

# Historische, wissenschaftstheoretische und kulturelle Wurzeln der angewandten Informatik

## Versuch einer Bestimmung in Form von Thesen

Christian Dahme, Humboldt-Universität zu Berlin

### 1. Wissenschaftshistorisch lassen sich mindestens 3 Wurzeln für die Nutzung des Computers und als Grundlagen für Software erkennen:

- a) Rechner als Mittel zur Unterstützung/Automatisierung von Berechnungen<sup>1</sup>
- b) Automatisierung von Tätigkeiten
- c) Software zur Problemlösung

Wissenschaftshistorisch ist a) m.E. die älteste. Sie ist u.a. mit dem Wunsch verbunden, aufwendige Berechnungen durch "Rechenwerkzeuge" später Rechenautomaten zu unterstützen und Fehler zu vermeiden.

Hier geht es in gewisser Weise um die **Automatisierung des Rechnens** oder allgemeiner um die **Automatisierung von Mathematik**

Dabei sind zwei Richtungen zu unterscheiden:

- a1) zur Unterstützung wissenschaftlicher Berechnungen  
als Mittel zum Berechnen von Modellen (Durchführung von Modellexperimenten)  
von Schickardts Rechner zur Unterstützung der Berechnungen von Kepler bis hin zur Nutzung von Computern für Modelle der modernen Physik und Naturwissenschaften und damit auch als Grundlage für die Entwicklung von Atombomben, Raketen, Raumfahrt u. dgl. (im Sinne von angewandten Naturwissenschaften)
- a2) zur Unterstützung kommerzieller Berechnungen  
als Mittel zum Berechnen von ökonomischen Sachverhalten und Modellen  
von Pascal, der die Verwaltungsarbeit seines Vaters unterstützen (rationalisieren) wollte, damit dieser mehr Zeit für die Familie hat, über Charles Thomas<sup>2</sup> bis hin zu ökonomischen Modellen (einschließlich Operations Research) wie z.B. Weltmodellen des Club of Rome bzw. IIASA<sup>3</sup>

b) bezieht sich auf die **Automatisierung** (von Teilen) **von Tätigkeiten** und geht insbesondere auf die Automatisierung des Webens (Lochplatte von Falcon, Jacquardscher Webstuhl) zurück.

Ausgehend von diesem Prinzip der "Programmsteuerung" entwickelte Herman Hollerith<sup>4</sup> ein elektromechanisches Lochkartensystem für die Auswertung der Volkszählung in den USA von 1890 - im Sinne von Datenerfassung, -Bereitstellung und -Auswertung - insbesondere großer Datenmengen<sup>5</sup>.

Ausgehend von den Erfahrungen bei der Volkszählung wurde dieses Verfahren von staatlicher bis hin zu betrieblicher Verwaltung wie unterschiedlichste Statistiken, Lagerhaltung, Kostenerfassung oder Lohnabrechnung angewandt.

c) ist wissenschaftshistorisch am jüngsten (entsteht etwa in den 50er Jahre des 20. Jahrhunderts). Sie ist mit der kognitiven Psychologie (insbesondere der kybernetischen Variante) verbunden und führte zum kognitivistischen Paradigma (Kognitivismus<sup>6</sup>): psychische Phänomene werden als „Informationsverarbeitung“ rekonstruiert.

---

<sup>1</sup> s. u.a. auch W.&U. Brauer "Änderungen von Forschungszielen und Denkgewohnheiten" in: W. Langenheder, G. Müller, B. Schinzel: Informatik cui bono ? ; GI-FB 8 Fachtagung, Freiburg, 23.-26. September 1992; Springer-Verlag, Berlin

<sup>2</sup> Charles Thomas beschäftigte als Chef zweier Versicherungsgesellschaften in Paris eine große Anzahl von "Rechnern" (d.h. Menschen, die für ihn rechneten). So kam er auf die Idee, eine Rechenmaschinenproduktion großen Stils aufzuziehen, um diese "Rechner" durch Rechenmaschinen zu ersetzen. Seine erstes "Arithmometer" war 1820 fertig. Insgesamt wurden ungefähr 1500 solcher Rechenmaschinen in seiner Werkstatt gebaut.

<sup>3</sup> International Institute for Applied Systems Analysis

<sup>4</sup> Seine Firma war Vorläufer von IBM

<sup>5</sup> das führte u.a. zur Entstehung von Datenbanken (Informationssystemen)

<sup>6</sup> einer der damals führenden kybernetisch-mathematischen Vertreter der kognitiven Psychologie war Friedhard Klix (Information und Verhalten; Berlin 1971); einen guten Überblick über den Kognitivismus findet man bei Günter Sämmer: Paradigmen der Psychologie; 8. Kapitel "Das Paradigma des Kognitivismus" (<http://home.arcor.de/saemmer/dis/diskap8.htm>).

## 2. Den in 1. dargestellten historischen Wurzeln können unterschiedliche Interpretationen von Software zugeordnet werden.

### a) Der (gegenstandsorientierte) Modell-Ansatz<sup>7</sup>

Software basiert auf einem Modell<sup>8</sup> des Gegenstandes, das in Software übertragen werden soll. Software kann dann als Automatisierung der "Berechnung" von (mathematisierten) Modellen verstanden werden (im Sinne von 1a).

Grundlage für die Software ist hier ein Modell des Gegenstandes, und damit die Disziplin<sup>9</sup> des Gegenstandsbereiches, zu dem der Gegenstand<sup>10</sup> gehört.

==> Aus dieser Sicht ist Softwareentwicklung ein interdisziplinärer Prozeß.

Dabei kann die Software selbst Rückwirkungen auf den ursprünglich abgebildeten Gegenstand haben, so dass man verkürzt auch spricht von:

Software als (Mittel zur) System- bzw. Organisationsgestaltung

### b) Der tätigkeitstheorieorientierte Ansatz<sup>11</sup>

Software als Automatisierung von operationalisierbaren Anteilen (innerer) menschlicher Tätigkeit<sup>12</sup> besser

Software als Resultat der Transformation von Anteilen realer bzw. möglicher menschlicher Tätigkeit in eine maschinelle Form

als Vergegenständlichung von menschlichen Fähigkeiten

Grundlage für die Softwareentwicklung ist hier die Tätigkeitstheorie.

In diesem Sinne könnte man die Tätigkeitstheorie als eine "Grundlagendisziplin" für die Softwareentwicklung ansehen.

Es wird danach gefragt: Welche Anteile der Tätigkeit lassen sich in Software transformieren.

### c) Der kognitivistische Ansatz

Aus der kognitiven Psychologie heraus entwickelte sich das Denkmuster:

Software als Problemlösung besser Software als Automatisierung einer Problemlösung oder

Software als Automatisierung (Maschinisierung) eines Algorithmus, der wiederum (selbst) eine Problemlösung repräsentiert.

Das führte u.a. zu GPS (General Problem Solver) und zur KI.

Diese Art der Charakterisierung von Software ist heute noch weit verbreitet und wird gerne in der theoretischen Informatik bevorzugt, da sie explizit auf dem Begriff des Algorithmus basiert.

## 3. Zwischen diesen Interpretationen kann ein Zusammenhang hergestellt werden.

Alle drei Ansätze haben (aus unterschiedlichen Perspektiven) ihre Berechtigung.

Sie stehen aber in einem gewissen Zusammenhang:

- Da Problemlösen eine innere Tätigkeit ist, schließt 2b) 2c) ein.

- Aus tätigkeitstheoretischer Sicht kann eine Zusammenhang zwischen 2a) und 2b) dadurch herstellen werden, dass man mit 2b) die Tätigkeit und mit 2a) den Gegenstand der Tätigkeit beschreibt.

Aus dieser Sicht sind beide Ansätze notwendig, indem sie sich ergänzen.

- Andererseits kann man die Tätigkeit selbst zum Gegenstand machen und dann wird 2b) ein Teil von 2a).<sup>13</sup>

Aus dieser Sicht lassen sich die Interpretationen 2b) und 2c) auf 2a) zurückführen.

---

<sup>7</sup> naturwissenschaftliches insbesondere physikalisches Denkmuster aber auch kybernetisch-systemtheoretisches Herangehen, s. u.a. Dahme, Ch.: Systemanalyse menschlichen Handelns - Grundlagen und Ansätze zur Modellbildung, Westdeutscher Verlag 1997, Opladen, insbesondere „1 Methodologische Positionen“

<sup>8</sup> unabhängig davon, ob das Modell bewußt oder unbewußt gebildet wurde

<sup>9</sup> das können auch mehrere Disziplinen sein.

<sup>10</sup> und die Fragestellung gehört, die mit Hilfe von Modellexperimenten beantwortet werden soll

<sup>11</sup> siehe: Dahme, Ch.; Raeithel, A.: Ein tätigkeitstheoretischer Ansatz zur Entwicklung von brauchbarer Software, in: Informatik-Spektrum · 20 · Heft 1 · Februar 1997, S. 5-12 (<http://waste.informatik.hu-berlin.de/Dahme/Da%2FRa.html>)

Dieser Ansatz präferiert das OO-Paradigma

<sup>12</sup> ein ähnlicher Ansatz bei F. Nake: Software als Maschinisierung von Kopfarbeit

<sup>13</sup> siehe Dahme, Ch.: Systemanalyse menschlichen Handelns - Grundlagen und Ansätze zur Modellbildung, Westdeutscher Verlag 1997, Opladen

**4. Softwareentwicklung kann als mehrstufiger Transformationsprozeß interpretiert werden.**

Hierbei wird vom Modell-Ansatz (siehe 2a)) ausgegangen:

Bei der ersten Transformation (von der realen in die symbolisch virtuelle Welt<sup>14</sup> - siehe Abb. unten) spielt die erkenntnistheoretische Position des "Modellentwicklers" eine wesentliche Rolle, d.h. hier geht es um die Frage: Wie reflektiert er Wirklichkeit als Grundlage für die Modellbildung und damit für die Softwareentwicklung ?

Dieser Transformationsprozeß ist nicht zu verwechseln mit einem Vorgehensmodell der Softwareentwicklung. Bei letzterem steht die Organisation der Herstellung von Software im Mittelpunkt. Der Transformationsprozeß charakterisiert primär das Ziel<sup>15</sup>, aber noch nicht „Wie man das Ziel erreicht“. Er legt aber Randbedingungen hierfür fest. Bei der Herstellung von Software können diese Transformationen mehrfach in unterschiedlicher Reihenfolge durchlaufen werden. Beim Prototyping werden manchmal sogar Teile der letzte Transformation (im Sinne von Human-Interface-Prototyping) vorgezogen, um diesen Prototyp als Katalysator zur Modellbildung und Spezifikation zu nutzen.

**5. Wenn Softwareentwicklung zu einer Dienstleistung<sup>16</sup> wird, dann wird Softwareentwicklung zu einem interkulturellen Prozeß.**

Wenn Softwareentwicklung zu einer Dienstleistung wird, stehen sich mindestens zwei Kulturen gegenüber - die der Anwender und die der Entwickler -, deren Vertreten mit einander kommunizieren und gegebenenfalls kooperieren müssen/sollten. Dabei können die unterschiedlichsten Kommunikations-, Kultur- und Koordinationsprobleme auftreten.

Das Kriterium brauchbar<sup>17</sup> - im Sinne von brauchbarer Software - ist ein typisches Kriterium für diesen interkulturellen Prozeß.

Solche interpersonellen, interkulturellen, kooperativen Prozesse finden in der Regel als Selbstorganisationsprozesse<sup>18</sup> statt. Selbstorganisationsprozesse lassen sich nicht direkt sondern nur über ihre Randbedingungen beeinflussen.

Fazit: In diesem Sinne kann Softwareentwicklung als ein Transformationsprozeß, der in einen Selbstorganisationsprozeß eingebettet ist, interpretiert werden.

---

<sup>14</sup> statt symbolisch virtuelle Welt könnte man auch semiotische Welt sagen.

<sup>15</sup> bzw. Teilziele bezogen auf die einzelnen Transformationen

<sup>16</sup> Wenn Software ingenieurmäßig/industriell entwickelt und vertrieben werden soll, dann wird Softwareentwicklung zu einer Dienstleistung.

<sup>17</sup> spiegelt sich insbesondere in der letzten Transformation in 4. wider.

<sup>18</sup> siehe Dahme, Ch.: Systemanalyse menschlichen Handelns; Kap. „5.4 Kooperation und Selbstorganisation“ – erläutert an Hand eines Beispiels

