

Der graphische Wissenseditor ZOO: Ein Interface zu objektorientierten Wissensbasen

Wolf-Fritz Riekert

Durchschaubarkeit und Adaptierbarkeit sind wichtige Kriterien für die Benutzerfreundlichkeit von Computersystemen. Diese Kriterien lassen sich nur schwer oder gar nicht erfüllen, wenn in komplexen Problembereichen traditionelle Softwaretechniken angewandt werden. Ein objektorientierter Systementwurf hingegen ermöglicht die Realisierung von Metasystemen, die den Benutzer befähigen, die Konzepte zu verstehen und zu verändern, die dem von ihm benutzten Anwendungssystem zugrunde liegen.

ZOO¹⁾ ist ein Metasystem für den Anwendungsexperten. ZOO visualisiert den Inhalt einer objektorientierten Wissensbasis in einer zweidimensionalen Darstellung. Dabei erscheint die Wissensbasis als ein Netz von graphischen Symbolen. Durch direkte Manipulation dieser graphischen Objekte ist es möglich, die Komponenten einer Wissensbasis zu untersuchen, zu verändern, zu erzeugen und zu vernichten.

Inhaltsübersicht

Einführung
Objektorientierte Darstellung von Wissen
Graphische Wissensdarstellung im System ZOO
Das Metasystem ZOO
Zusammenfassung
Literatur

Einführung

Computer werden zunehmend in Problembereichen eingesetzt, die sich schwer strukturieren lassen, beispielsweise in der Bürowelt, bei Planungsaufgaben, bei Designproblemen, in der Diagnose oder in der Konstruktion. Zwei Merkmale sind allen diesen Arbeitsbereichen gemeinsam: Es fehlt die formale Beschreibung des Problemgebiets, und es gibt keine wohldefinierten Handlungsziele. Es ist schwer oder gar unmöglich, eindeutig definierte Lösungsmethoden für die Problemstellungen in diesen Aufgabengebieten zu finden. Die Anwendung von Computersoftware in diesen Gebieten bereitet daher dem Benutzer häufig große Unannehmlichkeiten:

- Da die Ziele und Lösungswege nicht eindeutig sind, fällt es dem Benutzer schwer, die vom Computer gefundenen Lösungen zu akzeptieren. Die einer Computerentscheidung zugrundeliegenden Kriterien bleiben dem Benutzer meist verborgen.

1) Der Autor gehörte der Forschungsgruppe INFORM an der Universität Stuttgart an, als er das ZOO-System entwickelte. Diese Forschungsgruppe wurde im Rahmen des Verbundforschungsprojekts WISDOM vom Bundesminister für Forschung und Technologie (BMFT) und von deutschen Industriefirmen unterstützt. Die Entwicklung des ZOO-Systems wurde von der Firma TA Triumph Adler AG, Nürnberg, unterstützt. Der Name ZOO steht für „ZOO ordnet Objekte“.

- Da eine vollständige Problemspezifikation fehlt, ist jedes System zur Unvollständigkeit verurteilt. Der Benutzer erkennt sehr bald die Mängel des Systems, kann sie aber nicht beheben, weil dies besondere Programmierkenntnisse erfordert.

Es gibt also zwei wesentliche Qualitätsmerkmale für komplexe Softwaresysteme: die Durchschaubarkeit und die Adaptierbarkeit des Systems. Herkömmlich aufgebaute Systeme besitzen diese Vorzüge häufig nicht, vor allem wenn das Einsatzgebiet sehr komplex ist. Der Grund für diesen Mangel ist die Art, in der das Wissen in herkömmlichen Systemen repräsentiert ist: Weil das Wissen in einer Sammlung von Prozeduren und Datenstrukturen verschlüsselt ist, ist es nur für den Ablauf des Anwendungssystems nutzbar. Für alle anderen Zwecke ist Programmcode eine äußerst ungeeignete Wissensquelle. Weder lassen sich aus dem Programmcode Benutzeranleitungen, Erklärungen oder Hilfeinformation ableiten, noch unterstützt ein herkömmliches Programm seine eigene Erweiterung oder Veränderung.

Objektorientierte Anwendungssysteme eröffnen einen Ausweg aus diesem Problem. Das Verhalten eines solchen Systems leitet sich aus einer Wissensbasis ab, die aus Objekten besteht. Es gibt keine Beschränkung für die Interpretation des Wissens, das in Objekten repräsentiert ist. Daher ist es möglich, Systemkomponenten zu entwerfen, die den Inhalt einer objektorientierten Wissensbasis untersuchen und verändern können. Solche Systemkomponenten werden als Metasysteme bezeichnet, denn sie operieren auf einer höheren Abstraktionsebene als das zugrundeliegende Anwendungssystem. Metasysteme können das Verhalten eines Anwendungssystems erklären und steuern, da sie auf dessen Wissensquellen Zugriff haben.

Das Metasystem ZOO (siehe Abb. 1) dient zweierlei Zwecken. Zum einen ist es ein Instrument zur Untersuchung von Wissensbasen in einer zweidimensionalen graphischen Darstellung. Zum anderen ist es ein Werkzeug zur Bearbeitung von Wissensbasen mit Hilfe direkter Manipulation von Bildschirmobjekten. ZOO erscheint für seinen Benutzer ähnlich einfach wie ein Editor zur Erstellung von Bürographiken, seine Funktionalität ist jedoch weitaus größer. In jedem dargestellten graphischen Objekt wird eine Komponente einer systeminternen Wissensbasis sichtbar. Mit jedem vom Benutzer neu erzeugten graphischen Objekt wird auch ein Objekt der Wissensbasis angelegt.

Weil gleichzeitig mit der Graphik eine Wissensbasis entsteht, sind die Vorgänge des Designs, der Implementation und der Dokumentation eines wissensbasierten Systems parallelisiert. Die erstellten graphischen Spezifikationen stellen zugleich unmittelbar ablauffähige Softwaremodule dar. Das System ZOO kann deshalb als Werkzeug zum Rapid Prototyping angesehen werden.

ZOO läuft auf einer VAX 11/780 unter Berkeley UNIX 4.2 und benötigt ein BitGraph-Rastergraphikterminal der Firma BBN. ZOO verwendet das Fenstersystem WLISP [4] und die objektorientierte Sprache ObjTalk [12], die beide ebenso wie ZOO selbst in der Forschungsgruppe INFORM an der Universität Stuttgart entstanden sind. Die Implementationssprache für alle Softwarekomponenten ist Franz Lisp [5].

Objektorientierte Darstellung von Wissen

In den Softwaresystemen der Forschungsgruppe INFORM wird ObjTalk, eine objektorientierte Sprache, zur Repräsentation von Wissen verwendet. ObjTalk ist eine Sprache, die das „Actor“-Konzept und das Prinzip des „Message Passing“ von Hewitt [7]

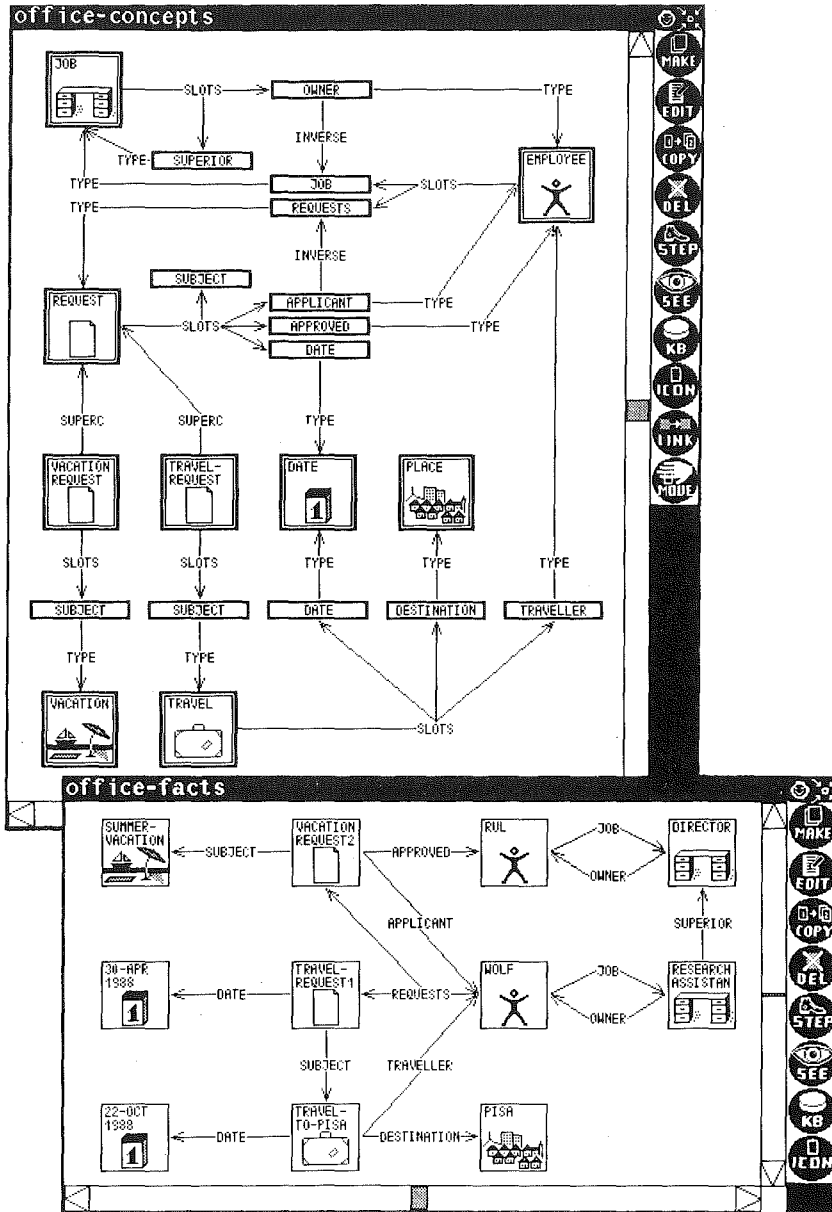


Abb. 1 Das Metasystem ZOO: Sichtbar sind die graphischen Darstellungen einer Metawissensbasis „office-concepts“ und einer Faktenwissensbasis „office-facts“, die einem Büroinformationssystem zugrunde liegen.

sowie die Frame-Techniken Minskys [11] in einer ähnlichen Art miteinander vereinigt wie die objektorientierten Systeme KL-ONE [2] und LOOPS [1].

ObjTalk ermöglicht gleichermaßen die Repräsentation von Faktenwissen über konkrete Sachverhalte wie die Repräsentation von Metawissen [3] über abstrakte Konzepte:

- Konkrete Sachverhalte aus der realen Welt werden in Faktenwissensbasen dargestellt, die aus ObjTalk-Objekten bestehen. ObjTalk-Objekte besitzen Slots, in denen Eigenschaften und Relationen zu anderen Objekten gespeichert werden können. ObjTalk ermöglicht die Klassifizierung von Objekten: Jedes Objekt gehört zu einer Klasse, als deren Instanz es bezeichnet wird.
- Abstrakte Konzepte werden in konzeptuellen Wissensbasen dargestellt. Diese bestehen aus einer besonderen Art von ObjTalk-Objekten, den sogenannten konzeptuellen Objekten. Zu den wichtigsten konzeptuellen Objekten zählen die Klassen; dies sind Objekte, die die gemeinsamen Eigenschaften ihrer Instanzen repräsentieren. Alle Klassen bilden eine Spezialisierungshierarchie. Klassen verfügen über Slotbeschreibungen, Regeln und Methoden, die ebenfalls wieder konzeptuelle Objekte sind.

Um ein Objekt zu betrachten und verändern zu können, benötigt man eine externe Darstellung. ObjTalk bietet ein derartiges Objektformat an, jedoch ist es für den Programmierer gedacht und stellt einen LISP-Ausdruck dar, der das Objekt definiert. Für den Endbenutzer eines in ObjTalk implementierten Anwendungssystems ist dieses Format nicht geeignet.

Mit einem hochauflösenden Bildschirm können die Objekte der Wissensbasis in einer dem Benutzer besser angemessenen Form dargestellt werden. Dies zeigte sich am Erfolg der neuen graphischen Benutzerschnittstellen, die erstmals im Büroinformationssystem Xerox Star [15] zum Einsatz kamen und heute in vielen kleinen und mittleren Computersystemen Standard geworden sind. Arbeitsgegenstände und Arbeitsmittel aus der Bürowelt werden durch graphische Objekte, sogenannte Icons, dargestellt und können mittels direkter Manipulation [8, 16] genutzt werden. Ikonische Benutzerschnittstellen sind sehr einfach zu erlernen und zu verstehen. Die Benutzerschnittstellen des Star-Systems und seiner Nachfolger erfüllen jedoch nicht alle Erfordernisse der externen Repräsentation von Wissen:

- Das Star-System ist auf die Bürowelt beschränkt. Ein allgemeines System muß hingegen Objekte aus beliebigen Anwendungsbereichen darstellen können.
- Das Star-System unterstützt nur hierarchische Relationen, z. B. zwischen einem Aktenordner und den in ihm enthaltenen Dokumenten. Ein universelles System sollte beliebige Relationen zwischen Objekten darstellen können.
- Es muß möglich sein, nicht nur konkrete Objekte (wie etwa die Akte „XYZ“), sondern auch abstrakte Begriffe (etwa den Begriff „Akte“) darzustellen.

Auf der anderen Seite gibt es Werkzeuge zur Bearbeitung von Wissensbasen (sogenannte Knowledge Engineering Tools), die die meisten dieser Erfordernisse erfüllen, aber nur sparsamen Gebrauch von Graphik machen. So werden Objekte vom Smalltalk-Browser [6] nur in textueller Form dargestellt. KEE [9] und LOOPS [1] zeigen nur Spezialisierungs- und Instantiierungshierarchien von Objekten, andere Relationen werden nicht unterstützt. KEE's SIMKIT-System beruht vollständig auf der Manipulation von graphischen Objekten, sein Einsatz ist jedoch beschränkt auf Simulationsanwendungen. Im Gegensatz dazu besteht die Grundidee des ZOO-Systems darin, die graphischen Fähigkeiten eines ikonischen Systems mit der Universalität eines Knowledge Engineering Tools zu verbinden.

Graphische Wissensdarstellung im System ZOO

Gewöhnliche Objekte und konzeptuelle Objekte der Wissensrepräsentationssprache ObjTalk können vom ZOO-System in einer graphischen Form dargestellt werden. Zu diesem Zweck bietet ZOO zwei Arten von graphischen Grundelementen an: Icons und Pfeile. Icons dienen zur graphischen Darstellung von Objekten. Üblicherweise besitzt solch ein Icon einen quadratischen Umriß. Das Icon besteht aus einem graphischen Symbol, das die Klassenzugehörigkeit des dargestellten Objekts verbildlicht. Das Icon ist beschriftet mit dem Namen des Objekts, für das es steht.²⁾ Relationen zwischen Objekten werden repräsentiert durch beschriftete Pfeile, die die entsprechenden Icons miteinander verbinden. Dabei erscheint der Name der Relation als Beschriftung der Pfeile. Abbildung 2 zeigt die graphische Repräsentation eines Objekts, eines deutschen Computersystems, in Relation zu anderen Objekten. Insgesamt erscheint das Faktenwissen als ein graphisches Netz, wobei Icons die Knoten und beschriftete Pfeile die Kanten darstellen.

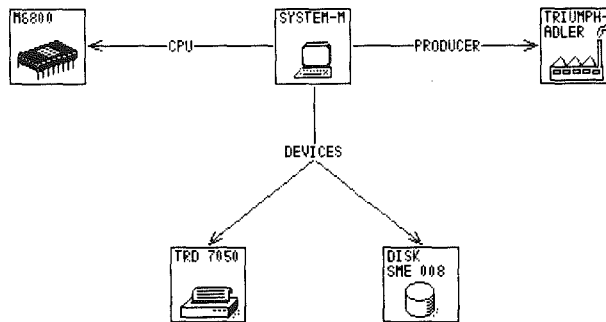


Abb. 2 Das Objekt SYSTEM-M in Relation zu anderen Objekten

Metawissen wird prinzipiell in derselben Art dargestellt wie Faktenwissen, lediglich mit dem Unterschied, daß konzeptuelle Objekte an die Stelle von gewöhnlichen Objekten treten. Klassen erscheinen als Icons, die mit einem dicken Rand umgeben sind (siehe Abb. 3). Defaultmäßig wird das graphische Symbol eines solchen Icons auf alle Subklassen in der Spezialisierungshierarchie und auf alle Instanzen vererbt. Die Spezialisierungshierarchie der Klassen wird von ObjTalk durch eine Relation mit dem Namen „super“ repräsentiert und läßt sich deshalb wie jede andere Relation durch beschriftete Pfeile darstellen. Die anderen konzeptuellen Objekte, wie Slotbeschreibungen, Methoden und Regeln, sind mit dem Netz der Klassen durch entsprechende Relationen verknüpft, die ebenfalls durch beschriftete Pfeile dargestellt werden.

2) Dieser Name braucht nicht eindeutig zu sein. Die Identität eines Objekts ergibt sich allein aus seiner Lage im Objektnetz, die für den Benutzer als geometrischer Ort sichtbar ist.

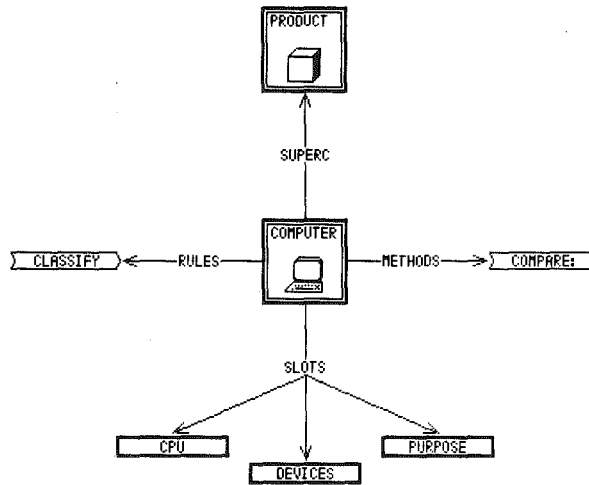


Abb. 3 Die Klasse COMPUTER und die sie umgebenden konzeptuellen Objekte

Slotbeschreibungen erscheinen als längliche Rechtecke, die den Slotnamen umschließen. Von solchen Rechtecken können wiederum beschriftete Pfeile ausgehen und auf andere Icons zeigen, die den Typ (eine Klasse), den Defaultwert (eine Instanz) und die inverse Relation (eine andere Slotbeschreibung) repräsentieren (siehe Abb. 4). In ähnlicher Weise sind auch Methoden und Regeln innerhalb des Netzes konzeptioneller Objekte dargestellt.

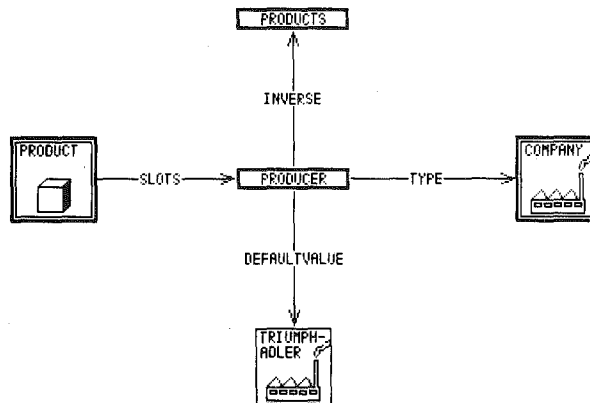


Abb. 4 Die Slotbeschreibung PRODUCER der Klasse PRODUCT

Das Metasystem ZOO

Das Metasystem ZOO läßt sich ansehen als eine Benutzerschnittstelle zum gesamten durch ObjTalk-Objekte repräsentierten Wissen eines Softwaresystems. Die Aufgabe

von ZOO ist es, den Inhalt einer Wissensbasis im oben beschriebenen graphischen Format darzubieten, wobei der Benutzer mittels direkter Manipulation Änderungen vornehmen kann. ZOO dient zur Bearbeitung von geladenen, ablauffähigen Wissensbasen. Es ist deshalb möglich, wissensbasierte Systeme zur Laufzeit zu verändern. ZOO ermöglicht gleichermaßen den Zugriff auf konzeptuelle Wissensbasen und Faktenwissensbasen.

Die graphische Anzeige von Wissensbasen erfolgt in rechteckigen Bildschirmbereichen, den sogenannten ZOO-Fenstern (siehe Abb. 1). Jedes Icon steht für ein Objekt der Wissensbasis, jeder beschriftete Pfeil stellt eine Relation zwischen Objekten dar. Durch Anklicken mit der Maus kann eines der graphischen Objekte (d. h. der Icons oder der beschrifteten Pfeile) selektiert werden. Diese Selektion ist auf dem Bildschirm erkennbar durch Inversdarstellung. Alle objektspezifischen Benutzerfunktionen des ZOO-Systems beziehen sich auf den selektierten Teil der Wissensbasis, der durch Inversdarstellung hervorgehoben ist. Die Benutzerfunktionen sind auf dem Rand des ZOO-Fensters durch Aktionssymbole dargestellt, die ebenfalls mit Hilfe der Maus aktiviert werden können. Das ZOO-System kann auf zwei Arten benutzt werden, wobei beliebige Mischformen möglich sind:

- ZOO kann als graphischer Browser³⁾ benutzt werden. Mit Hilfe der STEP-Funktion ist eine „Navigation“ zu benachbarten Objekten möglich. Dabei können auch bislang unsichtbare Teile der Wissensbasis aufgedeckt werden. Die Richtung der Navigation kann durch Auswahl aus einem Menü von Relationen festgelegt werden. Dieses Menü ist kontextabhängig, deshalb können nur Relationen ausgewählt werden, die auf das selektierte Objekt anwendbar sind.
- ZOO kann als graphischer Editor verwendet werden. Relationen zwischen Objekten können hergestellt werden, indem Pfeile zwischen den Icons gezeichnet werden. Neue Objekte können erzeugt werden, indem Kopien oder Instanzen bereits vorhandener Objekte gebildet werden. Schließlich gibt es Funktionen, um vorhandene Objekte zu löschen und um bestehende Relationen zwischen Objekten aufzulösen. Alle diese Operationen haben unmittelbare Auswirkung auf die zugrundeliegenden Wissensbasen.

Zusätzlich zu diesen zwei fundamentalen Funktionalitäten des Metasystems verfügt ZOO über eine Wissensbasisverwaltung zum Abspeichern und Laden von Wissensbasen, und das System besitzt Schnittstellen zu einem Icon-Editor [10] zur graphischen Gestaltung der Icon-Symbole.

Im System ZOO gibt es kein zentrales Verwaltungsprogramm zur graphischen Darstellung der Wissensbasisobjekte. Statt dessen erfolgt die Anzeigeverwaltung in objekt-orientierter Weise verteilt auf viele aktive Instanzen. Jedes Icon und jeder beschriftete Pfeil ist selbst durch ein ObjTalk-Objekt, ein sogenanntes Interaktionsobjekt, repräsentiert.

Interaktionsobjekte stellen die Verbindung her zwischen den internen Objekten und ihren graphischen Abbildern auf dem Bildschirm. In Interaktionsobjekten sind daher zwei Arten von Wissen repräsentiert:

- Ein Interaktionsobjekt besitzt graphisches Wissen, es beschreibt die Lage und die Erscheinung eines Bildschirmobjekts. Dieses Wissen wird benötigt zur Anzeige und zur Aktivierung der Bildschirmobjekte.
- Ein Interaktionsobjekt besitzt Wissen über die Existenz eines internen Objekts oder über eine einzelne Eigenschaft eines solchen Objekts. Dadurch werden Interaktionsobjekte zu Medien, über die ein Zugriff auf die internen Wissensbasisobjekte möglich ist.

3) Ein Browser ist ein Software-Werkzeug, mit dem es möglich ist, Datenstrukturen zu untersuchen. Dabei ist üblicherweise nur ein Teil der Datenstruktur im Blickfeld des Benutzers. Die Verschiebung des Blickfelds von einem Teil der Datenstruktur zu einem benachbarten Teil nennt man Navigation.

Innerhalb eines ZOO-Fensters gibt es verschiedene Klassen von Interaktionsobjekten, die zur Anzeige der verschiedenen Arten von Wissensbasisobjekten (gewöhnliche Objekte und konzeptuelle Objekte) und deren Relationen verwendet werden. Die Aktionssymbole am Rand eines ZOO-Fensters dienen nicht zur Visualisierung von Wissensbasisobjekten, sie stellen vielmehr Benutzerfunktionen des ZOO-Systems dar. Die meisten dieser Funktionen beziehen sich auf das selektierte graphische Objekt innerhalb des ZOO-Fensters. Die Wirkung dieser Funktionen hängt von der Klasse des selektierten Interaktionsobjekts ab. So kommt das System ZOO mit einer relativ kleinen Anzahl von Benutzerfunktionen aus, durch die Kontextabhängigkeit des Systemverhaltens ist die Benutzerschnittstelle trotz ihrer Einfachheit genügend mächtig.

Zusammenfassung

Die Berücksichtigung von Wissensaspekten ermöglicht eine neue Sichtweise der Mensch-Computer-Kommunikation. Die Forderung nach verständlichen und gut handhabbaren Benutzerschnittstellen läßt sich zurückführen auf das Problem einer benutzergerechten externen Wissensrepräsentation. Die hier vorgeschlagene Lösung dieses Problems beruht auf der zweidimensionalen Darstellung des Wissens durch Bildschirmobjekte, die der direkten Manipulation mit einem Zeigegerät unterworfen sind.

Große Vorteile bietet hierbei eine integrierte objektorientierte Systemarchitektur, die den Systemkern und die Benutzerschnittstelle gleichermaßen umfaßt. Eine objektorientierte Benutzerschnittstelle ermöglicht den Einsatz von Interaktionsobjekten, die sich nach außen hin als sensitive Bildschirmobjekte zur externen Wissensrepräsentation zeigen, nach innen hin aber aktive Datenstrukturen sind, die mit der internen Wissensbasis interagieren. Da das systeminterne Wissen selbst in einer objektorientierten Wissensbasis repräsentiert ist, haben die Interaktionsobjekte der Benutzerschnittstelle eine eindeutige interne Entsprechung, und eine enge Kopplung beider Darstellungen wird möglich.

Das Objekt als Paradigma der Benutzerschnittstellenprogrammierung und der Wissensrepräsentation ermöglicht die einfache Konstruktion von Metasystemen, mit denen Fakten und Konzepte des Systems visualisiert und vom Benutzer modifiziert werden können. Der objektorientierte, wissensbasierte Ansatz, verbunden mit fortgeschrittenen Techniken der Mensch-Computer-Kommunikation, ermöglicht so die Konstruktion besser durchschaubarer und in weiten Teilen vom Benutzer adaptierbarer Anwendungsprogramme und leistet so einen entscheidenden Beitrag zur Gestaltung benutzergerechter Computersysteme.

Literatur

- [1] *Bobrow, D. G.; Stefik, M.*: The LOOPS Manual. Technical Report KB-VLSI-81-83, Knowledge Systems Area, Xerox Palo Alto Research Center (PARC), 1981.
- [2] *Brachmann, R.* et al.: KL-ONE Reference Manual. BBN-Report 3848, BBN, Juli 1978.
- [3] *Davis, R.; Lenat, D. B.*: Advanced Computer Science Series: Knowledge Based Systems in Artificial Intelligence. McGraw-Hill, New York, 1982.
Hauptwerk zum Thema Metawissen und Metasysteme. Der zweite, von R. Davis verfaßte Teil des Buches befaßt sich mit dem Metasystem TEIRESIAS, das für Expertensysteme vom Typ MYCIN bestimmt ist.
- [4] *Fabian, F. Jr.; Lemke, A. C.*: Wlisp Manual. Technical Report CU-CS-302-85, University of Colorado, Boulder, Februar 1985. Translated by V. Patten and C. Morel.

-
- [5] *Foderaro, J. K.; Sklower, K. L.*: The Franz Lisp Manual. Technical Report, University of California, Berkeley, 1982.
 - [6] *Goldberg, A.*: SMALLTALK-80, The Interactive Programming Environment. Addison Wesley, Reading, Ma., 1984.
 - [7] *Hewitt, C.*: Viewing Control Structures as Patterns of Passing Messages. Artificial Intelligence Journal 8, S. 323-364, 1977.
 - [8] *Hutchins, E. L.; Hollan, J. D.; Norman, D. A.*: Direct Manipulation Interfaces. In: Norman, D. A.; Draper, S. (Hrsg.), User Centered System Design: New Perspectives on Human-Computer Interaction. Lawrence Erlbaum Associates Ltd., 1986.
 - [9] KEE Software Development System User's Manual. IntelliCorp, 1985.
 - [10] *Maier, D.*: Inform-Manual: Character-Editor, Version 3.1. WISDOM-Forschungsbericht FB-INF-85-26, Institut für Informatik, Universität Stuttgart, 1985.
 - [11] *Minsky, M.*: A Framework for Representing Knowledge. In: Winston, P. H. (Hrsg.), The Psychology of Computer Vision, S. 211-277. McGraw-Hill, New York, 1975.
Klassische Schrift zur frameorientierten Wissensrepräsentation aus psychologischer Sicht.
 - [12] *Rathke, C.*: ObjTalk. Repräsentation von Wissen in einer objektorientierten Sprache. Dissertation, Fakultät Mathematik und Informatik der Universität Stuttgart, Oktober 1986.
 - [13] *Riekert, W.-F.*: Werkzeuge und Systeme zur Unterstützung des Erwerbs und der objektorientierten Modellierung von Wissen. Dissertation, Fakultät Mathematik und Informatik der Universität Stuttgart, Oktober 1986.
Enthält eine umfassende Darstellung des ZOO-Systems.
 - [14] *Shneiderman, B.*: Direct Manipulation: A Step Beyond Programming Languages. IEEE Computer 16 (8), S. 57-69, August 1983.
Shneidermann führte mit diesem Beitrag den Begriff „direkte Manipulation“ in die Informatik ein. Eine Charakterisierung dieses Begriffs aus neuerer Sicht ist in [8] enthalten.
 - [15] *Smith, D. C.; Irby, Ch.; Kimball, R.; Verplank, B.*: Designing the Star User Interface. BYTE, April 1982.

H M D

HANDBUCH DER MODERNEN DATENVERARBEITUNG

Objektorientierte Systementwicklung

Fortdruck aus HMD 145

Forkel-Verlag

Heft 145
Januar 1989