

Georg Herzwurm *

Einsatz wissensbasierter Komponenten in CASE-Werkzeugen - Ergebnisse einer empirischen Untersuchung

Inhaltsverzeichnis

1	Einleitung	644
1.1	Wissensbasiertes CASE	644
1.2	Potential wissensbasierter Systeme im CASE	644
2	Konzept der empirischen Untersuchung	646
3	Existierende wissensbasierte CASE-Systeme	648
3.1	Entwicklerbezogene Merkmale wissensbasierter Systeme im CASE	648
3.2	Aufgabenbezogene Merkmale wissensbasierter Systeme im CASE	649
3.2.1	Unterstützte Phasen im Systemlebenszyklus	650
3.2.2	Unterstützte Aktivitäten im Systemlebenszyklus	651
3.2.3	Unterstützte Aufgabenklassen im Systemlebenszyklus	654
3.3	Systembezogene Merkmale wissensbasierter Systeme im CASE	655
3.4	Zusammenhänge zwischen Merkmalen wissensbasierter Systeme im CASE	659
4	Schlußfolgerungen für wissensbasiertes CASE	662
	Literatur	665

* *Dr. Georg Herzwurm, Lehrstuhl für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln*

1 Einleitung

1.1 Wissensbasiertes CASE

Trotz kontinuierlicher Bemühungen um eine Automation der Software-Entwicklung, die mit dem Aufkommen des Computer Aided Software Engineering (CASE) einen vorläufigen Höhepunkt erreicht hat, ist die Qualität der marktgängigen CASE-Werkzeuge derzeit nicht zufriedenstellend zu beurteilen¹.

Ein zunehmend häufig diskutierter Ansatz zur Verbesserung dieser Situation ist der Einsatz von wissensbasierten Systemen (WBS) - als integrierter Bestandteil von CASE-Werkzeugen oder als eigenständige Systeme -, die den Software-Entwickler auf einer höheren Ebene unterstützen können als dies mit Hilfe konventioneller Technologien möglich ist².

1.2 Potential wissensbasierter Systeme im CASE

Der Grundgedanke des wissensbasierten CASE geht von der These aus, daß sich eine Vielzahl der - trotz des Einsatzes moderner Werkzeuge nach wie vor bestehenden - aktuellen Probleme der Entwicklung konventioneller Software durch erfaßbares und formalisierbares Erfahrungswissen und somit durch den Einsatz wissensbasierter Systeme lösen lassen. [zum Stand der Kenntnis siehe z. B. Herzwurm /Wissensbasiertes CASE/ und Lowry, Duran /Knowledge-based Software Engineering/ 243-322]

Wissensbasierte Systeme beinhalten Potentiale, die sie für einen Einsatz im CASE als besonders geeignet erscheinen lassen³; z. B.:

- Anwendung des Wissens über die Software-Entwicklung zur gleichartigen Behandlung gleichartiger Vorgänge (Normierung bzw. Standardisierung) zwecks Erreichung objektiver, nachprüfbarer, d. h. ent

1 Berkau, Herzwurm /Auswahl PC-gestützter Software-Entwicklungsumgebungen/ 42-55

2 McClure /CASE is software automation/

3 Helsel /Was ist Expertentechnik?/

wicklerunabhängiger und nicht durch externe Einflüsse (z. B. Streß) beeinträchtigt, Ergebnisse.

- Bereitstellung des Wissens über die Software-Entwicklung durch Verfügbarmachung von Expertenwissen über die verschiedenen Methodenschritte und aufzurufenden Werkzeuge sowie durch die Erläuterung und Begründung von Entscheidungen über die Erklärungskomponente zwecks Unterstützung des Entwicklers bei seinen permanenten (Entwurfs-)Entscheidungen.
- Speicherung des Wissens über die Software-Entwicklung zur Verminderung der Konsequenzen des Ausscheidens von Mitarbeitern in laufenden Projekten und zur Vergrößerung der Wissensbasis des Unternehmens.
- Strukturierung des Wissens über die Software-Entwicklung zur Berücksichtigung zahlreicher potentieller Anpassungs-, Erweiterungs- und Fehlermöglichkeiten und somit zur Reduzierung des späteren Wartungsaufwands⁴.
- Vermittlung des Wissens über die Software-Entwicklung durch Lieferung von Expertenwissen an Anfänger zur Verkürzung der Schulungs- bzw. Einarbeitungszeit⁵.
- Vervielfältigung des Wissens über die Software-Entwicklung zur Entlastung hochrangiger Experten von Routinearbeiten und Bereitstellung von Expertise während des gesamten Projekts auch für minderqualifizierte Mitarbeiter⁶.

Die Fähigkeit wissensbasierter Systeme, den Entwickler bei seinen permanenten Entscheidungsproblemen zu unterstützen sowie vorhandenes Methodenwissen erfahrener Experten in CASE-Produkte zu integrieren und zu ergänzen, stellt somit ein bedeutendes Potential zur Lösung der aktuellen Probleme der Software-Entwicklung dar.

⁴ Harmon, King /Expertensysteme in der Praxis/

⁵ Schmitz /Expertensysteme/ 611-626

⁶ Regier /Expertensysteme für den betriebswirtschaftlichen Einsatz/ 16-25

2 Konzept der empirischen Untersuchung

Der nachfolgenden Untersuchung liegt die Ausgangsthese zugrunde, daß in Wissenschaft und Praxis bereits zahlreiche Versuche unternommen wurden, wissensbasierte CASE-Werkzeuge zu entwickeln. Untersuchungsziel ist die Erforschung des "state of the art" dieser Bemühungen und insbesondere die Klärung der Frage, ob sich wissensbasierte CASE-Werkzeuge zur Entwicklung konventioneller Software noch im Forschungsstadium befinden oder bereits Tendenzen zur betrieblichen Einsatzreife zeigen. Die Vorstellung erfolgreich entwickelter wissensbasierter Systeme im CASE kann zugleich als Anregung für neue Projekte in diesem Forschungsbereich fungieren und belegt die wissenschaftliche und praktische Relevanz des Themas wissensbasiertes CASE.

Aus Sicht der Sozialwissenschaften handelt es sich bei der nachfolgenden Materialsammlung um eine empirische, allerdings nicht prüfende, sondern beschreibende Untersuchung auf der Basis eines heuristischen Bezugsrahmens. Primäre Zielsetzung beschreibender Untersuchungen ist die Anregung neuer Hypothesen (Hypothesenerkundung) oder die Erforschung der Ausprägungen verschiedener Merkmale in einer Grundgesamtheit (Populationsbeschreibung)⁷.

Zu diesem Zweck wurden insgesamt 298 Organisationen in Deutschland, Österreich und der Schweiz in Form eines standardisierten Fragebogens nach von ihnen entwickelten wissensbasierten Systemen im CASE befragt. Ziel war es, möglichst alle Anbieter bzw. Hersteller von CASE-Produkten bzw. von Werkzeugen zur Entwicklung wissensbasierter Systeme im Untersuchungsraum zu befragen. Außerdem wurden Beratungsunternehmen, deren Tätigkeitsbeschreibung [als Grundlage fungierte v. a. Nomina Information Services (Hrsg.) /ISIS Firmen Report 1991/] den Bereich CASE und/oder wissensbasierte Systeme explizit als einen Schwerpunkt nennt, in die Erhebung einbezogen. Schließlich erfolgte die Auswertung von Studien- und Forschungsführern [z. B. Heinrich, Kurbel /Studien- und Forschungsführer Wirtschaftsinformatik /], um möglichst alle Forschungseinrichtungen und Hochschulen, die sich mit wissensbasierter Technologie

⁷ Bortz /Lehrbuch der empirischen Forschung für Sozialwissenschaftler/

und/oder CASE beschäftigen, zu involvieren. Die Zusammensetzung und das Antwortverhalten der befragten Organisationen im Rahmen der Umfrage zeigt die nachfolgende Abbildung.

Organisation	Versand		Rücklauf		Rücklauf- quote
	absolut	relativ	absolut	relativ	
Beratungsunternehmen	22	7,38%	14	7,82%	63,64%
Forschungseinrichtung	124	41,61%	78	43,58%	62,90%
Anbieter/Hersteller CASE-Branche	82	27,52%	57	31,84%	69,51%
Anbieter/Hersteller WBS-Branche	53	17,79%	20	11,17%	37,74%
Sonstige Organisation	17	5,70%	10	5,59%	58,82%
Summe	298	100,00%	179	100,00%	60,07%

Abb. 1: Quantitatives Antwortverhalten

Durch eine erfreulich hohe Rücklaufquote von über 60% kann davon ausgegangen werden, daß die Ergebnisse einen guten Überblick über den derzeitigen Stand wissensbasierter CASE-Systeme in Deutschland liefern. Dennoch werden neben dem Ergebnis der Befragung im folgenden auch die Resultate systematischer Recherchen anderer Datenquellen (Literatur [z. B. Mertens /Betriebliche Expertensystem-AnwendungenMertens/], Gespräche mit Fachleuten etc.) vorgestellt⁸, die in den folgenden Übersichten unter der Rubrik "Rest" subsumiert werden.

Als wissensbasierte Systeme im CASE werden hierbei alle Informationssysteme, die Wissen über die Software-Entwicklung in einer Wissensbasis speichern und über eine getrennte Inferenzkomponente verfügen, um aus dieser Wissensbasis problemspezifisches, neues Wissen abzuleiten, verstanden⁹.¹⁰

⁸ Die nach Datenquellen getrennt vorgenommene Auswertung führte nur in Ausnahmefällen zu unterschiedlichen Ergebnissen, so daß die Resultate gemeinsam vorgestellt werden. Sofern signifikante Differenzen zwischen den Datenquellen auftreten, erfolgt ein gesonderter Hinweis.

⁹ Schmitz /Expertensysteme/ 611-626

¹⁰ Unter einem konventionellen ADV-System werden dagegen v. a. für gut strukturierte Probleme in Sprachen der 3. oder 4. Generation realisierte daten- und prozedurorientierte Systeme zusammengefaßt.

3 Existierende wissensbasierte CASE-Systeme

Die wichtigsten charakteristischen Eigenschaften existierender wissensbasierter CASE-Systeme werden im folgenden nach entwicklerbezogenen, aufgabenbezogenen sowie systembezogenen Merkmalen differenziert, wobei sich lediglich auf die wesentlichen, den höchsten Erkenntnisgewinn versprechenden Merkmalen beschränkt wird.

3.1 Entwicklerbezogene Merkmale wissensbasierter Systeme im CASE

Von den insgesamt 298¹¹ entdeckten wissensbasierten CASE-Systemen wurden 95 (31,88%) in Deutschland entwickelt, während die U.S.A. mit 132 (44,30%) Systemen als (zumindest quantitativ) bedeutendster Entwickler von wissensbasierten Systemen im CASE bezeichnet werden kann.¹²

Bei den Entwicklern dominieren Forschungseinrichtungen, was einen Indikator für den derzeitigen Forschungscharakter dieser Domäne darstellt (Abb. 2). Aber auch die Anbieter stellen eine bedeutende Entwicklergruppe dar, woraus zu schließen ist, daß der Einsatz von wissensbasierten Systemen zur konventionellen Software-Entwicklung auch von der Industrie zunehmend als wettbewerbswirksamer Faktor anerkannt wird.¹³

¹¹ Die zahlenmäßige Übereinstimmung mit der Anzahl der Befragten ist zufällig.

¹² Infolge der Heterogenität der Quellen sowie der ausschließlichen Verwendung deutsch- und englischsprachiger Literatur sind die länderspezifischen Aussagen besonders kritisch zu bewerten.

¹³ Die Vermutung, daß die Existenz wissensbasierter Komponenten in CASE-Werkzeugen als zusätzliches Qualitätsmerkmal gesehen wird, zeigt sich auch in der überdurchschnittlichen Präsenz der von Anbietern entwickelten Systeme in der Umfrage gegenüber den aus anderen Quellen recherchierten Ergebnissen. So hat der Verfasser während der Befragung den subjektiven Eindruck gewonnen, daß auch in unberechtigten Fällen eine Wissensbasierung der Produkte von den Anbietern angegeben wurde.

Entwickler	Gesamt		Umfrage		Rest	
	absolut	relativ	absolut	relativ	absolut	relativ
Anbieter	100	33,56%	44	59,46%	56	25,00%
Anwender	10	3,36%	2	2,70%	8	3,57%
Forschungseinrichtung	113	37,92%	20	27,03%	93	41,52%
Gemeinschaftsarbeit	28	9,40%	8	10,81%	20	8,93%
Unbekannt	47	15,77%	0	0,00%	47	20,98%
Summe	298	100,00%	74	100,00%	224	100,00%

Abb. 2: Häufigkeitsverteilung des Merkmals Art des Entwicklers wissensbasierter Systeme im CASE

3.2 Aufgabenbezogene Merkmale wissensbasierter Systeme im CASE

Durch die Analyse existierender wissensbasierter CASE-Systeme können Tendenzaussagen bezüglich einer Eignung bestimmter Phasen oder Aktivitäten für eine Unterstützung durch wissensbasierte Systeme gewonnen werden.

Phase	Gesamt		Umfrage		Rest	
	absolut	relativ	absolut	relativ	absolut	relativ
Analyse	93	14,37%	34	13,71%	59	14,79%
Entwurf	117	18,08%	45	18,15%	72	18,05%
Realisierung	137	21,17%	44	17,74%	93	23,31%
Erprobung und Konsolidierung	42	6,49%	24	9,68%	18	4,51%
Pflege und Wartung	77	11,90%	34	13,71%	43	10,78%
Projektmanagement	48	7,42%	17	6,85%	31	7,77%
Software-Qualitätssicherung	60	9,27%	21	8,47%	39	9,77%
Dokumentation	42	6,49%	28	11,29%	14	3,51%
Unbekannt	31	4,79%	1	0,40%	30	7,52%
Summe	647	100,00%	248	100,00%	399	100,00%

Abb. 3: Häufigkeitsverteilung des Merkmals von wissensbasierten Systemen im CASE unterstützte Phasen im Systemlebenszyklus

Die These, daß infolge der im Verlaufe des Systemlebenszyklus zunehmende Strukturierungsgrad des zunächst diffusen Problembereichs Software-Entwicklung zu einer besonderen Eignung wissensbasierter Systeme für die frü-

hen Phasen Analyse und Entwurf führt, wird hierbei durch die Untersuchungsergebnisse bestätigt (Abb. 3).

3.2.1 Unterstützte Phasen im Systemlebenszyklus

Wissensbasierte Systeme in objektbezogenen Phasen

Die Ermittlung von Benutzeranforderungen und deren anschließende Transformation in ein DV-Konzept ist ein sehr informaler¹⁴ sowie wissensintensiver Prozeß¹⁵ und erfordert ein Höchstmaß an Erfahrung und Geschicklichkeit routinierter Experten, die jedoch nur selten verfügbar sind¹⁶. Infolgedessen stellt die Erhebung und Spezifikation von Benutzeranforderungen innerhalb der Entwurfs- und Analysephase eine für den Einsatz wissensbasierter Systeme offensichtlich besonders geeignete Aufgabenstellung dar. Einen weiteren Schwerpunkt stellt die Bereitstellung von Expertise für den logischen und physischen Datenbankentwurf dar. Die Gestaltung von Diagrammen, die Erstellung ausführbarer Spezifikationen sowie die Anleitung beim Erstellen bzw. beim Überprüfen von Prozeß- und Datenmodellierung sind häufig unterstützte Tätigkeiten in dieser Phase. Ferner existieren erste wissensbasierte Ansätze zur Programmsynthese aus einer formalen Spezifikation einerseits sowie zur Respezifizierung und Wiederverwendung der Entwürfe andererseits.

Die Realisierungsphase ist durch den Einsatz wissensbasierter Systeme gekennzeichnet, die den Programmierer bei der Eingabe des Quellcodes in einen Editor unterstützen, als Debugger fungieren oder die Qualität des erzeugten Codes hinsichtlich bestimmter Kriterien, wie z. B. Performance, prüfen.

Die Unterstützung der Konsolidierungs- und Erprobungsphase erfolgt zu- meist nur seitens größerer wissensbasierter Systeme, die auch Hilfestellung während phasenübergreifender Aktivitäten leisten.

Bei der Pflege und Wartung von Informationssystemen bildet die Analyse

¹⁴ Puncello /ASPIS/ 58-65

¹⁵ Humphrey /Improving the Software Development/ 28-30

¹⁶ Davis /Software Requierements Analysis/

mit dem Ziel des Programmverstehens einen Einsatzschwerpunkt für wissensbasierte Systeme.

Wissensbasierte Systeme während phasenübergreifender Maßnahmen

Die Unterstützung des Projektmanagements durch wissensbasierte Systeme umfaßt sowohl die Planung, Kontrolle und Steuerung von Ressourcen als auch die Beratung des Projektmanagers beim Auftreten bestimmter Probleme während eines Projekts.

Als verbreitete Einsatzgebiete wissensbasierter Systeme im Rahmen der SoftwareQualitätssicherung sind die Validierung von Software auf einer höheren Ebene als bei konventionellen CASE-Werkzeugen und die Beratung hinsichtlich der Anwendung von Testmethoden zu nennen.

Spezielle wissensbasierte Dokumentationssysteme bilden die Ausnahme innerhalb der Materialsammlung. Ähnlich wie bei Systemen der Erprobungs- und Konsolidierungsphase wird die Dokumentation vielfach nur von großen wissensbasierten Systemen zumeist indirekt, z. B. bei der Erstellung von Grafiken oder im Rahmen der Informationsverwaltung, unterstützt.

3.2.2 Unterstützte Aktivitäten im Systemlebenszyklus

Innerhalb des Aktivitätentyps Anforderungsanalyse dominieren dialogorientierte wissensbasierte Systeme zur Unterstützung der Entwicklung ausführbarer Spezifikationen.

Die Schätzung des von einem Software-Projekt verursachten Aufwandes wird von marktgängigen CASE-Werkzeugen nur unzureichend unterstützt und erfolgt in der Praxis häufig ohne Einsatz formaler Methoden in reinem Analogieschluß¹⁷. Deshalb wurden auch für den Aktivitätentyp Aufwandsschätzung unterstützende wissensbasierte Systeme entwickelt, die häufig auf einer Daten früherer Projekte beinhaltenden Erfahrungsdaten

¹⁷ Bons, Megen /Aufwandsschätzung in der DV-Anwendungsentwicklung/ 23-36

bank basieren und Heuristiken über den Zusammenhang zwischen bestimmten Projektmerkmalen sowie über den damit verbundenen Aufwand enthalten¹⁸.

Einen weiteren Einsatzschwerpunkt wissensbasierter Systeme bildet die Führung des Benutzers durch den Software-Entwicklungsprozeß. Die Hilfestellungen reichen dabei von Anleitungen bei bestimmten Software-Entwicklungsaktivitäten bis zu Leistungen, die die Software-Entwicklung ohne Kenntnis einer Programmiersprache ermöglichen sollen.

Unter dem Schlagwort "automatische Programmierung" wurden bereits in den 70er Jahren die ersten Versuche unternommen, aus einer formalen Spezifikation und einer Liste von Randbedingungen ein ausführbares Programm zu generieren¹⁹. Die Vielzahl codegenerierender wissensbasierter Systeme resultiert vermutlich z. T. noch aus diesen Bemühungen, was zahlreiche Systeme, die auf der Basis sehr formaler (zumeist algebraischer) Spezifikationen ADA- oder PROLOG-Programme produzieren, belegen. Da die verfügbaren Mittel zur unmittelbaren Synthese von Programmen aus überwiegend unvollständigen, unstrukturierten, widersprüchlichen Benutzeranforderungen jedoch i. d. R. nicht ausreichten, verlagerte sich die Zielsetzung von einer automatischen Software-Entwicklung zu einer intelligenten Assistenz des Software-Entwicklungsprozesses²⁰. Dies belegt die zunehmende Zahl von unterstützenden wissensbasierten Systemen zur Programm-Codeerzeugung in konventionellen Sprachen, wie C oder COBOL, auf der Grundlage grafischer Spezifikationen.

Den prominentesten Vertreter eines wissensbasierten Systems, das im Rahmen der Informationsverwaltung die Vollständigkeit und Konsistenz der gespeicherten Objekte gewährleisten soll, stellt die regelbasierte Enzyklopädie des CASE-Toolkit ADW/IEW dar. Außerdem sind die Verwaltung von Objekten wie beispielsweise Spezifikationen sowie die Transformati-

¹⁸ Seibt /Entwicklung eines Expertensystems/ 329-344

¹⁹ Davis /Software Requirements Analysis/
Ramamoorthy /Software Engineering and Artificial Intelligence/ 9-20

²⁰ Luqi /A Computer-Aided Prototyping System/ 66-72
Ramamoorthy /Software Engineering and Artificial Intelligence/ 9-20
Rego, Lima /A Tool for Automating Facts Analysis/ 57-80

on der in verschiedenen Systemlebenszyklusphasen gewonnenen Ergebnisse als weitere von existierenden wissensbasierten Systemen wahrgenommene Informationsverwaltungsaufgaben zu nennen.

Existierende CASE-Werkzeuge übernehmen zwar eine Reihe von Routineaufgaben und entlasten den Software-Entwickler von einer Reihe administrativer Tätigkeiten, sie enthalten jedoch beispielsweise keinerlei Hilfestellung zur Auswahl einer geeigneten Methode oder eines Werkzeugs für eine bestimmte Projektphase²¹. Aber auch die Anwendung bestimmter Methoden, bei denen der Entwickler aufgrund fehlender zwingender Methodenvorgaben auf der Basis seines Erfahrungswissens und seiner Interpretationen Entwurfsentscheidungen fällen muß, werden von existierenden CASE-Werkzeugen i. d. R. weder ausreichend unterstützt noch für eine spätere Wiederverwendung dokumentiert²², so daß das für die spätere Wartung wichtige Wissen über den Prozeß der Überführung von Benutzeranforderungen in eine Spezifikation und die anschließende Umsetzung in ein Programmsystem nicht verfügbar ist²³. Die Bereitstellung von Expertise bei der Anwendung bestimmter Software-Entwicklungsmethoden ist deshalb ebenfalls ein charakteristisches Einsatzgebiet existierender wissensbasierter CASE-Systeme. Besonders häufig unterstützte Methoden sind neben JSP/JSD v. a. das Entity-Relationship-Modell sowie zunehmend objektorientierte Ansätze.

Wissensbasierte Systeme zur Testunterstützung besitzen als Testobjekte zumeist Programme, vielfach aber auch Spezifikationen.

Eine weiteres typisches Einsatzfeld wissensbasierter Systeme ist die Unterstützung der Auswahl wiederverwendbarer Programmkomponenten. Ferner existieren zahlreiche Systeme, die Entwürfe mittels Rückübersetzung von Programmcode in (grafische) Spezifikationen auf der Basis von Wissen über das Anwendungsgebiet und der verwendeten Methoden wiederverwenden (Abb. 4).

21 Lempp /Expertensystem für Objekt-orientiertes Design/ VI-21-01 - VI-212-15

22 Budgen, Marashi /Knowledge Use in Software Design/ 163-179

23 Balzer /Software Technology in the 1990's/ 38-45

Aktivität	Gesamt		Umfrage		Rest	
	absolut	relativ	absolut	relativ	absolut	relativ
Anforderungsanalyse	50	10,71%	26	12,87%	24	9,06%
Aufwandsschätzung	15	3,21%	5	2,48%	10	3,77%
Benutzerführung	68	14,56%	31	15,35%	37	13,96%
Codegenerierung	79	16,92%	38	18,81%	41	15,47%
Informationsverwaltung	48	10,28%	27	13,37%	21	7,92%
Methodenunterstützung	35	7,49%	25	12,38%	10	3,77%
Testen	52	11,13%	19	9,41%	33	12,45%
Wiederverwendung	50	10,71%	27	13,37%	23	8,68%
Unbekannt	70	14,99%	4	1,98%	66	24,91%
Summe	467	100,00%	202	100,00%	265	100,00%

Abb. 4: Häufigkeitsverteilung des Merkmals von wissensbasierten Systemen im CASE unterstützte Aktivitäten im Systemlebenszyklus

3.2.3 Unterstützte Aufgabenklassen im Systemlebenszyklus

Unterteilt man die Anwendungsgebiete wissensbasierter Systeme in Aufgaben der Analyse (Diagnose und Interpretation), Aufgaben der Synthese (Konfigurierung und Planung) und Mischformen (z. B. Unterweisung und Vorhersage)²⁴ so wird evident, daß in der Domäne CASE Aufgaben der Analyse dominieren, da die Software-Entwicklung von gegebenen Benutzeranforderungen ausgeht und vielfach existierende Systeme als Ausgangspunkt der Entwicklungsaktivitäten fungieren. Dies belegt auch die Auswertung der Untersuchungsergebnisse, wobei Systeme, die der Aufgabenklasse Interpretation zuzuordnen sind, überwiegen (Abb. 5).

²⁴ Schmitz /Expertensysteme/ 611-626

Aufgabenklasse	Gesamt		Umfrage		Rest	
	absolut	relativ	absolut	relativ	absolut	relativ
Diagnose	60	20,13%	12	16,22%	48	21,43%
Interpretation	74	24,83%	21	28,38%	53	23,66%
Konfigurierung	37	12,42%	13	17,57%	24	10,71%
Planung	53	17,79%	8	10,81%	45	20,09%
Unterweisung	5	1,68%	0	0,00%	5	2,23%
Vorhersage	7	2,35%	2	2,70%	5	2,23%
Unbekannt	62	20,81%	18	24,32%	44	19,64%
Summe	298	100,00%	74	100,00%	224	100,00%

Abb. 5: Häufigkeitsverteilung des Merkmals von wissensbasierten Systemen im CASE unterstützte Aufgabenklasse

3.3 Systembezogene Merkmale wissensbasierter Systeme im CASE

Die Umfrageergebnisse zeigen, daß es sich bei dem weitaus größten Teil der untersuchten Systeme um Prototypen handelt, was darauf hinweist, daß wissensbasierte CASE-Systeme das Forschungsstadium derzeit noch nicht verlassen haben (Abb. 6).

Status	Gesamt		Umfrage		Rest	
	absolut	relativ	absolut	relativ	absolut	relativ
Projekt	25	8,39%	3	4,05%	22	9,82%
Prototyp	174	58,39%	27	36,49%	147	65,63%
Running System	31	10,40%	11	14,86%	20	8,93%
Produkt	50	16,78%	33	44,59%	17	7,59%
Unbekannt	18	6,04%	0	0,00%	18	8,04%
Summe	298	100,00%	74	100,00%	224	100,00%

Abb. 6: Häufigkeitsverteilung des Merkmals Status wissensbasierten Systeme im CASE

Die Unterstützung der Software-Entwicklung ausschließlich durch wissensbasierte Systeme ist offensichtlich nicht sinnvoll, sondern erfordert zusätzlich den Einsatz konventioneller Technologie unter Ausnutzung von

Synergieeffekten. Diese Vermutung legen jedenfalls die untersuchten Systeme nahe, die neben überwiegend wissensbasiertem meist auch aus konventionellem Code bestehen (Abb. 7).²⁵

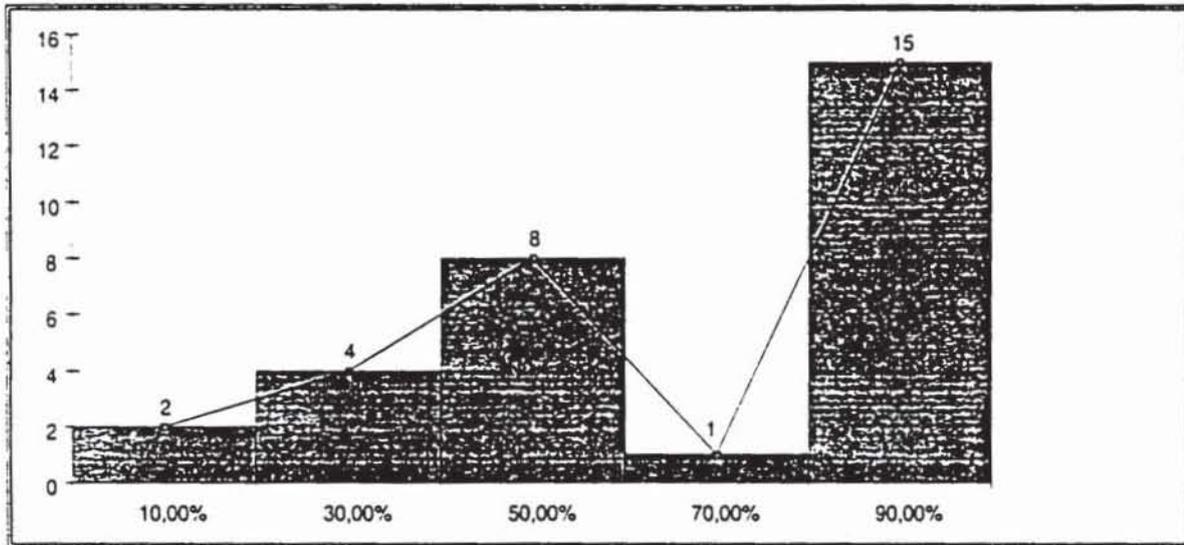


Abb. 7: Histogramm und Polygonzug zur Verteilung des Merkmals Verhältnis von wissensbasiertem und konventionellem Code bei wissensbasierten Systemen im CASE

Die bei den existierenden wissensbasierten CASE-Systemen verwendeten Wissensrepräsentationsformalismen sind von besonderer Bedeutung, da sie wertvolle Hilfestellungen bei der Realisierung ähnlicher Projekte bieten können.²⁶ Ähnlich wie in anderen betrieblichen Anwendungen wissensbasierter Systeme dominieren im CASE Regeln als Wissensrepräsentationsformalismus. Die Objektorientierte Programmierung folgt im deutlichem Abstand an der zweiten Stelle, während Frames, Prädikatenlogik und Semantische Netze weniger verbreitet sind (Abb. 8).

²⁵ Hierbei ist die kleine Stichprobe von 30 Systemen zu berücksichtigen, zu denen konkrete Zahlenangaben zum Verhältnis zwischen wissensbasiertem und konventionellem Code vorliegen, zu beachten.

²⁶ Für die hier vorgenommene Einordnung als wissensbasiertes System ist die Verwendung eines Wissensrepräsentationsformalismus eine notwendige, aber nicht hinreichende Bedingung. Es muß in jedem Falle eine getrennte Inferenzkomponente existieren, auf die an dieser Stelle allerdings nicht eingegangen wird.

Formalismus	Gesamt		Umfrage		Rest	
	absolut	relativ	absolut	relativ	absolut	relativ
Frames	27	6,87%	21	15,33%	6	2,34%
OOP	51	12,98%	26	18,98%	25	9,77%
Prädikatenlogik	27	6,87%	13	9,49%	14	5,47%
Regeln	111	28,24%	55	40,15%	56	21,88%
Semantische Netze	21	5,34%	14	10,22%	7	2,73%
Unbekannt	156	39,69%	8	5,84%	148	57,81%
Summe	393	100,00%	137	100,00%	256	100,00%

Abb. 8: Häufigkeitsverteilung des Merkmals Wissensrepräsentationsformalismen wissensbasierter Systeme im CASE

Die Komplexität der Anwendungen im CASE hat in zahlreichen Systemen zu einer hybriden Wissensdarstellung geführt, bei der die Verknüpfung von Regeln mit Objektorientierter Programmierung die am häufigsten verwendete Kombination von Wissensrepräsentationsformalismen darstellt (Abb. 9).

		Frames	OOP	Prädikatenlogik	Regeln	Semantische Netze	Zeilen-summe
Frames	absolut	-	14	6	23	7	50
	relativ	-	5,19%	2,22%	8,52%	2,59%	18,52%
OOP	absolut	14	-	7	29	6	56
	relativ	5,19%	-	2,59%	10,74%	2,22%	20,74%
Prädikatenlogik	absolut	6	7	-	23	4	40
	relativ	2,22%	2,59%	-	8,52%	1,48%	14,81%
Regeln	absolut	23	29	23	-	16	91
	relativ	8,52%	10,74%	8,52%	-	5,93%	33,70%
Semantische Netze	absolut	7	6	4	16	-	33
	relativ	2,59%	2,22%	1,48%	5,93%	-	12,22%
Spalten-summe	absolut	50	56	40	91	33	270
	relativ	18,52%	20,74%	14,81%	33,70%	12,22%	100,00%

Abb. 9: Häufigkeitsverteilung der Kombination verschiedener Wissensrepräsentationsformalismen wissensbasierter Systeme im CASE

Der Umfang der Wissensbasis der untersuchten Systeme ist sehr unterschiedlich und reicht beispielsweise von 42 bis 200.000 Regeln²⁷ (Median 175) bzw. von 500 bis 124.000 LOC (Median 14.628).

Bezeichnet man regelbasierte Systeme mit weniger als 500 Regeln als klein, Systeme mit 500 bis 1.500 Regeln als mittel und Systeme mit mehr als 1.500 Regeln als groß²⁸, so dominieren bei den 20 Systemen, deren Regelanzahl bekannt ist, kleine Systeme (Abb. 10).

Systemgröße	Gesamt		Umfrage		Rest	
	absolut	relativ	absolut	relativ	absolut	relativ
Kleines System	14	70,00%	12	75,00%	2	50,00%
Mittleres System	2	10,00%	1	6,25%	1	25,00%
Großes System	4	20,00%	3	18,75%	1	25,00%
Summe	20	100,00%	16	100,00%	4	100,00%

Abb. 10: Klassierte Häufigkeitsverteilung des Merkmals Umfang der Wissensbasis regelbasierter Systeme im CASE

Die Datenbasis beinhaltet neben 16 Systemen, deren Entwicklungsdauer mit "mehreren" Personenjahren angegeben ist, noch 34 weitere Systeme, zu denen konkrete Zahlenangaben vorliegen. Während 7 Systeme einen Aufwand von mehr als 200 Personenmonaten verursachten, blieb bei den meisten Systemen der Aufwand unter 200 Personenmonaten (Abb. 11). Dabei verursachten mehr als die Hälfte dieser Systeme mit maximal 20 Personengruppen einen relativen geringen Entwicklungsaufwand.

²⁷ Diese Zahl wurde für Transform, einer Programmierumgebung für IBM MVS-Rechner, angegeben.

²⁸ Harmon, King /Expertensysteme in der Praxis/

Personenmonate	absolute Häufigkeit	relative Häufigkeit	absolute Summenhäufigkeit	relative Summenhäufigkeit
bis 10	8	29,63%	8	29,63%
11 - 20	7	25,93%	15	55,56%
21 - 30	5	18,52%	20	74,07%
31 - 40	3	11,11%	23	85,19%
41 - 50	0	0,00%	23	85,19%
51 - 60	2	7,41%	25	92,59%
61 - 200	2	7,41%	27	100,00%

Abb. 11: Klassierte Häufigkeits- und Summenhäufigkeitsverteilung des Merkmals Erstellungsaufwand wissensbasierter Systeme im CASE in der Klasse bis 200 Personenmonate

3.4 Zusammenhänge zwischen Merkmalen wissensbasierter Systeme im CASE

Als Maß für die Untersuchung des Zusammenhanges zwischen Merkmalen wissensbasierter Systeme im CASE fungiert der aus der Gesamtdatenbasis unter Vernachlässigung unbekannter Merkmalswerte berechnete Kontingenzkoeffizient C, der für vollkommen unabhängige Merkmale den Wert 0 und für eindeutig abhängige Merkmale den Wert 1 annimmt²⁹.

Das deutliche Übergewicht der wissensbasierten Unterstützung der frühen Phasen des Systemlebenszyklus schlägt sich nur unterproportional in einsatzfähigen Systemen bzw. marktgängigen Produkten nieder (Abb. 12).

Während Systeme zur Unterstützung der Analyse-, Entwurfs- und Realisierungsphase sowie für das Projektmanagement die Schwelle zum betrieblichen Einsatz derzeit überwiegend noch nicht überschritten haben, dominieren sowohl in den Phasen der Erprobung und Konsolidierung, Pflege und Wartung als auch für die Dokumentation einsatzfähiger Systeme (Abb. 13).

²⁹ Es handelt sich hierbei um den korrigierten Kontingenzkoeffizienten von Pearson auf der Basis der Hilfsgröße χ^2 (Chi-Quadrat).

C = 0,28		Projekt	Prototyp	Running System	Produkt	Zeilen-summe
Analyse	absolut	12	45	9	24	90
	relativ	21,43%	15,73%	12,50%	13,19%	15,10%
Entwurf	absolut	12	57	12	33	114
	relativ	21,43%	19,93%	16,67%	18,13%	19,13%
Realisierung	absolut	12	73	12	35	132
	relativ	21,43%	25,52%	16,67%	19,23%	22,15%
Erprobung und Konsolidierung	absolut	3	12	6	20	41
	relativ	5,36%	4,20%	8,33%	10,99%	6,88%
Pflege und Wartung	absolut	5	29	9	30	73
	relativ	8,93%	10,14%	12,50%	16,48%	12,25%
Projektmanagement	absolut	4	31	4	8	47
	relativ	7,14%	10,84%	5,56%	4,40%	7,89%
Software-Qualitätssicherung	absolut	5	27	11	14	57
	relativ	8,93%	9,44%	15,28%	7,69%	9,56%
Dokumentation	absolut	3	12	9	18	42
	relativ	5,36%	4,20%	12,50%	9,89%	7,05%
Spalten-summe	absolut	56	286	72	182	596
	relativ	9,40%	47,99%	12,08%	30,54%	100,00%

Abb. 12: Kontingenztabelle der Merkmale unterstützte Phasen im Systemlebenszyklus und Status des Systems

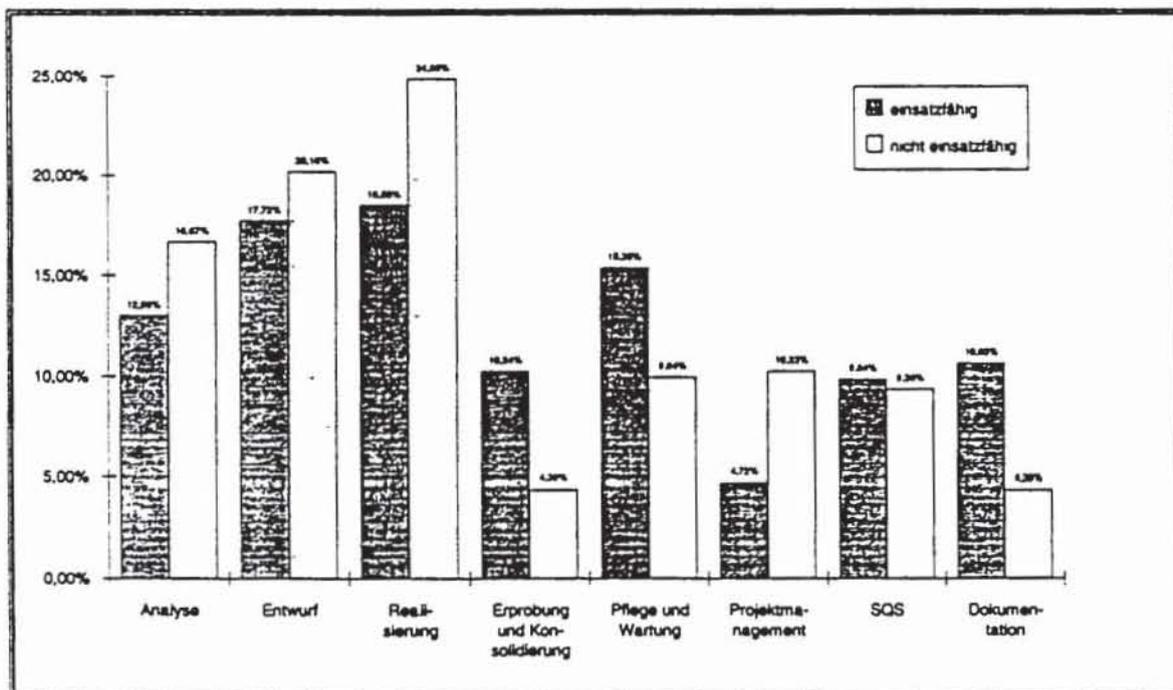


Abb. 13: Säulendiagramm zur Kontingenztabelle der Merkmale unterstützte Phasen im Systemlebenszyklus und Status des Systems

Ein schwacher bis mittelstarker Zusammenhang ist zwischen der Aufgabenklasse und dem verwendeten Wissensrepräsentationsformalismus festzustellen: In Abhängigkeit von der Aufgabenklasse werden offensichtlich unterschiedliche Wissensrepräsentationsformalismen präferiert. So werden beispielsweise Regeln als besonders geeignet für die Vorhersage- und Planungsprobleme, Objektorientierte Programmierung für Diagnoseaufgaben und Frames für Konfigurierungsprobleme angesehen (Abb. 14). Unterstellt man, daß aus der Beliebtheit eines Wissensrepräsentationsformalismus auf dessen Eignungsgrad geschlossen werden kann, so bieten diese Untersuchungsergebnisse potentiellen Entwicklern wissensbasierter Systeme im CASE erste Orientierungshilfen.

$C = 0,25$		Diagnose	Interpretation	Konfigurierung	Planung	Unterweisung	Vorhersage	Zeilen-summe
Projekt	absolut	3	6	4	6	0	0	19
	relativ	5,36%	8,45%	11,76%	11,76%	0,00%	0,00%	8,56%
Prototyp	absolut	34	42	20	36	2	6	140
	relativ	60,71%	59,15%	58,82%	70,59%	66,67%	85,71%	63,06%
Running System	absolut	7	10	3	5	1	1	27
	relativ	12,50%	14,08%	8,82%	9,80%	33,33%	14,29%	12,16%
Produkt	absolut	12	13	7	4	0	0	36
	relativ	21,43%	18,31%	20,59%	7,84%	0,00%	0,00%	16,22%
Spalten-summe	absolut	56	71	34	51	3	7	222
	relativ	25,23%	31,98%	15,32%	22,97%	1,35%	3,15%	100,00%

Abb. 14: Kontingenztafel der Merkmale Aufgabenklasse und Wissensrepräsentationsformalismen³⁰

Der Verbreitungsgrad eines Wissensrepräsentationsformalismus als Indikator für dessen Eignungsgrad erscheint recht problematisch. Dagegen ist der Reifegrad des entwickelten Systems ein aussagefähigerer Indikator für den Erfolg eines Wissensrepräsentationsformalismus. So sind nicht Regeln, sondern Objektorientierte Programmierung und Frames die in den untersuchten marktgängigen CASE-Systemen am häufigsten verwendeten Wissensrepräsentationsformalismen (Abb. 15). Dies führt zu der These,

³⁰ Die Aufgabenklasse Unterweisung fehlt, da bei keinem untersuchten System dieser Aufgabenklasse der verwendete Wissensrepräsentationsformalismus bekannt ist.

daß Regeln zwar den am häufigsten verwendeten, allerdings nicht den wirksamsten Wissensrepräsentationsformalismus darstellen.

$C = 0,37$		Frames	OOP	Prädi- katen- logik	Regeln	Seman- tische Netze	Zeilen- summe
Projekt	absolut	3	8	1	0	1	13
	relativ	11,11%	15,69%	3,85%	0,00%	5,00%	5,75%
Prototyp	absolut	9	21	15	60	10	115
	relativ	33,33%	41,18%	57,69%	58,82%	50,00%	50,88%
Running System	absolut	5	3	3	13	4	28
	relativ	18,52%	5,88%	11,54%	12,75%	20,00%	12,39%
Produkt	absolut	10	19	7	29	5	70
	relativ	37,04%	37,25%	26,92%	28,43%	25,00%	30,97%
Spalten- summe	absolut	27	51	26	102	20	226
	relativ	11,95%	22,57%	11,50%	45,13%	8,85%	100,00%

Abb. 15: Kontingenztabelle der Merkmale Wissensrepräsentationsformalismen und Status des Systems

4 Schlußfolgerungen für wissensbasiertes CASE

Bevor die Essenz der Untersuchung kritisch reflektiert wird, sei an dieser Stelle auf die Probleme der Datenerhebung hingewiesen, die die Aussagefähigkeit der Ergebnisse teilweise einschränken:

- Es sind keine Aussagen darüber möglich, inwieweit die untersuchten Systeme repräsentativ für die Gesamtmenge aller existierenden Systeme im CASE sind.
- Die Aufnahme von Systemen, bei denen die Klassifikation als wissensbasiert oder die Zuordnung zur Domäne CASE nicht mit Sicherheit konstatiert werden kann, führt u. U. zu einer zu umfangreichen Datenbasis.
- Die Angaben der Entwickler sind unabhängig von der Datenquelle i. d. R. nicht verifizierbar.
- Die entdeckten Systeme zeigen, daß wissensbasierte Systeme in CASE-Werkzeugen eingesetzt wurden. Es sind keine Aussagen über den beim Anwender erzielten Erfolg dieser wissensbasierten CASE-

Systeme möglich.

- Die durchgeführte Untersuchung unterliegt, wie andere Untersuchungen dieser Art auch, allen Schwächen und Problemen der empirischen Forschung.

Schließlich sei darauf hingewiesen, daß in der für einen Zeitschriftenaufsatz gebotenen Kürze nur die wesentliche Essenz der Untersuchungsergebnisse hervorgehoben werden und keine wissenschaftstheoretische oder ergebnisbezogene Detailanalyse erfolgen kann. Dies gilt gleichermaßen für die Darstellung des Bezugsrahmens sowie des Forschungsdesigns. [siehe hierzu Herzwurm /Wissensbasiertes CASE/]

In bezug auf die Zielsetzung lassen sich die Ergebnisse wie folgt resümieren:

- "State of the Art" wissensbasierter Systeme im CASE
Wissensbasierte Systeme im CASE befinden sich überwiegend noch im Forschungsstadium, zeigen aber bereits in einzelnen Bereichen Tendenzen zur Einsatzreife.
- Charakteristische Merkmale wissensbasierter Systeme im CASE
Das "typische" wissensbasierte System im CASE stammt aus den U.S.A., wurde von einer Forschungseinrichtung als Prototyp für die Phase Entwurf und Analyse zur Führung des Software-Entwicklers bei der Anwendung einer Methode zur Erhebung und Spezifikation von Benutzeranforderungen unter Einsatz von Regeln entwickelt, generiert aus der Spezifikation Code und ist der Aufgabenklasse Analyse zuzuordnen. Es bestehen tendenzielle Zusammenhänge zwischen unterstützten Phasen des Systemlebenszyklus, zugeordneter Aufgabenklasse, verwendeten Wissensrepräsentationsformalismen und dem Reifegrad wissensbasierter CASE-Systeme.
- Wissenschaftliche und praktische Relevanz des Themas wissensbasiertes CASE
Die Untersuchung existierender wissensbasierter Systeme zur Entwicklung konventioneller Software dokumentiert zahlreiche Entwicklungsaktivitäten von Anwendern, Software-Häusern sowie Forschungsein-

richtungen und somit zugleich das breite Interesse von Wissenschaftlern bzw. Praktikern am Themenkomplex wissensbasiertes CASE.

- Anregungen für Theorie und Praxis

Durch das Aufzeigen der grundsätzlichen Machbarkeit der Unterstützung der Software-Entwicklung durch wissensbasierte Systeme können Schwellenängste bei der Erprobung der wissensbasierten Technologie abgebaut werden. Die existierenden Anwendungen sind potentielle Anregungen für weitere Anstrengungen in dieser Domäne. Mit der Erkenntnis zwischen dem Zusammenhang von Aufgabenklasse und verwendeten Wissensrepräsentationsformalismus sind dem potentiellen Entwickler wissensbasierter Systeme im CASE einige Orientierungshilfen bei der Planung eines Projektes an die Hand gegeben worden. Außerdem bieten die genannten Zahlen bezüglich der Systemgröße und dem erforderlichen Erstellungsaufwand erste, grobe Eckdaten zur Abschätzung der erforderlichen Ressourcen für derartige Vorhaben.

Literatur

Balzer /Software Technology in the 1990`s/

Balzer, Robert et al.: Software Technology in the 1990's: Using a New Paradigm. In: Computer, Vol. 16, November 1983, S. 38-45

Bons, Megen /Aufwandsschätzung in der DV-Anwendungsentwicklung/

Bons, Heinz; Megen, Rudolf van: Aufwandsschätzung in der DV-Anwendungsentwicklung - Probleme und Lösungen. In: HMD, 21. Jg., Nr. 116, März 1981, S. 23-36

Bortz /Lehrbuch der empirischen Forschung für Sozialwissenschaftler/

Bortz, Jürgen: Lehrbuch der empirischen Forschung für Sozialwissenschaftler. Unter Mitarbeit von Dieter Bongers, Berlin u. a. 1984

Berkau, Herzwurm /Auswahl PC-gestützter Software-Entwicklungs-umgebungen/

Berkau, Dirk; Herzwurm, Georg: Kriterien für die Auswahl PC-gestützter Software-Entwicklungs-umgebungen. In: Information Management, 7. Jg., Heft 1, 1992, S. 42-55

Budgen, Marashi /Knowledge Use in Software Design/

Budgen, David; Marashi, Mustafa: Knowledge Use in Software Design. In: Spurr, Layzell (Hrsg.): CASE on Trial, Chichester u. a. 1990, S. 163-179

Davis /Software Requierements Analysis/

Davis, Alan M.: Software Requirements Analysis an Specification. Englewood Cliffs, New Jersey 1990

Harmon, King /Expertensysteme in der Praxis/

Harmon, Paul; King, David: Expertensysteme in der Praxis. Perspektiven, Werkzeuge, Erfahrungen. 3. Auflage, München-Wien 1989

Heinrich, Kurbel /Studien- und Forschungsführer Wirtschaftsinformatik

Heinrich, L. J.; Kurbel, Karl: Studien- und Forschungsführer Wirtschaftsinformatik. 3. Auflage, Berlin u. a. 1988

Hesel /Was ist Expertentechnik?/

Hesel, Gerhard: Was ist Expertensystemtechnik? In: Diebold (Hrsg.): Expertensystemtechnik in CASE-Umgebungen. Teilnehmerunterlagen des Diebold Seminars vom 28.06.1990 in Eschborn

Herzwurm /Wissensbasiertes CASE/

Herzwurm, Georg: Wissensbasiertes CASE. Theoretische Analyse, empirische Untersuchung und Entwicklung eines Prototypen. 2. Auflage, Braunschweig-Wiesbaden 1993

Humphrey /Improving the Software Development/

Humphrey, Watt S.: Improving the Software Development Process. In: Datamation, Vol. 35, No. 7, April 1989, S. 28-30

Kurbel, Pietsch /Expertensystem-Projekte/

Expertensystem-Projekte: Entwicklungsmethodik, Organisation und Management. In: Informatik Spektrum, Band 12, Heft 3, Juni 1989, S. 133-146

Lempp /Expertensystem für Objekt-orientiertes Design/

Lempp, Peter: Expertensystem für Objekt-orientiertes Design: Aktive Un-

terstützung und methodische Führung durch das XPS. In: Fähnrich, K.-P. (Hrsg.): ONLINE '89, 12. Europäische Kongreßmesse für Technische Kommunikation, Kongreß VI, Kongreßband, Velbert 1989, S. VI-21-01 - VI-21-15

Lowry, Duran /Knowledge-based Software Engineering/

Lowry, Michael; Duran, Raul: Knowledge-based Software Engineering. In: Barr, Avron et al. (Hrsg.): The Handbook of Artificial Intelligence, Volume IV, Reading u. a. 1989, S. 243-322

Luqi /A Computer-Aided Prototyping System/

Luqi, Mohammed Ketabchi: A Computer-Aided Prototyping System. In: IEEE Software, Vol. 5, March 1988, S. 66-72

McClure /CASE is software automation/

McClure, Carma: CASE is software automation. Engelwood Cliffs, New Jersey 1989

Mertens /Betriebliche Expertensystem-AnwendungenMertens/

Peter et. al.: Betriebliche Expertensystem-Anwendungen. Eine Materialsammlung. 2. Auflage, Berlin u. a. 1990

Nomina Information Services (Hrsg.) /ISIS Firmen Report 1991/

Nomina Information Services (Hrsg.): ISIS Firmen Report 1991. Die Unternehmen der Informationsindustrie. Deutschland, Österreich, Schweiz. 22. Jg., München 1991

Puncello /ASPIS/

Puncello, P. Paolo et al.: ASPIS: A Knowledge-Based CASE Environment.

IN: IEEE Software, Vol. 7, March 1988, S. 58-65

Ramamoorthy /Software Engineering and Artificial Intelligence/

Ramamoorthy, C. V. et al.: On Issues in Software Engineering and Artificial Intelligence. In: International Journal of Software Engineering and Knowledge Engineering, Vol. 1, No. 1, March 1991, S. 9-20

Regier /Expertensysteme für den betriebswirtschaftlichen Einsatz/

Regier, Hans J.: Expertensysteme für den betriebswirtschaftlichen Einsatz. Eigenschaften * Funktionen * Probleme * Nutzen. In: Office Management, 38. Jg., Heft 11, 1988, S. 16-25, S. 16-25

Rego, Lima /A Tool for Automating Facts Analysis/

Rego, Helena Barbosa; Lima, Jorge Reis: A Tool for Automating Facts Analysis. In: Spurr, Layzell (Hrsg.): CASE on Trial, Chichester u. a. 1990, S. 57-80

Schmitz /Expertensysteme/

Schmitz, Paul: Expertensysteme. In: Frese, Erich (Hrsg.): Handwörterbuch der Organisation, 3. Auflage, Stuttgart 1992, S. 611-626

Schwarze /Grundlagen der Statistik/

Schwarze, Jochen: Grundlagen der Statistik. Beschreibende Verfahren. Herne-Berlin 1981

Seibt /Entwicklung eines Expertensystems/

Seibt, Dietrich: Entwicklung eines Expertensystems zur Software-Aufwandschätzung - Erfahrungen aus einem Praxisprojekt. In: Horvath, P. (Hrsg.): Wirtschaftlichkeit neuer Produktions- und Informationstechnologien, Stuttgart 1988, S. 329-344