ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1992
Seria: MECHANIKA z. 109 Nr kol. 1171

International Conference on
COMPUTER INTEGRATED MANUFACTURING
Internationale Konferenz über
RECHNERINTEGRIERTE FERTIGUNGSSYSTEME
Zakopane, March 24-27 1992

Bernd SCHÜTZE

Institut für Konstruktions- und Tribotechnik
Technische Universität Otto von Guericke Magdeburg

WAS KANN DIE OBJEKTORIENTIERTE SOFTWAREENTWICKLUNG FÜR CAD BRINGEN ?

Zusammenfassung

An Hand eines CAD-Konzepts zur Gestaltung und Berechnung von Wellen werden die Vorzüge der objektorientierten Softwareentwicklung erläutert und Schlußfolgerungen für die zukünftige Anwendung der objektorientierten Denk- und Arbeitsweise gezogen.

Einleitung

In Bild 1 sind die Komponenten zusammengestellt, die bei der Entwicklung moderner CAD-Konzepte zu berücksichtigen sind. Obwohl das Bild 1 keinen Anspruch auf Vollständigkeit erhebt, können daraus zwei Schlußfolgerungen gezogen werden:

- In die Entwicklung von CAD-Systemen müssen wissenschaftliche Erkenntnisse aus unterschiedlichen Wissenschaftszweigen (Informatik, Konstruktionstechnik etc.) einfließen.
- Innerhalb jeder Komponente gibt es eine kaum noch überschaubare Vielzahl von Varianten.

Die Konstruktionswissenschaft kann in der Regel nur zu den drei unteren Komponenten einen Beitrag leisten. Für die Entwicklung moderner CAD-Konzepte ist es allerdings wichtig, neue Trends bei den in Bild 1 oben (gestrichelt) dargestellten, vor allem von der Informatik beeinflußten Komponenten zu erkennen. Eine solche Schlüsselrolle spielt zweifellos die objektorientierte Softwareentwicklung (/1/ und Bild 11). Die Vorteile dieser qualitativ neuen Methode sieht der Autor in

- der Überwindung entscheidender Mängel bei der Implemen-

tierung von CAD-Konzepten, die nach der Methode der *strukturierten Programmierung* entwickelt wurden.
- der Einführung der objektorientierten Denk- und Arbeitsweise bei der Entwicklung genauerer Modelle des konstruktiven Entwicklungsprozesses (KEP).

Die dabei am Institut für Konstruktions- und Tribotechnik gesammelten ersten Erfahrungen /2/ sollen im folgenden dargestellt werden.

1. Allgemeines Ziel der CAD-Forschung

Bild 2 zeigt das Magdeburger Modell des KEP /3/. Charakteristisch ist das Generieren des Endergebnisses (Konstruktionsdokumentation) über mehrere Zwischenergebnisse unterschiedlicher Abstraktionsstufe, wobei die Informationsgewinnung durch Operationen realisiert wird und umfangreiche Dateien einbezogen werden.
Das Ziel der CAD-Forschung zahlreicher Institutionen besteht nun darin, diesen Informationsverarbeitungsprozeß, der heuristische und determinierte Abschnitte besitzt, auf dem Rechner zu modellieren (z.B./4/,/5/,/3/).

2. Gegenwärtige Praxis

Das Bild 3 zeigt den Modellierungsprozeß, wie er gegenwärtig meist im Computer realisiert wird:
- Das Wissen zum Generieren und Manipulieren aller notwendigen Daten wird in Form von Algorithmen (Routinen) repräsentiert.
- Die Daten werden in einem strukturierten Produktmodell abgelegt.

Charakteristisch für dieses Vorgehen nach der Methode der strukturierten Programmierung sind
- die "starke" Trennung von Routinen und Daten sowie
- das Primat der Routinen gegenüber den Daten.

Diese Vorgehensweise wird am Beispiel eines CAD-Konzepts zum Gestalten und Berechnen von Wellen erläutert.
In Bild 4 ist der Ausgangspunkt und Lösungsweg deutlich erkennbar: die Hauptfunktion (Top), von der aus durch fortschreitende Dekompensation schließlich die erforderlichen Teilfunktionen, die Grundlage für die Modulbildung sind, abgeleitet werden.
Dieses Vorgehen wird üblicherweise als top-down-Entwurf bezeichnet.
Die funktionsorientierte Vorgehensweise durch Dekompensation

ist gut vergleichbar mit dem Erarbeiten der Funktionsstruktur im KEP (Bild 2). Für die Teilfunktion

Welle grob gestalten

liegt der Schwerpunkt auf dem Prädikat 'grob gestalten'. Die Dekompensation in Teilfunktionen führt zu stark reihenfolgeorientierten Konzepten.
Parallel, aber weitgehend unabhängig, werden die Daten nach der gleichen Methode strukturiert, Bild 5.
Mit der Realisierungsphase beginnt vielfach der "Leidensweg" des Entwicklungsteams. Die arbeitsteilige Implementierung der einzelnen Moduln M_i und deren Zugriff auf die einzelnen Dateien/Records D_i (Bild 6) führt zu den gefürchteten Seiteneffekten (unkontrollierte Datenänderungen), macht die Moduln überaus "empfindlich" gegen Datenstrukturveränderungen während der Realisierungsphase etc. - eine nur schwer beherrschbare Lösungsstrategie. Das gilt vor allem für den CAD-Bereich, für den unscharfes Wissen charakteristisch ist.
Es soll nun geprüft werden, inwieweit die objektorientierte Methode hier Abhilfe schaffen kann.

3. Objektorientierter Denk- und Arbeitsstil

Eine umfassende Darstellung findet man in /1/. Wodurch zeichnet sich objektorientierte Softwareentwicklung aus ?

- Es wird eine starke Bindung der Routinen an die Daten angestrebt. Ein Objekt - oder genauer gesagt: eine Klasse - enthält die Daten *und* die Routinen, mit denen diese Daten manipuliert werden. Dieses Prinzip heißt *Kapselung* und verfolgt das Ziel, daß die Daten und die darauf anwendbaren Operationen zu einer festen Einheit zusammengefügt werden. Ein Programmierer, der als Kunde eine Klasse nutzt, muß sich nicht mehr um Datenzugriffe oder die Reihenfolge der Routinen kümmern, sondern nimmt diese "Dienste", die eine Klasse zur Verfügung stellt, in Anspruch (sogenannter "Einkaufszettelansatz").
- Bild 7 zeigt exemplarisch einige Objekte, wie sie in 2D-CAD-Systemen benötigt werden. Um Objekte zu finden, wird von den existierenden Objekten der realen und abstrakten Welt ausgegangen, also Dreiecke, Kreise etc., in der Konstruktion Wirkflächen, Wirkräume etc. Aus der Darstellung geht auch das 2.Merkmal der objektorientierten Softwareentwicklung hervor: die *Vererbung*.

Ähnlich der Klassifizierung z.B.in der Zoologie (Bild 8) wird von den tatsächlich existierenden Objekten der realen Welt ausgegangen, um allgemeinere, häufig abstrakte Klassen zu finden. Diese Hierarchiebildung ermöglicht den Effekt der Vererbung. Die Vererbung basiert auf der "ist-ein(e)"-Relation, d.h. der Nachkomme hat mindestens ein Merkmal mehr als der Vorfahre. Dadurch ist eine weitere Spezialisierung (z.B. das Anfügen der Klasse 'Quadrat') und/oder eine Erweiterung (z.B. 'offene Figur', 'Segment') im nachhinein problemlos möglich, falls die Klassifizierung wissenschaftlich begründet ist. Es gilt die Regel:

Jeder Nachkomme erbt alle (öffentlichen) Merkmale des Vorfahren, d.h. alle Attribute und Routinen.

Am Beispiel eines einfachen Grafikpakets wird das in Bild 9 deutlich: Alle Attribute, die 'Punkt' besitzt, besitzt auch 'komplexes Objekt'. Ebenso verhält es sich mit den Routinen.

Eine Vererbungsrelation 1:n (wie in Bild 9) heißt Einfachvererbung, m:n dagegen Mehrfachvererbung (wie in Bild 10.3).

- Im Bild 9 wird auch der dritte mächtige Pfeiler der objektorientierten Softwareentwicklung deutlich, der *Polymorphismus* (viele Ausdrucksformen für die gleiche Sache). Z.B. hat die Routine 'zeige' in den Klassen 'Punkt' und 'komplexes Objekt' die gleiche Bedeutung (Semantik), nämlich die Darstellung auf dem Bildschirm. Die bei der Programmierung benutzten Routinen für Punkt, Linie, Kreisbogen sind jedoch unterschiedlich. Erst zur Laufzeit wird über das konkret gewählte Objekt festgestellt, welche Routine tatsächlich gestartet werden muß.

Mit den drei Säulen

- Kapselung
- Vererbung
- Polymorphismus

stehen neue Werkzeuge zur Verfügung, die in Verbindung mit dem Klassenbegriff eine gegenüber der strukturierten Programmierung völlig andere Denk- und Arbeitsweise erfordern und ermöglichen.

4. Beispiel 'Welle'

Näturlich ist das zu lösende Problem - nämlich die Modellierung des KEP im Computer - prinzipiell nicht leichter geworden als bei funktionsorientiertem Vorgehen:

- Finden von Klassen
- Beschreiben der Klassen als abstrakten Datentyp
- Finden einer geeigneten Klassenhierarchie durch Feststellen von Gemeinsamkeiten und Beziehungen zum Zwecke der Vererbung und zur Nutzung des Polymorphismus.

Allerdings sind die zu erwartenden Effekte beachtlich:
Die Wiederverwendbarkeit, Erweiterbarkeit und Kompatibilität werden qualitativ in viel höherem Maße befördert und gleichzeitig der Mangel an Flexibilität sowie zentralisierte Strukturen (Hauptprogramm) eingeschränkt.
In /2/ wurde das in Pkt.2. erläuterte Beispiel auf *objektorientierter* Grundlage teilweise implementiert (Turbo-Pascal 6.0), und zwar die 2D-Darstellung und einfache Berechnungen. Das Finden und Strukturieren der Klassen war dennoch schwierig. Die Bilder 10.1,10.2 zeigen zwei Varianten für Einfachvererbung, das Bild 10.3 eine Variante mit Mehrfachvererbung. Vor allem die ungewohnte Denkweise erwies sich als besonderes Hemmnis. Unbedingt anzuraten -aber das ist keine grundsätzlich neue Erkenntnis - ist die exakte Analyse des Problemfeldes, insbesondere

- die Zusammenstellung aller vermeintlich benötigter Daten
- die Bildung von Klassen unter Berücksichtigung der Vererbung
- die Beschreibung als abstrakten Datentyp.

Vor allem der letzte Anstrich erfordert tiefgründige Überlegungen und eine möglichst mathematische Beschreibung jedes Objekts entsprechend den Bestandteilen eines abstrakten Datentyps:

```
TYPEN
FUNKTIONEN
  . Erzeugen der Daten  ----->  Konstruktorfunktion
  . Ändern der Daten    ----->  Prozeduren
  . Zugriff auf Daten   ----->  Funktionen
VOR- UND NACHBEDINGUNGEN        |
  . Grenzen der Anwendbarkeit   | Semantik-
AXIOME                          | beschreibung
  . Regeln für Anwendung        |
```

Für praktische Empfehlungen diesbezüglich sind die vorliegenden Erfahrungen noch nicht ausreichend.

5. Schlußfolgerungen

In zwei Richtungen kann objektorientierte Softwareentwicklung für CAD-Forschung von Nutzen sein:

1. Für die Entwicklung durchgängiger CAD-Konzepte, die über

mehrere Jahre in Stufen bei zu erwartendem Wechsel von Teammitarbeitern implementiert werden, bietet der objektorientierte Ansatz ausreichend Gewähr für die Sicherstellung von Korrektheit, Robustheit, Erweiterbarkeit, Wiederverwendbarkeit und Kompatibilität der Software /1/.
Dabei trägt die Nutzung von Klassen-Bibliotheken (z.B. zur Realisierung der Benutzerschnittstelle) nicht nur zu einer effektiveren Implementierung bei (Was da ist, muß nicht noch einmal erfunden werden!), sondern kann bei zu erwartender steigender Anwenderzahl (Bild 11) bereits in naher Zukunft auch wesentliche Komponenten für CAD-Systeme (Grafik, Geometrie) enthalten.
Selbstverständlich muß eine objektorientierte Sprache gewählt werden, mit der vorhandene ingenieurtechnische Programme, die in anderen Sprachen implementiert sind, integriert werden können.

2. Objektorientierte Denk- und Arbeitsweise heißt, Klassen der realen und abstrakten Welt zu finden und sie als abstrakten Datentyp zu beschreiben:
Die enge Kopplung von

   Daten und Operationen unter Berücksichtigung ingenieurtechnischer Regeln und Bedingungen

kann nach Ansicht des Autors die weitere konstruktionswissenschaftliche Begriffshierarchie unter Vererbungsaspekten ebenso befruchten wie die stärkere Integration von Wissen im Sinne der Künstlichen Intelligenz.

Literaturverzeichnis:

/1/ Meyer,B.: Objektorientierte Softwareentwicklung, Carl Hanser Verlag München Wien 1990

/2/ Senge,A.: Objektorientierter Entwurf eines CAD-Systems Diplomarbeit TU Magdeburg/IKT 1992

/3/ Schütze,B.;Ratfisch,U.; Beitz,W.: Die "Kontaktmethode"- eine Methode zur Vereinfachung rechnerunterstützter Gestaltvariationen durch technisches Anordnen, Konstruktion 44(1992)2 S.43-50

/4/ Beitz,W.: Konstruktionsleitsystem als Integrationshilfe. VDI-Bericht Nr.812. Düsseldorf: VDI-Verlag 1990. S.181-201

/5/ Pahl,G.: Konstruieren mit 3D-CAD Systemen, Springer-Verlag Berlin... 1990

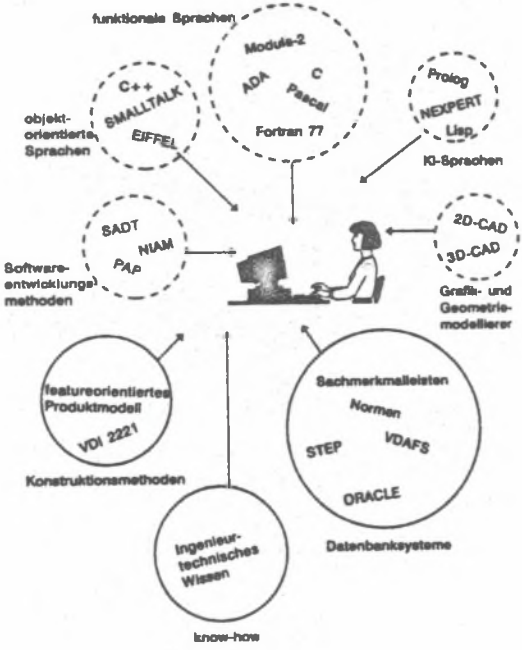

Bild 1: Komponenten für moderne CAD- Konzepte

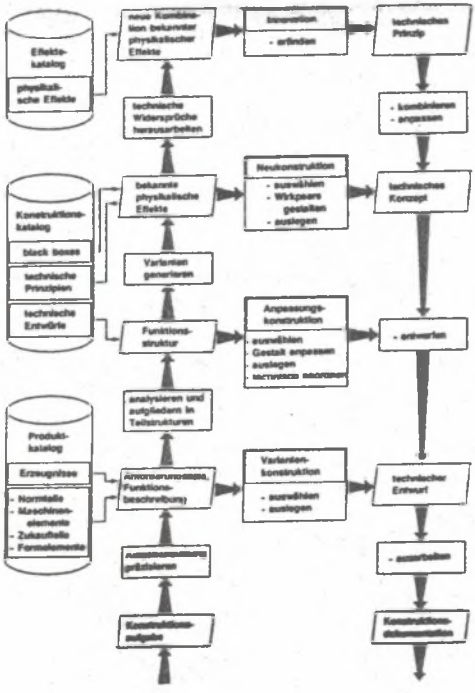

Bild 2: Magdeburger Modell des konstruktiven Entwicklungs-prozesses

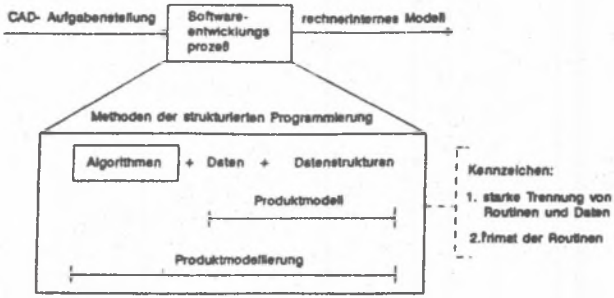

Bild 3 Charakteristik der Methode der strukturierten Programmierung

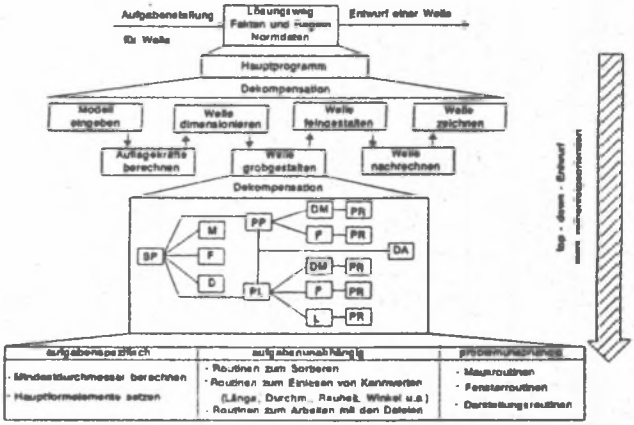

Bild 4: Funktionsorientierte Softwareentwicklung

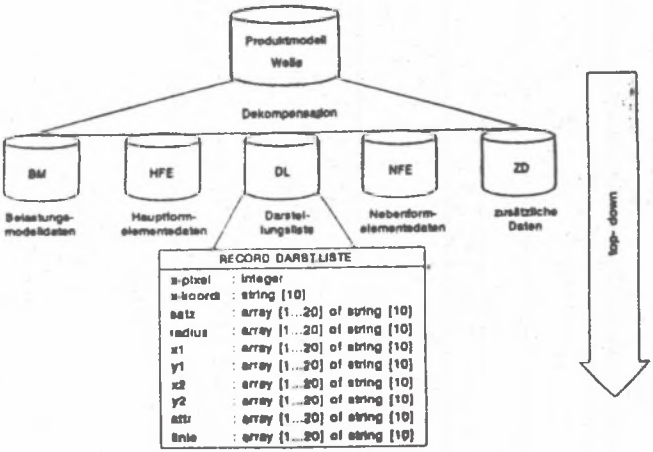

Bild 5: Dekompensation der Daten

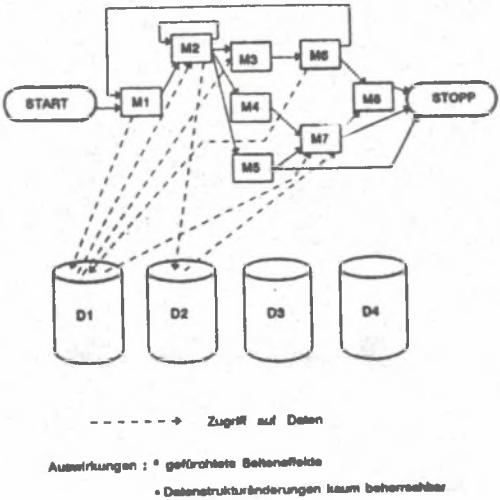

Bild 6: Probleme bei der arbeitsteiligen Implementierung

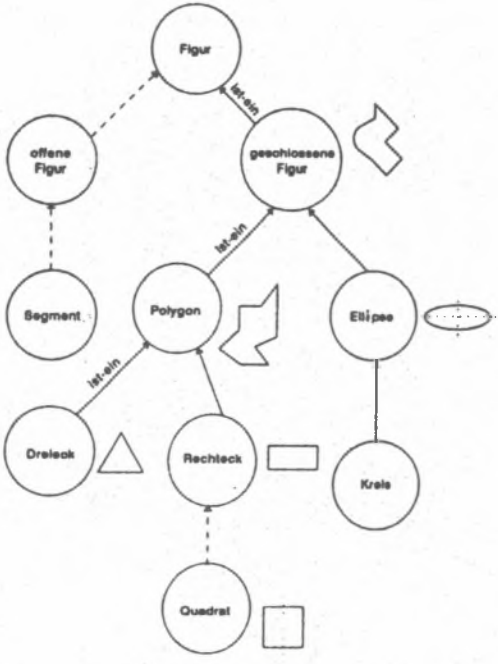

Bild 7: Klassen für 2D-Objekte

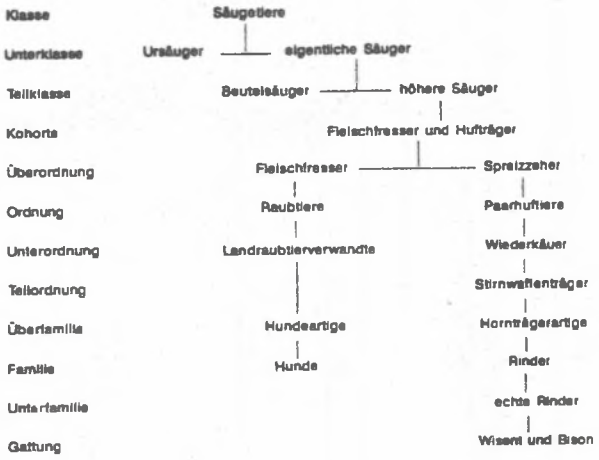

Bild 8: Klassifizierung in der Zoologie

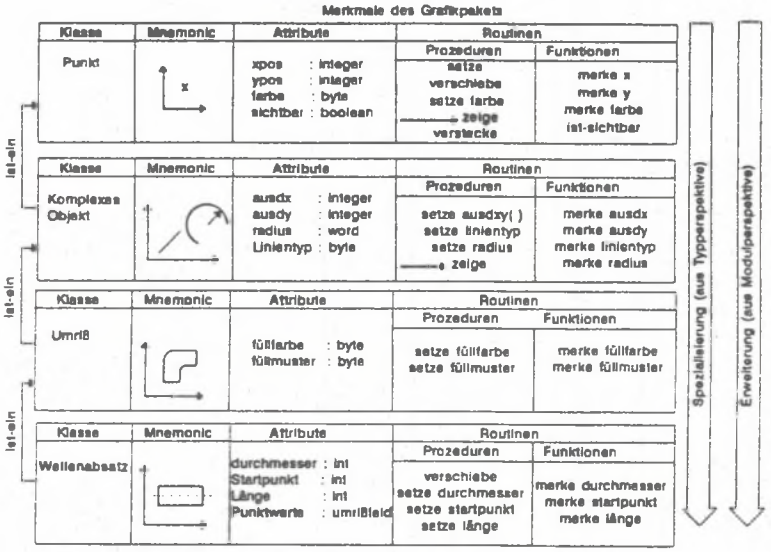

Bild 9: Vererbung und Polymorphismus

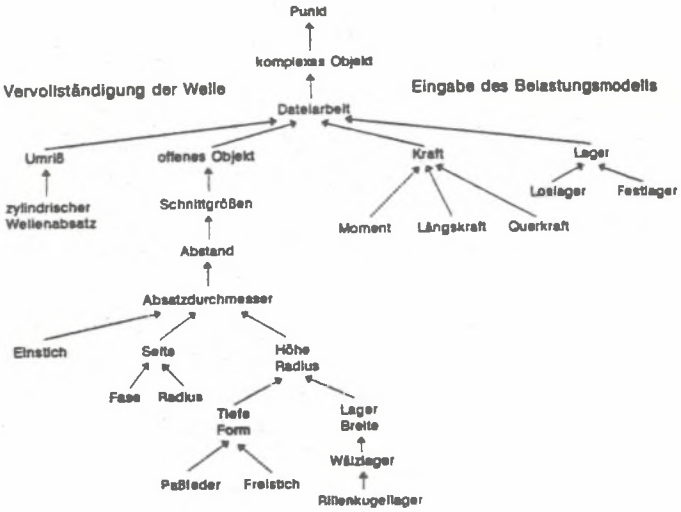

Bild 10.1: 1. Variante für Klassenbildung 'Welle'

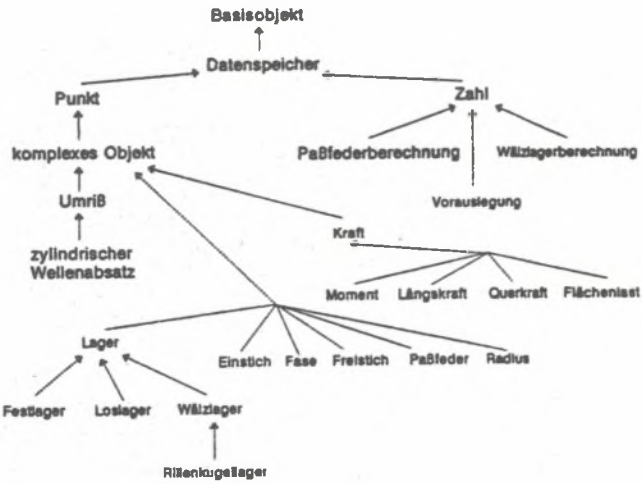

Bild 10.2: 2. Variante für Klassenbildung 'Welle'

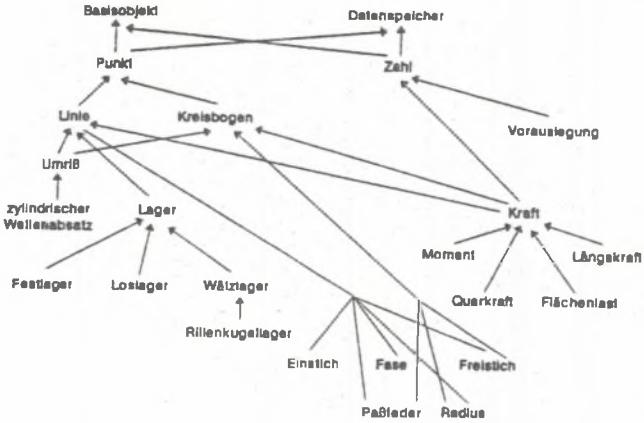

Bild 10.3: 3. Variante für Klassenbildung 'Welle'

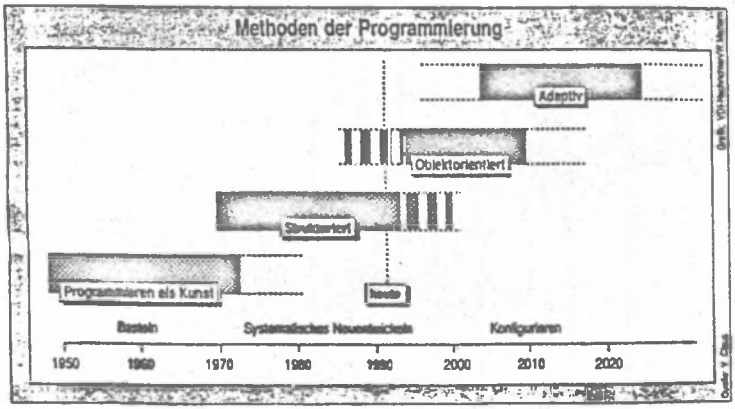

Bild 11: Trends für Methoden der Programmierung
(aus VDI-Nachrichten)

JAKIE MA ZNACZENIE ROZWÓJ OBIEKTOWO ZORIENTOWANEGO OPROGRAMOWANIA DLA CAD

Streszczenie

W pracy wyjaśniono rozwój oprogramowania obiektowo zorientowanego służącego do określania postaci konstrukcyjnej i obliczeń wałków. Przedstawiono wnioski przyszłych zastosowań obiektowo zorientowanych sposobów myślenia i pracy.

HOW IMPORTANT ARE OBJECT ORIENTED SYSTEMS FOR CAD

Summary

In the paper was presented the development of object-oriented systems for description the construction and calculation of shafts. Conclussions for future using the object-oriented way of thinking and working were presented.

Wpłynęło do redakcji w marcu 1992r. Recenzent:Ryszard Knosala