</i>
<i>Frame</i>
<i>frame headings [<kind> <name> [<p>arameter]</i>
Activity diagrams (see UML chapter 12)
<i>action, pin, named pin, effect at pin, state at pin, streaming fill, arrow in pin, parameter set</i>
<i>pin multiplicity, comment at pin, <<selection>>, exception at pin, in/out semantics considered for directed edges, {write}, {create}, {read}, {stream}</i>

<i>action types: addstructuralfeaturevalue, add variable feature, broadcast signal, call behavior, call operation, clear association, clear structural feature, clear variable, create link, create link object, create object, destroy link, destroy object, send object, send signal, unmarshall, value specification, accept event</i>
<i>object node, datastore, central buffer, state, port, state at port</i>
<i>activity, parameter, constraints at parameter, parameter sets, edges to individuals</i>
<i>events accept, send, timer</i>
<i>pseudo final, flow final, initial, decision, merge, fork, join</i>
<i>activity edge, weight, object flow, selection, guard, connector notation</i>
<i>partitions 1dimensional, 2dimensional, sub grouped, partition name in activities</i>
<i>interruptible region, exception handler, exception pin</i>
<i>expansion region, shorthand, structured region</i>
<i>local pre conditions, postconditions</i>
<i>Note free, attachable, member attachable</i>
<i>Frame act</i>
<i>frame headings [<kind> <name> [<p>arameter]</i>
<i>tabular notation {annex E}</i>
Sequence diagrams (see UML chapter 14 especially UML chapter 14.22)
<i>lifeline, execution specification, self messages, nested lifelines/recursive</i>
<i>state invariant</i>
<i>Frame sd, ref, ref syntax/parameter, alt, coregion, consider, assert, par, loop, critical, neg, strict, seq, ignore, opt</i>
<i>frame headings [<kind> <name> [<p>arameter], break frame, ref semantics considered, part decomposition</i>
<i>local attributes, constraints</i>
<i>continuation</i>
<i>duration constraints,time observation, diagonal messages</i>
<i>message asynchronous, synchronous, return, lost, found, general ordering, gate, create, destroy + x</i>
<i>message syntax ident, assignment, arguments, return value, any type, undefined argument</i>
<i>Note free, attachable</i>
<i>default stereotypes <<create>>, <<destroy>></i>
<i>multiple stereotypes at one element supported</i>

<i>tabular notation {UML Annex E}</i>
Communication diagrams (see UML chapter 14 especially UML chapter 14.27)
<i>frame sd</i>
<i>frame headings [<kind] <name> [<p>arameter]</i>
<i>lifeline, {the Object/Instance specification} with state</i>
<i>message, flat sequence numbers, nested sequence numbers, communication direction, name, recurrence, iteration clause, guard</i>
<i>Note free, attachable</i>
Interaction overview diagrams specialized activity diagrams (see UML chapter 14 especially UML chapter 14.28)
<i>frame sd</i>
<i>frame headings [<kind] <name> [<p>arameter]</i>
<i>interaction kind</i>
<i>interaction use as ref</i>
<i>pseudo: decision, merge, fork, join, initial, final, continuation</i>
<i>duration constraints, time observation</i>
<i>properly nesting of branches and joins</i>
<i>frame heading may name invisible lifelines</i>
<i>Note free, attachable</i>
Timing diagrams (see UML chapter 14 espter UML chapter 14.31)
<i>frame sd</i>
<i>frame headings [<kind] <name> [<p>arameter]</i>
<i>message asynchronous, call, reply, label</i>
<i>state axis/timeline:, time ruler/ tick marks, states, state changes, lifelines with object name</i>
<i>value lifeline, value changes</i>
<i>time constraint, duration constraint, state/condition</i>
<i>general ordering</i>
<i>destruction event</i>
<i>Note free, attachable</i>
State Machines (see UML chapter 15)

<i>simple state, refinement state list, regions, tab, compartments, activities in compartment, composite state, decomposition icon, invariant notation, constraint</i>
<i>internal transitions</i>
<i>state machine, submachine state ":"</i>
<i>transition, constraint, guard, event, action, reflective</i>
<i>final, initial, history, deep history, junction, choice, terminate, fork, join</i>
<i>entry point, exit point</i>
<i>alternative entry, alternative exit</i>
<i>frame stm, extended</i>
<i>frame headings [<kind> <name> [<p>arameter]</i>
<i>signal send, receipt, action, deferred trigger</i>
<i>protocol state machines {semantic difference}</i>
<i>protocol transition pre, event, post</i>
<i>Note free, attachable, member attachable</i>
<i>tabular notation (UML Annex E)</i>
Use cases (see UML chapter 16)
<i>use case, classifier</i>
<i>actor class, icon</i>
<i>extension, extension point, constraint in note</i>
<i>include</i>
<i>system, stereotype subsystem</i>
<i>association, multiplicity</i>
<i>generalization usecase, actor</i>
<i>in packages or frames</i>
<i>Note free, attachable, member attachable</i>
<i>Frame uc</i>
<i>frame headings [<kind> <name> [<p>arameter]</i>
Information flows (see UML chapter 17.2)
<i>information item, stereotype flow, black triangle</i>
<i>flow dependency</i>
<i>representation dependency</i>

<i>connectors named</i>
<i>associations with information direction named information item</i>
Model management (see UML chapter 17.3)
<i>Model name, stereotype, properties, constraints</i>
<i>dependencies among models, multiple stereotypes supported</i>
<i>default stereotypes <<metamodel>>, <<systemModel>></i>
Templates (see UML chapter 17.5)
<i>on classifiers, packages, operations, attributes, collaborations</i>
<i>typed/bound by classifier, valueSpecification</i>
<i>template names, types, values, bound types shown as constrained classifier</i>
<i>bind instantiation template assignment in <></i>
Profiles (see UML chapter 18)
<i>extension notation, constraints</i>
<i><<stereotype>>, attributes, constraints, <<metaclass>>, <<profile>></i>
<i>dependencies reference, apply</i>
<i>icons</i>
<i>own stereotypes in diagrams leads to explicit profile?</i>
Traceability and links
<i>linked diagrams class-state, class sequence, state-state, activity-behavior, class-package, class-activity, operation-attribute, sequence-sequence ref, class-class</i>
<i>traceability by hyperlinks</i>
Code Generation / Model Driven Support
<i>code generation class diagrams, component diagrams, composition diagrams, deployment diagram, activity diagram, sequence diagram, interaction overview, timing, state machine</i>
<i>Languages java, c, C++, ada, C#, delphi, pHP, Python, VB.net, visual Basic, Perl, objectiveC, CORBA, iDL, eiffel, emf Ecore</i>
<i>association properites have influence on generated data types</i>
<i>template code class diagram, component, deployment, composition diagrams</i>
<i>behavior activity, statechart, sequence, timing, interaction overview, class</i>
<i>document generation use cases, for classes, test cases, other</i>
<i>debug/simulate state chart</i>
<i>custom transformation template, explicit source code transformation, visitor, qVT, other</i>

<i>code synchronization: automatic, manual</i>
<i>MetaModel changeable, profile application, stereotypes only</i>

B Reference Diagrams

In this section we show the remaining reference diagrams not given in Section 3.3. The two reference diagrams from the previous UML tool evaluation [12] were also used as reference diagrams in this evaluation. Furthermore, additional reference diagrams were created in order to cover the remaining UML diagram types. The evaluator was requested to model all reference diagrams and store the results as screen shots in the evaluation repository.

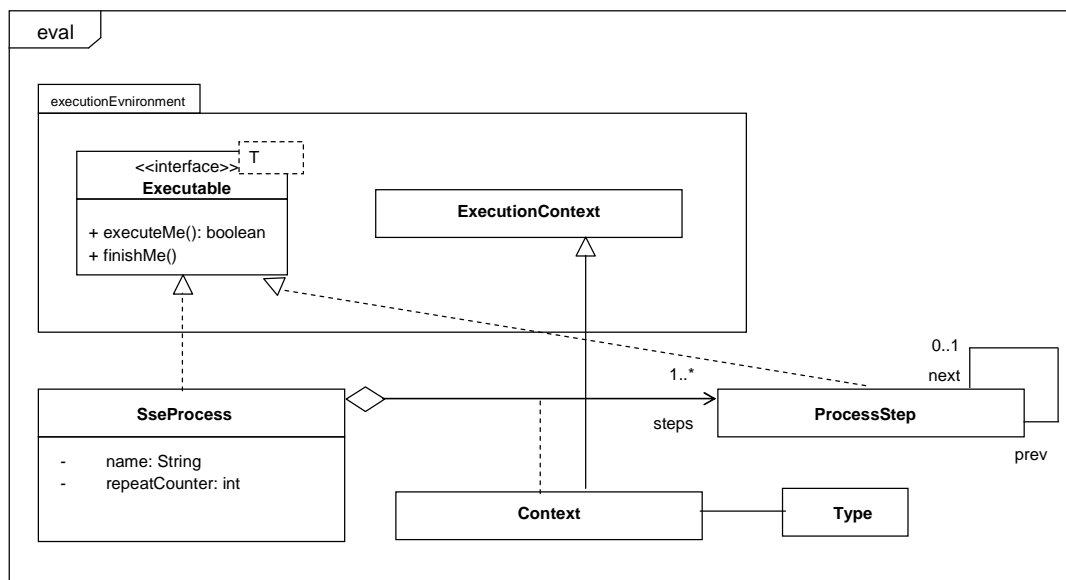


Figure 81: Additional Reference Class Diagram

Deriving the association class “Context” to an additional ordinary class **must not be allowed**.

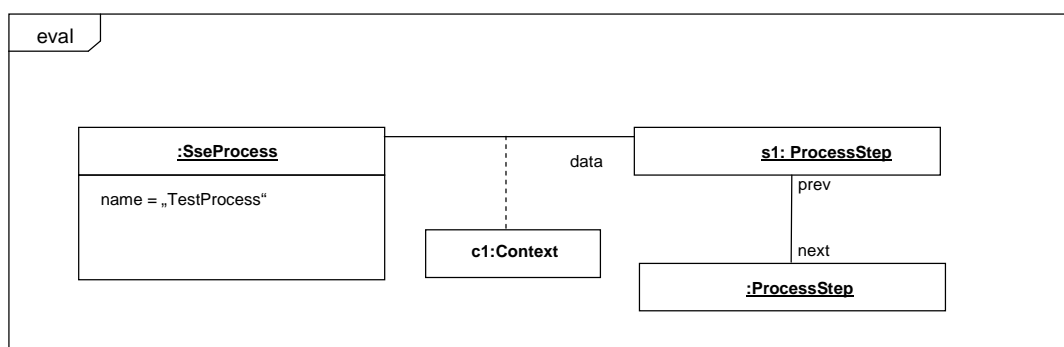


Figure 82: Reference Instance Diagram
based on the class diagram in Figure 81.

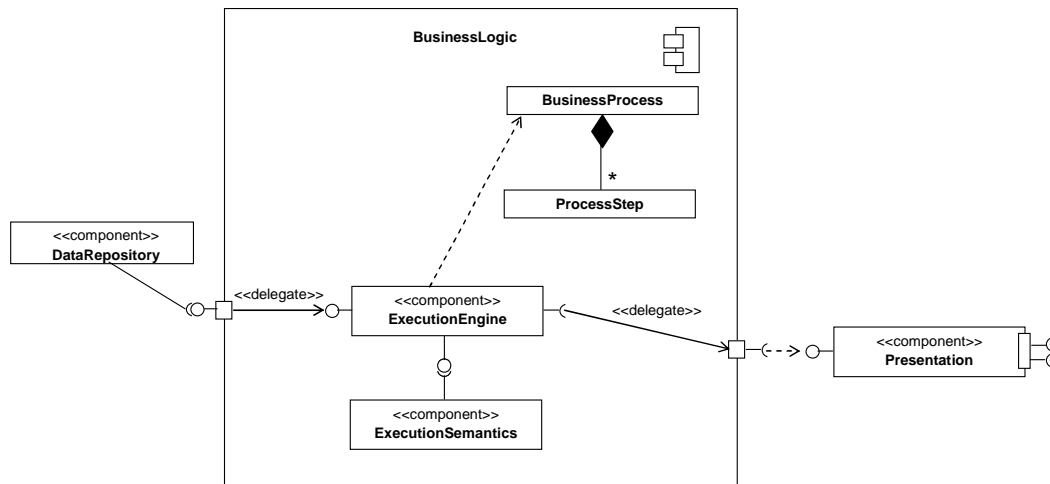


Figure 83: Reference Component Diagram

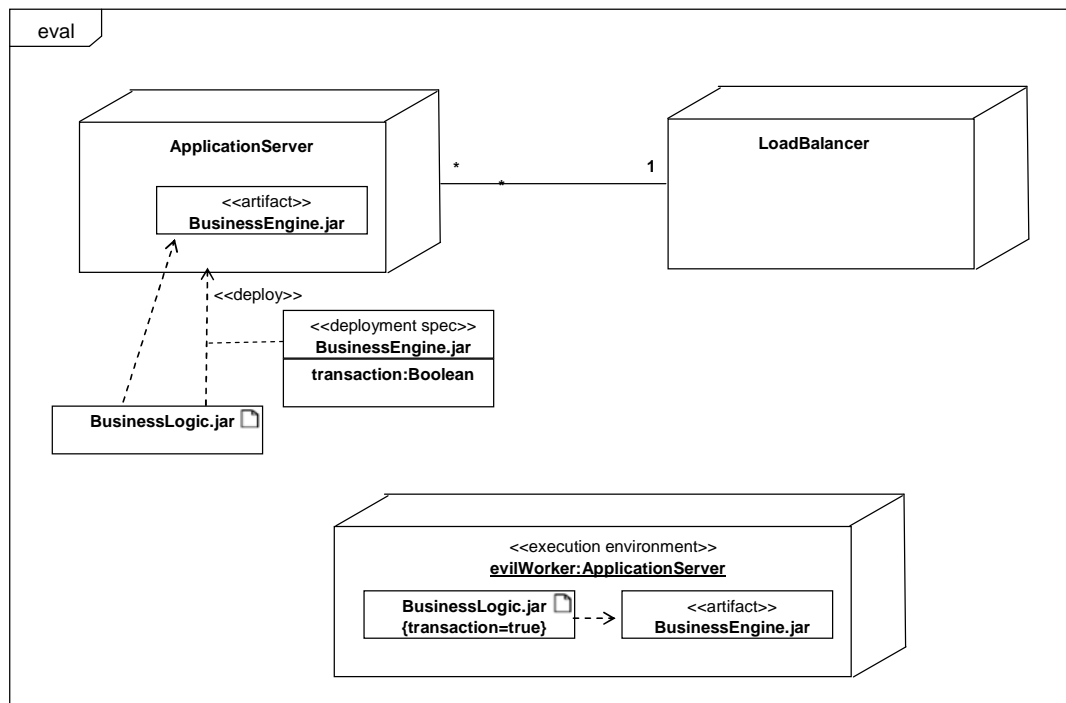


Figure 84: Reference Deployment Diagram

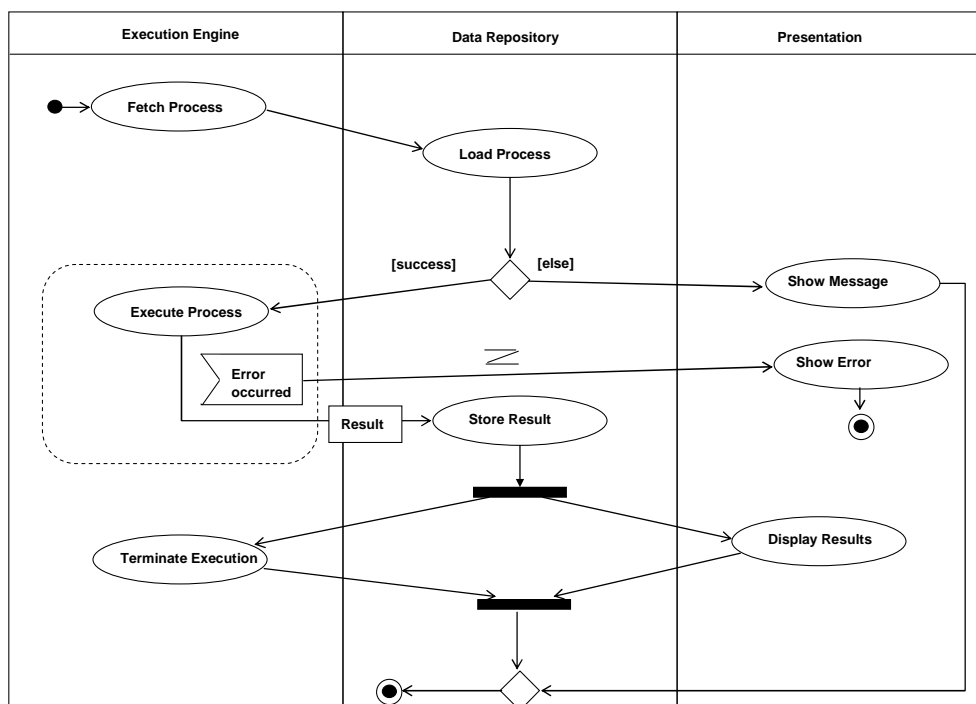


Figure 85: Reference Activity Diagram

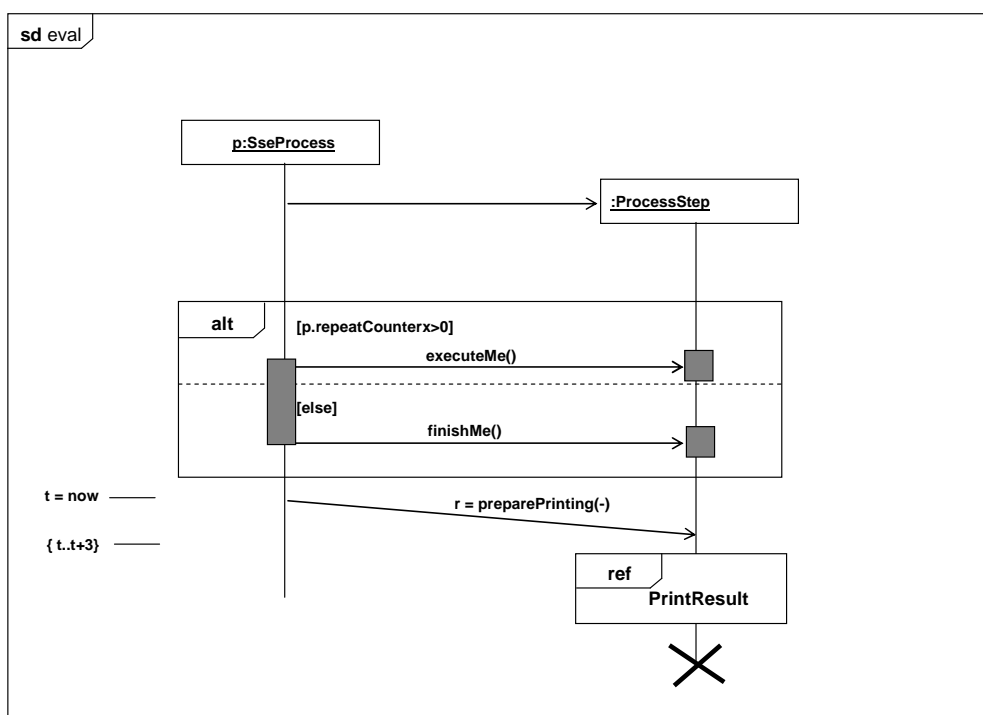


Figure 86: Reference Sequence Diagram

A tool may require to create the operation “preparePrinting” and a second sequence diagram “PrintData” to refer to.

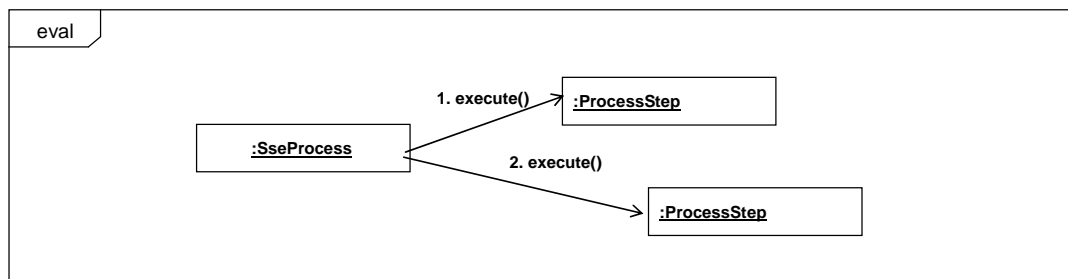


Figure 87: Reference Communication Diagram

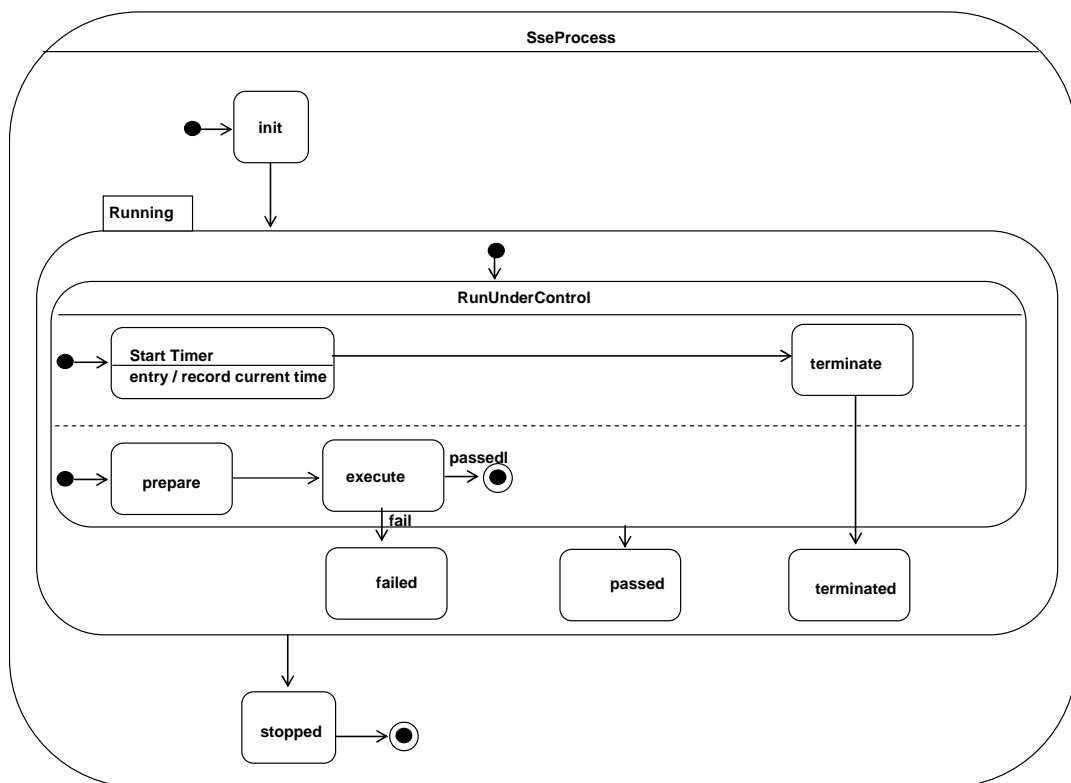


Figure 88: Reference State Machine Diagram
(to be **attached** to the class `SseProcess`)

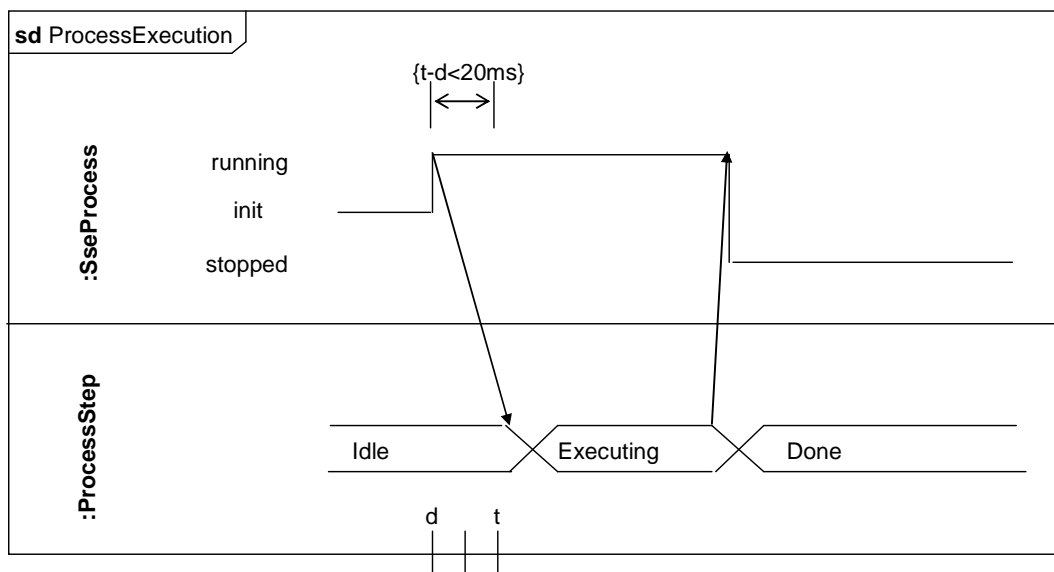


Figure 89: Reference Timing Diagram

The substates in SseProcess should be known from Figure 88 while the substates in ProcessStep should be added as new states.

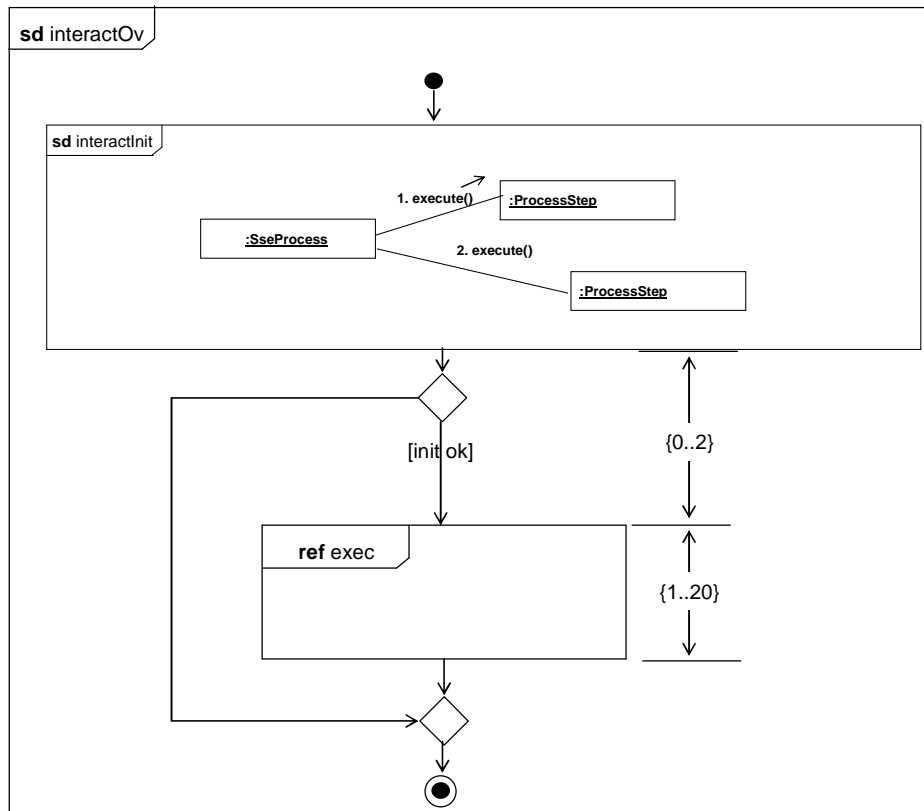


Figure 90: Reference Interaction Overview Diagram

This diagram refers to Figure 87.

References

- [1] Architecture and Design: Unified Modeling Language (UML), 2003. Online available at: http://www.cetus-links.org/oo_uml.html#oo_uml_utilities_tools.
- [2] Object-Oriented Analysis & Design: Tools, 2003. Online available at: http://www.cetus-links.org/oo_ooa_ood_tools.html.
- [3] Objects by Design, UML Products by Company, 2005. Online available at: http://www.objectsbydesign.com/tools/umltools_byCompany.html.
- [4] UML-Werkzeuge, 2008. Online available at: <http://wwwswt.informatik.uni-rostock.de/deutsch/Projekte/RoseTutorial/node101.html>.
- [5] Wikipedia, List of UML tools, 2008. Online available at: http://en.wikipedia.org/wiki/List_of_UML_tools.
- [6] BRÖCKER, C., FERRANDINA, F., OESTEREICH, B., VERSTEEGEN, G., AND WEILKIENS, T. *iX-Studie "Bessere Software!" Konfigurations- und Änderungsmanagement sowie UML*. Heise Verlag, 2005. (in German).
- [7] BUNYAKIATI, P., FINKELSTEIN, A., SKENE, J., AND CHAPMAN, C. Using JULE to Generate a Compliance Test Suite for the UML Standard. In *ICSE '08: Proceedings of the 30th international conference on Software engineering* (New York, NY, USA, 2008), ACM, pp. 827–830.
- [8] DASSO, A., FUNES, A., PERALTA, M., AND SALGADO, C. Metrics Development for UML Tools evaluation. In *Proceedings of WICC 2005* (2005), pp. 286 – 290.
- [9] DOBING, B., AND PARSONS, J. How UML is used. *Communications of the ACM* 49, 5 (2006), 109–113.
- [10] EGYED, A. Automated Abstraction of Class Diagrams. *ACM Transactions on Software Engineering and Methodology* 11, 4 (2002), 449—491.
- [11] EICHELBERGER, H. Evaluation-Report on the Layout Facilities of UML Tools. Technical Report 298, Institut für Informatik, University of Würzburg, July 2002.
- [12] EICHELBERGER, H., ELDOGAN, Y., AND SCHMID, K. A Comprehensive Analysis of UML Tools, their Capabilities and their Compliance. Technical report, Institut für Informatik, University of Hildesheim, 2009. available on request only.
- [13] EICHELBERGER, H., AND VON GUDENBERG, J. W. UML Class Diagrams - State of the Art in Layout Techniques. In *Proceedings of Vissoft 2003, International Workshop on Visualizing Software for Understanding and Analysis* (2003), pp. 30–34.
- [14] FUNES, A., DASSO, A., SALGADO, C., AND PERALTA, M. UML Tool Evaluation Requirements. In *Proceedings of ASIS 2005* (2005).
- [15] HESSE, W., AND MAYR, H. C. Modellierung in der Softwaretechnik: eine Bestandsaufnahme. *Informatik-Spektrum* 31, 5 (October 2008), 377–393. (in German).
- [16] JECKLE, M. UML Tools (CASE & Drawing), 2004. Online available at: <http://www.jeckle.de/umltools.htm>.
- [17] KIRCHNER, L., AND FRANK, U. Evaluierung von UML-Modellierungswerkzeugen. *OBJEKTspektrum*, 1 (2004). (in German).

-
- [18] KIRCHNER, L., AND JUNG, J. Ein Bezugsrahmen zur Evaluierung von UML-Modellierungswerkzeugen. Tech. Rep. 26, Institut für Wirtschaftsinformatik, Universität Koblenz-Landau, 2001. (in German).
- [19] KOBRYN, C. Will UML 2.0 Be Agile or Awkward? *Communications of the ACM* 45, 1 (2002), 107—110.
- [20] LESTER, N., AND WILKIE, F. Evaluating UML Tool Support for Effective Coordination and Communication across Geographically Disparate Sites. In *Proceedings of the 2004 International Workshop on Software Technology and Engineering Practice (STEP'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 57–64.
- [21] LUNDELL, B., LINGS, B., PERSSON, A., AND MATSSON, A. UML Model Interchange in Heterogeneous Tool Environments: An Analysis of Adoptions of XMI 2. In *Model Driven Engineering Languages and Systems, 9th International Conference, MoDELS 2006, Genova, Italy, October 1-6, 2006, Proceedings* (2006), O. Nierstrasz, J. Whittle, D. Harel, and G. Reggio, Eds., no. 4199 in Lecture Notes in Computer Science, Springer-Verlag Inc., pp. 619—630.
- [22] MELLOR, S., RIOUX, L., AND BÉZIVIN, J., Eds. *MDA-components: Is there a Need and a Market?* (2003). Proceedings of the common meeting to the OMG MDA Working Group, the OMG MDA User Group and the OFTA Software Model Engineering Group, Online available at: <http://www.sciences.univ-nantes.fr/lina/atl/www/papers/OMG.OFTA.pdf>.
- [23] MELLOR, S. J., WILKIE, I., FONTAINE, J. S., AND STATEZNI, D. E. An Approach to Adding Precision and Clarity to UML, 1999. Online available at: <http://www.omg.org/docs/ad/99-12-25.pdf>.
- [24] (MIWG), M. I. W. G. Model interchange wiki, 2011. online available at http://www.omgwiki.org/model-interchange/doku.php#test_suite.
- [25] MWALUSEKE, G. W., AND BOWEN, J. P. Evaluation of some of the current UML tools. Tech. rep., Apr. 2002. <http://myweb.lsbu.ac.uk/mwalusgw/publications.html>.
- [26] NIST. Nist validator, 2011. online available at <http://syseng.nist.gov/se-interop/sysml/validator>.
- [27] OMG. Unified Modeling Language: Infrastructure version 1.3. Specification v1.3 00-03-01, Object Management Group, Mar. 2000. Available online at: <http://www.omg.org/docs/formal/00-03-01.pdf>.
- [28] OMG. Unified Modeling Language: Infrastructure version 1.4. Specification v1.4 01-09-67, Object Management Group, Sept. 2001. Available online at: <http://www.omg.org/docs/formal/01-09-67.pdf>.
- [29] OMG. Diagram interchange, v1.0, Apr. 2006. Online available at: <http://www.omg.org/docs/formal/06-04-04.pdf>.
- [30] OMG. Object Constraint Language. Specification v2.00 2006-05-01, Object Management Group, May 2006. Available online at: <http://www.omg.org/docs/formal/06-05-01.pdf>.
- [31] OMG. MOF 2.0/XMI Mapping, v2.1.1, Dec. 2007. Online available at: <http://www.omg.org/spec/XMI/2.1.1/>.
- [32] OMG. Unified Modeling Language: Infrastructure version 2.1.1. Specification v2.11 2007-02-06, Object Management Group, Feb. 2007. Available online at: <http://www.omg.org/docs/formal/07-02-06.pdf>.
-

- [33] OMG. Unified Modeling Language: Superstructure version 2.1.1. Specification v2.1.1 2007-02-05, Object Management Group, Feb. 2007. Available online at: <http://www.omg.org/cgi-bin/doc?formal/2007-02-05>.
- [34] OMG. OMG UML Vendor Directory Listing, 2008. Online available at: <http://uml-directory.omg.org/vendor/list.htm>.
- [35] OOSE. UML Tools, 2008. Online available at: <http://www.oose.de/umltools.htm>.
- [36] PERSSON, A., GUSTAVSSON, H., LINGS, B., LUNDELL, B., MATTSSON, A., AND ÄRLIG, U. OSS tools in a heterogeneous environment for embedded systems modelling: an analysis of adoptions of XML. In *Open Source Application Spaces: Fifth Workshop on Open Source Software Engineering (5-WOSSE) May 17, 2005, St Louis, MO, USA* (2005), pp. 39 – 42.
- [37] PERSSON, A., GUSTAVSSON, H., LINGS, B., LUNDELL, B., MATTSSON, A., AND ÄRLIG, U. Adopting Open Source Development Tools in a Commercial Production Environment: Are We Locked-In? In *Open Source Application Spaces: Fifth Workshop on Open Source Software Engineering (5-WOSSE) May 17, 2005, St Louis, MO, USA*. 2006.
- [38] SUN, D., AND WONG, K. On Evaluating the Layout of UML Class Diagrams for Program Comprehension. In *Proceedings of the 13th International Workshop on Program Comprehension (IWPC'05)* (2005), IEEE.
- [39] WEILKIENS, T. Die wogen glätten sich. *OBJEKTSpektrum*, 4 (2007), 36–37. (in German).