

Umgebungsmodellierung im Wandel der Zeit:

Ein Rückblick und Ausblick

Fabian Kneer und Erik Kamsties und Klaus Schmid

University of Applied Sciences and Arts Dortmund / University of Hildesheim

23. November 2017

Contents

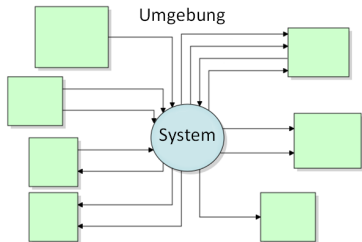
- 1 Motivation
- 2 Rückblick Umgebungsmodellierung
- 3 Adaptivität und Umgebungsmodellierung
 - Zielorientiert
- 4 Ausblick
- 5 Fazit

Motivation

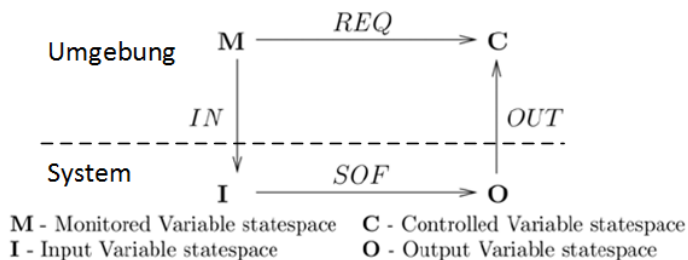
- ▶ In den 1970er Jahren wurden erste Ansätze zur systematischen Modellierung des Systems und der Abgrenzung zur Systemumgebung vorgestellt
- ▶ Kenntnis der Umgebung ist wichtig, um Anforderungen zu verstehen
- ▶ Unser Fokus liegt im Requirements Engineering für adaptive Systeme
- ▶ Umgebungsmodelle helfen die Umgebung zu verstehen und Strategien zur Lösung von Konflikten und Abweichungen in dem Umgebungsverhalten zu identifizieren

Kontextdiagramm - Tom DeMarco [1]

- ▶ Teil der strukturierten Analyse
- ▶ Interagierende Komponenten werden als Rechteck gekennzeichnet (Datenquellen und -senken)
- ▶ Verbindungen repräsentieren Datenflüsse
- ▶ Das Kontextdiagramm kann alternativ mit Modellen der UML erstellt werden

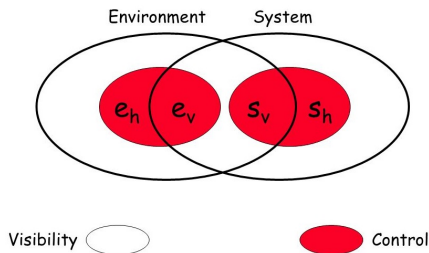


Vier Variablen Modell - Parnas [5]



- ▶ Beschreibung der Umgebung mit Hilfe von *Variablen*
- ▶ Input und Output entsprechen *Registern* auf die ein System zugreifen kann
- ▶ *Monitored* und *controlled* Variablen stellen physikalische Größen in der realen Welt dar

WRSPM-Modell - Jackson [2]

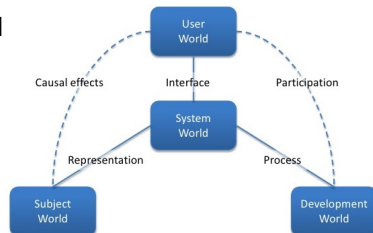


- ▶ Umgebung und System werden als Menge von *phenomena* dargestellt
- ▶ Phenomena besitzen zwei Eigenschaften *visibility* und *control*

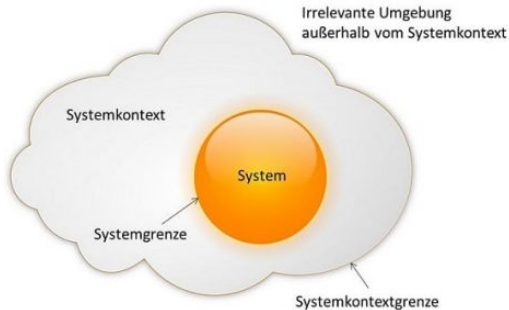
Four Worlds - Jarke & Pohl [3]

Vier Kontextfacetten:

- ▶ Gegenstandsfacette: materielle und immaterielle Gegenstände über die das System Informationen abbildet
- ▶ Nutzungsfacette: Systemverwendung durch Menschen und/ oder andere Systeme
- ▶ IT-Systemfacette: IT-Systemumgebung und die entsprechenden IT-Strategien
- ▶ Entwicklungsfacette: enthält Aspekte, die die Entwicklung des Systems beeinflussen und den Entwicklungsprozess



Systemkontext - IREB [6]



System besteht aus:

- ▶ System und seiner Systemgrenze
- ▶ Systemkontext und der Systemkontextgrenze
- ▶ Irrelevante Systemumgebung

Was fällt auf?

- ▶ Unterschiedlich abstrakte Elemente in der Umgebung
- ▶ Unterschiedlich starker Bezug zur Domäne des Systems
- ▶ Unterschiedliche Sichten auf die Umgebung:
 - ▶ Entwicklungssicht - enthält Elemente, die während der Entwicklung relevant sind
 - ▶ Laufzeitsicht - enthält die Elemente, die während der Laufzeit eines Systems relevant sind
- ▶ Sichtbarkeit und Kontrolle über Elemente

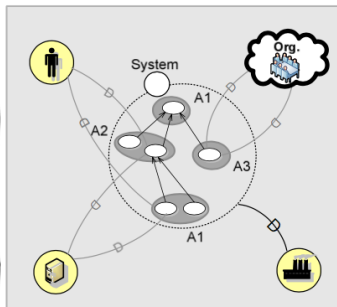
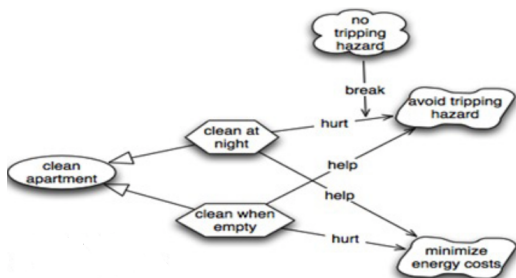
Umgebungsmodellierung und Adaptivität

- ▶ Ziel ist die Anpassung des Systems an Veränderungen in der Umgebung
- ▶ Lösungsstrategien für Adaptivität basieren auf Informationen über Zustände bzw. Änderungen in der Umgebung
- ▶ Wir haben unterschiedliche Paradigmen für die Modellierung der Umgebung für diese Systemklasse in der Literatur identifiziert: zielorientiert, Agenten-orientiert, regelbasiert, sowie Zustandsautomaten

Zielorientierte Modellierung

- ▶ Identifizierung von Akteuren im System oder seiner Umgebung
- ▶ Beziehungen zwischen den Akteuren
- ▶ Ziele und Intentionen der Akteure
- ▶ *Ressource* Element wird von Akteuren verwendet, um Ziele zu erreichen
- ▶ In der Umgebungsmodellierung wird dieses erweitert oder spezialisiert z.B. *Monitored* oder *Controlled* Ressource
- ▶ Annotationen von Elementen in der Umgebungen
- ▶ Anpassung des Modells mittels Laufzeitmonitoring von Annahmen über die Umgebung

Darstellungsformen für die Umgebung in i*



Was fällt auf?

- ▶ Unterscheidung zwischen zwei Arten von Umgebungen
 - ▶ *statische* Umgebung, die vorausgesehen werden kann
 - ▶ *dynamische* Umgebung, über die Annahmen getroffen werden müssen
- ▶ Separate, isolierte Modellierung von Laufzeit- und Entwicklungssicht
 - ▶ Anpassung des Modells über Laufzeitinformationen
 - ▶ Benötigt Feedback-System zur Überprüfung der Annahmen in der Umgebung
- ▶ Unabhängigkeit von der Domäne

Zukünftige Systeme - Beispiel Automobil/Mobilität

- ▶ Trends: *electrified, autonomous, connected, shared and updated*
- ▶ Zentralisierte Hochleistungsrechnerarchitekturen ("Centralized Domain Control Units - DCUs") mit Multi-/Manycore-Prozessoren und GPU-Unterstützung
- ▶ Systeme bilden komplexe Systeme (System of systems), z.B. Platooning bei LKWs
- ▶ Zukünftige Systeme tendieren zur *Heterogenität*
- ▶ Komplexität der realen Welt





Implikationen für die Umgebungsmodellierung

- ▶ Domänenübergreifende Modellierung der Umgebung
- ▶ Verbindung von Entwicklungszeit- und Laufzeitmodellen der Umgebung
- ▶ Umgebungsmodelle als eigenständige Assets unabhängig von Systemmodellen
- ▶ Vollständigkeit des Umgebungsmodells
 - ▶ zur Entwicklungszeit?
 - ▶ zur Laufzeit?

Zukünftige Arbeit

- ▶ Empirischer Vergleich der Ansätze im Bereich Adaptivität zur Ermittlung von Vor- und Nachteilen
- ▶ Bestimmung der Mächtigkeit der Ansätze
- ▶ Ausnutzung der Vorteile zur Entwicklung eines integrierenden Meta-Modells
- ▶ Methodiken zur Unterstützung der Modellierung



-  **T. DeMarco.** *Structured Analysis and System Specification.* Prentice Hall PTR, 1979.
-  **C. A. Gunter, E. L. Gunter, M. Jackson und P. Zave.** “A Reference Model for Requirements and Specifications”. In: *IEEE Software* 17.3 (2000), S. 37–43.
-  **M. Jarke und K. Pohl.** “Establishing Visions in Context: Toward a Model of Requirements Processes”. In: *Proceedings of the Fourteenth International Conference on Information Systems, Orlando, Florida, USA, December 5-8, 1993.* Hrsg. von J. I. DeGross, R. P. Bostrom und D. Robey. 1993, S. 23–34.
-  **A. Lopez-Lorca, G. Beydoun, R. Valencia-Garcia und R. Martinez-Béjar.** “Automating the reuse of domain knowledge to improve the modelling outcome from interactions between developers and clients”. In: *Computing* 98.6 (2016), S. 609–640.



D. L. Parnas und J. Madey. “Functional Documents for Computer Systems”. In: *Sci. Comput. Program.* 25.1 (1995), S. 41–61.



K. Pohl. *Requirements Engineering - Fundamentals, Principles, and Techniques.* 2010.