

Michael Jackson's Problem Frames

PdW, 17.12.2002

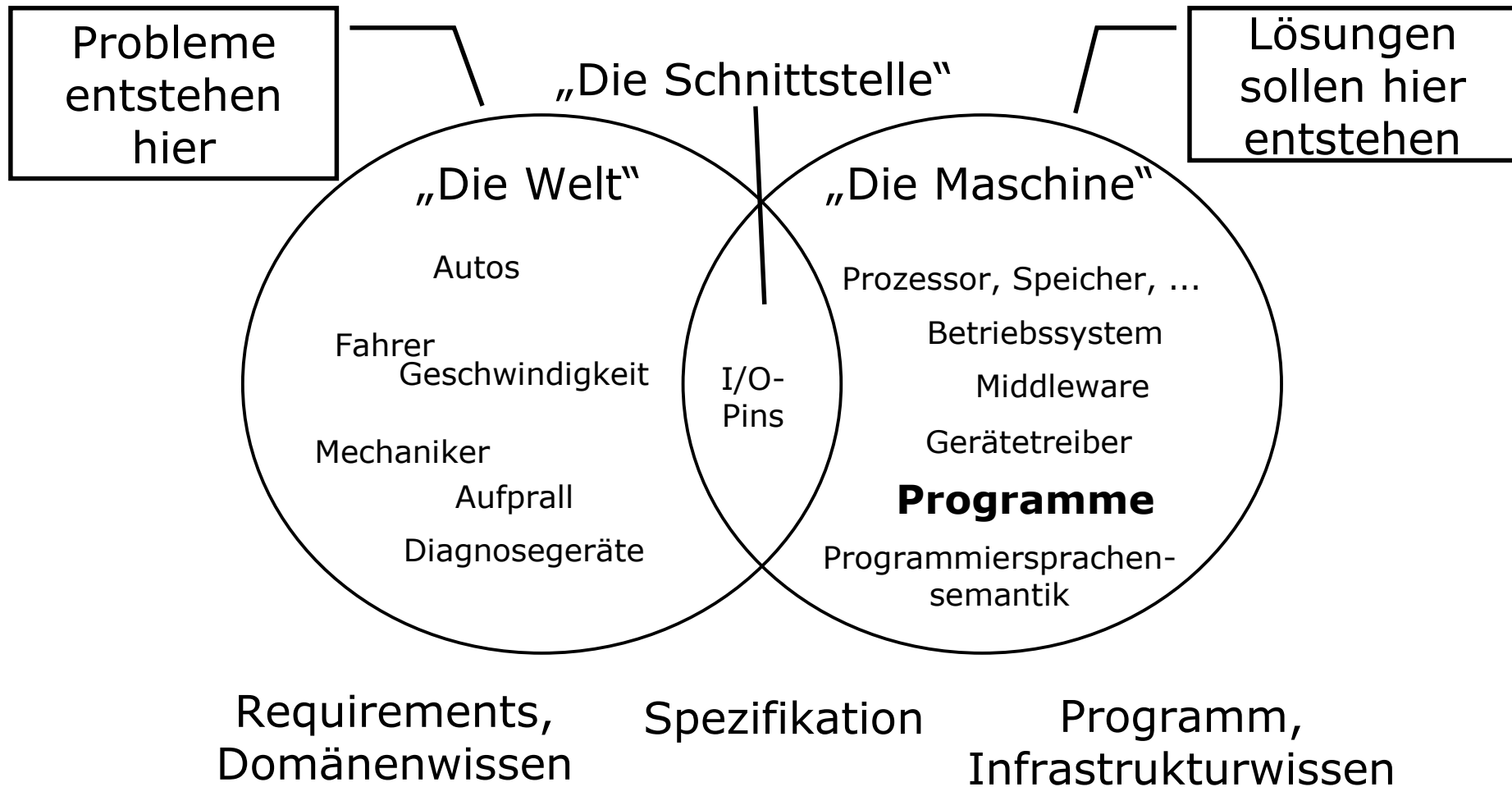
Jan Philipps

 *validas*

Worum es geht

- ▶ „Lösungen“
 - ▶ Viele Methoden, Notationen, Heuristiken, Werkzeuge
- ▶ „Probleme“
 - ▶ Welche Methode anwenden?
 - ▶ Welche Notationen sind geeignet?
 - ▶ Wie Probleme in handlichere Unterprobleme aufteilen?
- ▶ Problem Frames
 - ▶ Notationen & Konzepte zur Abstraktion von Klassen von SW-Entwurfsproblemen
 - ▶ Strukturierung von Requirements-Dokumenten
 - ▶ Hinweise für geeignete Lösungstechniken

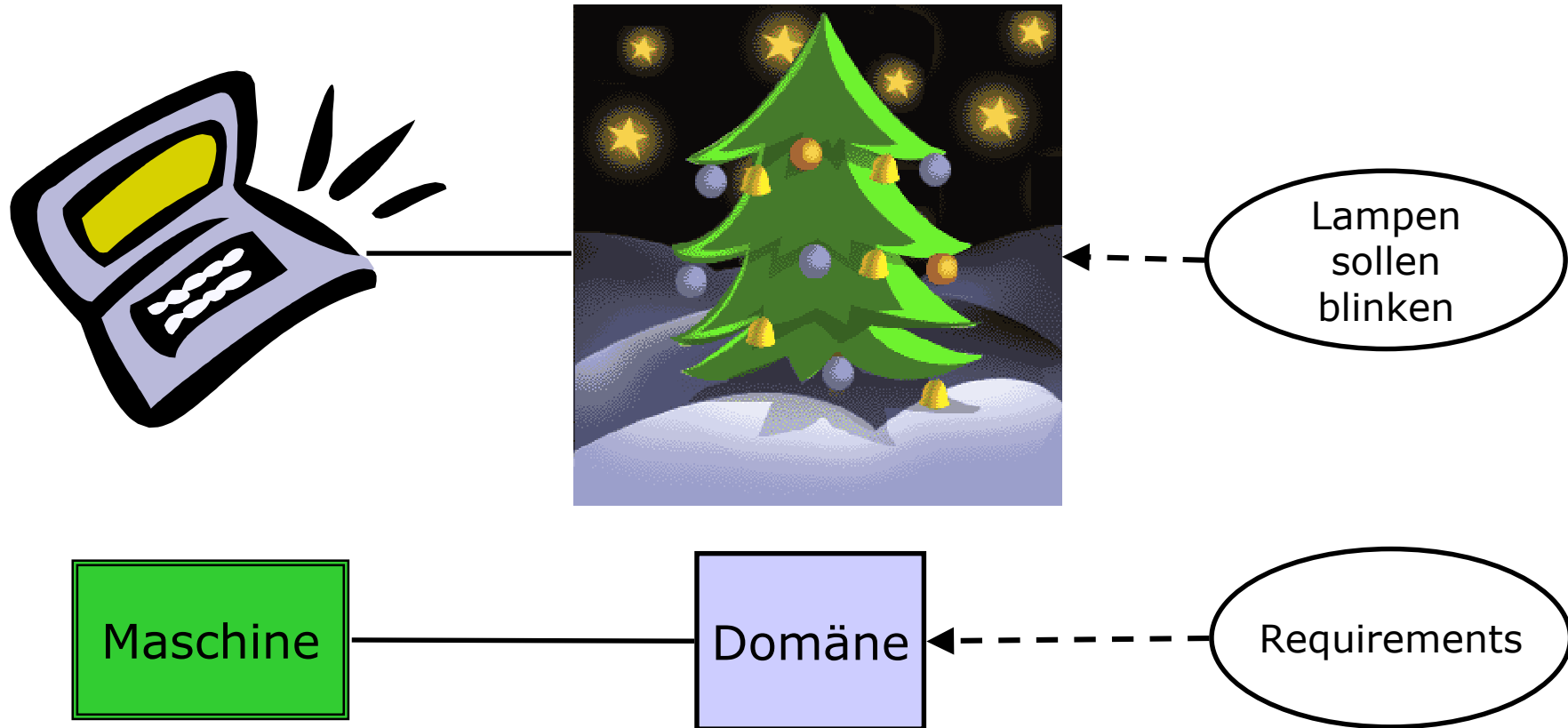
Wozu brauchen wir Software?



- ▶ Ziel eigentlich: „ $S \wedge D \Rightarrow R$ “ und „ $P \wedge I \Rightarrow S$ “
- ▶ Heute aber eher: „ $P \wedge I \wedge D \Rightarrow R$ “

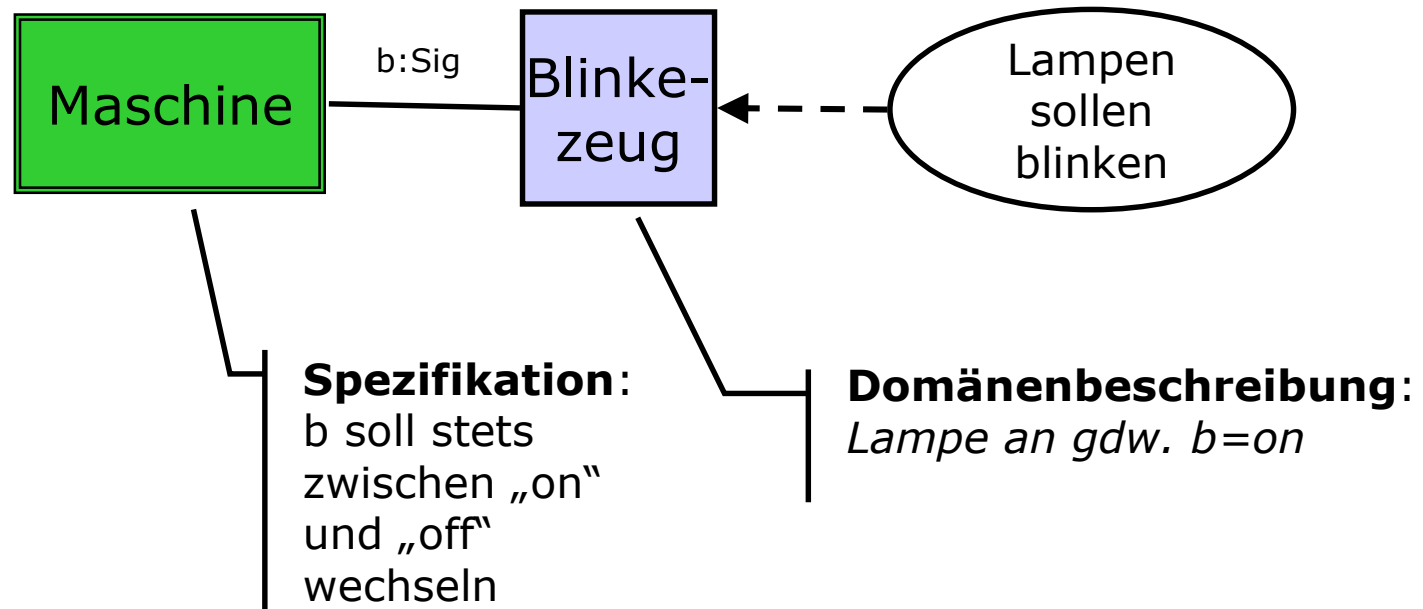
Problemdiagramme

- ▶ Kontextdiagramm („Problemgrenzen“)
- ▶ Problembeschreibung („Requirements“)



Fragestellungen („Concerns“)

- ▶ Was muß ich noch wissen, um nachzuweisen, daß meine Maschine funktioniert?
- ▶ Wie sieht das Korrektheitsargument aus?



*Wg. Spezifikation wechselt b,
wg. Domänenbeschreibung ist die Lampe mal aus und mal an,
also blinkt die Lampe.*

Nötige Beschreibungen

- ▶ Domänenbeschreibungen („indikativ“)
 - ▶ „Lampe ist an, gdw. wenn $b=on$ “
- ▶ Requirements und Spezifikationen („optativ“)
 - ▶ „Lampe soll blinken“
 - ▶ „ b soll abwechselnd on und off sein“
- ▶ **Voraussetzung:** Vernünftige gemeinsame Terminologie
 - ▶ Bezeichnungen („Lampen sind die Glühbirnen am Leuchtezeug“)
 - ▶ Definitionen („Eine Lampe blinkt, wenn sie abwechselnd an und aus ist“)

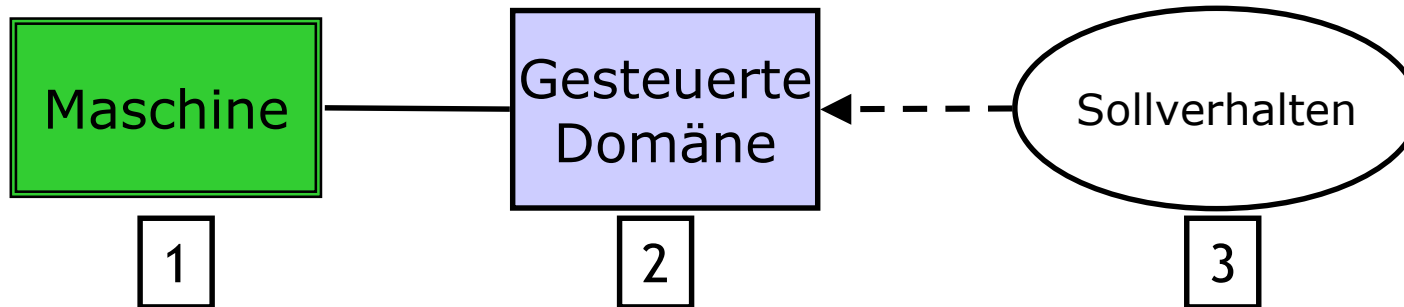
Domänen-Beschreibungen

- ▶ **Kausale Domänen** (Geräte, physikalische Umgebung)
 - ▶ State Transition Diagrams, Entscheidungstabellen, Regelungstechnik
 - ▶ **Symbolische Domänen** (Daten)
 - ▶ Datenbankschema, Klassendiagramme, ADTs, ...
 - ▶ **Verhandelbare Domänen** (Benutzer)
 - ▶ Kausalität kann nicht garantiert werden
 - ▶ Benutzerhandbücher?
-
- ▶ Entsprechende Beschreibungstechniken für Requirements...
 - ▶ Die Maschine ist auch eine „kausale Domäne“

Vorgehen

- ▶ Problemdiagramme aufzeichnen (Kontext + erste, vage Requirements)
- ▶ Teilprobleme finden, Problemgerippen („Problem Frames“) zuordnen
- ▶ Beschreibungstechniken zuordnen
- ▶ Domänen, Requirements, Spezifikation beschreiben

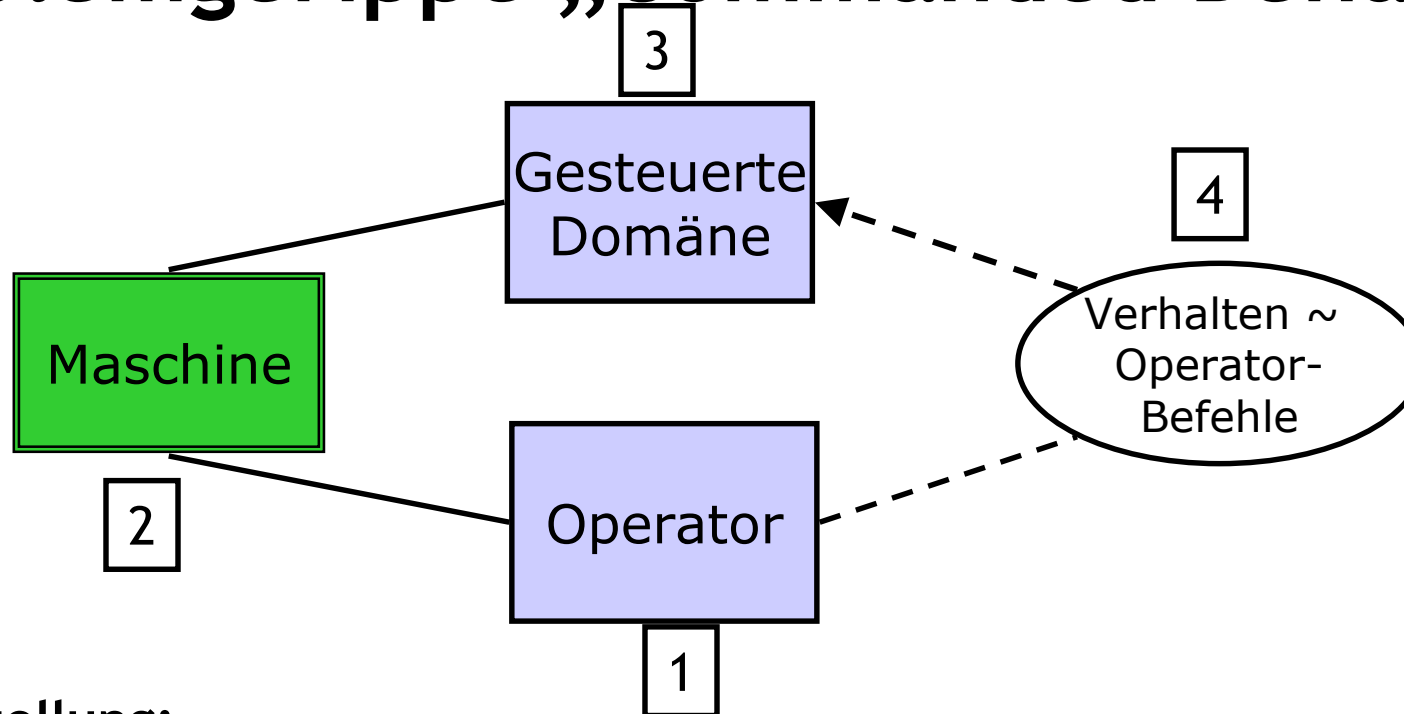
Problemgerippe „Required Behavior“



Fragestellung:

1. Wg. Spezifikation,
2. Verhält sich die Domäne wie folgt,
3. Deshalb ist die Anforderung erfüllt

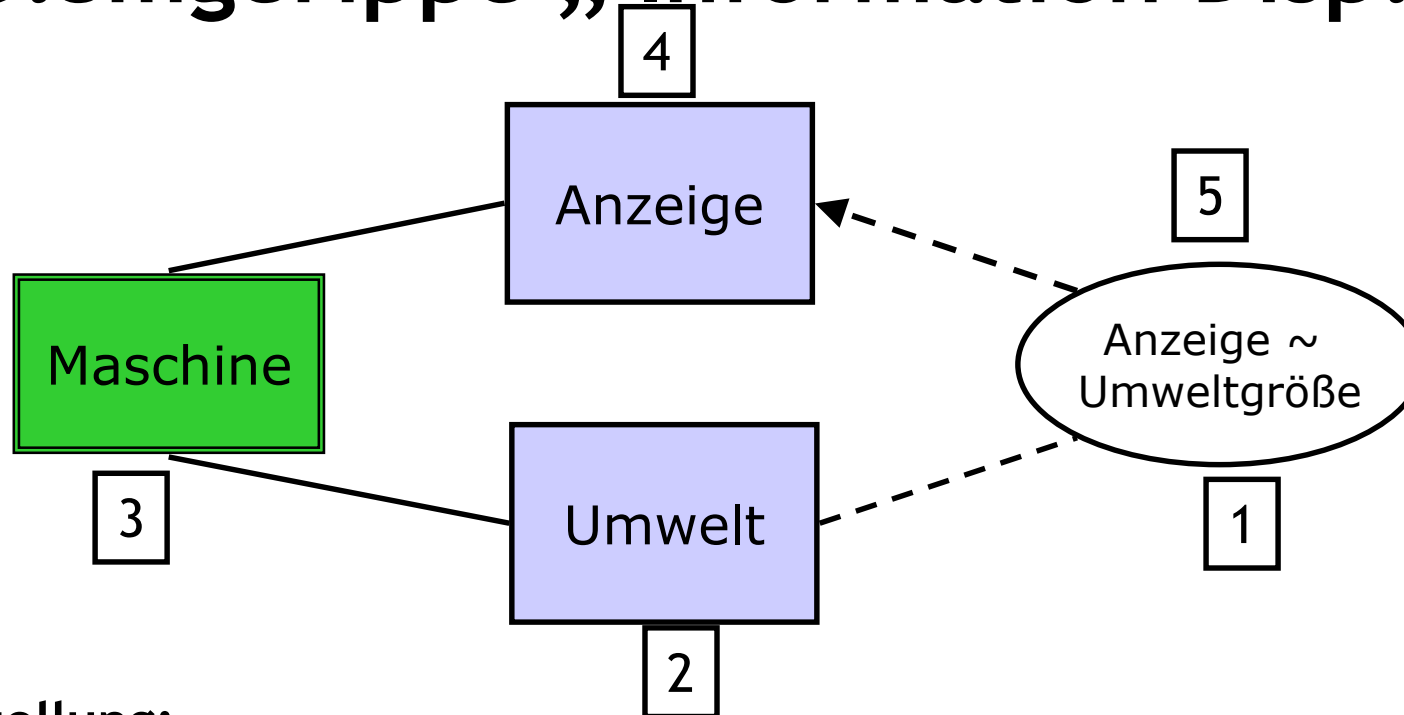
Problemgerippe „Commanded Behavior“



Fragestellung:

1. Wenn der Operator einen Befehl gibt
2. Wird er von der Maschine wie folgt umgesetzt,
3. Deshalb verhält sich die Domäne wie folgt,
4. Deshalb ist die Anforderung erfüllt

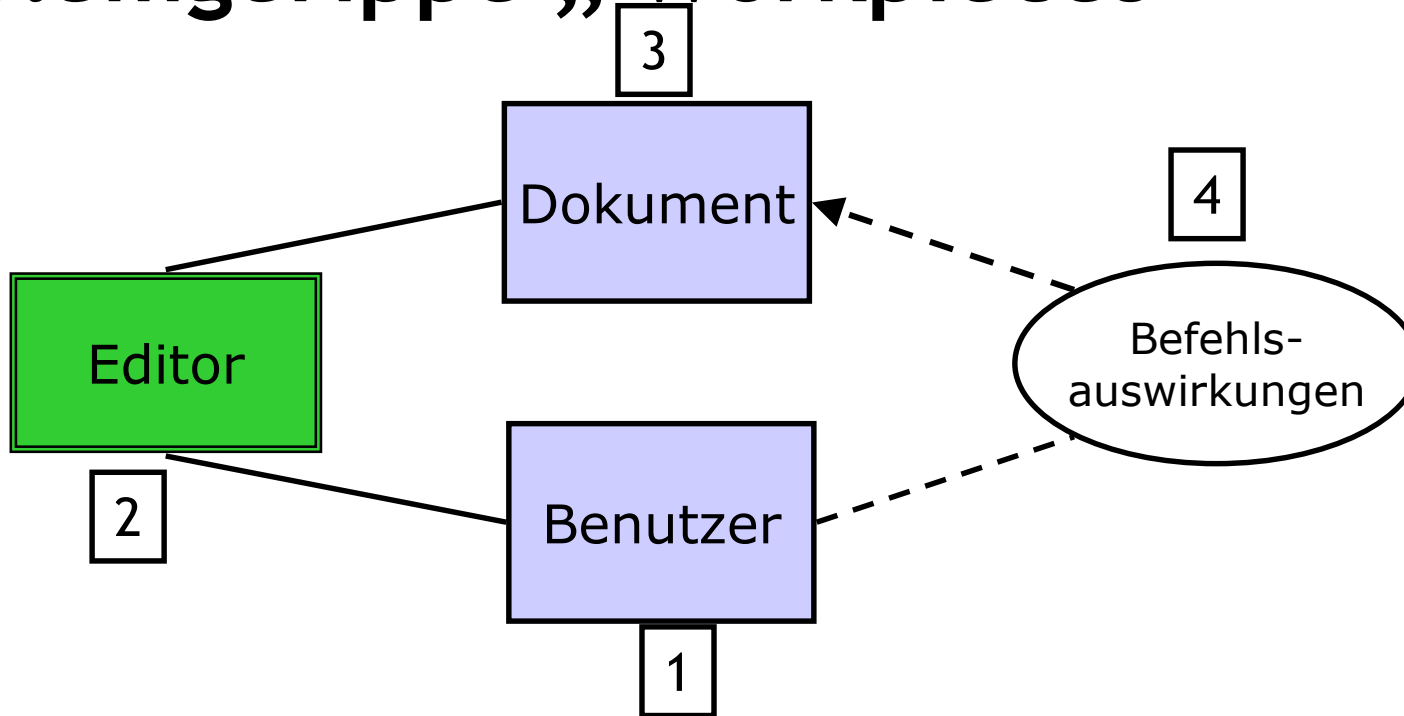
Problemgerippe „Information Display“



Fragestellung:

1. In diesem Umweltzustand,
2. Liefern die Sensoren folgende Werte,
3. Weshalb die Maschine diese Anzeigebefehle auslöst,
4. Die zu dieser Anzeige führen,
5. Weshalb die Anforderung erfüllt ist.

Problemgerippe „Workpieces“

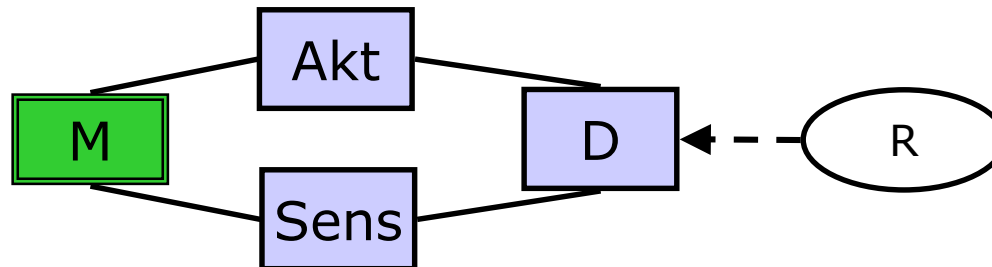


Fragestellung:

1. Wenn der Benutzer diesen Befehl gibt,
2. Wird von der Maschine diese Operation ausgelöst,
3. Die das Dokument wie folgt ändert,
4. Was die gewünschte Auswirkung ist.

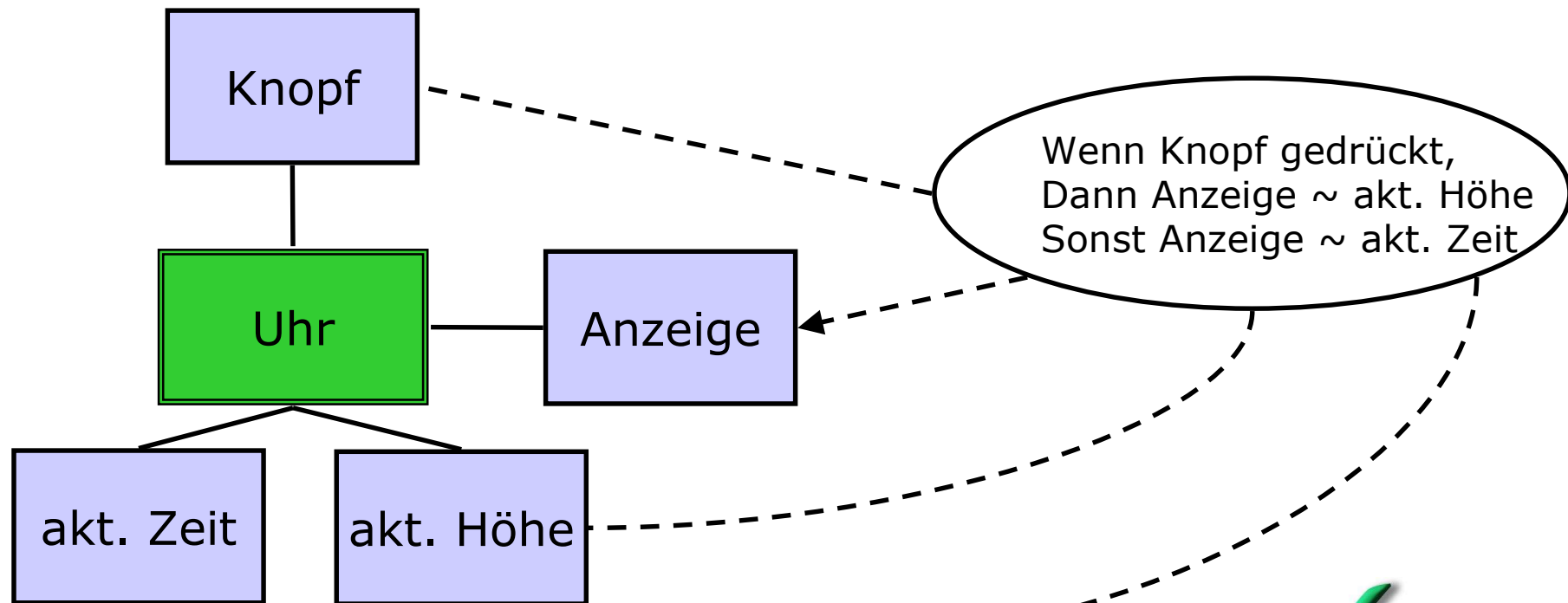
Problemgerippe

- ▶ Häufig vorkommende Problemmuster
- ▶ Triviale Einfachheit
- ▶ Weitere Gerippe: Verbindungen, Transformation
- ▶ Zahlreiche Varianten
 - ▶ Indirekte Verbindungen zwischen Domänen (Benutzer, Telekom, Sensoren/Aktuatoren)
 - ▶ Indirektes Required Behavior (vgl. SCR):

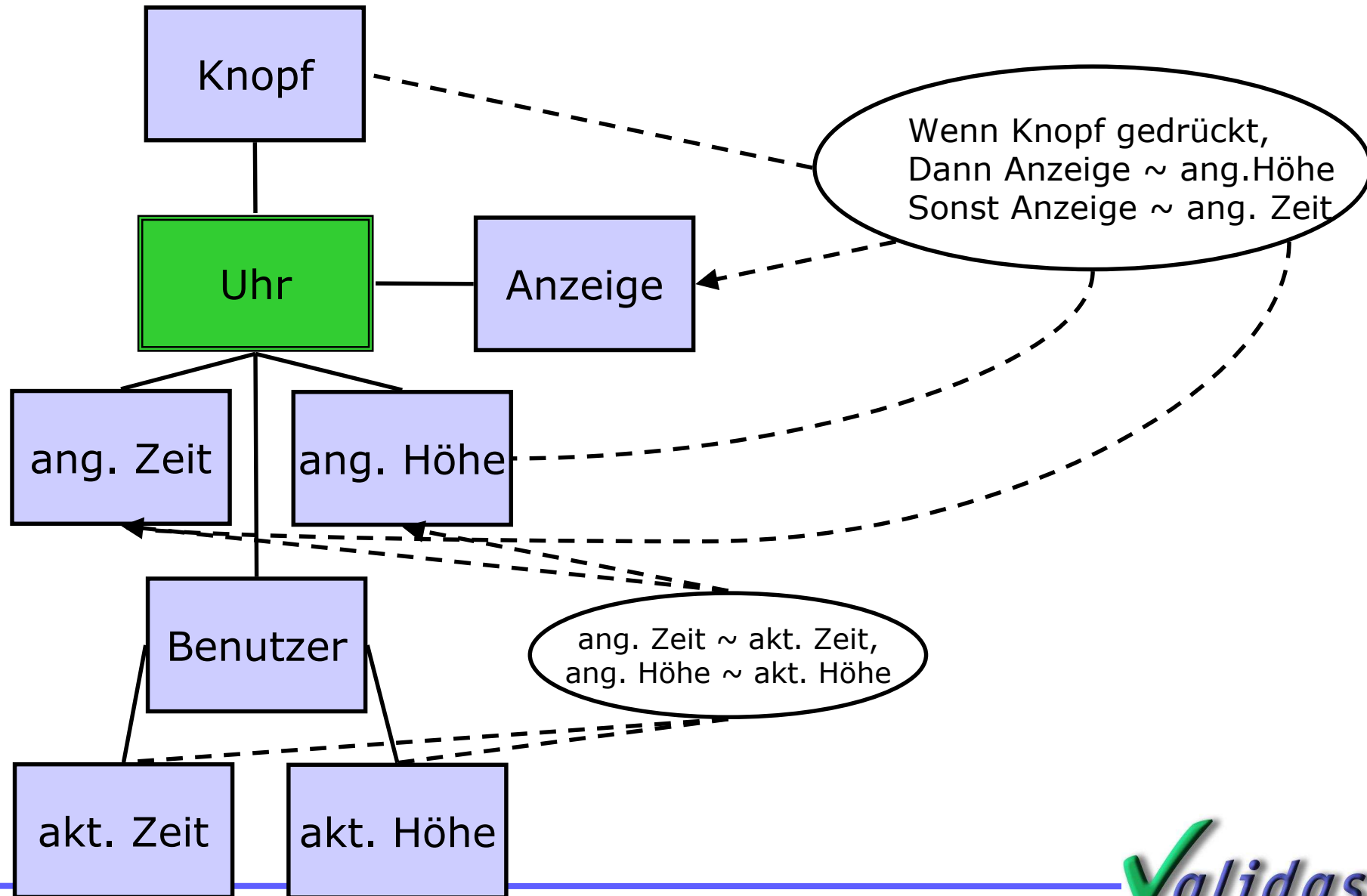


Armbanduhr

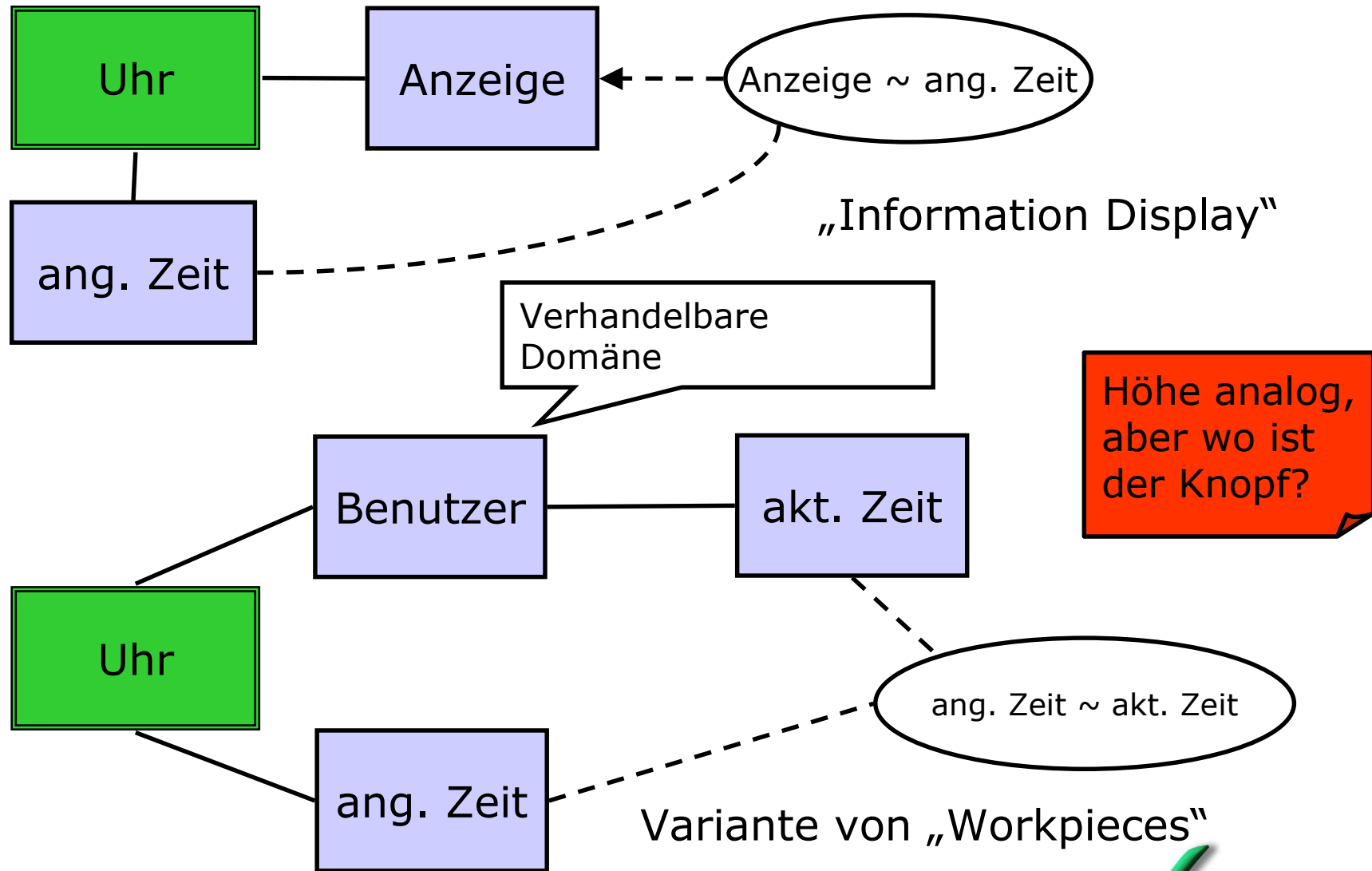
- ▶ Armbanduhr soll Uhrzeit + Höhe über Null anzeigen



Armbanduhr - Modelldomänen



Armbanduhr - Teilprobleme



Problem-Dekomposition

- ▶ Problemdiagramme als **Überlagerung** von Problemgerippen
- ▶ Projektion zum Finden von Teilproblemen

- ▶ Standard-Gerippe suchen?
 - ▶ Häufig: Controlled Behavior, Information Display
- ▶ Verschiedene Tempi innerhalb des Problems?
 - ▶ Diagnoseschnittstellen vs. Regelungsverhalten
- ▶ Modell-Domänen?
 - ▶ Aufspaltung in Informations-Sammlung und -Verarbeitung

- ▶ *Aber wie kriegt man die Teillösungen der Teilprobleme wieder zusammen?*

Problemübergreifende Fragestellungen

- ▶ Initialisierung
 - ▶ Stimmt initialer Zustand meiner Beschreibung mit der wirklichen Welt überein?
- ▶ Zerbrechlichkeit
 - ▶ Kann irgendetwas in dem betrachteten Weltausschnitt kaputt gehen?
- ▶ Identifizierung
 - ▶ Sind Modell- und Welt-Entitäten richtig verbunden?
- ▶ Zuverlässigkeit
 - ▶ Gibt es Abweichungen vom beschriebenen Domänenverhalten?
 - ▶ Sind Fehlermodelle oder eigene Verbindungsdomänen nötig?
- ▶ Interferenz, Atomarizität
 - ▶ Stören sich Teillösungen gegenseitig?
 - ▶ Ist Abstraktion des Teilproblems zu grob?

Anmerkungen

- ▶ Problemdiagrammerstellung oft knifflig
- ▶ Teilproblemsuche nichttrivial
- ▶ Bis jetzt kaum Zuordnung von Lösungstechniken zu Teilproblemen
 - ▶ Zuordnung zu Standardarchitekturen, Design Patterns?
- ▶ Integration von Teillösungen schwierig
 - ▶ Wie kombiniert man Regelungstechnik- mit Interpreter- mit Blackboard-Architektur?

- ▶ Immerhin aber ergiebiger als „Transformationelle“ vs. „reaktive“ Systeme

Referenzen

- ▶ Michael Jackson. *Problem Frames*. Addison Wesley, 2001
- ▶ Michael Jackson. *Software Requirements & Specifications*. Addison Wesley, 1995
- ▶ B.L. Kovitz. *Practical Software Requirements*. Manning, 1999.

- ▶ Diverse Papers von Michael Jackson, Daniel Jackson, Pamela Zave