

Exkurs: Problem Frames

Blick vom Problembereich auf die Architektur

Burkhardt Renz

Fachbereich MNI
Technische Hochschule Mittelhessen

Wintersemester 2017/18

Übersicht

- Wie findet man eine Architektur?
 - Fragestellung
 - Die Idee der Problem Frames
 - Beispiel Ampelschaltung
- Problem Frames
 - Arten von Bereichen
 - 5 grundlegende Problem Frames
- kwic revisited
 - Problemstellung
 - Zerlegung und Architektur

Wie findet man eine Architektur?

Auf diese Frage gibt es keine allgemein gültige Antwort – auch keine auf jeden Fall zutreffende Vorgehensweise.

Zu bedenken sind

- Es gibt verschiedene Arten von Problemen. Meist wird nicht die richtige Architektur gewählt, weil nicht das wirkliche Problem betrachtet wird!
- Je nach Problem braucht man die passende Beschreibung. Je allgemeiner ein Ansatz ist, desto weniger wird er im Spezialfall nützen!
- Dem Problem angemessene Lösung; Architektur steckt in den Paradigmen des Problems.

Vorgehensweisen

- Kataloge von Stilen, Mustern
- Heuristiken und Faustregeln
- Vergleich mit kompletten Architekturen – wie etwa Middleware, SAP R/3 usw.
- Methode von Jan Bosch
- Architekturanalyse als Teil des Softwareentwicklungsprozesses etwa durch Architektur Reviews wie ATAM (Architecture Tradeoff Analysis Method)

Idee

Wie findet man eine geeignete Architektur? Diese Frage stellt sich Michael Jackson auf eine neue Weise:

Klar auseinanderhalten, was man meint:

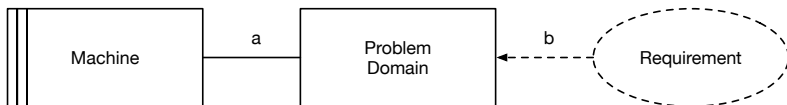
System und Computer (*system* und *machine*);

beschreibendes, analytisches Modell und analoges Modell
(*description* und *model*)

Nicht in den Computer und seine Software schauen, sondern rauswärts in die Welt, denn *dort* ist das Problem.

Dazu muss man die Problembereiche finden: sie sind nicht im Computer, sondern in der Welt, für die der Computer (*the machine*) ein Werkzeug sein soll.

Idee bebildert

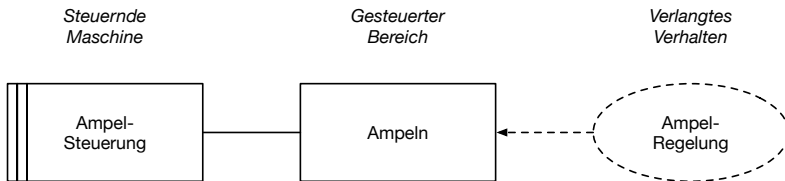


- Der Computer und der Problembereich haben eine Schnittstelle a gemeinsamer Phänomene (Zustände, Werte, Ereignisse)
- Die Anforderungen beziehen sich auf eine Menge b von Phänomenen des Problembereichs

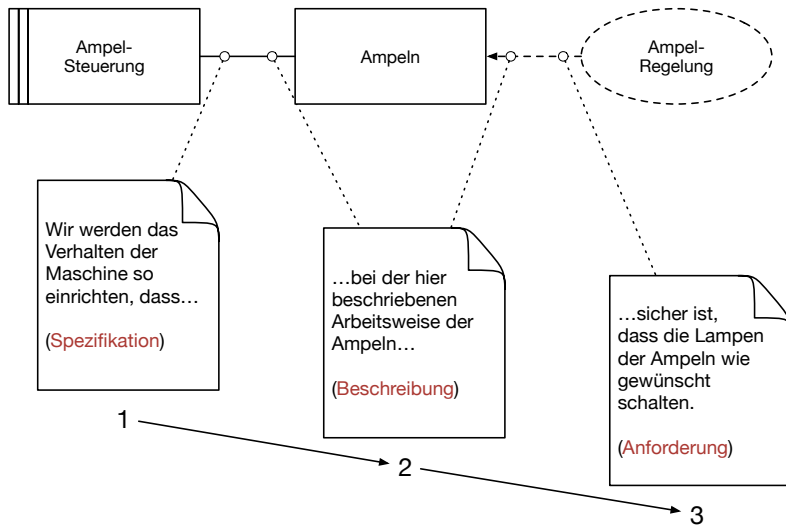
Die Anforderungen richten sich auf den Problembereich, nicht an den Computer!

Problemdiagramm Ampelschaltung

Das Problemdiagramm zur Ampelschaltung bei einspuriger Verkehrsführung



Schritte zur Problemlösung



Was hat das mit Architektur zu tun?

- Die Schritte unterschieden sich je nach Art des Problems
- Wirkliche Problembereiche haben verschiedene Arten von Problemstellungen
- ... lassen sich *nicht* hierarchisch zerlegen
- ... bestehen aus grundlegenden Arten von Problemstellungen (Problem Frames)
- ... haben bestimmte orthogonale Anforderungen

Idee: Lösungen für grundlegende Problem Frames zusammensetzen zu Architektur

Übersicht

- Wie findet man eine Architektur?
 - Fragestellung
 - Die Idee der Problem Frames
 - Beispiel Ampelschaltung
- Problem Frames
 - Arten von Bereichen
 - 5 grundlegende Problem Frames
- kwic revisited
 - Problemstellung
 - Zerlegung und Architektur

Bereiche

Arten von Bereichen

- die Maschine (*machine domain*)
- die gestaltbaren Bereiche (*designed domain*)
- die vorgegebenen Bereiche (*given domain*)

Charakteristik von Bereichen

- Kausale Bereiche, deren Phänomene Wirkung einer bekannten Ursache sind
- Lexikalische Bereiche, der Phänomene symbolische Informationen sind
- Gebietende (autonome) Bereiche (*biddable domains*), deren Phänomene zwar Regeln folgen (können), dieses aber nicht müssen

Verlangtes Verhalten (*Required Behaviour*)

Problemstellung

Es gibt einen Teil der wirklichen Welt, dessen Verhalten so gesteuert werden muss, dass bestimmte Bedingungen erfüllt sind. Das Problem ist, eine Maschine zu bauen, die diese Steuerung ausübt.

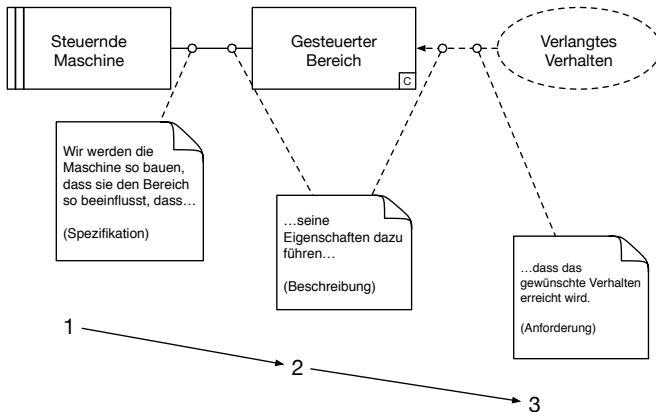
Beispiel

- Ampel an Baustelle mit einspuriger Verkehrsführung

Problem Frame Verlangtes Verhalten



Argumentation beim „Verlangten Verhalten“



Befohlenes Verhalten (*Commanded Behaviour*)

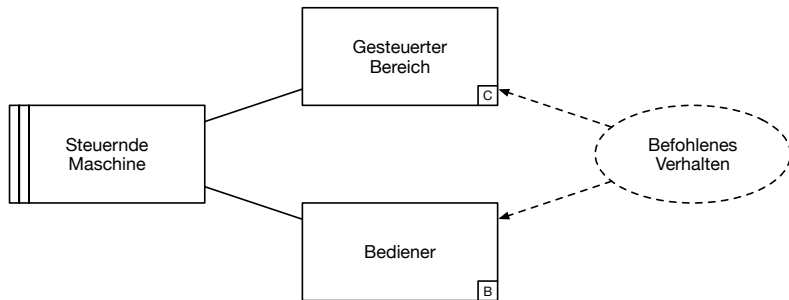
Problemstellung

Es gibt einen Teil der wirklichen Welt, dessen Verhalten entsprechend der Befehle eines Bedieners gesteuert werden muss. Das Problem ist, eine Maschine zu bauen, die die Bedienerbefehle entgegennimmt und die befohlene Steuerung ausübt.

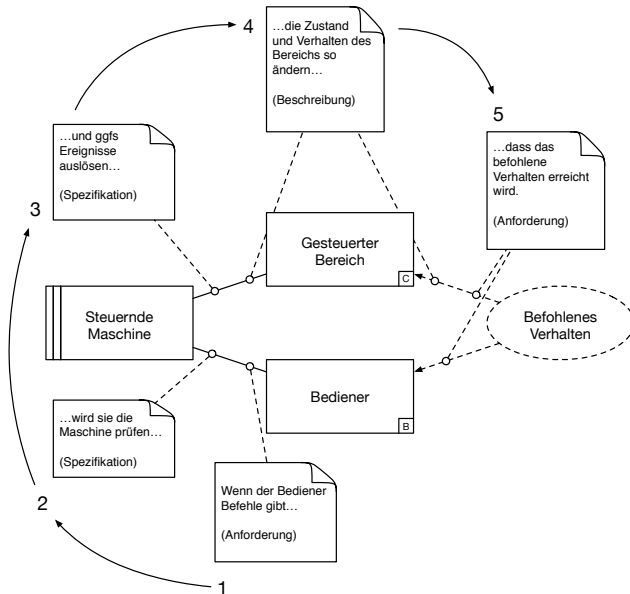
Beispiel

- Schleusentor

Problem Frame Befohlenes Verhalten



Argumentation beim „Befohlenen Verhalten“



Anzeige von Information (*Information Display*)

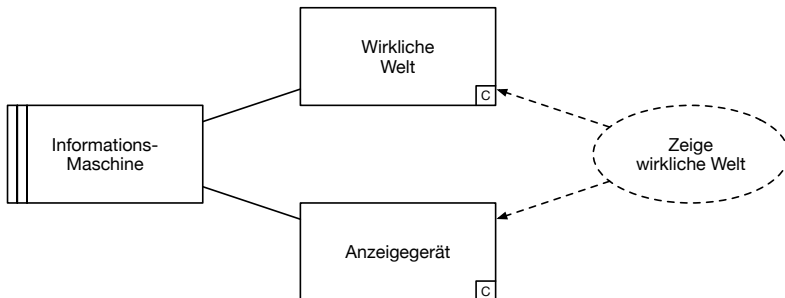
Problemstellung

Es gibt einen Teil der wirklichen Welt, über dessen Zustände und Verhalten fortlaufend gewisse Informationen gebraucht werden. Das Problem ist, eine Maschine zu bauen, die diese Informationen ermittelt und am verlangten Ort und in der verlangten Form darstellt.

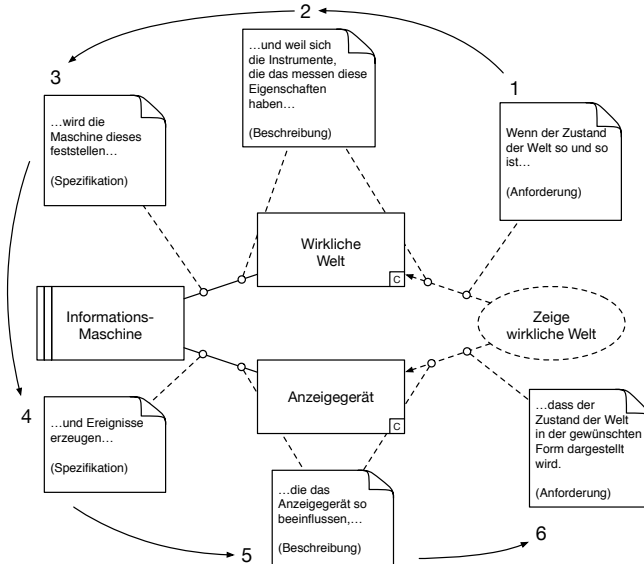
Beispiel

- Tachometer

Problem Frame Anzeige von Information



Argumentation bei der „Informationsanzeige“



Einfache Werkstücke (*Simple Workpieces*)

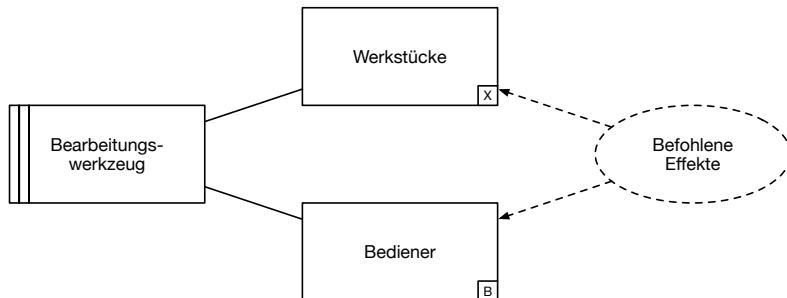
Problemstellung

Man braucht ein Werkzeug, mit dem ein Benutzer eine bestimmte Sorte von computerverarbeitbaren Objekten (wie Texte, Graphiken oder ähnliches) anlegen und bearbeiten kann, so dass sie dann weiterverarbeitet werden können (kopiert, gedruckt, analysiert, usw.). Das Problem ist, eine Maschine zu bauen, die als solches Werkzeug dient.

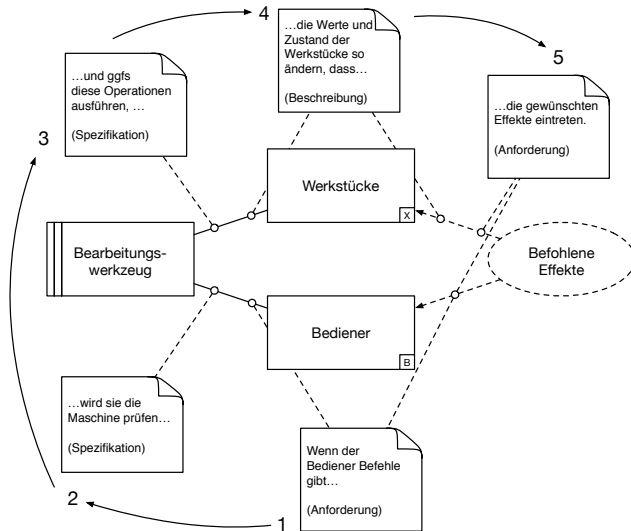
Beispiel

- Einladungsliste

Problem Frame Einfache Werkstücke



Argumentation bei „einfachen Werkstücken“



Transformation

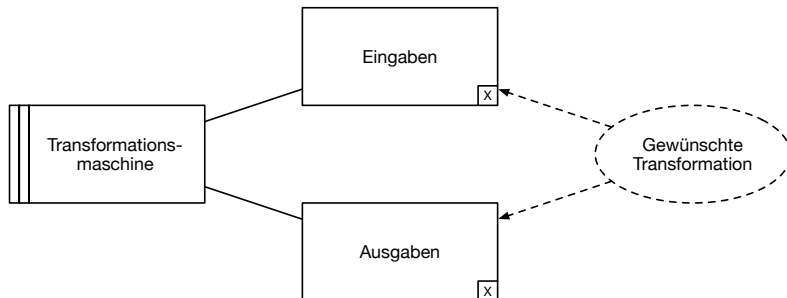
Problemstellung

Es gibt computerlesbare Eingabedateien, deren Daten so transformiert werden müssen, dass man bestimmte Ausgabedateien erhält. Die Ausgabedaten müssen ein bestimmtes Format haben und gewissen Regeln folgend aus den Eingabedaten erstellt werden. Das Problem ist, eine Maschine zu bauen, die die verlangten Ausgaben aus den Eingaben produziert.

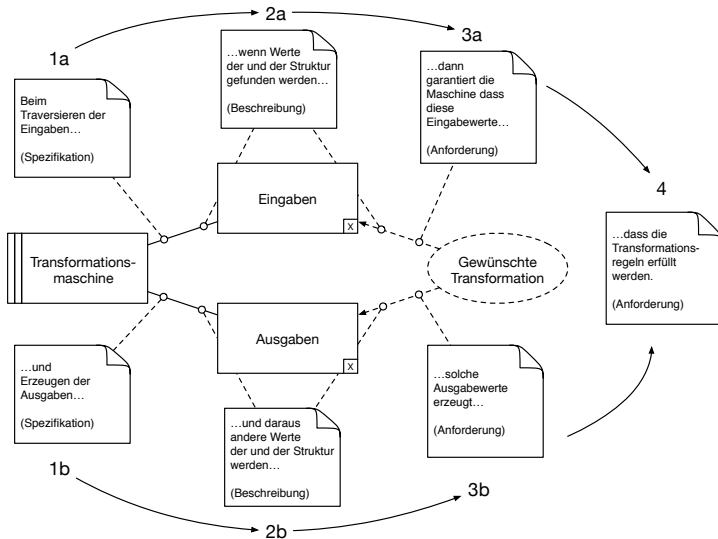
Beispiel

- E-Mail-Dateien auswerten

Problem Frame Transformation



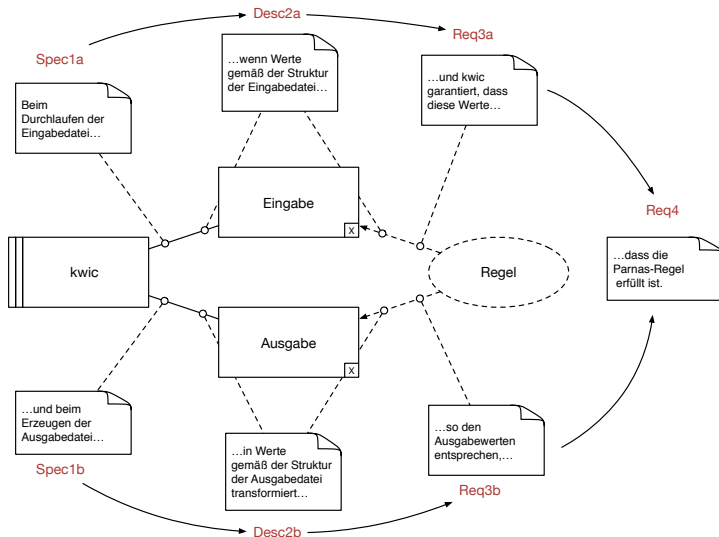
Argumentation bei der „Transformation“



Übersicht

- Wie findet man eine Architektur?
 - Fragestellung
 - Die Idee der Problem Frames
 - Beispiel Ampelschaltung
- Problem Frames
 - Arten von Bereichen
 - 5 grundlegende Problem Frames
- kwic revisited
 - Problemstellung
 - Zerlegung und Architektur

Kwic als Problem Frame



Analyse von Kwic

Benötigt werden

- Req4 – Anforderungen an das Gesamtergebnis
- Req3a, Req3b – Anforderungen an den Eingabebereich und den Ausgabebereich
- Desc2a, Desc2b – Beschreibungen der lexikalischen Bereiche
- Spec1a, Spec1b – Spezifikation der Maschine, *wie* sie die Eingabe und die Ausgabe durchläuft

Beschreibung von Ein- und Ausgabe

Zu Eigenschaften der Eingabe gehören

- Codierung
- Größe der Dateien
- Länge der Zeilen
- Aussagen über signifikante Werte (Füllworte?)

Zu Eigenschaften der Ausgabe gehören

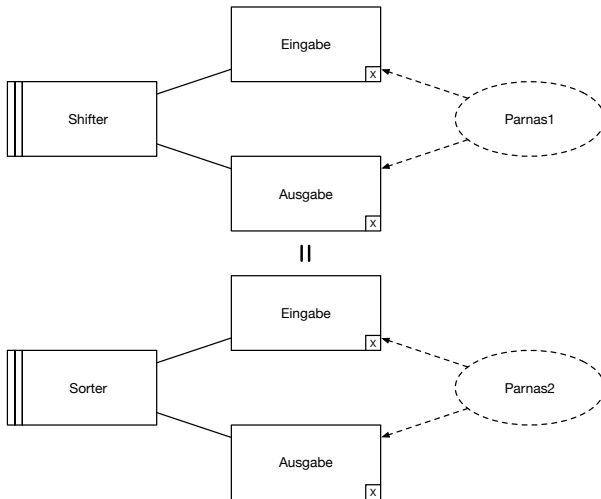
- Codierung
- Formatierung (perm. Index?)

Verhalten der Transformation

Sind Anforderungen und Eigenschaften des Problembereichs verstanden, ist die Aufgabe das geeignete Verhalten der Maschine zu spezifizieren

- *Effizientes* Traversieren von Ein- und Ausgabe
- *Variabilität* von vorneherein berücksichtigen

Zerlegung des Problem Frames für Kwic



Folgerungen für die Architektur

- Zerlegung eines Problems vom Typ Transformation in zwei solche
- Einfaches Zusammensetzen der Problem Frames: die Ausgabe des einen ist die Eingabe des anderen
- Pipes & Filters entspricht als Architektur der Lösung dieser Zerlegung des Problems
- Pipes & Filters in diesem Sinne die „natürliche“ Architektur für kwic
- Alternative: Modell, das Transformation beschreibt
- Anforderungen können Problem Frame sprengen: z.B. Interaktion mit Benutzer
- Aufwand der Änderungen hängen davon ab, ob neue Anforderungen in Problem Frame passen

Literatur



Jan Bosch

Design and Use of Software Architectures

Harlow, UK: Addison-Wesley, 2000.



Michael Jackson

Problem Frames: Analyzing and structuring software development problems

Harlow, England: Addison-Wesley, 2001.



Benjamin L. Kovitz

Practical Software Requirements: A Manual of Content & Style

Greenwich, CT: Manning, 1999.



Maritta Heisel et al

Systematic Architectural Design based on Problem Patterns

Relating Software Requirements and Architecture, Springer 2011.