

## PROGRAMMIEREN MIT MENTALEN MODELLEN?!

Möbus, Claus<sup>1</sup>; Schröder, Olaf und Colonius, Hans

Oldenburg

Die Entwicklung von Software kann als kognitive Transformation angesehen werden, die eine Aufgabenbeschreibung, ein Anforderungsprofil oder gar eine Funktionsspezifikation in die strukturierte Beschreibung des Computerprogramms überführt.

Das Zielgebiet des Transformationsprozesses (die Menge aller Computerprogramme) kann dabei geordnet werden nach Programmierstilen, die sich im Laufe der Entwicklung in Informatiklehre und Forschung herauskristallisiert haben. Wir kennen hier den imperativen, den funktionalen, den prädiktiven sowie den objektorientierten Stil. Taxonomiemerkmale sind u.a. Charakteristika der Ablaufsteuerung und der Nebeneffektfreiheit der Berechnungen. Diese wirken sich auf die Belastung des Arbeitsgedächtnisses ("working memory load") und damit auf die Fehlerfreiheit der Programme aus, was fatale Konsequenzen haben kann. Um das Erlernen dieser Stile zu erleichtern, schlagen Psychologen wie Informatiker Modelle vor, die vom Syntaxwissen abstrahieren und vorwiegend Semantik- bzw. Berechnungswissen im Sinne operativer Semantik transportieren. Dabei wird auf dem naiven Vorwissen der angehenden Programmierer aufgebaut. Diese Modelle, die die Rolle von mentalen Modellen spielen sollen, können den einzelnen Programmierstilen zugeordnet werden: das "Schuhkarton"- und das "Tapezierrollenmodell" dem imperativen Stil, das "Keller"- und das "Trichtermodell" dem funktionalen Stil sowie das "Uhrenmodell" dem prädikativen Stil. Es stellt sich die Frage, ob einer dieser Vorschläge den von DuBoulay, O'Shea & Monk (1980) für Surrogate geforderten Kriterien "Einfachheit" und "Sichtbarkeit der wesentlichen Abläufe" entspricht und damit geeignet ist, als mentales Modell gelehrt und gelernt zu werden. Darüber hinaus ist zu fragen, ob ein derartiger Kandidat für ein mentales Modell so tragfähig ist, auch als konkrete Programmiersprache eingesetzt zu werden. Ist die Programmierung mit einem mentalen Modell möglich?

Wir werden in dem Vortrag erste Ansätze in dieser Richtung, die in unserem DFG-Projekt erarbeitet wurden, vorstellen. Basis hierfür soll eine abstrakte Maschine sein, die auch unter dem Namen "Formularmaschine" bekannt geworden ist und die zum mentalen Alltagsmodell des Trichters korrespondiert. Wir werden die Programmierung dieser "Gedankenmaschine" mit Hilfe von Zeichnungen an konkreten Beispielen zeigen, Bezüge zu bekannten Programmiersprachen (LISP) herstellen und aus ersten Untersuchungen an Hand von Fehlerprotokollen über Mißkonzeptionen von Schülern und Studenten berichten.

### Literatur:

DuBoulay, B., O'Shea, T. & Monk, J. (1980). The black box inside the glas box. Presenting computing concepts to novices. University of Edinburgh: Department of Artificial Intelligence

---

**Manfred Amelang**

---

**Bericht über den 35. Kongreß  
der Deutschen Gesellschaft für Psychologie  
in Heidelberg 1986**

**Band 1**

**Kurzfassungen**

herausgegeben  
im Auftrag der Deutschen Gesellschaft für Psychologie



---

**Verlag für Psychologie · Dr. C. J. Hogrefe  
Göttingen · Toronto · Zürich**

---