

Stefan Krause

Vergleich von CASE-Werkzeugen zur
Modellierung von Softwaresystemen mittels
UML für KMU

Diplomarbeit

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Copyright © 2003 Diplomica Verlag GmbH
ISBN: 9783832476373

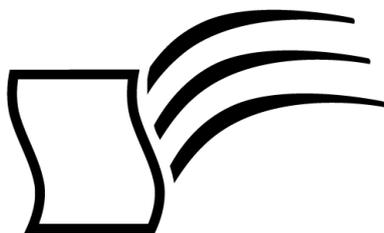
Stefan Krause

Vergleich von CASE-Werkzeugen zur Modellierung von Softwaresystemen mittels UML für KMU

Stefan Krause

Vergleich von CASE- Werkzeugen zur Modellierung von Softwaresystemen mittels UML für KMU

Diplomarbeit
Friedrich-Schiller-Universität Jena
Fachbereich Mathematik und Informatik
Abgabe Dezember 2003



Diplom.de

Diplomica GmbH _____
Hermannstal 119k _____
22119 Hamburg _____

Fon: 040 / 655 99 20 _____
Fax: 040 / 655 99 222 _____

agentur@diplom.de _____
www.diplom.de _____

ID 7637

Krause, Stefan: Vergleich von CASE-Werkzeugen zur Modellierung von Softwaresystemen mittels UML für KMU

Hamburg: Diplomica GmbH, 2004

Zugl.: Friedrich-Schiller-Universität Jena, Universität, Diplomarbeit, 2003

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden, und die Diplomarbeiten Agentur, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Diplomica GmbH

<http://www.diplom.de>, Hamburg 2004

Printed in Germany

Inhaltsverzeichnis

1	Einleitung	5
2	Einführung in UML 1.4	6
2.1	Anwendungsfalldiagramm	6
2.2	Aktivitätsdiagramm	7
2.3	Klassendiagramm	10
2.4	Sequenzdiagramm	23
2.5	Kollaborationsdiagramm	24
2.6	Zustandsdiagramm	26
3	CASE-Werkzeuge	31
3.1	Definition und Ziele	31
3.2	Allgemeine Anforderungen an CASE-Werkzeuge	32
3.3	Vorgehensweise bei der Auswahl eines CASE-Werkzeugs	33
3.4	Modifizierte Vorgehensweise	33
4	Kriterienkatalog	35
4.1	Ausschlusskriterien	35
4.2	Allgemeine Kriterien für CASE-Werkzeuge	36
4.3	Spezielle Kriterien für CASE-Werkzeuge	39
4.4	Spezielle Kriterien bzgl. der Modellierung mit der UML	41
5	Referenzmodell	44
5.1	Anwendungsfälle	44
5.2	Aktivitätsdiagramme	44
5.3	Beschreibung des Klassenmodells	47
5.4	Beschreibung des Änderungskatalogs	50
5.4.1	Änderungen am Klassendiagramm (Abb. 5.6)	50
5.4.2	Änderungen am Sequenzdiagramm (Abb. 5.7)	50
5.4.3	Änderungen am Kollaborationsdiagramm (Abb. 5.8)	51
5.4.4	Änderungen am Zustandsdiagramm (Abb. 5.9)	51
6	Evaluierung der Werkzeuge	52
6.1	Vorauswahl	52
6.2	Bewertungsnotation	52
6.3	INNOVATOR 8.0	53
6.3.1	Einführung und Beschreibung	53
6.3.2	Bewertung gegenüber dem Kriterienkatalog	54
6.3.3	Bewertung gegenüber dem Änderungskatalog	65
6.4	Together Control Center 6.1	68
6.4.1	Einführung und Beschreibung	68

6.4.2	Bewertung gegenüber dem Kriterienkatalog	69
6.4.3	Bewertung gegenüber dem Änderungskatalog	78
6.5	Rational Rose Enterprise Edition	83
6.5.1	Einführung und Beschreibung	83
6.5.2	Bewertung gegenüber dem Kriterienkatalog	83
6.5.3	Bewertung gegenüber dem Änderungskatalog	92
6.6	Poseidon for UML Professional Edition 2.0.2	97
6.6.1	Einführung und Beschreibung	97
6.6.2	Bewertung gegenüber dem Kriterienkatalog	97
6.6.3	Bewertung gegenüber dem Änderungskatalog	105
7	Auswertung	110
7.1	Ausschlusskriterien	110
7.2	Allgemeine Kriterien für CASE-Werkzeuge	110
7.3	Spezielle Kriterien für CASE-Werkzeuge	111
7.4	Spezielle Kriterien bzgl. der Modellierung mit der UML	112
7.5	Modellierung des Referenzmodells	112
7.6	Zusammenfassung	112
8	Zusammenfassung und Ausblick	113
	Literaturverzeichnis	114
	Abbildungsverzeichnis	114
	Anhang	117

1 Einleitung

Die stetig zunehmende Komplexität von Softwareprojekten hat zur Entwicklung von Werkzeugen geführt, die den Software-Entwicklungsprozess unterstützen. Sogenannte CASE¹-Werkzeuge helfen dabei, die Arbeit in sämtlichen Phasen des Entwicklungsprozesses zu erleichtern. Als CASE-Werkzeuge bezeichnet man alle Software-Produkte, die Funktionen zur Entwicklung von Software bereitstellen. Durch sie können Routineabläufe automatisiert und das Software-Management erleichtert werden. Außerdem tragen CASE-Werkzeuge dazu bei, die Software-Produktivität zu erhöhen und die Software-Qualität zu verbessern.

Um aus der Vielzahl der am Markt verfügbaren CASE-Werkzeuge dasjenige auswählen zu können, das den gestellten Anforderungen am nächsten kommt, ist eine genaue Betrachtung der Leistungsmerkmale eines Werkzeuges erforderlich. Dabei können Vergleichsstudien von CASE-Werkzeugen den Entscheidungsprozess vereinfachen und beschleunigen. Aus Aufwandsgründen können Vergleichsstudien meistens nur eine geringe Anzahl aus der Vielzahl von Werkzeugen einbeziehen.

In der vorliegenden Arbeit werden vier CASE-Werkzeuge unter dem Gesichtspunkt des Einsatzes in Klein- und mittelständischen Unternehmen (KMU) verglichen. Besonderer Wert wird außerdem auf die Modellierungsmöglichkeiten mit der UML gelegt. Dazu wird im zweiten Kapitel zunächst eine Einführung in UML 1.4 gegeben. In Kapitel drei werden Anforderungen beschrieben, die an CASE-Werkzeuge gestellt werden und es wird eine Vorgehensweise zur Bewertung von CASE-Werkzeugen entwickelt. Die Bewertung wird anhand eines Kriterienkatalogs durchgeführt, der in Kapitel vier aufgeführt ist. Im fünften Kapitel erfolgt die Beschreibung eines Referenzmodells und eines zugehörigen Kataloges von Änderungen, welche mit jedem der untersuchten Werkzeuge in Kapitel sechs modelliert werden. Hier erfolgt außerdem die Evaluierung der Werkzeuge anhand des in Kapitel vier aufgestellten Kriterienkatalogs. Schließlich werden im siebten Kapitel die Bewertungsergebnisse gegenübergestellt. Kapitel acht fasst die Arbeit zusammen und gibt einen Ausblick.

¹Computer Aided Software Engineering

2 Einführung in UML 1.4

Die UML ist eine Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme. Sie deckt ein breites Spektrum von Anwendungsgebieten ab und eignet sich u.a. für konkurrierende, verteilte, zeitkritische und sozial eingebettete Systeme.

Im Folgenden werden die wichtigsten Modellierungselemente der UML vorgestellt. Dabei wird von der Version 1.4 ausgegangen. Eine ausführliche Beschreibung der UML findet man in [Oestereich01], [Stevens00], [Rumbaugh99] sowie unter [OMG]. Für Begriffe aus der Objektorientierung sei auf die einschlägige Literatur verwiesen.

2.1 Anwendungsfalldiagramm

In einem Anwendungsfalldiagramm werden die Beziehungen zwischen einer Menge von Anwendungsfällen und den daran beteiligten Akteuren gezeigt. Ein Anwendungsfalldiagramm bildet somit einen Kontext und eine Gliederung für die Beschreibung von Geschäftsvorfällen.

Ein Beispiel für einen Geschäftsvorfall ist die schriftliche Schadensmeldung eines Hausratversicherten. Der gesamte Ablauf für die Verarbeitung dieses Ereignisses wird durch einen Geschäftsprozess (z.B. 'Hausratschaden melden') beschrieben. Dabei ist zu beachten, dass auch Aktivitäten dargestellt werden können, die nicht durch die zu entwickelnde Anwendung unterstützt werden.

Anwendungsfalldiagramme sind in erster Linie ein Hilfsmittel zur Anforderungsermittlung und kein Designansatz oder eine Beschreibung des internen Verhaltens eines zu entwickelnden Systems. Primär werden Anwendungsfälle dazu verwendet die Kommunikation mit Anwendern und Auftraggebern zu unterstützen. Sie beschreiben das externe Systemverhalten, also die Leistungsmerkmale eines Systems. Darüber wie dieses Systemverhalten realisiert wird, treffen Anwendungsfälle keine Aussage.

Abbildung 2.1 zeigt die verschiedenen Notationselemente eines Anwendungsfalldiagramms.

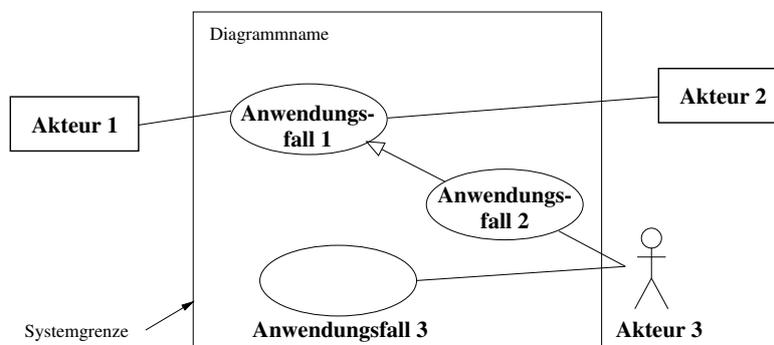


Abbildung 2.1: Notationselemente des Anwendungsfalldiagramms.

2.2 Aktivitätsdiagramm

Aktivitätsdiagramme beschreiben die Ablaufmöglichkeiten eines Systems mit Hilfe von Aktivitäten. Eine Aktivität ist ein Zustand mit einer internen Aktion und einer oder mehreren ausgehenden Transitionen (Zustandsübergängen), die automatisch dem Abschluss der internen Aktion folgen. Sie stellt somit einen einzelnen Schritt in einem Verarbeitungsablauf dar. Im Falle von mehreren ausgehenden Transitionen müssen diese durch Bedingungen unterscheidbar sein.

Aktivitätsdiagramme unterstützen die Beschreibung nebenläufiger Aktivitäten. Dabei ist die Reihenfolge von parallel laufenden Aktivitätspfaden irrelevant, d.h. sie können nacheinander, gleichzeitig oder abwechselnd laufen.

Aktivitäten und Aktivitätsdiagramme sind entweder einer Klasse, einer Operation oder einem Anwendungsfall zugeordnet und beschreiben die internen Ablaufmöglichkeiten dieser Modellelemente.

In einem Aktivitätsdiagramm kann man Aktivitäten hierarchisch schachteln, so dass eine einzelne Aktivität aus mehreren Detailaktivitäten besteht. Dabei müssen die eingehenden und ausgehenden Transitionen der zusammengesetzten Aktivität und des Detailmodells übereinstimmen.

Aktivitätsdiagramme lassen sich in Verantwortungsbereiche unterteilen. Dadurch können die Aktivitäten anderen Elementen oder Strukturen zugeordnet werden. Man kann damit z.B. ausdrücken, zu welcher Klasse oder Komponente die Aktivitäten gehören.

Multiple Transitionen bzw. Auslöser bieten eine weitere Möglichkeit parallele Abläufe darzustellen.

Aufgrund der genannten Darstellungselemente sind Aktivitätsdiagramme geeignet die unterschiedlichsten Arten von Abläufen darzustellen. Fachliche Zusammenhänge und Abhängigkeiten, die sich hinter einem Anwendungsfall verbergen, lassen sich mit ihnen vollständig beschreiben. Ein Aktivitätsdiagramm kann außerdem Sachverhalte aus mehreren Anwendungsfällen beschreiben sowie Geschäftsregeln und Entscheidungslogik abbilden.

Die Darstellung einer Aktivität (Abb. 2.2) erfolgt durch ein benanntes Rechteck mit gerader Ober- und Unterseite und runden Seitenlinien. Es enthält eine Aktivitätsbeschreibung, die aus einem Namen, einer frei formulierten Beschreibung, Pseudocode oder Programmiersprachencode bestehen kann. Transitionen werden wie in einem Zustandsdiagramm als Pfeile dargestellt. Eine Ereignisbeschriftung entfällt, da die Transitionen implizit durch den Abschluss der Aktivität ausgelöst werden.



Abbildung 2.2: Aktivitäten und Transition.

Bedingungen und Verzweigungen

Ausgehende Transitionen können mit in eckigen Klammern stehenden Bedingungen (boolsche Ausdrücke) versehen werden. Als Alternative können Verzweigungen, dargestellt durch Rauten (Abb. 2.3), verwendet werden. Eine solche Raute stellt ebenfalls eine Entscheidungsaktivität dar.

Und- und Oder-Synchronisation

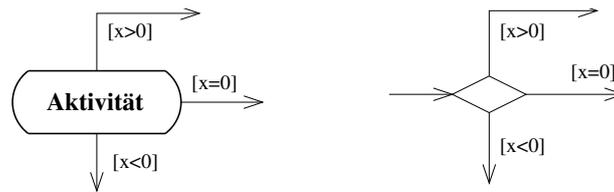


Abbildung 2.3: Entscheidung/Verzweigung.

Transitionen können synchronisiert, zusammengeführt und aufgeteilt werden (Abb. 2.4). Bei einer Zusammenführung von mehreren Transitionen setzt sich der Aktionsfluss fort, sobald eine Transition eingeht. Die Synchronisation setzt dagegen voraus, dass alle eingehenden Transitionen vorliegen.

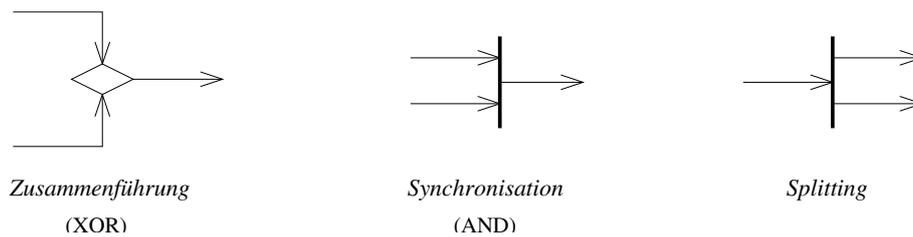


Abbildung 2.4: Synchronisation.

Zusammengesetzte Aktivitäten

Zusammengesetzte Aktivitäten werden in folgender Form dargestellt (Abb. 2.5).

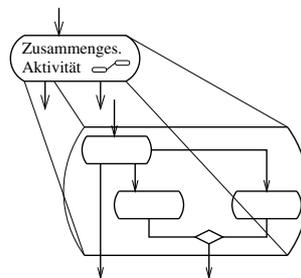


Abbildung 2.5: Zusammengesetzte Aktivität.

Schleifen

Für eine Schleife ist lediglich eine Abbruchbedingung und eine zurückführende Transition notwendig (Abb. 2.6). Innerhalb der Schleife sind beliebig viele Aktivitäten zulässig.

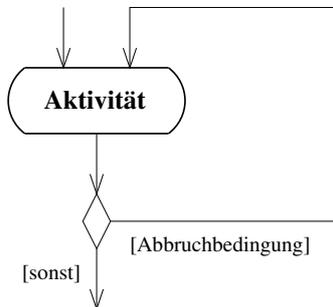


Abbildung 2.6: Wiederholung/Schleife.

Optionale Aktivitäten

Vor einer optionalen Aktivität (Abb. 2.7) wird eine bedingte Verzweigung notiert. Die Zweige werden anschließend wieder zusammengeführt.

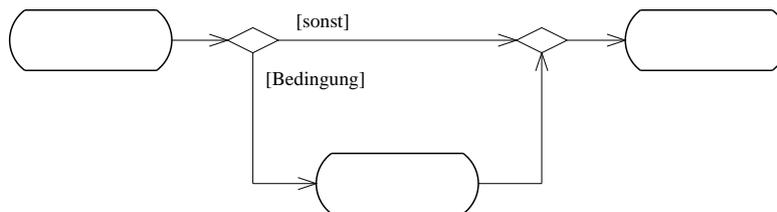


Abbildung 2.7: Optionale Aktivität.

Signale senden und empfangen

Signale können beim Übergang zwischen zwei Aktivitäten an ein Objekt gesendet werden. Ebenso kann der Start einer Aktivität das Eintreffen eines Signals voraussetzen. Abbildung 2.8 zeigt die zugehörigen Notationselemente. Mit ihrer Hilfe lassen sich z.B. nebenläufige Prozesse synchronisieren.