

Umgebungsmodellierung im Wandel der Zeit - Rückblick und kritischer Ausblick

Fabian Kneer¹ und Erik Kamsties¹ und Klaus Schmid²

¹ Fachhochschule Dortmund

{erik.kamsties, fabian.kneer}@fh-dortmund.de

² Universität Hildesheim

schmid@sse.uni-hildesheim.de

Motivation. In den 70er Jahren wurde die Notwendigkeit erkannt Systeme zu modellieren, um die Erfolgsrate von Projekten zu erhöhen und eine Brücke zwischen Nutzern und Entwicklern zu schlagen [2]. Hierbei wurden erstmals Überlegungen zum Thema Umgebung, System und ihrer Abgrenzung getätigt. Ein Verständnis der Umgebung ist Voraussetzung für die Formulierung von Anforderungen im Requirements Engineering (RE).

In diesem Beitrag geben wir einen historischen Rückblick auf verbreitete Modelle, Ansätze und Paradigmen zur Umgebungsmodellierung. Als Beispiel für die heutige Ansätze haben wir Ansätze der Umgebungsmodellierung für adaptive Systeme betrachtet. Mit Blick in die Vergangenheit und einem beispielhaften Überblick über die jetzige Nutzung, geben wir einen kritischen Ausblick auf die zukünftige Herausforderungen und Forschungsmöglichkeiten.

Rückblick. Ab 1970 wurden verschiedene Modelle für die Systemumgebung vorgeschlagen. Der früheste Ansatz ist die strukturierte Analyse, vorgeschlagen durch Tom DeMarco [2]. Dieser Ansatz verfügt über ein Kontextdiagramm (context diagram). Dieses Diagramm wurde als Start für die hierarchische Analyse verwendet und enthält Informationen darüber, welche Aspekte innerhalb des Systems und welche in der Umgebung liegen. Damit wurden die Aspekte auf die für die Analyse Relevanten eingeschränkt (s. Abb. 1a).

Ein weiteres Modell wurde von Parnas [5] vorgeschlagen: das 4-Variablen-Modell wurde durch die Regelungstechnik beeinflusst. Es beschreibt die Umgebung mit Hilfe von *quantities*, welche als mathematische Variable beschrieben werden (s. Abb. 1b). Die Variablen können als *monitored*, *controlled* oder beides charakterisiert werden. Weiter definiert das Modell *input* und *output* Variablen, die I/O-Register des Rechners darstellen. Zum Beispiel beschreibt die *IN*-Relation zwischen einer *monitored* und einer *input* Variablen das Verhalten eines Eingabegerätes. Ob die *IN*- oder *OUT*-Relation ein Teil der Umgebung ist, hängt von der Systemgrenze ab. Diese kann zwischen Software und Hardware oder zwischen System und der realen Welt liegen.

Ausgangspunkt des Umgebungsmodells von Jarke und Pohl [4] ist der *domain of interest*. Die Autoren schlagen darauf basierend vier Teilbereiche (sub-

domain) vor, welche sie *Worlds* nennen. Die *subject-World* enthält alle Informationen über die das System verfügt. Die *usage-World* beschreibt die direkten, beziehungsweise indirekten Nutzer des Systems. In der *system-World* werden alle Technologien, auf denen das System beruht, dokumentiert. Schließlich enthält die *development-World* alle Aspekte der Entwicklung (s. Abb. 1c).

Jackson hat mit dem WRSPM-Modell [3] eine weitere Sicht auf die Umgebung vorgeschlagen (s. Abb. 1d). Hier werden die Umgebung und das System als zwei Mengen von *phenomena* dargestellt, die eine Schnittmenge besitzen. Phenomena besitzen zwei Eigenschaften *visibility* und *control*. Die Phenomena der Umgebung werden durch sie gesteuert und sind teilweise sichtbar für das System. Das Artefakt *World* erfasst Wissen über die reale Welt und enthält die Phenomena der Umgebung und ihre Beziehungen.

Pohl et al. [6] haben eine weitere Beschreibung der Umgebung vorgeschlagen. Dieses Modell verfeinert das Vier-Welten-Modell mit der Notation für den *Kontext* (context): "Der Systemkontext ist der Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen an das System relevant ist." Hierfür werden Grenzen definiert, welche das System von dem (relevanten) Kontext und der (irrelevanten) Umgebung trennt (s. Abb. 1e). Des weiteren definieren die Autoren den Inhalt des Kontext weiter: "Kontextaspekte sind materielle und immaterielle Bestandteile des Systemkontext, wie z.B. Personen, Systeme, Prozesse oder physikalische Grundgesetze." Bei den Kontextaspekten wird zwischen drei Arten unterschieden: (1) Anforderungsquellen (z.B. Stakeholder, existierende Dokumente oder existierende Systeme), (2) Betrachtungsgegenstände (Menschen und materielle oder immaterielle Objekte) und (3) Eigenschaften und Beziehungen von Betrachtungsgegenständen.

Im Kontextmodell von Tom DeMarco werden Informationsflüsse zwischen System und Umgebung betrachtet, in den späteren Modellen wird diese Sicht der realen Welt Schritt für Schritt angereichert, wie wir im Vortrag näher zeigen werden.

Adaptive Systeme. Als Beispiel für die heutige Nutzung von Umgebungsmodellen, stellen wir Beispiele aus dem Bereich der adaptiven Systeme vor.

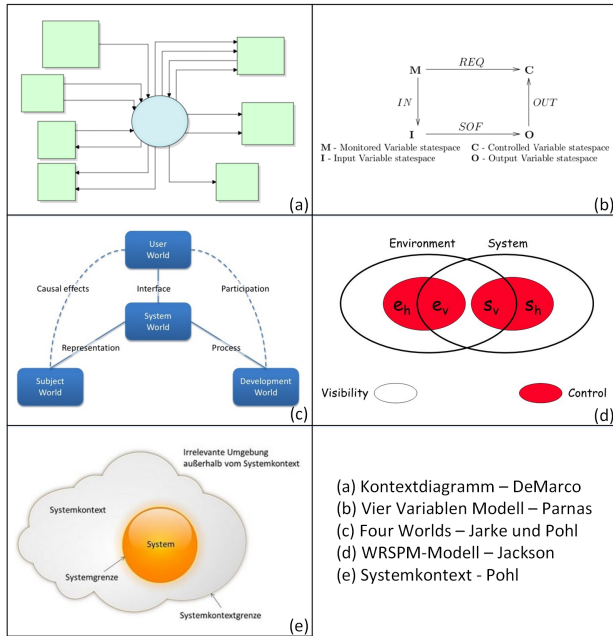


Abbildung 1: Übersicht der Modelle

Ein adaptives System besitzt nach Cheng et al. [1] zwei wichtige Charakteristiken: “(1) the design decisions are partially made at run-time, and (2) the system itself reasons about its state and environment”. Dies bedeutet, dass eine genaue Beschreibung der Umgebung ein wichtiger Faktor für ein adaptives System ist. Diese Art von System reagiert auf eine Veränderung in der Umgebung (Monitor), indem es die Situation analysiert (Analyze) und mögliche Anpassungen ermittelt (Plan) und anschließend ausführt (Execute).

Durch eine Literaturstudie haben wir vor allem *goal-based*, *agent-based*, *rule-based* und *state machines* als relevante Modellierungsparadigmen identifiziert.

Am Beispiel der zielorientierten Ansätze wollen wir diese neue Sichtweise auf die Umgebung vorstellen. Die zielorientierten Ansätze im allgemeinen versuchen basierend auf einem Kontextdiagramm Akteure innerhalb eines Systems und der Umgebung zu identifizieren. Diese Akteure haben Ziele und Intentionen, welche sie auf unterschiedliche Weise erfüllen können. Ein Beispiel für die Notation bietet i*. Diese Notation enthält ein Element *Ressource*, welches physische oder virtuelle Elemente enthalten kann. Diese Ressource wird von einem Akteur benötigt, um seine Ziele zu erreichen. In der Umgebungsmodellierung wird dieses Element erweitert durch Spezialisierungen, wie z.B. *monitored resource* oder *controlled resource*, was eine typische Unterscheidung ist, wie bereits bei Parnas gesehen. Des weiteren werden wichtige Elemente für die Umgebung zur klaren Erkennung mit Annotationen versehen [7]. Dies ermöglicht zwei Sichten auf die Umgebung (1) einfache Einbindung der Umgebungselemente als Ressource oder (2) Darstellung von Aspek-

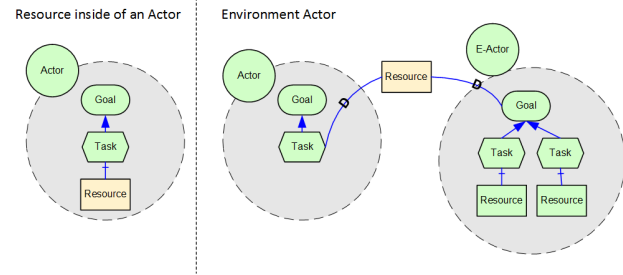


Abbildung 2: Beispiel Zielmodellierung

ten der Umgebung als Akteure mit einer detaillierten Beschreibung, wie die einzelnen Umgebungselemente erschlossen werden und welche Ziele und Intentionen die Umgebungselemente haben (s. Abb. 2).

Ausblick. Im Vortrag wird – basierend auf der historischen Entwicklung – die aktuelle Nutzung von Umgebungsmodellen untersucht. Weiterhin werden offene Fragen für weitere Forschung an Umgebungsmodellen und die damit verbundenen Herausforderungen diskutiert.

Literatur

- [1] B. H. C. Cheng et al. *Software Engineering for Self-Adaptive Systems: A Research Roadmap*, pages 1–26. 2009.
- [2] T. DeMarco. *Structured Analysis and System Specification*. Prentice Hall PTR, 1979.
- [3] C. A. Gunter, E. L. Gunter, M. Jackson, and P. Zave. A reference model for requirements and specifications. *IEEE Software*, 17(3):37–43, 2000.
- [4] M. Jarke and K. Pohl. Establishing visions in context: Toward a model of requirements processes. In J. I. DeGross, R. P. Bostrom, and D. Robey, editors, *Proceedings of the Fourteenth International Conference on Information Systems, Orlando, Florida, USA, December 5-8, 1993*, pages 23–34, 1993.
- [5] D. L. Parnas and J. Madey. Functional documents for computer systems. *Sci. Comput. Program.*, 25(1):41–61, 1995.
- [6] K. Pohl. *Requirements Engineering - Fundamentals, Principles, and Techniques*. 2010.
- [7] N. A. Qureshi, I. J. Jureta, and A. Perini. Towards a requirements modeling language for self-adaptive systems. In *Proceedings of the 18th International Conference on Requirements Engineering: Foundation for Software Quality, REFSQ'12*, pages 263–279, 2012.