

Proseminar  
Sommersemester 2006

## Human-Computer Interaction

# Design of the Conceptual Model

Basierend auf dem Interview mit David Liddle  
Auszug aus „Bringing Design to Software“  
von Terry Winograd

RWTH Aachen  
Informatik Lehrstuhl iX

media  
computing  
group



Prof. Dr. Jan Borchers  
Betreuer:  
Dipl.-Inform. D. Spelmezan

Autoren:  
Violett Panahabadi  
Neda Nazari

## Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>2</b>
<b>2. Hintergrundinformationen</b>	<b>3</b>
<b>3. Conceptual Model</b>	<b>4</b>
<b>3.1 Was ist das Conceptual Model?</b>	<b>4</b>
<b>3.2 Relevanz des Conceptual Model beim Software Design</b>	<b>6</b>
<b>3.3 Umsetzung des Conceptual Model bei der Entwicklung des Xerox Star</b>	<b>7</b>
<b>4. Der Zeit zu weit voraus</b>	<b>9</b>
<b>5. Fazit</b>	<b>10</b>
<b>Literaturverzeichnis</b>	<b>11</b>
<b>1. Literarische Quellen</b>	<b>11</b>
<b>2. Internetquellen</b>	<b>11</b>

## 1. Einleitung

Die Human-Computer Interaction (HCI) als Teilgebiet der Informatik beschäftigt sich mit der benutzergerechten Gestaltung von interaktiven Systemen und ihren Mensch-Maschine-Schnittstellen, mit denen der Mensch über eine Benutzeroberfläche interagieren kann. Diese Benutzerschnittstellen müssen, um optimal bedienbar zu sein, speziell auf die Bedürfnisse des Benutzers angepasst sein.

In der folgenden Ausarbeitung wird das Conceptual Model in Bezug auf Software Design vorgestellt. Es soll bewirken, dass der Benutzer sich ein klares Bild von einem System machen und es intuitiv bedienen kann. Dies soll zum schnelleren Erlernen des Systems und zur Vermeidung von möglichen Fehlern führen.

David Liddle<sup>i</sup>, ein Wissenschaftler der Informatik, veranschaulicht in einem Interview mit der Association of Software Design den Begriff des Conceptual Model. Dabei beschreibt Liddle seine Erfahrungen bei der Entwicklung des Xerox Star und begründet damit die Bedeutsamkeit des Conceptual Model beim Entwurf einer benutzerfreundlichen grafischen Schnittstelle. Das Interview verschafft Einblicke in das Software Design und in die Ideen, die durch Programmierer, Psychologen und Designer entstanden sind, und durch deren Engagement sich ein neuer Weg für die Interaktion mit einem Computer eröffnet hat.

Diese Arbeit hat zum Ziel, näher zu untersuchen, was das Conceptual Model genau ist, warum es bei der Entwicklung einer Software solch hohen Rang hat und welche Vor- und Nachteile sich bei der Erarbeitung eines guten Conceptual Model ergeben.

---

<sup>i</sup> David Liddle: Mitgründer und Präsident des Forschungsinstituts ‚Interval Research Corporation‘ in Palo Alto. Beratender Professor an der Universität Stanford, wo er Vorlesungen im Bereich der HCI und Computerindustrie hält. Erfinder der metaphorischen Computer-Systeme und beauftragter Leiter bei der Entwicklung des Xerox Star.



Abbildung 1.1:  
David Liddle

## 2. Hintergrundinformationen

Die **Xerox Corporation** ist ein Technologie- und Dienstleistungsunternehmen in Stamford, Connecticut, welche sich zum Ziel gesetzt hat, Anwendern zu helfen, durch intelligente Dokumenten-Management-Strategien effizientere Arbeitsprozesse zu gestalten.

Xerox produziert und vermarktet eine umfassende Palette an Produkten wie z.B. Drucker, Faxgeräte und Kopierer.

Die Firma ist zusätzlich im Besitz einer Forschungsanlage, dem **Palo Alto Research Center (PARC)**. Viele der uns heute Bekannten und im Nachfolgenden erwähnten Technologien und Innovationen sind dort entstanden. Das PARC wurde 1970 vom Chefwissenschaftler Jack Goldman in Palo Alto, Kalifornien gegründet. Zu dieser Zeit verlor Xerox den Patentschutz für Xerografie, das Verfahren der Trockenkopie, und fürchtete, dass es durch Länder wie Japan große Marktanteile verlieren würde. Infolgedessen wurde PARC geschaffen, um neue Bürotechnologien für Xerox zu entwickeln. Heute ist PARC Technologie-Entwickler nicht nur für Xerox, sondern auch für viele andere Unternehmen.

Die mit Abstand größte im PARC entstandene Innovation war der erste Personal Computer mit einer grafischen Benutzeroberfläche (GUI = Graphical User Interface). Dieser trug den Namen **Xerox Alto** und besaß Softwarekomponenten, die dem Benutzer die Interaktion mit der Maschine durch grafische Elemente, wie z.B. Symbole, ermöglichte. Der Xerox Alto hatte bereits eine 3-Tasten Maus, nutzte Bitmap-Grafik und Menüs. Durch seine bitorientierte Ausgabe war es möglich die Anzeige auf dem Bildschirm zu drucken, nach dem Prinzip „WYSIWYG = What you see is what you get“. Der Alto wurde jedoch nur in der Forschung verwendet. Deswegen hat man im PARC einen PC entwickelt, der auch kommerziellen Nutzen haben sollte, den Xerox Star.



Abbildung 2.1 :Xerox Alto



Abbildung 2.2 Xerox Star

Der **Xerox Star** sollte entwickelt werden, damit auch Normalverbraucher, die nicht das technische Know How hatten, den Computer als nützliches Medium verwenden konnten. Ziel war es, den Star in den Büros der Unternehmen zur Verfügung zu stellen. Deswegen beinhaltete er hauptsächlich ein „Office Automation System“, das dem User das Ausführen von Bürotätigkeiten (wie Textverarbeitung, Tabellenkalkulation etc.) vereinfachen sollte. Der Star sollte folgende

- Anzeige der Benutzeroberfläche: Hierbei geht es um die optische Darstellung der Objekte, die auf dem Bildschirm zu sehen sind, wie z.B. Fenster, Buttons oder Icons.
- Kommando-Aufruf-Mechanismus: Die Befehle so sollten so konstruiert sein, dass sie innerhalb verschiedener Anwendungen gleich aufzurufen waren.
- Conceptual Model.

### 3. Conceptual Model

#### 3.1 Was ist das Conceptual Model?

Das Conceptual Model übersetzt man mit ‚konzeptionelle Modellierung‘. Es ist im Bereich der HCI und speziell beim Software Design, ein Modell, welches auf dem Konzept beruht, dem Benutzer die Arbeit mit einem Computer so angenehm wie möglich zu gestalten.

In dem Buch „The Design of Everyday Things“ beschreibt der Autor Donald Norman<sup>ii</sup> das Conceptual Model als Konzept, das der Designer für sein System entwickelt und das als Design Modell des Designers bezeichnet wird. Zusätzlich gibt es den Benutzer, der mit dem System arbeitet und sich ein mentales Modell bildet. Mentale Modelle sind Vorstellungen, die der Mensch von sich selbst, von Anderen, von seiner Umgebung oder von Dingen mit denen er arbeitet, hat. Diese mentalen Modelle werden durch Erfahrungen oder Instruktionen mit Menschen bzw. Objekten geformt, die der Mensch insbesondere visuell und mental verinnerlicht. Der Benutzer hat laut Norman somit eine Vorstellung, wie das vom Designer zu entwerfende System aussieht, was für Aufgaben damit ausgeführt werden können und welcher persönlicher Nutzen sich der Benutzer dadurch definieren kann. Donald Norman hat dafür den Begriff der *Affordance* eingeführt. Dieser Begriff wurde aus dem englischen Verb „to afford“ abgeleitet und kann sinngemäß mit „Nutzen“ oder „Angebotscharakter“ definiert werden, z.B. wissen wir, dass der Stuhl den Angebotscharakter hat sich darauf zu setzen oder darauf zu stellen.

Der Fehler, der bei dem Conceptual Model häufig gemacht wird, ist, dass der Designer davon ausgeht, dass das mentale Modell des Benutzers identisch ist mit seinem Design Modell. Da Designer und User nicht direkt miteinander sondern nur über das System kommunizieren, kann es passieren, dass das

---

<sup>ii</sup> Donald Norman : em. Professor der Informatik, Psychologie und Kognitionswissenschaft. Er ist Usability-Spezialist und war Berater vieler Unternehmen, u.a. Vizepräsident der Advanced Technologie Group bei Apple Computers, und Leiter des Appliance Design Center bei Hewlett-Packard.

System das Conceptual Model des Designers nicht klar genug veranschaulicht. Folglich kann es im passieren, dass der Benutzer sich ein falsches Mental Model bildet und den richtigen Nutzen des Systems nicht erkennt oder - im schlimmsten Fall - sich das System für ihn als nutzlos erweist.

Um dies zu vermeiden sollte ein Designer die Kriterien für ein „gutes“ Conceptual Model beachten. Ein gutes Conceptual Model hilft dem unerfahrenen Benutzer, sich eine nützliche und klare mentale Vorstellung des Systems zu bilden, damit er in der Benutzung möglichst wenige Fehler macht und das System schnell erlernen kann. Dabei ist wichtig, dass das Conceptual Model der Aufgabe des Benutzers bzw. dessen gedankliche Vorstellung der Aufgabe angepasst ist. Daher liegt dem Conceptual Model eine ausführliche und genaue Aufgabenanalyse zugrunde.

Zusätzlich sollte der Designer eine gründliche Interpretation des Mentalen Models vornehmen, das die Verstrickung der menschlichen Verbindungen zu Objekten beinhaltet. Diese Verbindung zwischen Objekten bzw. Anwendungen und ihren Auswirkungen bezeichnet Donald Norman als „Mapping“. Ein gutes Mapping kann erzielt werden, in dem die Funktion durch Metaphern bzw. Symbole, dargestellt wird. Z.B. sollte ein Bedienknopf eines Gerätes mit einer Abbildung oder einem Symbol versehen sein, die den Nutzen des Schalters darstellt.

Das Conceptual Model ist in vielen Bereichen anwendbar. Dabei weisen alle Modelle eine Gemeinsamkeit auf: Sie stellen die Wünsche und Vorstellungen des Benutzers in den Vordergrund.

Um den Begriff des Conceptual Model zu demonstrieren, verwenden wir als Beispiel das Bediensystem der Klimaanlage in einem Auto. Hierbei ist ein gutes Conceptual Model gewählt worden, weil durch Symbole und Farben leicht und intuitiv die Funktionen der



Abbildung 3.1: Klimaanlage Auto

Klimaanlage zu bedienen sind. Zudem ist jedem Schaltknopf genau eine Funktion zugewiesen. Selbst wenn man nichts mit den Symbolen assoziieren kann, kann man die Schalter ausprobieren und, wie Norman sagt, erhält man dann ein *Feedback* von dem Auto, d.h. man kriegt immer direkt eine Antwort oder kann die Auswirkungen sehen, nachdem man einen Schalter betätigt hat.

Nach David Liddle verkörpert das „User’s Conceptual Model“ die gedankliche Verbindung eines Benutzers und eines Systems, in dem er ein Objekt bearbeiten möchte, d.h. der Benutzer selbst hat eine eigene bildliche Vorstellung von seiner Aufgabe (Task) und kennt ein System (Computer), mit dem er seine Aufgabe bewältigen kann (z.B. durch Applikationen). D. Liddles Formulierung dazu lautet: *“This model represents what the user is likely to think, and how the user is likely to respond”*<sup>iii</sup> Bei der Entwicklung des Xerox Star spielte für David Liddle das User’s Conceptual Model des Software Designs die zentrale Rolle.

<sup>iii</sup> Quelle: Bringing Design to Software, Seite 21 Zeile 8 f.

### 3.2 Relevanz des Conceptual Model beim Software Design

*„Software Design is the act of determining the user’s experience with a piece of software. It has nothing to do with how the code works inside, or how big or small the code is. The designer’s task is to specify completely and unambiguously the user’s whole experience.“<sup>iv</sup> [...]*

*“The most important thing to design is the user’s conceptual model. Everything else should be subordinated to making that model clear, obvious, and substantial. That is almost exactly the opposite of how most software is designed. “<sup>v</sup>*

*(David Liddle)*

Wie von David Liddle angesprochen, werden oft Lösungen entwickelt, die nicht wirklich auf die realen Probleme zugeschnitten sind. Es wird zu spät erkannt, dass die entwickelte Lösung nicht den wirklichen Anforderungen der Benutzer entspricht, weil die Analyse vernachlässigt wurde. Ein sorgfältiger Designprozess sollte also immer mit der Analyse der Ausgangssituation beginnen.

Eines der Hauptanliegen des Gebietes Human Computer Interaction ist es, ein User Interface so zu gestalten, dass es eine möglichst hohe Benutzerfreundlichkeit aufweist.

*„Designers make their greatest user-interface errors when they don't think about users in terms of what those users are doing in their jobs.“<sup>vi</sup>*

Nur wenn man weiß, welches Ziel der Benutzer verfolgt bzw. welche Vorstellung er von seiner zu lösenden Aufgabe hat, kann man eine geeignete Benutzerschnittstelle entwickeln, die dem Benutzer die Möglichkeit bietet, seine Aufgabe unkompliziert und effizient durch einen Computer umzusetzen. Eine genaue Benutzer- und Aufgabenanalyse ist laut Liddle daher grundlegend und oft unumgänglich. Selbst gegenüber seinem Team, das teilweise Schwierigkeiten bei der Implementierung seiner Ideen hatte, verhartete David Liddle auf seine Meinung:

*„let's make this work for the user. If we find we can't implement something, then we'll go back to the drawing board. But we're not going to pick things that we think will be small or fast to build, and then bully the user into accepting them.“<sup>vii</sup>*

---

<sup>iv</sup>Quelle: Bringing Design to Software, Seite 30 Zeile 19-22

<sup>v</sup>Quelle: Bringing Design to Software, Seite 21 Zeile 3-6

<sup>vi</sup>Quelle: Bringing Design to Software, Seite 28 Zeile 23 ff.

<sup>vii</sup>Quelle: Bringing Design to Software, Seite 25 Zeile 4-8

### 3.3 Umsetzung des Conceptual Model bei der Entwicklung des Xerox Star

Der Xerox Star wurde mit der Zielsetzung entwickelt, erstmals einen PC auf dem Markt zu verbreiten. Unerfahrene Nutzer, die zum ersten Mal mit dem PC arbeiteten und nicht das nötige Hintergrundwissen besaßen, sollten in der Lage sein, solch ein System selbstständig zu erlernen und anzuwenden. Um dies möglich zu machen, ist das Conceptual Model von zentraler Bedeutung. Bei der Modellierung von Mensch-Maschine-Systemen muss sowohl die technische als auch die menschliche Komponente berücksichtigt werden. Bei der Entwicklung vom Xerox Star, war für David Liddle die menschliche Komponente, also das User's Conceptual Model, jedoch vordergründig, natürlich weil es in erster Linie von den oben genannten unerfahrenen Benutzern einfach zu bedienen sein sollte.

Das User's Conceptual Model bei der Entwicklung vom Xerox Star fand in zwei Teilen seine Umsetzung:

Zunächst waren natürlich Innovationen aus der Vergangenheit ausschlaggebend. Dabei bezog man sich insbesondere auf den Vorgänger des Xerox Star, dem Xerox Alto. Die Entwicklung des ersten Computers mit einem GUI (1973) bezog sich auf Alan Kays Konzept eines Dynabooks. Die Idee des Dynabooks beruhte auf der Vision der benutzerfreundlichen Bedienung, nicht nur auf symbolischer, sondern auch auf sensomotorischer Ebene (z.B. Touchscreen).

Die Entstehung der ersten Computermaus (1964) von Douglas C. Engelbart war ebenfalls eine große Anregung für die Entwicklung eines PC mit einer GUI, da es noch keine grafischen Benutzeroberflächen gab, bei der die Maus als Zeigegerät eingesetzt werden konnte. Die Computermaus wurde somit zum ersten mal beim Xerox Alto eingesetzt.

Engelbarts Hypertext-System<sup>viii</sup>, das eher der Funktionsweise des menschlichen Denkens als dem eines linearen Textes entsprach, war ebenfalls maßgebend für die Entwicklung des Xerox PCs, da die Annahme bestand, dass unser vernetztes Denken ähnlich abläuft wie die hypertextbasierten Strukturen.

Jedoch wurde nicht nur das Conceptual Model der vergangenen Innovationen bei der Entwicklung vom Xerox Star involviert, sondern auch David Liddle und sein Team, bestehend aus Programmierern, Psychologen und Designer, leisteten neuen Beitrag zum Conceptual Model. Die neuen zentralen Elemente für eine benutzerfreundliche und einfache Bedienung des Xerox Star, waren:

---

<sup>viii</sup> System zur Speicherung und inhaltlichen Verknüpfung von Textdokumenten durch hierarchische Verweisstruktur. Es entstand aus dem Wunsch heraus, im Rechner gespeicherte Texte, wie sie von einem Editor erzeugt werden, aufzubereiten, inhaltlich zu strukturieren und Beziehungen innerhalb von Texten in unterschiedliche Dateien herzustellen (Quelle: Duden Informatik)

---



### ♣ Symbole/Metaphern

David Liddle und sein Team führten erstmals Symbole ein, die der Benutzer auf der graphische Benutzeroberfläche leicht mit seiner anstehenden Aufgabe verbinden konnte. Somit konnte er erkennen welches Symbol zu welchem Programm, Dokument o.Ä. gehörte. Denn in der Psychologie ist es zum Einem bekannt, dass der Mensch besser Dinge erlernt, wenn er sie sieht und mit ihnen praktisch arbeitet und zum Anderen kann er Dinge oder Objekte besser wieder erkennen als sich daran zu erinnern („*Recognition is generally easier than recall*“<sup>ix</sup>

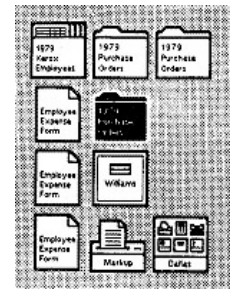


Abbildung 3.1:  
Desktop Metapher

by Anderson). Für das Design der Icons benutze man Desktop-Metaphern (Schreibtisch-Metaphern), wie z.B. Ordner, Dokumente o.Ä. Dies sind alles Metaphern, die sich in der Umgebung des Benutzers, vor allem im Büro oder am Schreibtisch, wieder finden. Sie verkörpern visuell das passende physikalische Objekt, zu dem ein Xerox Star User leicht eine Beziehung zwischen dem angezeigten Symbol auf dem Bildschirm und einem Gegenstand, der in seinem Büro wieder zu finden war, herstellen konnte. Wenn der Benutzer z.B. das Symbol eines Druckers sah, sollte er sofort erkennen, dass er durch einen Mausklick darauf sein gewünschtes Dokument ausdrucken konnte.

### ♣ Property Sheets

Property Sheets sind heute mehr unter dem Begriff der Menüoption/Eigenschaften bekannt. Anstatt in einer Shell den gewünschten Befehl für die Bearbeitung eines Objektes zu suchen und dann erst auszuführen, bot sich durch die Property Sheets die Möglichkeit, sich mit der Maus eine Liste von Eigenschaften des zu bearbeitenden Objekts ausgeben zu lassen, welche man umändern konnte. Dabei waren die Property Sheets so ausgelegt, dass sich die Option der Befehle nur auf das Objekt selbst begrenzte

### ♣ Direct Manipulation

Direct Manipulation ist ein Bearbeitungskonzept, wobei ein Resultat direkt nach der Bearbeitung durch den Mausfeil sichtbar wird.

### ♣ WYSIWYG - What you see is what you get

Dabei wird ein Dokument auf dem Bildschirm genauso angezeigt, wie es bei einem Ausgabegerät, z. B. einen Drucker, erscheinen wird. Die Herkunft des Begriffs WYSIWYG stammt aus dem PARC, und wurde von den Forschern in den späten 70ern bei der

<sup>ix</sup> Quelle: Human Factors Testing in the Design of Xerox's 8010 „Star“ Office Workstation

Entwicklung des ersten Editors, Bravo, geprägt. Der Bildschirm des Xerox Alto konnte eine ganze Seite Text darstellen und diese auf den damals ersten Laserdruckern ausdrucken.

#### ♣ **Übereinstimmung von Befehlen innerhalb verschiedener Anwendungen**

Hierbei kam es darauf an, die Befehle so zu konstruieren, dass sie innerhalb verschiedener Anwendungen gleich aufzurufen waren. Die Realisation dieses Prinzips erfolgte dadurch, dass man auf der Tastatur des Xerox Stars extra Kommando-Tasten zur Verfügung stellte, dessen Funktionalität innerhalb verschiedener Anwendungen konstant war.



Abbildung 3.2:  
Kommando-Tasten

Hinter diesem Conceptual Model verbirgt sich monatelange Analyse und Forschung. Ständig wurden Prototypen entwickelt und diese an Benutzern getestet. Hatte der Benutzer Probleme damit, so wurde das Konzept optimiert und erneut ein Prototyp entwickelt, solange bis sich keine Hindernisse mehr in den Weg stellten.

Obwohl die oben aufgegriffenen Punkte für die heutigen Systeme selbstverständlich sind, blieb der Erfolg des Stars zu seiner Zeit aus.

#### 4. Der Zeit zu weit voraus

Der Weg des Stars vom Labor auf den kommerziellen Markt verlief alles andere als erfolgreich.

Nachdem die Entwicklung des Stars abgeschlossen war, drohte das Projekt ins Wasser zu fallen. Xerox sah keinen kommerziellen Nutzen vom Star. Die Gründe dafür waren unterschiedlicher Natur. Erstens kostete die Entwicklung durch zahlreiche Tests ein Vermögen, und zum anderen schien Xerox der Star als ein kommerzieller PC mit graphischer Benutzeroberfläche für den Markt nicht geeignet zu sein, da zu der Zeit nur Computer zu Forschungszwecken verwendet wurde und nicht zur Bearbeitung von Bürotätigkeiten, weil der Kostenaufwand viel zu hoch war.

Somit musste David Liddle sehr viel Überzeugungsarbeit leisten, um sein Produkt zu vermarkten. Letztendlich stimmte er Xerox um, so dass diese bereit war den Star der Öffentlichkeit zu präsentieren. 1981 wurde der Xerox Star als erster Personal Computer mit einem GUI auf dem Markt verkauft.

Der Erfolg des Stars blieb aus. Es wurden rund 100 000 Computer verkauft, doch der Wunsch auf jedem Tisch eines Büros einen Computer stehen zu sehen, wurde nicht erfüllt. Die Zeit für den PC war noch nicht gekommen und der Preis von 15 000 \$ war viel zu hoch, so dass nur große und erfolgreiche Unternehmen sich einen Xerox Star leisten konnten.

David Liddle sah noch mehr Gründe für den Misserfolg. Der Star war ein öffentlich zugängliches Medium aus denen andere Computerfirmen, z.B. IBM, ihren Vorteil zogen. Dadurch dass Xerox seine

Konzepte nicht patentieren ließ, war es für die konkurrierenden Firmen ein leichtes Spiel Xerox Innovationen zu übernehmen.

Ein weiterer Kritikpunkt war, dass der Star keine Buchhaltungsmethoden wie Tabellenkalkulation besaß, da man bereits 1978 mit der Entwicklung des Stars begonnen hatte, das erste Tabellenkalkulationssystem jedoch erst 1979 von einer anderen Firma erstellt wurde. Man hatte demnach nicht die Möglichkeit ein Programm in der Form beim Star zu implementieren. Der Star hatte ein ähnliches Programm, das zur Tabellenkalkulation diente, es wurde jedoch von niemandem erkannt, da es nicht dem Standard entsprach, und laut David Liddle, die Metapher schlecht gewählt war.

Die mangelnde Erfahrung auf dem Computermarkt resultierte aus der Tatsache, dass Xerox bis 1981 keine PCs verkaufte und somit branchenfremd war. Der Vertrieb von Xerox war dementsprechend überfordert, weil er es bisher gewohnt war nur Kopierer und Drucker zu vermarkten. Der Vertrieb wusste viel zu wenig über die Funktionen und Aufgaben, die der Star ausführen konnte, und war somit nicht in der Lage, den Xerox Star den potenziellen Käufern schmackhaft machen.

Ein weiterer Grund für den Misserfolg des Stars war sein isoliertes System. Wenn man den PC um weitere Komponenten erweitern wollte, egal ob Hard- oder Software, so musste diese ein Produkt der Firma Xerox sein. Produkte anderer Hersteller waren nicht kompatibel.

Da man sich bei der Entwicklung des Stars zu sehr auf ein perfektes GUI konzentriert hatte, wurde die technische Ebene vernachlässigt. Die Anwendungen vom Xerox Star waren im Vergleich zu den konkurrierenden Systemen, wie z.B. IBM, viel zu langsam.

## **5. Fazit**

Trotz des wirtschaftlichen Misserfolges war der Xerox Star eine große Bereicherung für die Computer-Industrie. Er schaffte Grundlagen für fast alle uns heute bekannten Systeme, wie z.B. Apple Macintosh oder Microsoft Windows

Obwohl sich die Bemühungen bei der Umsetzung des Conceptual Model für die heutigen Systeme ausgezahlt haben, machte man bei der Herstellung des Xerox Star den Fehler, sich zu stark auf die Interessen des Benutzers zu konzentrieren. Dadurch wurde die technische Ebene vernachlässigt und der Star konnte dies bezüglich bei den anderen Systemen nicht mithalten. Schlussfolgernd kann man jedoch sagen, dass das Conceptual Model beim Software Design weiterhin eine zentrale Rolle spielen sollte.

## Literaturverzeichnis

### 1. Literarische Quellen

- Terry Winograd: *Bringing Design to Software*, Addison-Wesley, 1996.
- Donald Norman : *The Design of Everyday Things*, Basic Books ,2002
- Jeff Johnson, Terry Roberts, William Verplank, David C. Smith, Charles Irby, Marian Bread, und Kevin Mackey, “*Xerox Star, a retrospectpective.*” IEEE Computer 22:9(September 1989) 11-26.
- Xerox Office Division: William L. Bewley, Teresa L. Roberts, David Schroit, William L. Verplank, “*Human Factors Testing in the Design of Xerox’s 8010 ‘Star’ Office Workstation*”, Conference on Human Factors in Computing Systems: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, ACM Press.
- DUDEN Verlag, *Duden Informatik*, September 2003

### 2. Internetquellen

- [http:// www.xerox.com](http://www.xerox.com)
- <http://xeroxstar.tripod.com>
- <http://live.interactivesystems.info/lenya/Lehre/WiSe0506/ias0405/LehreFolienInteraktiveS/InSys-01-Einfuehrung.pdf>