

CHRISTIAN DAHME

## Wissenschaftstheoretische Positionen in bezug auf die Gestaltung von Software

*Aus welcher wissenschaftlichen, wissenschaftstheoretischen bzw. erkenntnistheoretischen Position (heraus) reflektieren wir Wirklichkeit als Grundlage für die Gestaltung (Design) von Software und ihrer Integration in den Anwendungszusammenhang?*

Dieser Frage wurde schon an verschiedenen Stellen aus unterschiedlichen Sichten bzw. Positionen heraus nachgegangen.<sup>1</sup> Dabei sind die unterschiedlichsten Positionen bzw. Herangehensweisen zu beobachten, wie z.B.:

1. empirisches, deduktives, hypothetisches Herangehen (auf der Basis von Beobachtungen und Experimenten - auch naturwissenschaftliches Herangehen genannt).  
Hierzu gehört u.a. die Modellmethode einschließlich Modellbildung, sowie die Phasen des Gegenstandsverständnisses einer Wissenschaftsdisziplin, aber auch Begriffe wie System, Veränderung in der Zeit (Historizität, Entwicklung, Evolution).<sup>2</sup>
2. kulturhistorische Position.<sup>3</sup>  
hier mit der Tätigkeitstheorie und der ethnographischen Methode.
3. (radikal) konstruktivistische Position.<sup>4</sup>
4. semiotische Position<sup>5</sup> (einschließlich der Kommunikation).

1 u.a.: Sichtweisen der Informatik. Hrsg. v. W. Coy et al. Braunschweig, Wiesbaden: Vieweg 1992; Software Development and Reality Construction. Hrsg. v. C. Floyd et al. Berlin: Springer 1992.

2 s. u.a. Dahme, Ch., Systemanalyse menschlichen Handelns - Grundlagen und Ansätze zur Modellbildung. Westdeutscher Verlag: Opladen 1997.

3 s. u.a.: Dahme, Ch. / Raeithel, A., Ein tätigkeitstheoretischer Ansatz zur Entwicklung von brauchbarer Software. - In: Informatik-Spektrum. 20 (1997)1, S. 5 - 12; Raeithel, A., Selbstorganisations, Kooperation, Zeichenprozess. Hrsg. v. Ch. Dahme. Opladen: Westdeutscher Verlag 1998.

4 Floyd, C. (Fußnote 1).

5 s. u.a.: Die erträgliche Leichtigkeit der Zeichen. Hrsg. v. F. Nake. Baden-Baden: Agis Verlag 1993; Raeithel, A. (Fußnote 3); Hesse, W. et al., Terminologie der Softwaretechnik - Ein Begriffssystem für die Analyse und Modellierung von Anwendungssystemen. - In: Informatik-Spektrum 17(1994)1.

Diese Diskussion wird hier aus einer anderen Perspektive wieder aufgenommen u.a. mit folgendem Anspruch:

*Wie kommt man zu einer relativ leicht kommunizierbaren, transparent nachvollziehbaren Transformation von den Wünschen und der realen Situation zu brauchbarer Software?*

Dabei werden 3 unterschiedliche Positionen bzw. Sichten betrachtet:

1. Der Gegenstand, der in Software transformiert werden soll, steht im Mittelpunkt.
2. Software als (reduziertes) Abbild einer Tätigkeit.
3. Softwareentwicklung als interkultureller/kommunikativer Prozeß.

Diese drei Positionen sind nicht als sich gegenseitig ausschließende zu verstehen, sondern sie stellen unterschiedliche Blickwinkel bzw. Dimensionen dar, die sich gegeneinander/ zueinander ergänzen können und zwar:

1. die gegenstandsbezogene Dimension.
2. die tätigkeitsbezogene Dimension.
3. die interkulturelle bzw. kommunikative Dimension.

### 1. *Die gegenstandsbezogene Dimension*

Der Gegenstand, der in Software transformiert werden soll, und das von diesem Gegenstand gebildete Modell stehen im Mittelpunkt dieser Betrachtung. Dabei wird Softwareentwicklung als Transformationsprozeß verstanden, der eine Reihe von Transformationen umfasst, vgl. Abb. 1.

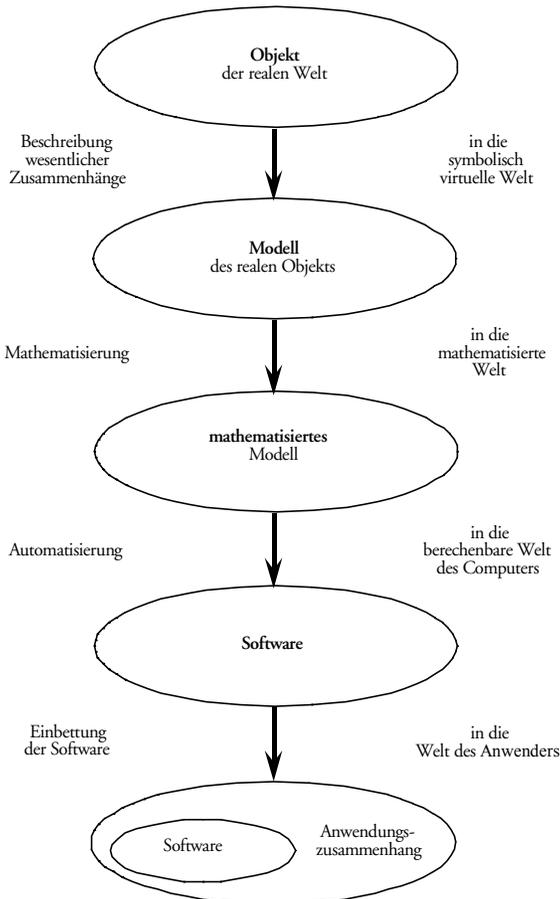
Mit jeder dieser Transformationen ist eine gewisse Reduktion<sup>6</sup> des ursprünglichen Untersuchungsobjektes verbunden, dessen sich die Beteiligten an diesem Prozeß bewußt sein sollten.

Frage: *Inwieweit läßt sich dieser Transformationsprozeß bzw. lassen sich diese Transformationen methodisch kontrollieren?*

Betrachten wir hierfür stellvertretend die Transformation „von der realen Welt (reales Objekt) in die symbolisch virtuelle Welt (Modell des realen Objektes)“.<sup>7</sup>

6 Fuchs-Kittowski, K., Reduktive Methode und Reduktionismus in den Biowissenschaften. - In DZfPh. 5(29)1981, S. 503 - 516.

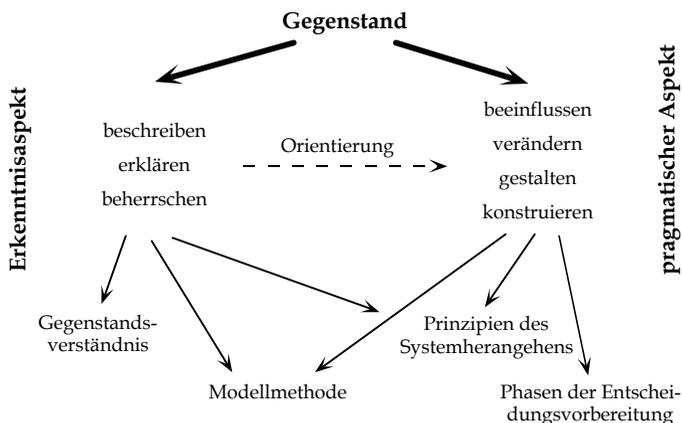
7 Zu den anderen Transformationen: s. u.a. Dahme, Ch. / Raeithel, A. (Fußnote 3), Dahme, Ch., Softwareentwicklung mit HyperCard - Benutzerfreundliche Interfacegestaltung, Addison-Wesley 1995 (bis Herbst 1995: Berlin: Verlag Technik.); Dahme, Ch., Zu einigen Voraussetzungen und Möglichkeiten der Synergetik in den Sozialwissenschaften. - In: Ethik und Sozialwissenschaften. 7(1996) 4, S. 605 - 607.

Abbildung 1 *Softwareentwicklung als Transformationsprozeß*

Wenn wir uns wissenschaftlich mit einem Objekt beschäftigen, kann es zwei unterschiedliche Richtungen für unser Interesse geben, die jedoch nicht losgelöst voneinander existieren (s. Abb. 2):

1. in der ersten Richtung steht der Erkenntnisaspekt und
2. in der zweiten der pragmatische Aspekt im Vordergrund.

Letzteres ist typisch für die Softwareentwicklung. Aus wissenschaftstheoretischer Sicht gibt es nun verschiedene Herangehensweisen, Methoden und Prinzipien,

Abbildung 2 *Interesse für einen Gegenstand*

die unser Handeln in solchen Situationen orientieren können. Die aus systemanalytischer Sicht interessantesten sind hier (s. Abb. 2) genannt.<sup>8</sup>

Hier nur kurz ein paar Anmerkungen zum Gegenstandsverständnis und zur Modellbildung<sup>9</sup>:

In der Entwicklung einer Wissenschaftsdisziplin gibt es unterschiedliche Phasen in der Durchdringung und damit auch im Verständnis eines Gegenstandes, die aufeinander aufbauen, sich teilweise wiederholen, aber auch gegenseitig durchdringen und ergänzen. Die wichtigsten Phasen in bezug auf die Entwicklung des Gegenstandsverständnisses sind:

1. Die beschreibende bzw. ganzheitlichorientierte Phase.
2. Die Phase der Orientierung auf die Teile des Ganzen bzw. die analytische Phase.
3. Die Phase der Re-Orientierung auf das Ganze bzw. die synthetische Phase.<sup>10</sup>

8 siehe Dahme, Ch. (Fußnote 2), S. 16 ff.

9 ausführlich s. dito (Phasen des Gegenstandsverständnisses: S. 16 ff.; Modellmethode: S. 21 ff.)

10 „Während es in der 1. Phase um die Beschreibung des Objekts und seine Abgrenzung von der Umwelt geht, wird in der 2. Phase erkannt, daß das Objekt zerlegbar ist, und es werden Erkenntnisse über die Teile dieses Objektes gewonnen. In der 3. Phase ist wieder das Ganze der Gegenstand der Untersuchung, jedoch jetzt unter dem Aspekt der Erklärung und Beherrschung des Ganzen, aus dem Zusammenwirken der Teile.“ dito S. 18.

Frage: *Wie kann ein Außenstehender (Beobachter, Softwareentwickler) nun erkennen, in welcher Phase sich eine Wissenschaftsdisziplin bzw. Anwender dieser Disziplin befinden?*

Für unsere Zwecke ist hier der Zusammenhang zwischen der Phase und den für diese Phase typischen Methoden interessant. Charakteristisch für die erste Phase ist die Methode der Beobachtung (einschließlich statistischer Auswertungsmethoden), für die zweite Hypothesentestung im Experiment, während in der dritten Phase Systemeffekte (auch kooperative, bzw. synergetische Effekte genannt) im Mittelpunkt stehen. So kann man aus den genutzten bzw. beobachteten Methoden in gewisser Weise auf die Phase schließen.

Ein Modell<sup>11</sup> muß nicht immer bewußt gebildet worden sein! So kann man z.B. durch Probieren, Skizzieren, „Basteln“ auf indirekte Weise - im Hintergrund - zu einem Modell kommen, ohne daß man sich dieses Modells bewußt werden muß.

Darüber hinaus gibt es zwei Hauptwege, um direkt (aber auch methodisch kontrolliert und wissenschaftlich begründet) zu einem solchen Modell zu kommen<sup>12</sup>: durch *phänomenologische Modellbildung oder durch theorieorientierte Modellbildung*.

*Konsequenz: Softwareentwicklung ist eigentlich ein interdisziplinärer Prozeß, in dem neben der Informatik die Disziplin zu integrieren ist, zu der der Gegenstand gehört, der (zum Teil) in Software transformiert werden soll, d.h. die Disziplin, die die Grundlagen für die Modellbildung liefert. Das können gegebenenfalls auch mehrere wissenschaftliche Disziplinen sein.*

## 2. Die tätigkeitsbezogene Dimension

Kommen wir nun zur 2. Dimension, die bei der Softwareentwicklung zu berücksichtigen ist - im Sinne von Automatisierung menschlicher Tätigkeit.

Diese Dimension beschreibt den tätigkeitstheoretisch begründeten Anteil der Softwareentwicklung. Auch diese Dimension soll hier nur auszugsweise - schlagwortartig - umrissen werden<sup>13</sup>.

11 Statt „Modell“ sagt man auch, daß man eine „Vorstellung“ davon braucht, was man in der und mit der Software abbilden will.

12 dito, S. 26 ff.

13 ausführlich siehe: Dahme, Ch. / Raeithel, A. (Fußnote 3) sowie Dahme, Ch., An Activity-theoretical Approach - a Way to Useful Software - in memory of Arne Raeithel. - In: Mind, Culture, and Activity. Lawrence Erlbaum Associates, Mahwah New Jersey (in print).

Wie könnte man aus dieser Sicht Software beschreiben?

1. Software ist
  - Resultat der Transformation von Anteilen realer bzw. möglicher menschlicher Tätigkeit in eine maschinelle Form,
  - Vergegenständlichung von menschlichen Fähigkeiten.
2. Im Rahmen einer Anwendung<sup>14</sup> ist Software Mittel einer Tätigkeiten. Als Mittel einer Tätigkeiten kann Software Medium, Werkzeug oder Automat sein.
3. Der Sinn der Nutzung eines solchen Mittels (Software) kann darin bestehen, die Tätigkeiten der Anwender
  - zu erleichtern und zu unterstützen oder
  - deren Effektivität zu erhöhen, aber auch
  - bestimmte neuartige Tätigkeiten überhaupt erst zu ermöglichen.

Frage: *Was läßt sich von einer menschlichen Tätigkeit in Software übertragen und wie kommt man zu diesem Anteil?*<sup>15</sup>

Der potentiell in Software übertragbare Anteil ist durch folgendes charakterisiert:

1. Er bezieht sich auf Anteile einer inneren<sup>16</sup>, orientierenden<sup>17</sup> Tätigkeit oder läßt sich in solche transformieren.
2. Diese Anteile liegen als Operationen<sup>18</sup> vor oder lassen sich von Handlungen in Operationen überführen (Operationalisierbarkeit).
3. Es liegt Wissen vor, mit dem sich diese Operationen (vollständig) reproduzieren lassen – reproduzierbares Wissen.
4. Dieses Wissen ist mitteilbar.
5. Es kann in öffentliches Wissen<sup>19</sup> überführt werden.

14 Wenn ich von Anwendung von Software spreche, dann schließe ich die technische Realisierung mit ein!

15 Hier haben wir das Transformationsproblem, was ich am Anfang erwähnt habe, wieder. Welcher Anteil der Realität (beschrieben als Tätigkeit) läßt sich in Software abbilden/übertragen/transformieren?

16 In der Tätigkeitstheorie unterscheidet man zwischen der äußeren, praktischen Tätigkeit und der inneren, psychische Tätigkeit.

17 Ergebnisse der orientierenden Tätigkeiten dienen vor allem dazu, andere Tätigkeiten zu regulieren, sie auf den sozialen und gegenständlichen Kontext auszurichten und ihre innere Struktur in Richtung einer besseren, leichter vermittelbaren Organisation zu ordnen (s. u.a. Dahme Ch. / Raeithel A. - Fußnote 3 - S. 8).

18 Operationen: Verfahren zur Realisierung einer Handlung.

19 Mitteilbares Wissen, das nicht mehr an die Person gebunden ist, die es hervorgebracht hat, weil es in einer mitgeteilten, öffentlichen, kopierbaren und für das angesprochene Publikum verständlichen Form vorliegt, nennen wir *öffentliches Wissen*.

Nur öffentliches Wissen, das sowohl ein reproduzierbares Verfahren als auch das Modell des Gegenstandes, auf das sich das Verfahren bezieht, betrifft und das bezogen auf die jeweilige Expertengemeinschaft allgemein verständlich ist, kann transparent und nachvollziehbar in Software übertragen werden.

Die Tätigkeitstheorie hat die auch für die Softwareentwicklung wichtigen drei Analyseebenen einer Tätigkeit<sup>20</sup> herausgearbeitet (s. Tabelle 1). Eine Tätigkeit wird durch ein Motiv, dem ein Bedürfnis zugrunde liegt, ausgelöst, ist auf einen Gegenstand gerichtet, der zur Bedürfnisbefriedigung geeignet ist, und ist mit der Befriedigung dieses Bedürfnisses beendet.

Tabelle 1 <i>Bedeutung der drei Analyseebenen einer Tätigkeit für die Softwareentwicklung</i>		
Analyseebene	Leitfrage	Beispiele für softwareentwicklungsrelevante Gestaltungsziele
Tätigkeit	Warum (Motiv)	Benutzer- und Fehlerfreundlichkeit Einbettung in die Anwendertätigkeit
Handlung	Was (Ziel)	Dialoggestaltung, Handhabbarkeit, visuelle Rückmeldungen, Hilfen
Operation	Wie (Verfahren)	direkt in ein Programm transformierbare, effiziente und revidierbare Objekte, Methoden und Bot- schaften bzw. Algorithmen und Datenstrukturen

Dabei bezieht sich das Motiv einer Tätigkeit auf die Frage: „Warum tut ein Mensch das?“.

Um ausgehend von einem Motiv zu einer Bedürfnisbefriedigung zu kommen, muß die Person eine Situation erreichen, die dieses ermöglicht - angestrebende Situation. Wird diese Situation bewußt zielgerichtet angestrebt, so wird dieser Prozeß *Handlung* genannt. Bei einer Handlung ist zwischen dem *Ziel* - was man erreichen will - und dem *Verfahren* - wie man dieses Ziel erreichen will - zu unterscheiden. Die Verfahren, die eine Handlung konkret realisieren, werden als *Operationen* bezeichnet. Da Operationen nur von den Bedingungen abhängen, unter denen sie stattfinden können, haben sie die Potenz zur *Automatisierung*, d.h., sie können gegebenenfalls völlig aus einer Tätigkeit herausgenommen und einem Automaten übertragen werden.

20 s. Leontjew, A. N., Tätigkeit, Wissen, Persönlichkeit. Stuttgart: Klett 1977.

Beispiele für die Automatisierung menschlicher Tätigkeit sind u.a.:

- Jacquardscher Webstuhl
- Analytical Engine von Babbage
- Holleriths Zähl- und Sortiermaschine
- Geld- und Fahrkartenautomaten
- Automatisierte Steuerungs- (CNC) und Ablaufsysteme (Workflow)

Dabei sind die Operationen die für die funktionelle Seite der Softwareentwicklung wichtigste Ebene, während die anderen beiden Ebenen für die Einbettung der Software in den Anwendungszusammenhang wesentlich sind.

Zusammenhang zwischen der 1. und 2. Dimension

1. Menschliche Tätigkeit bezieht sich immer auf einen Gegenstand (Gegenstand der Tätigkeit). Man spricht daher auch von einer gegenständliche Tätigkeit. Während sich die 1. Dimension mit dem Gegenstand<sup>21</sup> beschäftigt, wendet sich die 2. Dimension der Tätigkeit an sich zu.
2. Innere, orientierende Tätigkeit findet auf einer symbolisch virtuellen Ebene, d.h. in einer symbolisch virtuellen Welt, statt. Mittel und Gegenstand einer solchen Tätigkeit sind Bilder, Symbole und Sprache, insbesondere aber auch gegenständliche Modelle - d.h. Repräsentanten realer Objekte auf symbolischer Ebene - sowie Software.

Der Sinn aller virtuellen Welten besteht darin, uns in den anderen realen Welten zu orientieren, sie verstehen, prognostizieren und gestalten zu können. Geeignete Software, zweckmäßig eingesetzt, kann hier einen guten Beitrag leisten, man denke etwa an entscheidungsunterstützende Systeme oder Simulationen von physikalisch-chemisch-geologischen Prozessen wie Wetter, Ökosystemen, Verbrennungsprozessen und so fort.

Ein möglicher Weg, um zu einer solchen symbolisch virtuellen Welt zu kommen, ist die Bildung oder die Aneignung von Modellen und Theorien im entsprechenden Gegenstands- bzw. Objektbereich.

### 3. *Die interkulturelle bzw. kommunikative Dimension*

Versteht man Softwareentwicklung als eine Dienstleistung, so erfordert dieses eine Kommunikation zwischen den Dienstleistung in Anspruch nehmenden (z.B. den Anwendern, den Bestellern, den Betreibern) und den Dienstleistenden (z.B. den Softwareentwickler). Dabei wird häufig unterstellt, daß diese Kommunikation „problemlos“ erfolgt, d.h., das durch die Kommunikation selbst keine

21 Dabei kann dieser Gegenstand selbst wieder eine Tätigkeit sein.

Probleme erzeugt werden. Diese Annahme ist jedoch mit einem hohen Risiko behaftet. So ist es meistens falsch, anzunehmen, daß

1. der Anwender bei Projektbeginn genau weiß, was er will,
2. der Anwender das, wovon er weiß, daß er es will, vollständig mitteilen kann,
3. der Entwickler ausreichend verstanden hat, was der Anwender mitteilen konnte,
4. das kommunizierte Wissen ausreicht, um die vom Anwender gewollten Funktionen produzieren zu können,
5. der Anwender versteht, was der Entwickler außer den vorgelegten Beispielen noch leisten könnte,
6. der Anwender also wüßte, welche Software möglich wäre, wenn der Entwickler besser über seine Bedürfnisse unterrichtet wäre.

Eine recht naive und gefährliche Annahme versteckt sich außerdem in der Zuversicht, daß Entwickler und Anwender es schon rechtzeitig merken werden, wenn Mißverständnisse auftreten, und diese durch Nachfragen leicht beseitigen können. Vielmehr ist es die Regel, daß Verständnislücken auf beiden Seiten viel zu spät und manchmal gar nicht aufgeklärt werden, so daß für den Anwender Unnützes mit viel Mühe implementiert wird, und zugleich zentrale, aber bislang implizit gebliebene Anforderungen gerade noch nicht erfüllt sind.<sup>22</sup>

Damit eine solche „Kooperation“ (u.a. im Sinne einer Dienstleistung) funktioniert, wird eine gemeinsame Sprache benötigt, die beide Parteien verstehen. Dies erfordert einen gemeinsamen Kulturbereich, der das Verständnis bezüglich des Untersuchungsobjektes (einschließlich dessen Umgang) überdeckt. Häufig ist es aber so, daß Softwareentwickler und Anwender unterschiedliche Kulturen herausgebildet haben, die sich kaum überdecken.

Um brauchbare Software herstellen zu können, sollte die Kultur der Anwender berücksichtigt werden, da die Software für den Anwender entwickelt wird und in seine Kultur eingebettet werden soll. Dazu müssen sich aber beide dieses Situation bewußt sein.

Es gibt nun verschiedene Möglichkeiten bzw. Ansätze, auf dieses Phänomen zu reagieren bzw. mit diesem umzugehen, d.h. einen möglichst großen „gemeinsamen Kulturbereich“ bzw. Verständnis für die Kultur des anderen zu erreichen:

1. Herausbildung bzw. Förderung einer gemeinsamen Sprache.  
- Zukunftswerkstätten.<sup>23</sup>

22 s. Dahme, Ch. / Raethel, A. (Fußnote 3), S. 5.

23 Kuhnt, B. / Müllert, N.R., Moderationsfibel: Zukunftswerkstätten verstehen, anleiten, einsetzen. Münster: Ökotopia 1996.

- Prototyping<sup>24</sup> - dabei kann der Prototyp selbst als Katalysator für die Herausbildung einer gemeinsamen Sprache dienen.
- 2. Kultur eines „Fremden“<sup>25</sup> erkennen.
  - mit Hilfe ethnographischer Methoden.<sup>26</sup>
- 3. zwischen Kulturen vermitteln.<sup>27</sup>

Bei 2. und 3. geht es u.a. darum, die Kultur des „Anderen“ wahrzunehmen, etwa in dem Sinne:

1. Wie reflektiert der Softwareentwickler Wirklichkeit als Grundlage für die Softwareentwicklung?
2. Wie reflektiert der Anwender Wirklichkeit als Grundlage für die Problem-, Anforderungs- bzw. Modellbeschreibung?

Dabei finden solche inneren, orientierenden Tätigkeiten auf einer symbolischen Ebene bzw. in einer symbolischen Welt statt (semiotischer Aspekt), die sich einerseits auf eine „private Welt“ rein geistiger Vorstellungen und Vorgänge bezieht. Bei einer Kooperation erweitert sich dieses „Innen“ - in Abhängigkeit von der gemeinsamen, kulturellen und sozialen Einbettung der *kooperativ* tätiger Personen - zusätzlich zu einem sozialen Raum, in dem sich die Beteiligten aneinander und miteinander orientieren können. Solche interpersonellen, interkulturellen, kooperativen Prozesse finden in der Regel als Selbstorganisationsprozesse statt.<sup>28</sup>

Das Kulturproblem spiegelt sich aber auch in der letzten Transformation (s. Abb. 1) wider. Diese bezieht sich auf die Oberflächen- und Interaktionsebene, die mit der Funktionsebene<sup>29</sup> harmonisieren sollte. Dazu sollten die virtuellen Objekte der Designphase am besten so umgebaut werden, daß sie für den Anwender als leicht verständliche und wirksame Software-Objekte ihre orientierende, virtuelle Kraft entfalten können. Hier spielen u.a. Benutzer- und Fehlerfreundlichkeit, sowie die Einbettung der Software in die existierende

24 Floyd, C., A Systematic Look at Prototyping. - In: Approaches to Prototyping. Hrsg. v. R. Budde / K. Kuhlenkamp / L. Mathiassen / H. Züllighoven. Proceedings of the Working Conference on Prototyping, Berlin, Heidelberg: Springer Verlag 1984, S. 1 - 18.

Floyd, C., Software-Engineering - und dann?. - In: Informatik Spektrum. 1(1994)17, S. 29 - 37.

25 aus der Sicht des Softwareentwickler der Anwender und umgekehrt.

26 s. Raeithel A. (Fußnote 3) insbesondere „Zur Ethnographie der kooperativen Arbeit“.

27 Kuhnt, B., Systemische Beratung in kooperativen Softwareprojekten. - In: Informatik-Spektrum. 20 (1997)1, S. 29 - 32.

Kuhnt, B., Softwareentwicklung als systemische Intervention in Organisationen. Dissertation an der Uni. Zürich 1998.

28 s. Dahme Ch. (Fußnote 2), S. 276 ff., sowie Raeithel, A. (Fußnote 3) S. 151 ff.

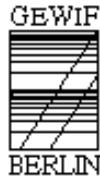
29 die sich auf die Transformationen vom „Objekt der realen Welt“ bis „Software“ bezieht.

Anwendertätigkeit eine wichtige Rolle (s. auch Tabelle 1) Die Anwender begrüßen in der Regel die Übernahme ihrer spezifischen Sprache, ihrer üblichen Arbeitsschritte in Dialoge und Menüs. Dies erfordert die Nachbildung ihrer gewohnten Mittel und Materialien als Software-Objekte.



---

Gesellschaft für  
Wissenschaftsforschung



Klaus Fuchs-Kittowski,  
Heinrich Parthey,  
Walther Umstätter,  
Roland Wagner-Döbler (Hrsg.)

**Organisationsinformatik  
und Digitale Bibliothek  
in der Wissenschaft**

Wissenschaftsforschung  
Jahrbuch 2000

**Sonderdruck**

Mit Beiträgen von:

*Manfred Bonitz • Christian Dahme • Klaus  
Fuchs-Kittowski • Frank Havemann •  
Heinrich Parthey • Andrea Scharnhorst •  
Walther Umstätter •  
Roland Wagner-Döbler*

Wissenschaftsforschung  
Jahrbuch **2000**

---

**Organisationsinformatik und Digitale Bibliothek in  
der Wissenschaft:**

Wissenschaftsforschung Jahrbuch 2000 / Klaus  
Fuchs-Kittowski; Heinrich Parthey; Walther  
Umstätter; Roland Wagner-Döbler (Hrsg.). Mit  
Beiträgen von Manfred Bonitz ... - Berlin:  
Gesellschaft für Wissenschaftsforschung 2001.

Das Werk ist in allen seinen Teilen urheberrechtlich  
geschützt.

Jede kommerzielle Verwertung ohne schriftliche  
Genehmigung des Verlages ist unzulässig. Dies gilt  
insbesondere für Vervielfältigungen, Übersetzungen,  
Mikroverfilmungen und die Einspeicherung und  
Verarbeitung in Systeme(n) der elektronischen  
Datenverarbeitung.

© Gesellschaft für Wissenschaftsforschung,  
1. Auflage 2001  
Alle Rechte vorbehalten.

Verlag:  
Gesellschaft für Wissenschaftsforschung  
c/o Prof. Dr. Walther Umstätter, Institut für  
Bibliothekswissenschaft der Humboldt-Universität zu  
Berlin, Dorotheenstr. 26, D-10099 Berlin

Druck: BOOKS on DEMAND GmbH,  
Gutenbergring, D-22848 Norderstet

ISBN 3-934682-34-0

Preis: 14,00 €