

*Georg Herzwurm, Andreas Hierholzer, Michael Kunz**

Tendenzen der CASE-Tool Evaluierung

Inhaltsverzeichnis

1. Überblick	228
2. Tendenzaussagen bei einer Gliederung nach der Entwicklungsumgebung	230
2.1. PC-(LAN-)Werkzeuge	230
2.2. UNIX-Werkzeuge	232
2.3. Werkzeuge für heterogene Umgebungen	233
3. Tendenzaussagen bei einer Gliederung nach der Entwicklungsmethodik	234
3.1. Konventionelle Werkzeuge	234
3.2. Objektorientierte Werkzeuge	234

* *Dr. Georg Herzwurm, Dipl.-Kfm. Andreas Hierholzer, Dipl.-Kfm. Michael Kunz, Lehrstuhl für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln*

1. Überblick

Betrachtet man den CASE-Markt, so erscheint das Angebot an Produkten, verfügbaren Entwicklungs- und Ablaufumgebungen, unterstützten Vorgehensweisen, Methoden und Beschreibungsmitteln, Generierungsmöglichkeiten und Projektmanagementfunktionen auf den ersten Blick ausgesprochen unübersichtlich. Abbildung 1 gibt einen diesbezüglichen Überblick über die im Rahmen der CASE-Studie evaluierten Produkte.

Dieser Tatbestand bleibt trotz oder vielleicht auch gerade wegen einer gewissen Marktberreinigung, die mit dem Nachlassen der „CASE-Euphorie“ verbunden ist, bestehen. Ursache hierfür ist die Tatsache, daß viele Anbieter mit einer neuen Positionierung ihrer Produkte versuchen, neue Kundenkreise zu erschließen. Dies geschieht in der Regel dadurch, daß neue Funktionen im Bereich objektorientierte Systementwicklung oder Entwicklung von Client-Server-Systemen hinzugefügt werden und dementsprechend neue Produkte und Produktkomponenten entstehen.

Es scheint somit schwierig, generelle Tendenzen der Marktentwicklung aus der Fülle der teilweise widersprüchlichen Einzelinformationen abzuleiten. Solche Tendenzen stellen aber - trotz aller nicht zu vermeidenden Vereinfachungen, die mit ihnen einhergehen - eine unverzichtbare Orientierungshilfe bei der Auswahl und Einführung von CASE-Umgebungen dar.

Produkt	Hersteller	Betriebssysteme				grafische Beschreibungsmittel	Methoden	besondere Vorgehensweisen	Generierung				PM	KM
		Dos	MS-Windows	OS/2	UNIX				Sonstige	DB-Schemata	Code	maskenor. UI		
ADW	Knowledgeware	•	•	•	•	M-Spek., ERD, AD, DFD, SC	SA, SD, IE	IE	•	•	•	•		
case 4/0	mikromTools	•				FRD, JD	ERM, SA, SD		•	•	•	•		•
DDB-Conceptor/ DDB-CASE	Delta	•	•	•	•	DTD, ERD, M-Spek., FB	ERM, SA	Merise, SDMS	•	•	•	•		
Exaccelerator	Intersolv	•				DTD, ERD, K-Spek	ERM, RT	SA, SC	•				•	
IEF	TI Information Engineering	•	•	•	MVS	AD, SC, ELIJ, ERD, DDD	ERM	IE	•	•	•	•		•
Innovator	MIT	•	•	•	•	DTD, SERM, SC, M-Spek	ERM, SA, SD		•	•	•	•		
Maestro II	Softlab	•	•	•	•	ERD, FB, FKUM	ERM	V-Modell, SETec, Merise, SSADM	•	•	•	•		•
ObjecTool	Ritech Consulting	•	•	•	•		OOA, OOD							
OM-Tool	IOPProducts	•	•	•	•		OOA, OOD							
NEW (PredictCASE)	Software AG	•	•	•	•	DTD, ERD, BTD, KTD	ERM, SA	ISOTEC						•

Legende:

• : verfügbar

AD: Action Diagram
 BFD: Business Function Diagram
 CAD: Klassenstrukturdiagramm
 DDD: Dialog Design Diagram
 DFD: Datenflußdiagramme
 ELIJ: Entity Life History
 ERD: Entity-Relationship-Diagramme
 ERM: Entity-Relationship-Modellierung
 FB: Funktionsbaum
 IE: Information Engineering

Fkflld: Funktionsflußdiagramme
 FND: Function Net Diagram
 KFD: Kontrollflußdiagramme
 KM: Konfigurationsmanagement
 K-Spek: Kontroll-Spezifikation
 M-Spek: Mini-Spezifikation
 MS: Modulstruktur
 OAOD: OO Analysis/Design (Condo/ Yourdon)
 OOA: objektorientierte Analyse
 OOD: objektorientiertes Design
 OOSA: OO Systemanalyse (Shlaacr/ Mellor)

OMT: Object Modelling Technique
 PM: Projektmanagement
 RT: Real Time
 SA: Strukturierte Analyse
 SC: Structure Charts
 SD: strukturiertes Design
 SERM: strukturierte ERM
 STD: State Transition Diagram
 VSOM: Vorgehensmodell für semantisches Objektmodell

Abb. 1: Überblick über die evaluierte Werkzeuge (Teil I)

Produkt	Hersteller	Betriebssysteme					grafische Beschreibungsmittel	Methoden	besondere Vorgehensweisen	Generierung			PM	KM
		Dos	MS-Windows	OS/2	UNIX	Sonstige				DB-Schemata	Code	maskenorientierte UI		
ProMod	CAP & bit SSP		•	•	•	VMS	DFD, ERD, FB, K-Spek, M-Spek, SC	SA, SD, ERM						
Software through Pictures	IDE				•		DFD, ERM, OOD, SC	SA, AD, OOA, OOD						•
SDW	SDW Software	•	•				DFD, ERD, SC	ERM, SA, SD	•					•
SOM-CASE	Ual Bamberg	•	•		•		SERM, SOM	OOA, OOD		•				
Systems Engineer	LBMS		•				DFD, ELH, ERD, MS	SA, SD, ERM	•	•				•
Teamwork	Instrumac				•		ERM, M-Spek, SC	ERM, SA, SD	•	•				•
Westmount-I-CASE (Yourdon, OMT)	Westmount Technology B. V.				•	VMS	CAD, DFD, ERD, M-Spek, STD	ERM, OMT, SA, SD	•	•				•

Legende:

- : verfügbar
- AD: Action Diagram
- BFD: Business Function Diagram
- CAD: Klassenstrukturdiagramm
- DDD: Dialog Design Diagram
- DFD: Datenflußdiagramme
- ELH: Entity Life History
- ERD: Entity-Relationship-Diagramme
- ERM: Entity-Relationship-Modellierung
- FB: Funktionsbaum
- IE: Information Engineering
- Fktfl: Funktionsflußdiagramme
- FND: Function Net Diagram
- KFD: Kontrollflußdiagramme
- KM: Konfigurationsmanagement
- K-Spek: Kontroll-Spezifikation
- M-Spek: Mini-Spezifikation
- MS: Modulstruktur
- OAOD: OO Analysis/Design (Coad/Yourdon)
- OOA: objektorientierte Analyse
- OOD: objektorientiertes Design
- OOSA: OO Systemanalyse (Shlaer/Mellor)
- OMT: Object Modelling Technique
- PM: Projektmanagement
- RT: Real Time
- SA: Strukturierte Analyse
- SC: Structure Charts
- SD: strukturiertes Design
- SERM: strukturierte ERM
- STD: State Transition Diagram
- VSOM: Vorgehensmodell für semantisches Objektmodell

Abb. 1: Überblick über die evaluierten Werkzeuge (Teil 2)

Ausgangspunkt solcher Tendenzen ist die Art und Weise der Aufteilung der betrachteten CASE-Produkte, die es erst ermöglicht, sinnvolle Aussagen zu Gruppen von CASE-Werkzeugen zu machen. In diesem Beitrag wird bei dieser Aufteilung auf formalere Verfahren verzichtet und eine Gruppierung vorgenommen, die sich aus den Erfahrungen im Verlauf der Studie ergeben hat.

Diese Aufteilung gliedert in zwei Dimensionen: Zum einen wird eine Unterscheidung nach der Art der Entwicklungsumgebung vorgenommen, für die die Werkzeuge zur Verfügung stehen. Hier wird differenziert nach:

- PC-(LAN-)Werkzeugen
- UNIX-Werkzeugen
- Werkzeugen für heterogene Umgebungen.

Zum anderen werden anhand der zu Grunde liegenden Entwicklungsmethodik

- Werkzeuge für konventionelle Methoden
- von objektorientierten Werkzeugen abgegrenzt.

Für die so gebildeten Gruppen sollen in diesem Beitrag für die im Rahmen der CASE-Studie evaluierten Werkzeuge Tendenzen und Trends aufgezeigt sowie idealtypisch Stärken und Schwächen beschrieben werden.

2. Tendenzaussagen bei einer Gliederung nach der Entwicklungsumgebung

2.1. PC-(LAN-)Werkzeuge

PC-Werkzeuge laufen mit Ausnahme von CASE-Werkzeugen, die nicht teamfähig sind, in (lokalen) Netzen ab. Die Vorteile von PC-Werkzeugen lassen sich folgendermaßen charakterisieren:

- PC-Tools zeichnen sich gegenüber den anderen Gruppen durch die höchste Benutzerfreundlichkeit aus. Sie bedienen sich ausnahmslos grafischer Oberflächen und können sich auf Vorkenntnisse der Benut-

zer mit diesen Oberflächen stützen. Betrachtet man die gebotene Benutzerfreundlichkeit, kommt man zu dem Schluß, daß sie offensichtlich von Entwicklern entworfen und hergestellt wurden, die durch ihren vorherigen Kontakt mit der Entwicklung von PC-Standardsoftware der Benutzerfreundlichkeit eine höhere Bedeutung beimessen als dies bei den beiden anderen betrachteten Gruppen der Fall ist. Zu beobachten ist weiterhin, daß die benutzerunfreundlichsten PC-Tools benutzerfreundlicher sind als in diesem Punkt „schlechte“ heterogene und UNIX-Tools.

- CASE-Werkzeuge im PC-Bereich sind vergleichsweise preiswert, vor allen Dingen ausgesprochen preiswerte Werkzeuge sind ausschließlich hier zu finden. Nichtsdestotrotz gibt es auch hier Systeme, die für komplett ausgestattete Arbeitsplätze zu Softwarekosten von über 50.000,- DM führen.
- PC-Werkzeuge sind häufig in den frühen Phasen besonders stark, bieten aber in der Implementierungsphase meist eine geringere Leistungsfähigkeit. Schwächen bestehen auch im „Randbereich“ der Funktionalität: Leistungsfähige Komponenten zur Unterstützung des Konfigurations- und Projektmanagements, Testwerkzeuge u.ä. sind selten anzutreffende Ausnahmen.
- Durch die Nähe zu Standardapplikationen bieten sie in der Regel gute, über DDE usw. hervorragend integrierte Dokumentationsmöglichkeiten.
- Im konventionellen Bereich ist die Anzahl und Vielgestaltigkeit der angebotenen Programme und damit die Auswahl am besten.

Nachteile ergeben sich in den folgenden Bereichen:

- Bezüglich der Datenhaltung läßt sich feststellen, daß Multiuserfähigkeit nicht selbstverständlich ist (alle evaluierten objektorientierten PC-Werkzeuge waren nicht multiuserfähig), und die Eignung für größere Entwicklergruppen (> 20 Personen) noch seltener anzutreffen ist. Gut schneiden in diesem Zusammenhang Systeme in Client-Server-Architektur ab, deren Datenhaltung sich auf einem Datenbankserver mit

schnellem Dateisystem befindet (etwa als Netware Loadable Module auf einem Novell-Server): Hier sind bei einem leistungsfähigen PC als Server auch Entwicklerteams von bis 50 Entwicklern pro Server praktikabel. Ansonsten ist die Performance - auch auf leistungsfähigster Hardware - häufig unterdurchschnittlich bis schlecht.

- Aus dem bisher dargestellten ergibt sich, daß für große bis sehr große, nicht zerlegbare Entwicklungen PC-Tools ausscheiden und daß evtl. der Einsatz anderer Werkzeuge in den späten Phasen notwendig ist, was entsprechende Schnittstellenprobleme schafft.
- Qualität und Funktionalität objektorientierter PC-Werkzeuge war sehr unbefriedigend

2.2. UNIX-Werkzeuge

Stärken und Schwächen bei den CASE-Tools unter UNIX liegen in den folgenden Bereichen:

- Vor allen Dingen objektorientierte UNIX-Werkzeuge bauen ihre Datenhaltung auf relationalen Datenbanksystemen auf. Die Vorteile dieser Architektur (Datensicherheit, Datenschutz, einfacher Zugriff auf das Repository über SQL, etwa für die Erstellung von Dokumentationen) überwiegen die Nachteile (Performance, Speicherbedarf, evtl. Kosten) eindeutig. Pluspunkte können Werkzeuge dieses Segments typischerweise außerdem durch ihre Stabilität, breite Funktionsabdeckung und Offenheit in bezug auf eine Erweiterung durch Fremdwerkzeuge sammeln. Nicht zu vernachlässigen sind schließlich die weitgehenden Verteilungsmöglichkeiten im Netz, die eine differenzierte Lastverteilung gestatten und große Flexibilität bei sich verändernden Anforderungen bieten. Die Möglichkeit, floatende Lizenzen bis auf Teilfunktionsebene nutzen zu können, erlaubt eine beliebige netzweite Nutzung mit vergleichsweise weniger erworbenen Lizenzen. Alle im Rahmen dieser Studie als überdurchschnittlich beurteilten objektorientierten Werkzeuge liefen unter UNIX ab.
- Schwächen bestehen im Bereich der Bedienung, wo oftmals selbst einfache Arbeitserleichterungen wie Auswahllisten beim Öffnen von Pro-

jekten oder Teilmodellen nicht angeboten werden. Außerdem sind die Preise der CASE-Tools unter UNIX vergleichsweise hoch. Bedingt durch die Architektur der Systeme (Ablauf in einer vernetzten, verteilten und grafischen Umgebung unter UNIX, Datenhaltung in einem relationalen Datenbanksystem) ist die Installation auch bei guten Systemkenntnissen aufwendig und komplex. Es sollte daher erwogen werden, diese vom Anbieter durchführen zu lassen. Es ist in der Regel keine Entwicklung für andere Plattformen und Betriebssysteme möglich. Die meisten konventionellen Werkzeuge basieren auf proprietären Datenhaltungssystemen.

2.3. Werkzeuge für heterogene Umgebungen

Als heterogen sollen hier Werkzeuge bezeichnet werden, an deren Ausführung mindestens zwei, unter unterschiedlichen Betriebssystemen laufende Rechner beteiligt sind. Zielgruppe von Werkzeugen für heterogene Umgebungen sind eindeutig Großanwender, die das Bedürfnis haben, eine Softwareentwicklung in und für unterschiedliche Hardwareplattformen in größeren Entwicklungsteams durchzuführen. Dementsprechend sind die anzutreffenden Programme in diesem Segment durch leistungsfähige Datenhaltungskomponenten, Projektmanagementkomponenten, Anpaßbarkeit an individuelle Bedürfnisse und die Offenheit für die Integration von Fremdprodukten gekennzeichnet. Allerdings existieren auch bei Werkzeugen, die ausdrücklich den Anspruch erheben, auf die dargestellte Entwicklungssituation zugeschnitten zu sein, teilweise erschreckende Defizite in diesen Bereichen.

Eine konzeptionelle Schwäche besteht außerdem darin, daß bei den PC-Host-Systemen, die nicht in Client-Server-Architektur realisiert sind, meist immer noch keine Online-Datenhaltung erfolgt und daher aufwendige und in ihrer Funktion letztlich immer unbefriedigend bleibende Up- und Downloadvorgänge erforderlich sind. Aus der komplexen Systemumgebung entstehen potentielle Hard- und Softwareprobleme. Schließlich sind auch die Werkzeuge für heterogene Umgebungen vergleichsweise teuer.

3. Tendenzaussagen bei einer Gliederung nach der Entwicklungsmethodik

3.1. Konventionelle Werkzeuge

Konventionelle CASE-Werkzeuge befinden sich typischerweise bereits erheblich längere Zeit am Markt als die nun neu hinzukommenden Vertreter, die die objektorientierte Systementwicklung unterstützen. Mit dieser Tatsache geht eine Ausreifung des Marktes einher, die sich folgendermaßen äußert:

- Standardmethoden und -beschreibungsmittel (ERM, SA/SD usw.), die den Wechsel des Werkzeugs durch den geringeren Einarbeitungsaufwand und bessere Austauschmöglichkeiten zwischen den Werkzeugen erleichtern. Dies ermöglicht auch die Kopplung mehrerer Werkzeuge.
- Stabilität und Ausgereiftheit
- Breite Funktionsabdeckung
- Breite Palette von Entwicklungs- und Ablaufumgebungen

Bedingt durch die Methodenbrüche der zu Grunde liegenden Methoden sind konventionelle Werkzeuge hingegen in der Durchgängigkeit des Entwicklungsprozesses tatsächlich bereits heute weniger stark.

3.2. Objektorientierte Werkzeuge

Als derzeit dominierendes Phänomen im Markt für objektorientierte CASE-Werkzeuge läßt sich die Dynamik und Unreife des Marktes feststellen: Es existiert im Gegensatz zum konventionellen Bereich im Moment kein Methodenstandard; nimmt man allein als Maßstab, wie viele Werkzeuge sich an den einzelnen Methoden orientieren, läßt sich eine Dominanz von Werkzeugen feststellen, die die Object Modelling Technique nach Rumbaugh et. al. unterstützen. Drei der evaluierten Programme arbeiten nach dieser Methode, jeweils eines mit OOA/OOD nach Coad/Yourdon und eines mit OOSA nach Shlaer/Mellor. Die Unreife wird weiterhin an den kurzen Zeiträumen deutlich, in denen neue Versionen von Werkzeugen auf den Markt kommen; so ist es durchaus nicht unüblich,

daß allein für das Jahr 1994 drei neue Versionen angekündigt sind.

Zeigt der vorgenannte Punkt, daß Hersteller teilweise noch mitten in der Entwicklung ihrer Programme stecken und somit in kurzem Abstand für den Kunden jeweils erheblich verbesserte Versionen verfügbar sind, ist ein anderer Aspekt dieser Unreife für Anwender erheblich ungünstiger: Es treten nämlich nicht nur in schneller Folge neue Hersteller und Vertrieber in diesem Marktsegment auf, es ist derzeit auch noch nicht auszuschließen, daß die Weiterentwicklung oder der Vertrieb eines Produktes durch Hersteller bzw. Vertrieber eingestellt werden, weil es sich als zu unausgereift und leistungsschwach erwiesen hat. Wenn man bedenkt, welche Kosten mit Auswahl und Einführung von CASE-Werkzeugen verbunden sind und welche Migrationskosten zu einem anderen objektorientierten Werkzeug sich nach einigen durchgeführten Projekten ergeben, läßt sich ermessen, was ein derartiger Entschluß für die betroffenen Anwender bedeutet. Es ist daher für Anwender unbedingt notwendig, die derzeitige und zukünftige Leistungsfähigkeit des in Aussicht genommenen Herstellers oder Vertriebers und seiner Produkte besonders genau zu untersuchen, wenn mehr als nur Pilotprojekte durchgeführt werden sollen.

Diese kritische Durchleuchtung ist für die Produkte auch noch aus einem anderen Grund heraus dringend angeraten: Die objektorientierten CASE-Produkte fallen im Vergleich zu den konventionellen Vertretern ihrer Art durch eine deutlich breitere Streuung in Funktionalität und Qualität auf; neben reinen Zeichenwerkzeugen sind bereits ausgesprochen leistungsfähige Programme zu finden; es gibt sowohl Produkte, die sich noch im „Experimentierstadium“ befinden, als auch ausgereifte Werkzeuge. Die Situation ist damit ähnlich der, die im konventionellen Bereich zu Anfang der 80er Jahre geherrscht hat. Im Rahmen dieser Studie konnte kein Werkzeug gefunden werden, das zum jetzigen Zeitpunkt auch „Randbereiche“ wie Projektmanagement, Konfigurationsmanagement usw. zufriedenstellend abdeckt. Grundsätzlich sollte man bei Auswahl und Einführung objektorientierter Werkzeuge keine gewünschte Funktion als selbstverständlich voraussetzen: Selbst elementare Dinge wie Teamfähigkeit sind häufig nicht realisiert.

Objektorientierte Vorgehensweisen erheben den Anspruch, die Wiederverwendung besonders gut zu unterstützen. Diese Unterstützung beschränkt sich bei allen untersuchten Systemen auf die Vererbung. Darüber hinausgehende Wiederverwendungskonzepte, die für geplante, unterneh-

mensweite Wiederverwendung mit Sicherheit unentbehrlich sind, wurden nicht vorgefunden: Es existieren keinerlei Funktionen, die das Ablegen und Auffinden von Klassen oder von Analyse- und Designergebnissen systematisch unterstützen.

Die oft angesprochene höhere Durchgängigkeit des Systementwicklungsprozesses ist tatsächlich eine Stärke objektorientierter Systementwicklung. Iterative Vorgehensweisen werden dadurch praktikabler und Werkzeuge leisten hier einen sinnvollen Beitrag.

Die besten objektorientierten CASE-Werkzeuge bieten bereits heute eine sehr gute Anbindung an Datenbanken und C++-Compiler. Die Anbindung an Oberflächen läßt hingegen bedingt durch die mangelnde Standardisierung und Verfügbarkeit zu wünschen übrig, da viele vormals nicht objektorientierte Werkzeuge aus dem Datenbankumfeld erst jetzt in Richtung Objektorientierung weiterentwickelt werden.

Insgesamt läßt sich sagen, daß bereits zum heutigen Zeitpunkt Werkzeuge verfügbar sind, die für Pilotprojekte oder einzelne phasenspezifische oder phasenübergreifende Aktivitäten objektorientierter Systementwicklung (etwa die Dokumentation) vollauf geeignet sind. Vor einem entsprechenden Einsatz sollte man sich der im Moment bei allen Werkzeugen anzutreffenden Schwächen bewußt sein und prüfen, ob diese akzeptabel sind. Mit für den gesamten Entwicklungsprozeß tragfähigen Lösungen, die einen vollständigen Ersatz für konventionelle CASE-Umgebungen darstellen, ist nicht vor Ende 1994 zu rechnen.