

Aktuelles Schlagwort

Ygamt

Was sind eigentlich Modelle?

Roland Kaschek

Universität Klagenfurt, Institut für Informatik
Email: roland.kaschek@ifit.uni-klu.ac.at

Modelle, ihre Herstellung und Nutzung, d.h. das Modellieren, spielt in der Informatik eine große Rolle. Diese große Rolle wird dem Begriff „Modell“ zwar nicht von allen Autoren zugebilligt, in [Or97] wird er eher als ein Begriff untergeordneter Bedeutung behandelt. Andererseits entsteht im Umfeld der GI zur Zeit eine Konferenzreihe, deren Gegenstand das Modellieren in der Informatik ist. Ein Tagungsbericht zur Modellierung 98 war im EMISA-Forum 2, 1998 zu finden. Im Heft 2, 1999 wird wohl ein Tagungsbericht von der Modellierung 99 abgedruckt werden. Für die EMISA sind Modelle wichtig, weil sie Entwicklungsmethoden für Informationssysteme und deren Anwendung behandelt und in viele derartige Methoden, wie etwa OMT, vgl. [R*91], Modelle dazu verwendet werden das Vorgehen bei der Entwicklung von Informationssystemen zu organisieren.

Der Begriff „Modell“ wird in der Informatik unterschiedlich verstanden. Im Englischen redet man vom „conceptual model“, also begrifflichem oder gedanklichem Modell. Im Deutschen wird vielfach der Ausdruck konzeptionelles, d.h. ein Konzept betreffendes, oder konzeptuelles, d.h. ein Konzept aufweisende, Modell verwendet. Solche Modelle müssen nicht gedanklich sein. Diese Bedeutungsverschiebung macht es zusätzlich interessant den Modellbegriff der Informatik zu untersuchen. Im deutschen Sprachkreis sind Modelle von Stachowiak besonders gründlich untersucht worden, vgl. etwa [St73, St92]. In [Pe98, S. 119-121] wurde seine „Allgemeine Modelltheorie“ jüngst wieder aufgegriffen.

Manche Autoren, wie etwa Fowler, vgl. [Fo97, S.2], bestimmen Modelle (in der Informatik) als etwas gedankliches. Eine gleichartige Auffassung wird auch z.B. von Falkenberg et al., vgl. [F*98, S. 213] sowie Lockemann und Mayr in [LM78] vertreten. Für diese Autoren ist ein Modell (in der Informatik) eine Vorstellung, die sich ein Individuum von etwas, z.B. Dingen oder Vorgängen in seiner Umgebung bildet.

Andererseits sagt Larman in [La98, s. 29]: „Ein System (sei es real oder ein Softwaresystem) ist gewöhnlich von überwältigender Komplexität, daher ist es notwendig das System derart in Teile zu zerlegen, daß man es verstehen und seine Komplexität beherrschen kann. Diese Teile können als Modelle repräsentiert werden, die wesentliche Aspekte des Systems beschreiben und hervorheben.“ In ähnlicher Weise äußern sich Blaha und Premerlani in [BP98, S. 6]: „In entitätsbasierten Ansätzen, die in [Ch76] ihren Ursprung haben, registriert man Eintitäten in der Realität, beschreibt sie und stellt Beziehungen zwischen ihnen fest. Die OMT-Methodik verfolgt einen entitätsbasierten

Ansatz.“ In diesen Auffassungen ist ein Modell etwas physisch vorhandenes oder schriftlich fixiertes, ein Muster oder (komposites) Zeichen.

Wieder andere, wie etwa die Autoren in [K*98] oder Vossen, vgl [Vo94], verwenden zwar das Wort Modell definieren aber den Begriff „Modell“ nicht. Überhaupt ist natürlich die Bedeutung unterschiedlicher Ansichten zu derartigen Fragen in der Praxis durch andere dort auftretende Probleme stark relativiert. Ein klares Verständnis der wichtigsten Fachbegriffe schadet aber auch in der Praxis nicht.

Meyer, vgl. [Me97, S. 230] sagt: „Ihnen ist vielleicht aufgefallen, daß in der obigen Diskussion (wie auch in früheren Diskussionen zu verwandten Themen) jede Referenz auf die ‚Realität‘ vermieden wurde. Statt dessen wurde oben mit Blick auf das, was die Software repräsentiert lediglich von dem ‚modellierten System‘ gesprochen. Diese Unterscheidung wird gewöhnlich nicht getroffen. In vielen Diskussionen im Umfeld der Informationsmodellierung wird von ‚Modellierung der Realität‘ geredet. In Büchern über objektorientierte Analyse wird eine Vielzahl ähnlicher Ausdrücke verwendet. Wir sollten daher kurz über diesen Begriff nachdenken. Es ist nämlich aus wenigstens 4 Gründen irreführend über die ‚Realität‘ hinter einem Software System zu reden.“

Meyer fährt dann fort und nennt seinen ersten Grund, daß nämlich Realität im Auge des Betrachters liege, daß, ich ergänze hier, also Wahrnehmung von Vorerfahrung und Wahrnehmungssituation bestimmt und erst, vgl. dazu auch [F*98, S. 31], durch einen Akt des Vorstellens zu einer Vorstellung, etwas bewußt wahrgenommenem, wird. Als zweiten Grund hält Meyer fest, daß Software oft auf Software angewandt werde, daß also Software und das worauf sie sich bezieht nicht grundsätzlich verschiedener Art sein müßten. Der dritte Grund Meyers ist, daß Computer und Software heute zum Bestandteil des Lebens vieler Menschen geworden und daher genau so real seien wie andere Aspekte des modernen Lebens. Schließlich sagt Meyer, daß Software nicht Modell der Realität, sondern Modell eines Modells eines Ausschnitts der Realität sei und bringt einige Beispiele für diese Behauptung, ohne aber den Begriff „Modell“ zu definieren.

Die Semiotik untersucht die menschliche Kommunikation, insoweit, als diese sich Zeichen bedient, vgl. etwa [Ec94, Kap. 2, insbes. S. 139]. Die Semiotik spielt in der Informatik eine Rolle, weil im Zuge der Entwicklung von Software anschauliche Darstellungen, sogenannte Schemata, etwa ER-, Datenfluß- oder Zustandsdiagramme, also Zeichen, als Mittel zur Organisation des Software Entwicklungsprozesses verwendet werden. Die Semiotik bestätigt Meyers ersten Grund, indem sie der menschlichen Kommunikation ein Sender - Empfänger Modell unterlegt, in dem der Empfänger in einem Wechselspiel von Codeauswahl, codegemäßer Interpretation einer empfangenen Botschaft und Beurteilung des Ergebnisses der Interpretation sowie gegebenenfalls Modifikation der Botschaft zum Zwecke der Auffindung eines geeigneten Codes zunächst einen Code auswählt und mit diesem dann die Botschaft interpretiert.

Meyers vierten Grund muß man zum Teil ebenfalls akzeptieren und damit einen in der Praxis häufig noch wunden Punkt im „Requirements Engineering“ berühren, daß nämlich der Analytiker (in der Regel) nicht seine Weltsicht beschreiben, sondern diejenige der Bedarfsteller erheben, analysieren und integrieren (für den Fall, daß es sich um mehrere Bedarfsteller handelt) sowie einer Formalisierung zugänglich machen soll.

Implizit ist in Meyers 4. Grund die Festsetzung enthalten, daß Software ein Modell sei. Aus Sicht der Semiotik ist diese Festsetzung nicht zweckmäßig, da sie zur Untersuchung von Vorgängen des Zeichengebrauchs 3 nicht aufeinander reduzierbare, einander überlappende aber voneinander relativ unabhängige Teildisziplinen und Fragestellungen identifiziert. Es handelt sich dabei um Syntaktik (Wie sind Zeichen zusammengesetzt?), Semantik (Worauf verweisen Zeichen?) und Pragmatik (In welcher Beziehung steht der Zeicheninterpret zum Zeichen?). Meyers implizite Festsetzung macht es unmöglich den Zweck des Modellgebrauchs, der von der Semiotik im Rahmen der Pragmatik untersucht würde, zu behandeln, weil sich mit dieser Festsetzung alleine kein Bezug zum Interpretieren herstellen läßt. Faßt man Modelle hingegen als Vorstellungen von Individuen und die Schemata bzw. die Software als Darstellungen, d.h. eine einem Medium aufmodulierte und dadurch wahrnehmbar gemachte Vorstellung auf, was in der Folge als geschehen vorausgesetzt wird, so können die wesentlichen Aspekte der Modellierung vermittelt durch den Zeichengebrauch untersucht werden.

Hinsichtlich seines zweiten Grundes kann man Meyer nicht zustimmen: nicht die Software, etwa im Falle eines Compilers, der, wie man verkürzend sagt, ein Programm übersetzt, wirkt auf dieses Programm ein, sondern, der Computer, in dem sich die betreffende Übersetzung ereignet. Die Software ist lediglich ein Mittel zur Steuerung einer Maschine, die bestimmte Operationen ausführen, etwa Zeichen manipulieren kann. Mit Hilfe der Software wählt ein Mensch einen der Abläufe aus, zu denen die betreffende Maschine fähig ist. Insofern hat sich seit der Zeit der mittels Lochkarten gesteuerten Webstühle, vgl. dazu [Ra94, S. 7, 8] nicht viel geändert: Lediglich die Art in der programmiert wird, d.h. Zeichensysteme zur Maschinensteuerung hergestellt werden, und die Leistungsfähigkeit der Maschinen hat sich (und zwar dramatisch) geändert.

Da jedes Stück Software ein nach vorgegebenen Regeln zusammengesetztes Zeichen ist und Menschen heute oft Software verwenden, muß man sich bei der Herstellung von Software auch mit Software befassen. Insofern hat Meyer auch mit seinem dritten Grund Recht. Er berücksichtigt an der besprochenen Stelle allerdings nicht, daß im „Requirements Engineering“ zur Analyse der Weltansicht der Bedarfsteller neben der Konsistenz dieser Sicht auch ihre Kompatibilität mit der Wirklichkeit analysiert werden muß und daher unterschiedliche Arten des Seins berücksichtigt werden müssen, nämlich dasjenige der Dinge und das der Gedanken.

Nach [D*89] ist eine Vorstellung ein „Bild, das sich jmd. in seinen Gedanken von etw. macht, das er gewinnt, indem er sich eine Sache in bestimmter Weise vorstellt.“ Damit ist eine für das alltägliche Leben ausreichende Erklärung des Begriffs „Modell“ gegeben, um ihn wissenschaftlich zu formulieren hat Stachowiak, vgl. dazu [St73, S.304 ff.], versucht ihn formal zu explizieren. Dies wird hier nicht nachvollzogen, sondern es wird (um den Begriff „Modell“ klarer zu machen) nur erklärt, welche Anforderungen von Stachowiak an das Bilden von Modellen gestellt werden.

Zunächst identifiziert Stachowiak, vgl. [St73, S. 131-133], drei Hauptmerkmale des Allgemeinen Modellbegriffs, nämlich:

1. das *Abbildungsmerkmal*: Modelle beziehen sich auf etwas, ein sogenanntes Modelloriginal, sie repräsentieren es, und
2. das *Verkürzungsmerkmal*: Modelle unterscheiden sich vom Modelloriginal in einer Weise, die vom Modellnutzer akzeptiert wird, sowie

3. das *pragmatische Merkmal*: Modelle sind ihren Originalen nicht per se, sondern nur für bestimmte Individuen, für einen bestimmten Zeitraum und hinsichtlich bestimmter Zwecke, Operationen und Untersuchungen zugeordnet.

Um genauer sagen zu können, was diese Merkmale bedeuten und wie sie sich im Rahmen einer Vorstellung ergänzen, und was deswegen beim Modellieren im Einzelnen zu tun ist definiert Stachowiak, [St92], ein fünfstelliges Prädikat $:(M,O,I,T,Z)$, gelesen als: *M ist Modell des Originals O für das Individuum I während des Zeitraums T zum Zweck Z*. Um den abbildungsartigen Zusammenhang zwischen **O** und **M** beschreiben zu können führt Stachowiak den Begriff des Ikomorphismus ein. Er verwendet allerdings dieses Wort nicht, sondern redet von ikomorphen Bildern. Ein Ikomorphismus ist eineindeutige partielle Abbildung 4 aus einer Menge P_O in eine Menge P_M (von **O** bzw. **M** beschreibenden Prädikaten). Stachowiak nennt Prädikate des Komplements des Definitionsbereichs von 4 in P_O *präteriert* und Prädikate des Komplements des Wertebereichs von 4 in P_M *abundant* und etabliert damit die Prädikate, die zwar dem Original **O**, nicht aber dem Modell **M** zugeschrieben werden als Präteritionsklasse bzw. umgekehrt als Abundanzklasse und behandelt so das Verkürzungsmerkmal. Dabei wird unterstellt, daß der Ikomorphismus die von **I** gemeinte Beziehung von Eigenschaften des Modelloriginals zu Eigenschaften des Modells herstellt wodurch das Abbildungsmerkmal behandelt wird. Stachowiaks Ansicht etwas vereinfachend kann man nun sagen, daß ein Individuum **I** eine Vorstellung vom Objekt **O** hat, wenn das Prädikat $:(M,O,I,T,Z)$ gilt. D.h. das Objekt **M** ein Modell eines Objektes **O** für **I** während **T** zum Zweck **Z** ist. Wenn also **M** und **O** durch Mengen P_M bzw. P_O von Prädikaten beschrieben, ein Ikomorphismus $4: P_O \rightarrow P_M$ festgelegt, an **M** gewisse dem Zweck **Z** dienliche Untersuchungen bzw. Operationen durchgeführt wurden und dadurch ein Objekt **M*** erzeugt wurde, dieses Objekt durch eine Menge P_{M^*} von Prädikaten beschrieben und mittels einer sogenannten Reversen 4^* (eine Fortsetzung der Inversen 4^{-1} des Ikomorphismus 4) auf eine Menge $(P_O)^*$ von Prädikaten abgebildet wurde, die als Beschreibung des Originals **O** gilt und so schließlich auch das pragmatische Merkmal berücksichtigt wurde.

Literatur

- [BP98] Blaha M., Premerlani W., Object-Oriented Modeling and Design for Database Applications, Prentice-Hall, Inc., Upper Saddle River, N.J., 1998.
- [Ch76] P.P Chen, The Entity – Relationship Model: Toward a Unified View of Data, ACM Transactions on Database Systems 1,1 (1976):9-37.
- [D*89] Drosdowski G., Dose M., Eckey W., Folz J., Hartmann H., Mang D., Mangold M., Schrupp C., Trunk-Nußbaumer M., Thyen O.M.A., Wermke M., Duden Deutsches Universal Wörterbuch, Bibliographisches Institut, Mannheim, 1989.
- [Ec94] Eco U., Einführung in die Semiotik, Wilhelm Fink Verlag, München, 1994.
- [Fo97] Fowler M., Analysis Patterns: Reusable Object Models, Addison-Wesley Longman Inc., Menlo Park, CA, 1997.
- [F*98] Falkenberg E.D., Hesse W., Lindgreen P., Nilson B.E., Han Oei J.L., Rolland C., Stamper R.K., Van Assche F.J.M., Verrijn-Stuart A.A., Voss K., A Framework of Information System Concepts, The FRISCO Report (Web Edition), IFIP, 1998.

- [K*98] Kneuper R., Müller-Lschnat G., Oberweis A.(Hrsg.), Vorgehensmodelle für die betriebliche Anwendungsentwicklung, B.G. Teubner Verlag, Stuttgart, Leipzig, 1997.
- [La98] Larman G., Applying UML and Patterns, Prentice Hall PTR, Upper Saddle River, N.J., 1998.
- [LM78] Lockemann P.C., Mayr H.C., Rechnergestützte Informationssysteme, Springer-Verlag, Berlin, Heidelberg, 1978.
- [Me97] Meyer B., Object-Oriented Software Construction, SECOND EDITION, Prentice Hall PTR, Upper Saddle River, N.J., 1997.
- [OR97] Ortner E., Methodenneutraler Fachentwurf, B.G. Teubner Verlagsgesellschaft, Stuttgart, Leipzig, 1997.
- [Pe98] Petkoff B., Wissensmanagement, Addison-Wesley Longman Verlag GmbH, Bonn et al., 1998.
- [Ra94] Randell B., The Origins of Computer programming, IEEE Annals of the History of Computing, Vol 16,4(1994):6-14.
- [R*91] Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorenzen W., Object-Oriented Modeling and Design, Prentice-Hall, Inc. A Division of Simon & Schuster, Englewood Cliffs, N.J., 1991.
- [St73] Stachowiak H., Allgemeine Modelltheorie, Springer-Verlag, Wien, New York, 1973.
- [St92] Stachowiak H., Modell, in Seiffert H., Radnitzky G. (Hrsg), Handlexikon zur Wissenschaftstheorie, Deutscher Taschenbuch Verlag GmbH & Co. KG, München, 1992.
- [Vo94] Vossen G., Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme, Addison-Wesley (Deutschland) GmbH, Bonn et al., 1994.