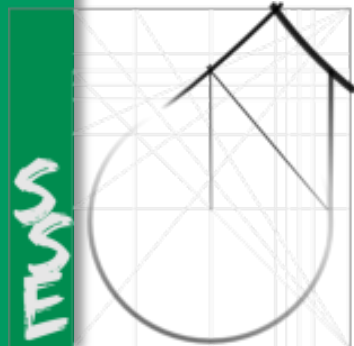


Operationalisierung des Projektcontrollings

Informatik 2008

Edward Fischer



Technische Universität Clausthal
Institut für Informatik - Software Systems Engineering
Lehrstuhl von Prof. Dr. Andreas Rausch
Julius-Albert-Str. 4
38678 Clausthal-Zellerfeld

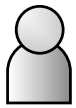


Agenda

- **Motivation & Zielsetzung**
- Problemstellung
- Lösung
- Was offen bleibt
- Zusammenfassung



Motivation

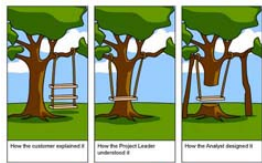
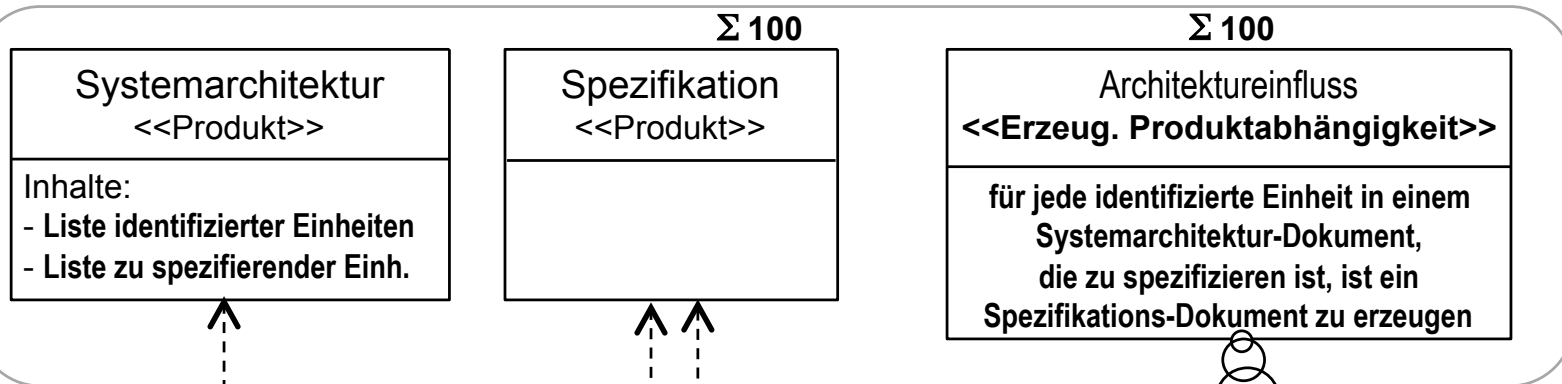


Projektleiter

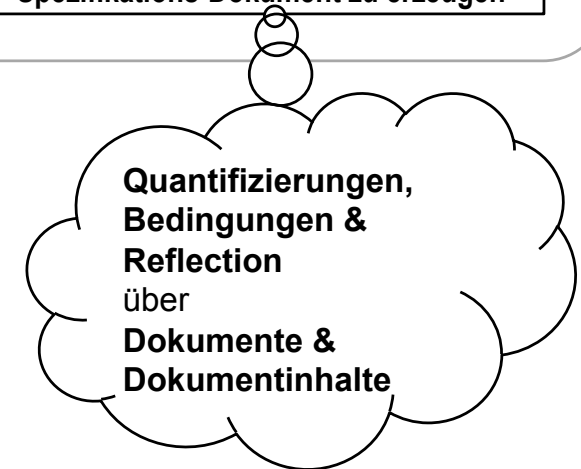
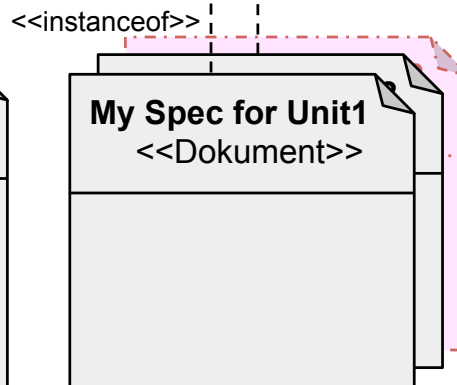
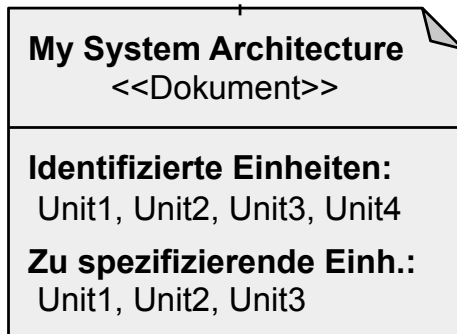
- Überwacht Termine, Erfüllungsgrad Arbeitspakete, Mittelabfluss,
- Berichtet im Lenkungsausschuss, ...



Vorgehensmodell



Projekt

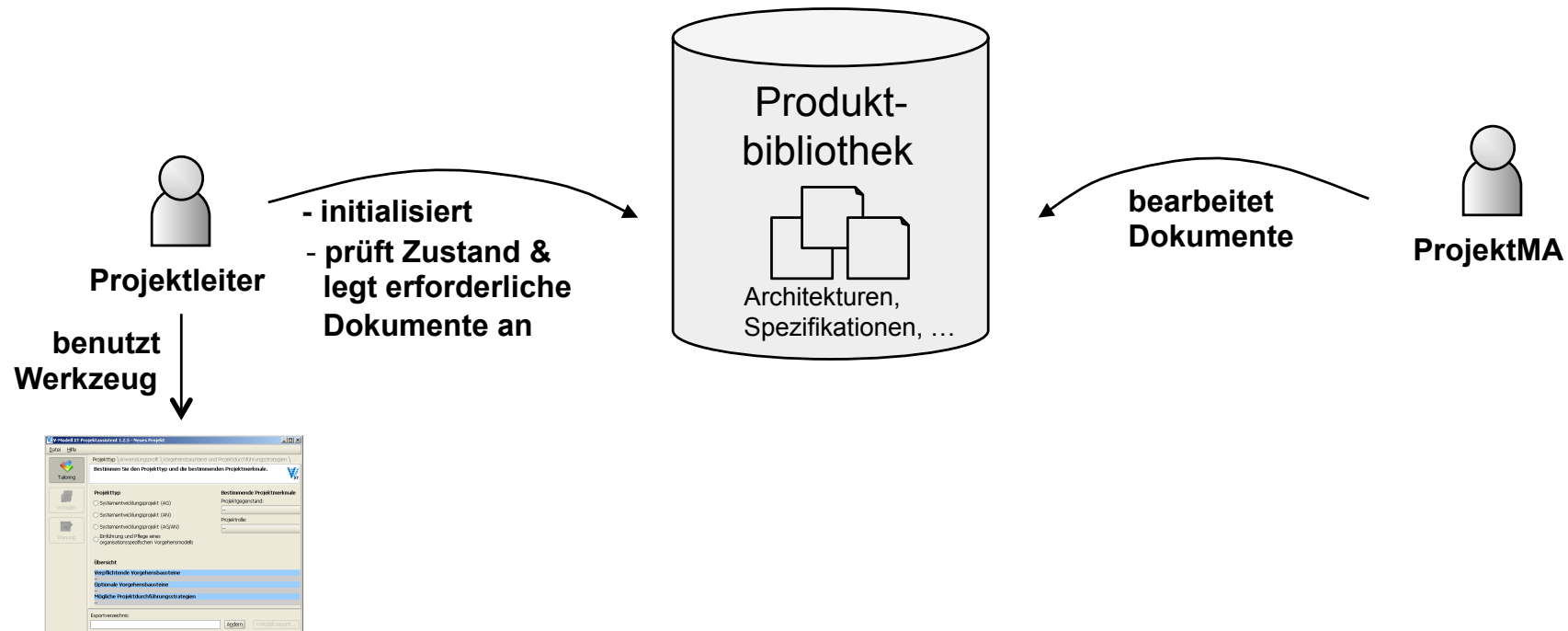


$\Sigma 1000$

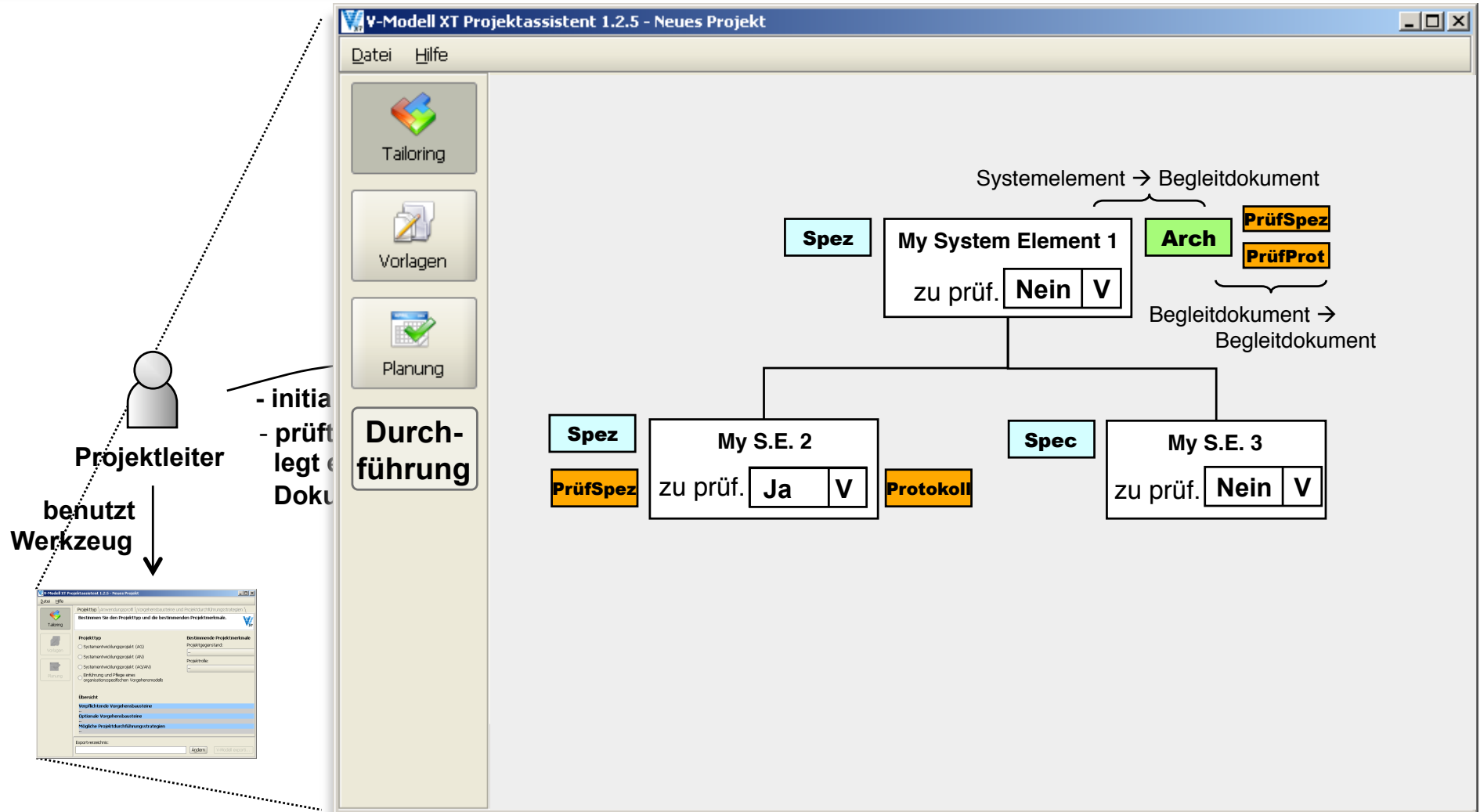


Zielsetzung: Werkzeugunterstützung

- **Werkzeugunterstützung für folgende Projektkonstellation:**



Zielsetzung: Werkzeugunterstützung (Vision)



Zielsetzung: zentrale Fragestellungen

- **Wie kann automatisiert beantwortet werden..**
 - A) .. ob Dokumente & -inhalte korrekt bzgl. eines Vorgehensmodell sind?
 - B) .. ob im betrachteten Projektzustand Dokumente fehlen, bzw welche Dokumente noch zu erzeugen sind, und mit welchen Inhalten?
 - C) .. wie mit Änderungen von Dokumenten & -inhalten umzugehen ist?



Agenda

- Motivation & Zielsetzung
- **Problemstellung**
- Lösung
- Was offen bleibt
- Zusammenfassung



Probleme

A) .. ob Dokumente & -inhalte korrekt bzgl. eines Vorgehensmodell sind?

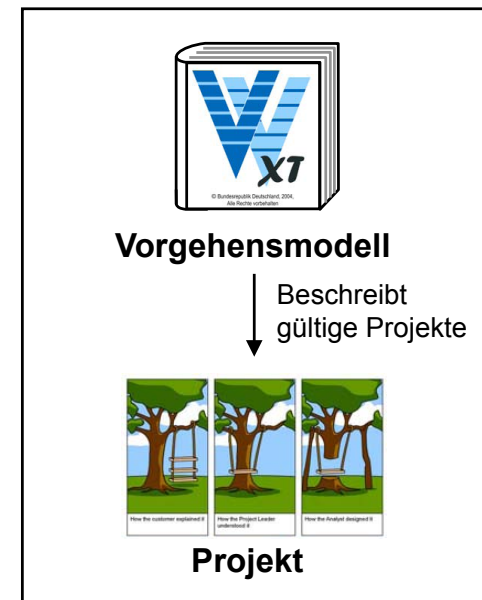
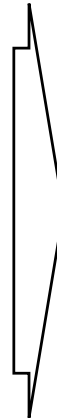
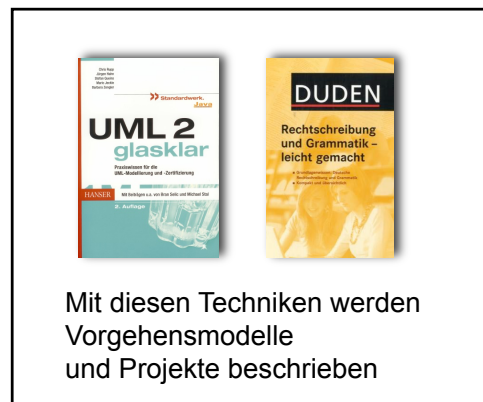
B) .. ob im betrachteten Projektzustand Dokumente fehlen, bzw welche Dokumente noch zu erzeugen sind, und mit welchen Inhalten?

C) .. wie mit Änderungen von Dokumenten & -inhalten umzugehen ist?



Probleme im Detail: A

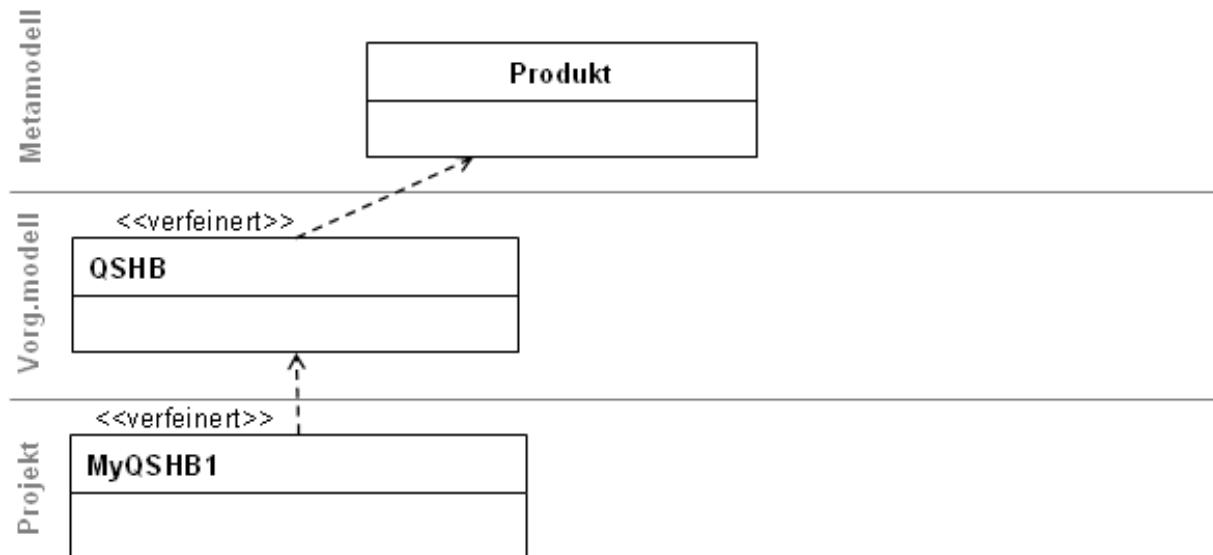
- Vorgehensmodelle werden derzeit beschrieben durch...
 - Formale Sprachanteile (UML, OCL, ...)
 - Natürliche Sprachanteile
- Produktinhalte & -abhängigkeiten sind verstärkt in natürlicher Sprache dargestellt
- Damit keine automatische Prüfung der Eingaben möglich



Probleme im Detail: A

Welche Anforderungen gibt es beim Beschreiben von Produktstrukturen?

- Verfeinerungskonzept benötigt (meist 3 „Ebenen“):
 - höchste Ebene beschreibt, was allen Projekten gemein ist – „Metamodell“
 - mittlere Ebene beschreibt, was Projekte einer Organisation* gemein ist – „Vorg.modell“
 - unterste Ebene beschrieben, was für genau ein konkretes Projekt gilt – „Projekt“



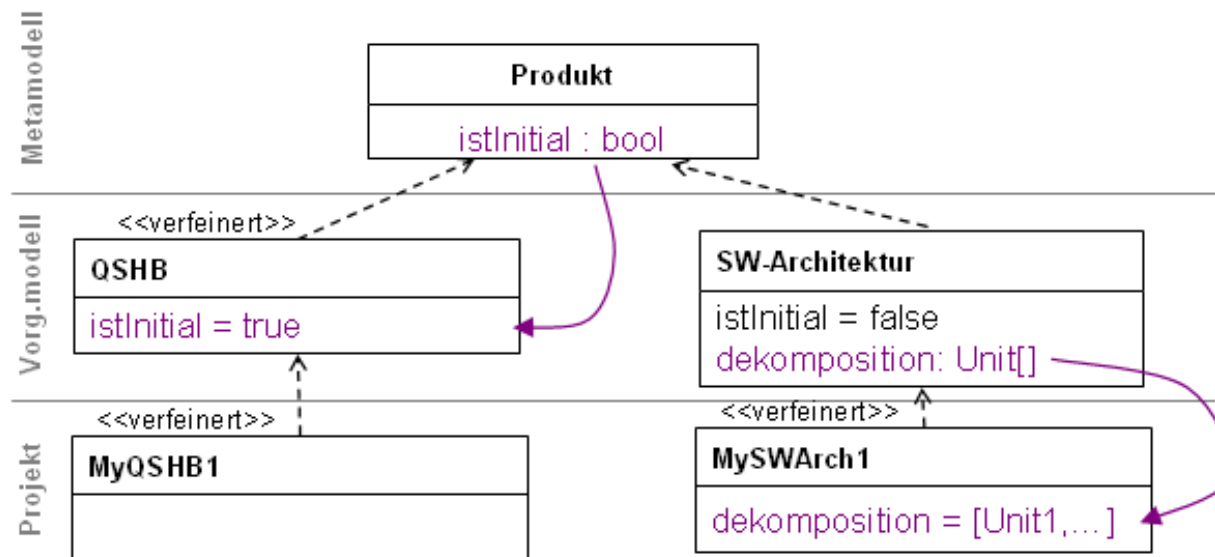
*wahlweise auch andere Kriterien (z.B. Branche)



Probleme im Detail: A

Welche Anforderungen gibt es beim Beschreiben von Produktstrukturen?

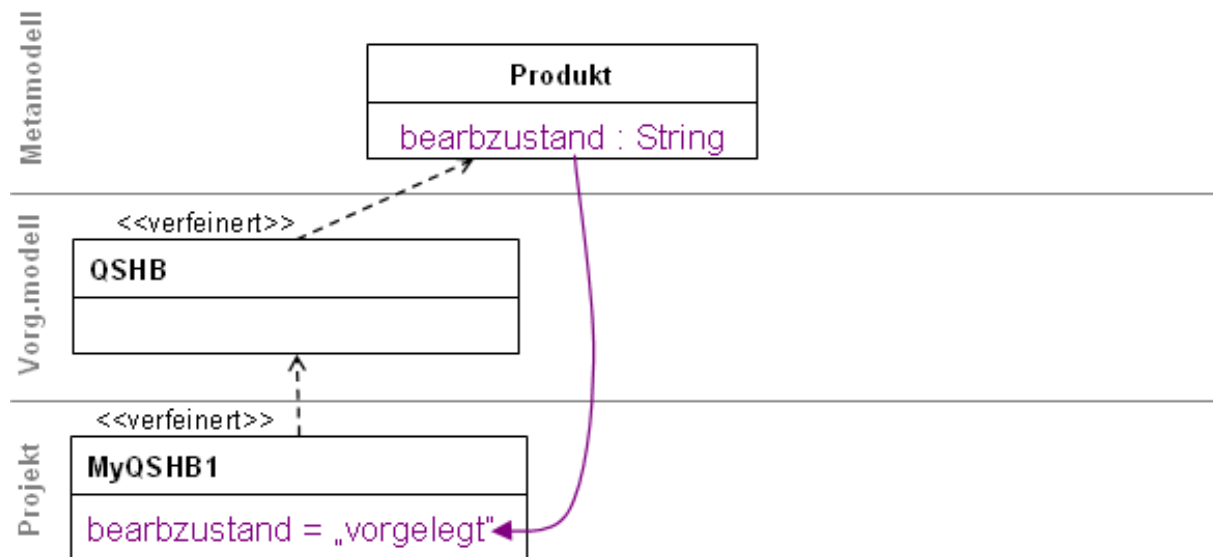
- Vorgaben für Verfeinerungen (1.Ordnung)
 - z.B. im Metamodell „Für jeden Produkttyp ist festzulegen, ob er initial ist oder nicht“
 - z.B. im Modell „jede SW-Architektur enthält eine Dekomposition als Liste von Einheiten...“



Probleme im Detail: A

Welche Anforderungen gibt es beim Beschreiben von Produktstrukturen?

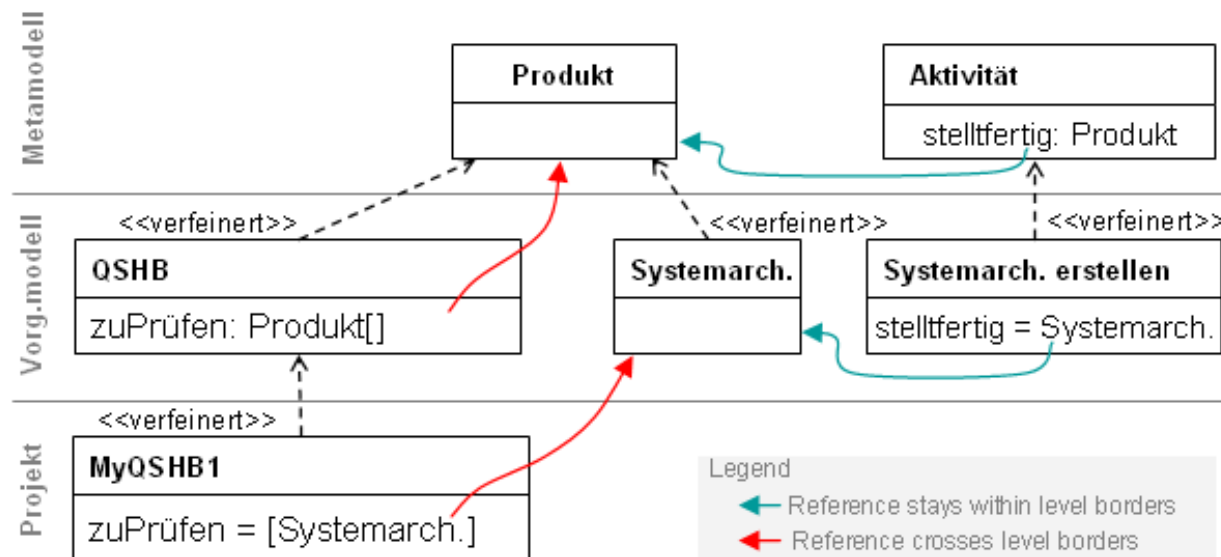
- Vorgaben für Verfeinerungen (n.Ordnung)
 - z.B. im Metamodell „in allen Projekten haben alle Dokumente einen Bearbeitungszustand“



Probleme im Detail: A

Welche Anforderungen gibt es beim Beschreiben von Produktstrukturen?

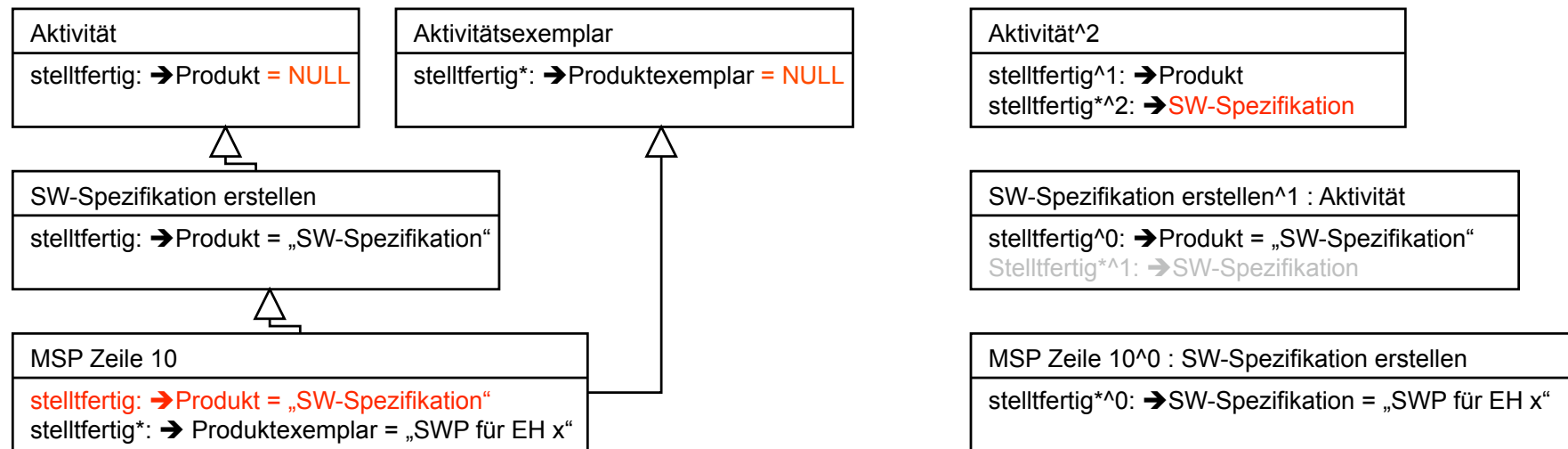
- Reflection
 - z.B.: „für alle Dokumente eines im QSHB auftauchenden Produkttyps, muss eine Prüfspezifikation erzeugt werden“



Probleme im Detail: A

Welche Probleme gibt es mit existierenden Beschreibungsansätzen?

- Metamodellierung nach MOF: (! Reflection), ! Verfeinerungsvorgaben n.Ord.
- Metamodellierung nach DeepInst. ! Reflection, ! echte Verfeinerung
- Modularisierung/Spezialisierung Artefaktbelastete Verfeinerungsvorgaben n.Ord.



Modularisierung / Spezialisierung

Deep Instantiation



Probleme

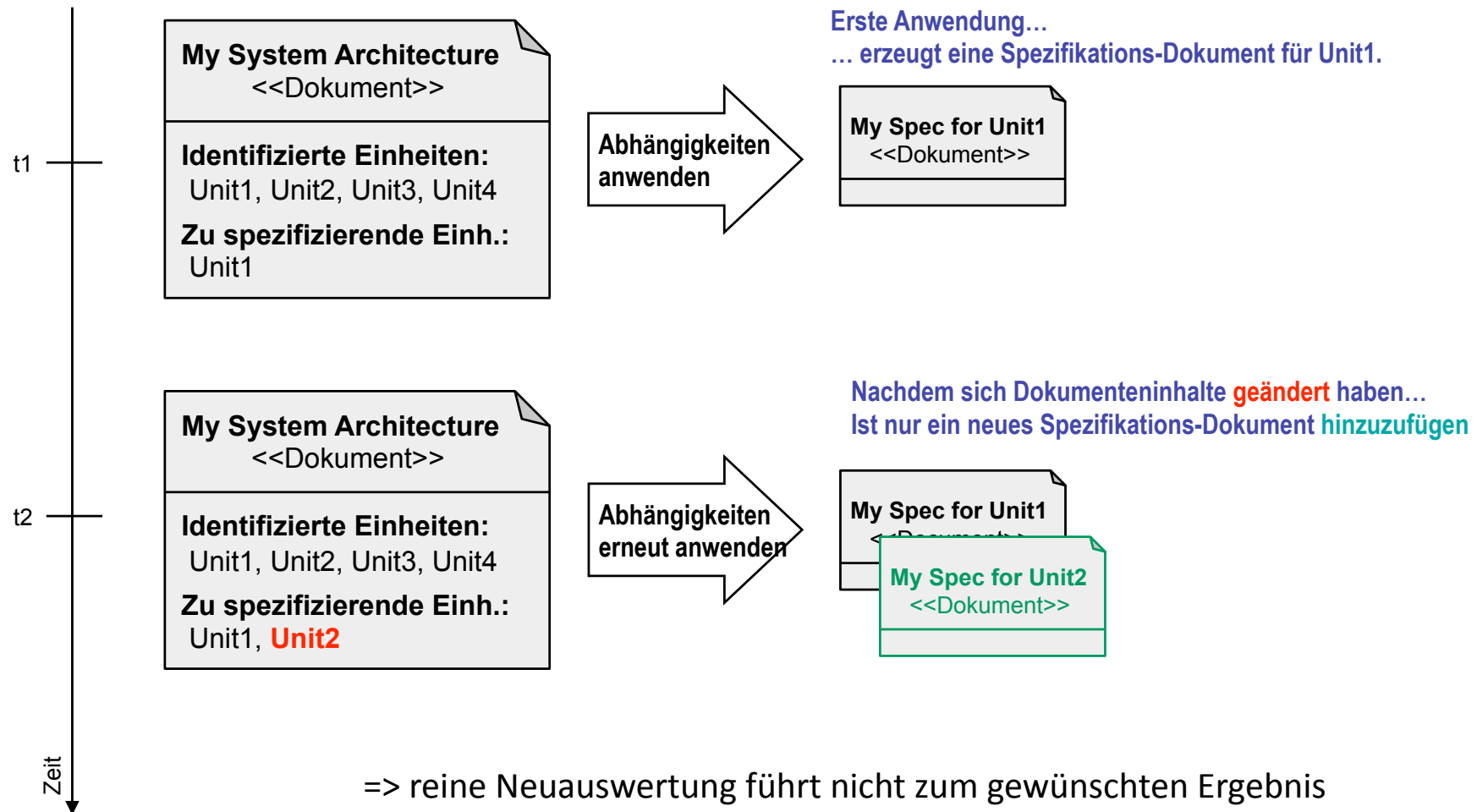
A) .. ob Dokumente & -inhalte korrekt bzgl. eines Vorgehensmodell sind?

B) .. ob im betrachteten Projektzustand Dokumente fehlen, bzw welche Dokumente noch zu erzeugen sind, und mit welchen Inhalten?

C) .. wie mit Änderungen von Dokumenten & -inhalten umzugehen ist?



Probleme im Detail: B&C



=> reine Neuauswertung führt nicht zum gewünschten Ergebnis

=> Doublettenerkennung keine Option (wegen Veränderungen)



Agenda

- Motivation & Zielsetzung
- Problemstellung
- **Lösung**
- Was offen bleibt
- Zusammenfassung

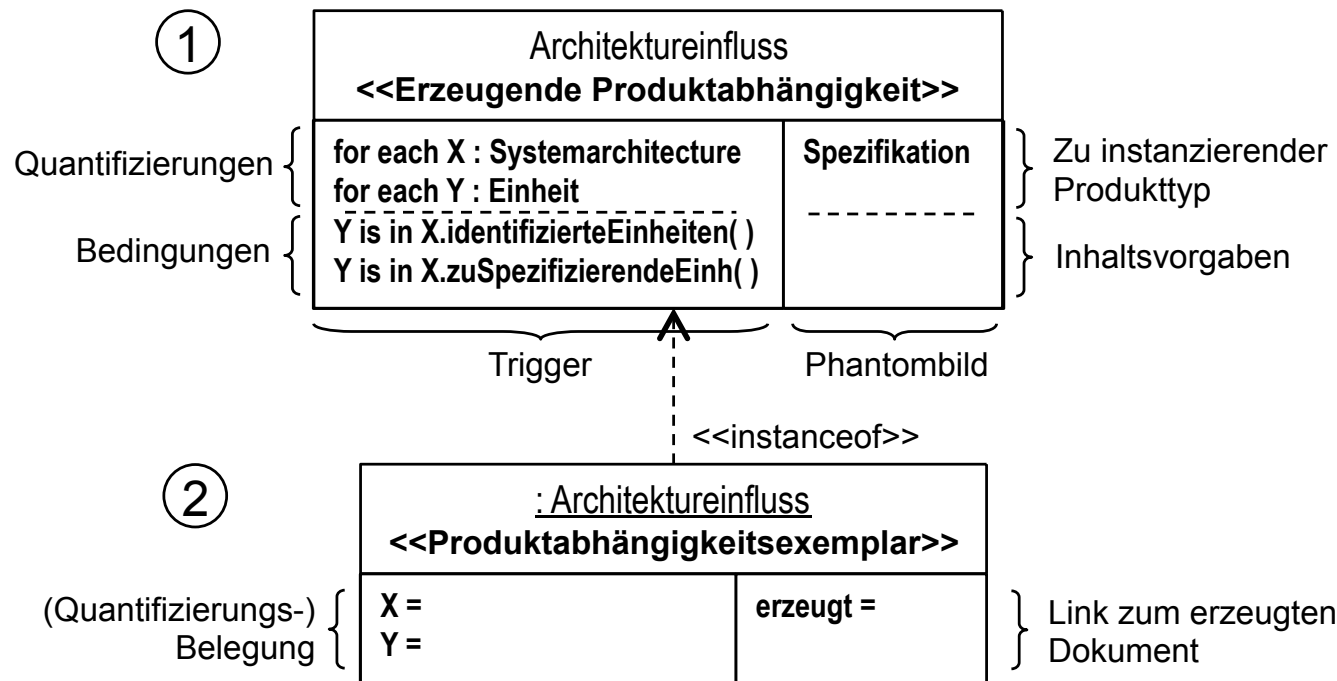


Lösung

- Lösungsidee B&C – Umgang mit Änderungen
- Lösungsidee A – Beschreibung von Produktstrukturen
- Integration der Lösungsideen



Lösungsidee B&C

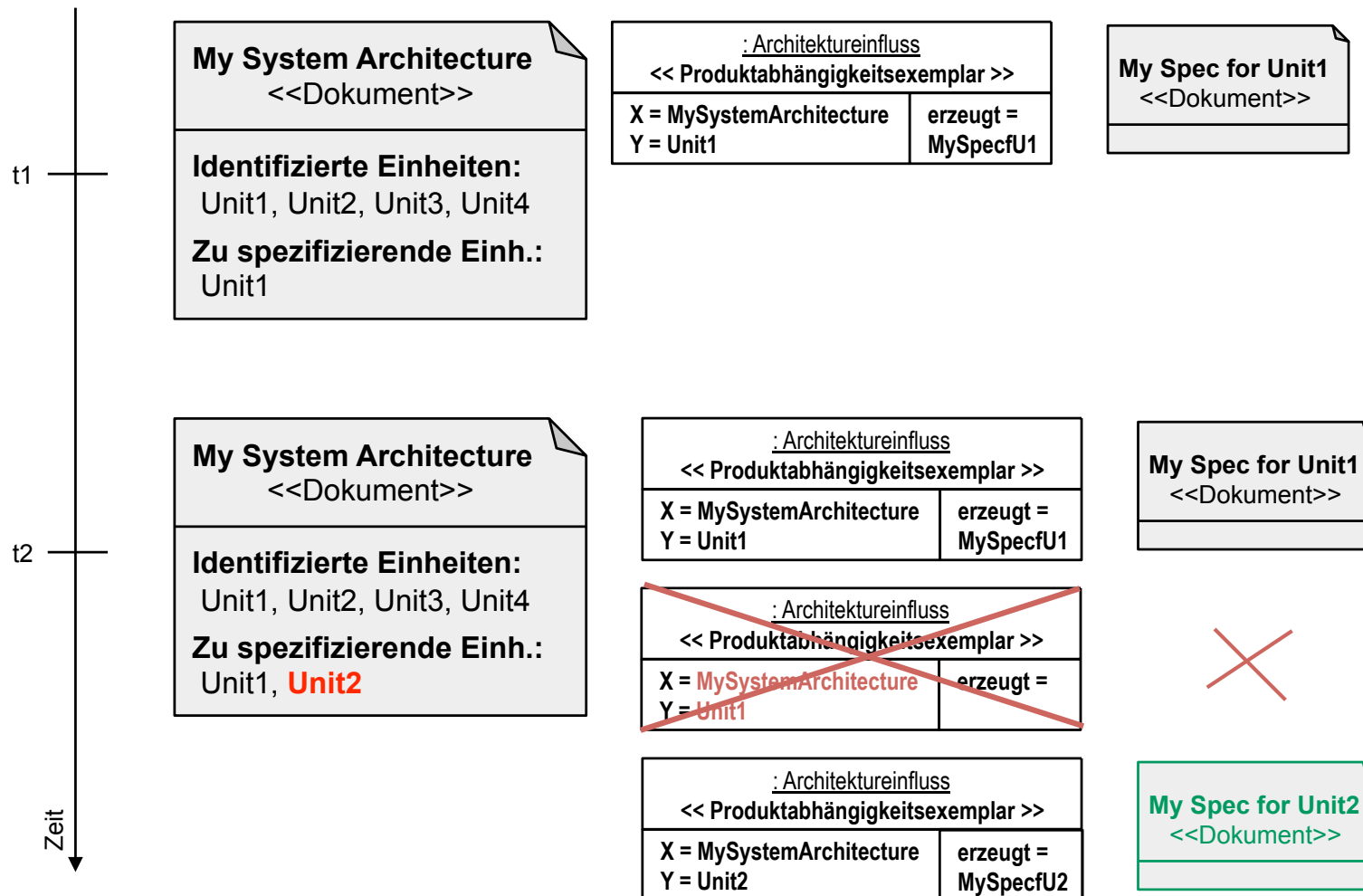


- entsteht für jede mögliche Quantifizierungsbelegung

③ - Produktabhängigkeitsexemplare als Teil des Projektzustands,
mit Unique-Constraint über der Quantifizierungsbelegung



Lösungsidee B&C – am Beispiel



Lösung

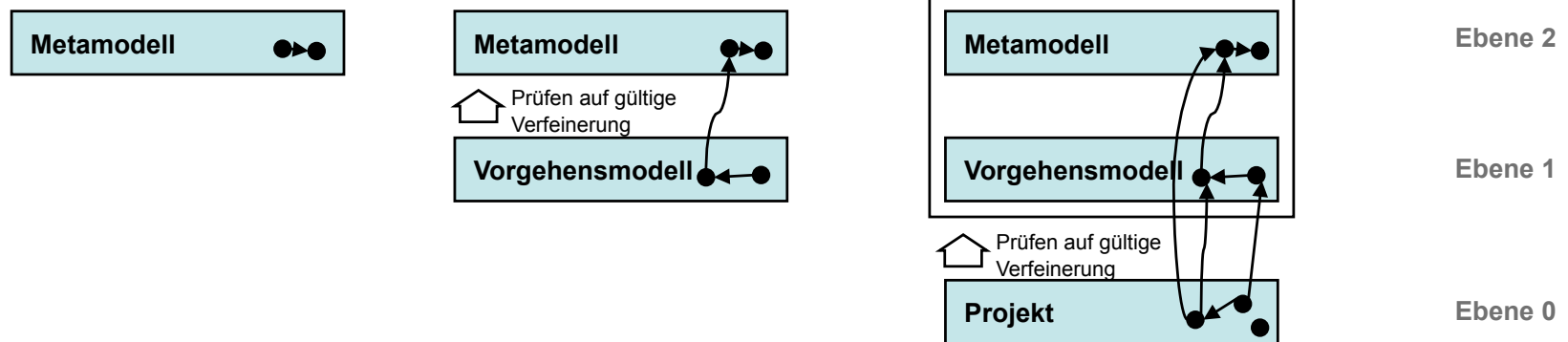
- Lösungsidee B&C – Umgang mit Änderungen
- Lösungsidee A – Beschreibung von Produktstrukturen
- Integration der Lösungsideen



Lösungsidee A

- Ebenen mit hierarchischer Ordnung

- Konsistenzprüfung immer von einer Ebene gegen die Summe aller hierarchisch höheren (für Reflection & schrittweise Verfeinerungen)



- Ebenenbestandteile

- bestehen aus Elementen mit Eigenschaften (Werte, Links auf andere Elemente, Instanzbeschreibungen)

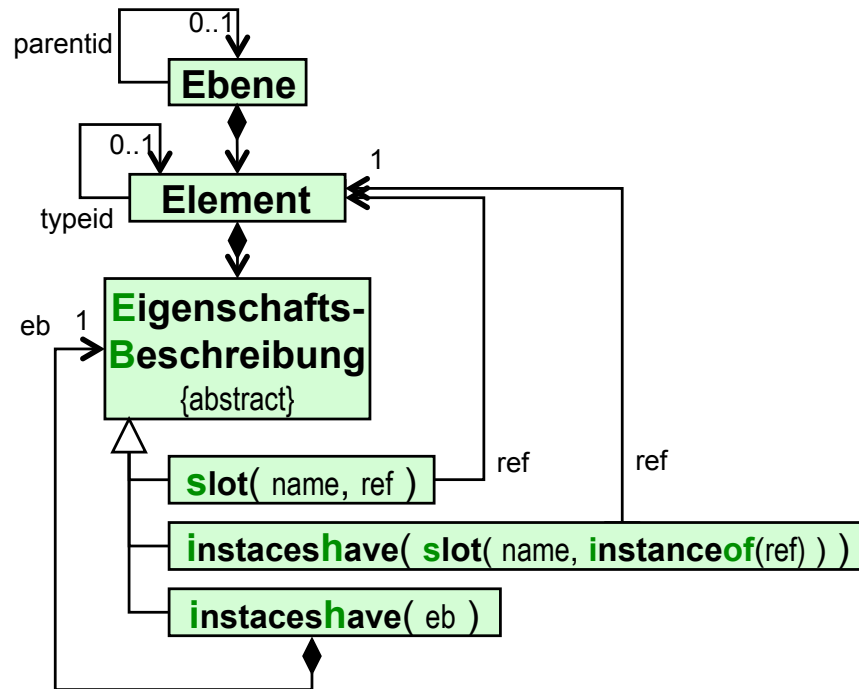
- Instanzierung als Ebenentrennung

- Jedes Element einer Ebene Instanz eines Elements der unmittelbar höheren Ebene
- alle Vorgaben höherer Ebenen umsetzen, Ergänzungen möglich

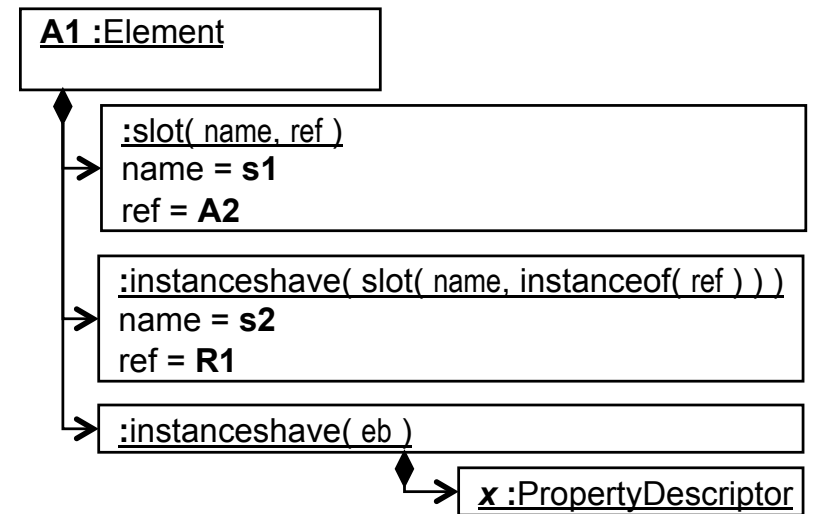


Lösungsidee A

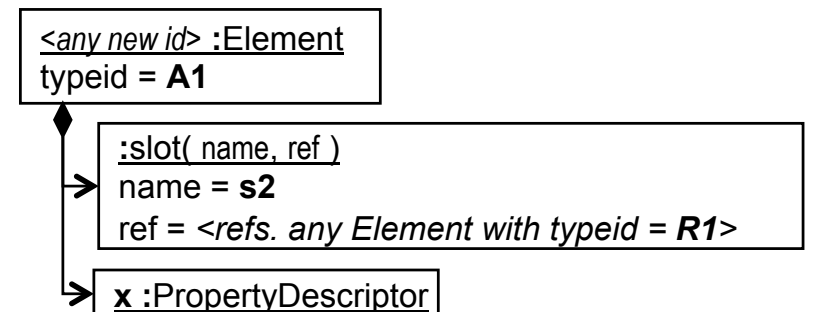
Struktur



Instanziierungssemantik

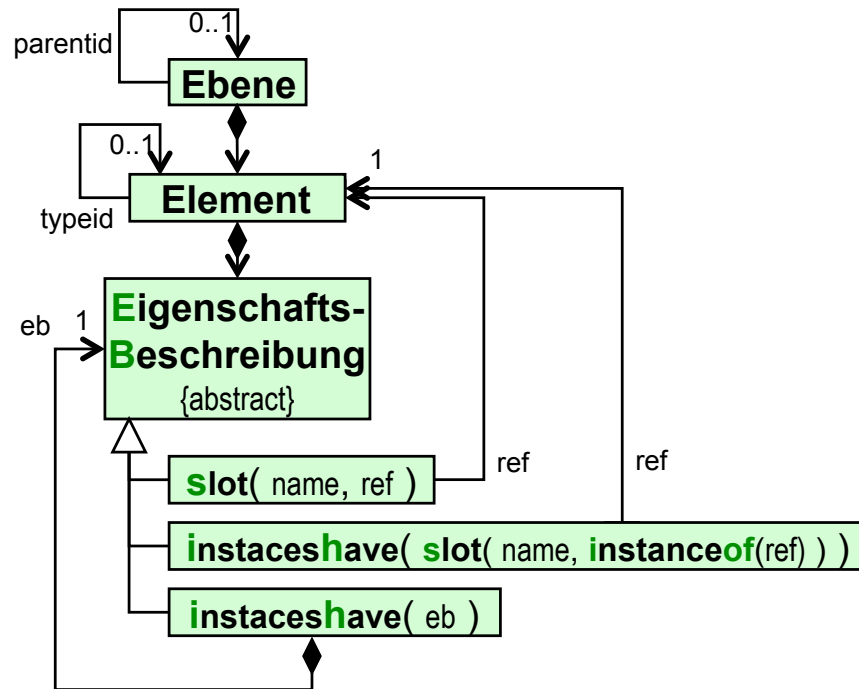


instanzieren



Lösungsidee A

Struktur



Instanziierungssemantik

A1

slot(**s1**, **A2**)
 instaceshave(slot(**s2**, instanceof(**R1**)))
 instaceshave(**x**)

instanzieren

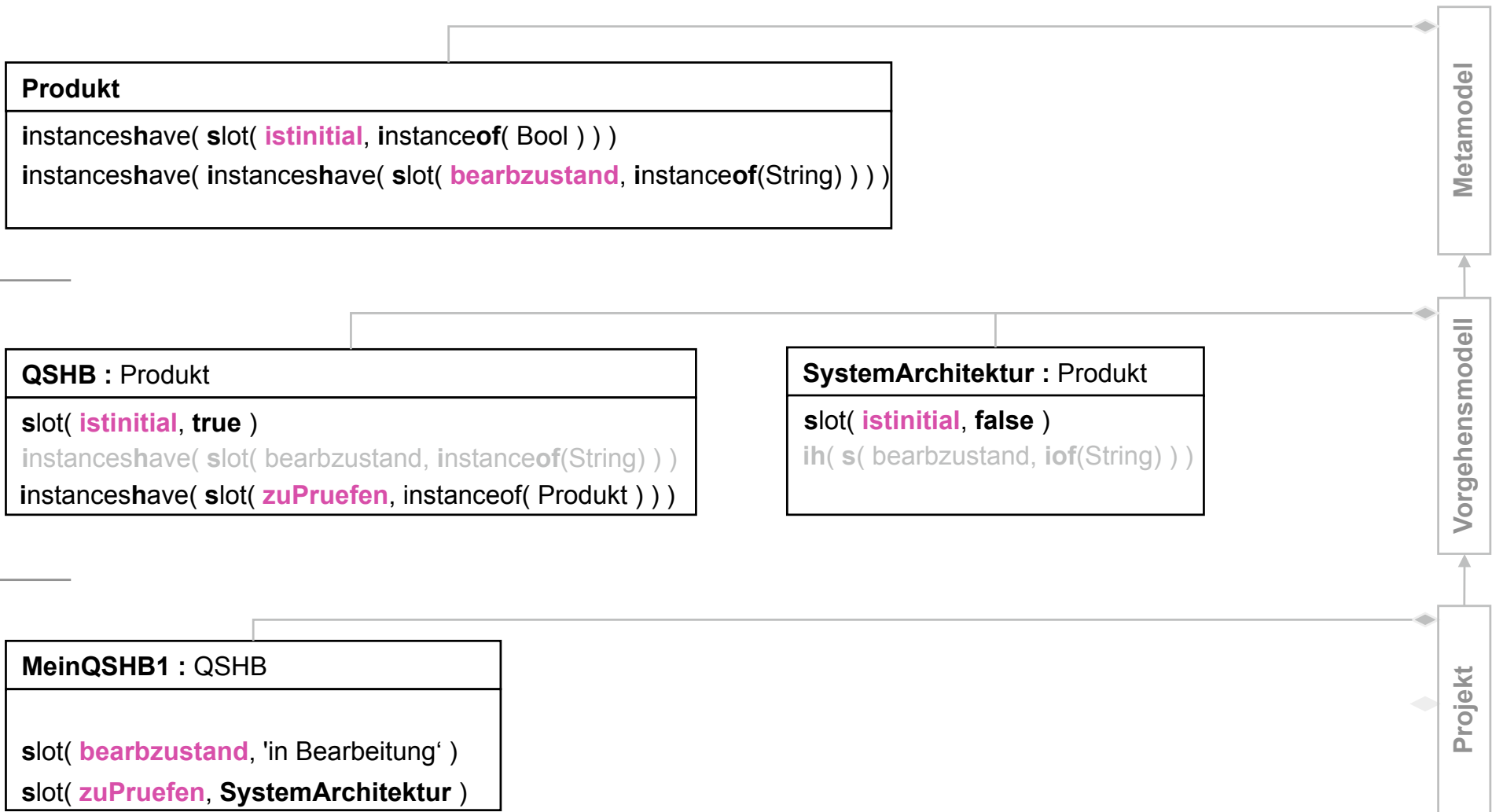
Id des Elements Typindikator

<any new id> : **A1**

slot(**s2**, <ref. to any Element with typeid = **R1**>)
x



Lösungsidee A

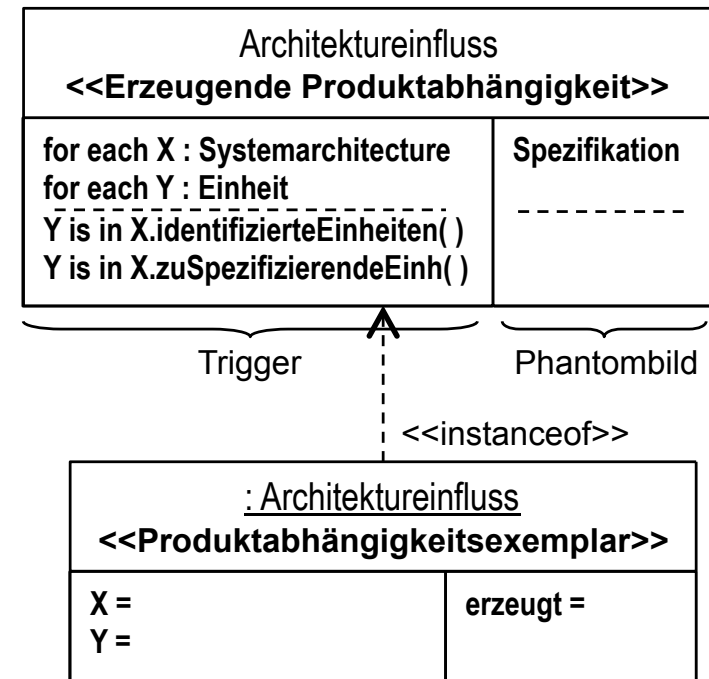
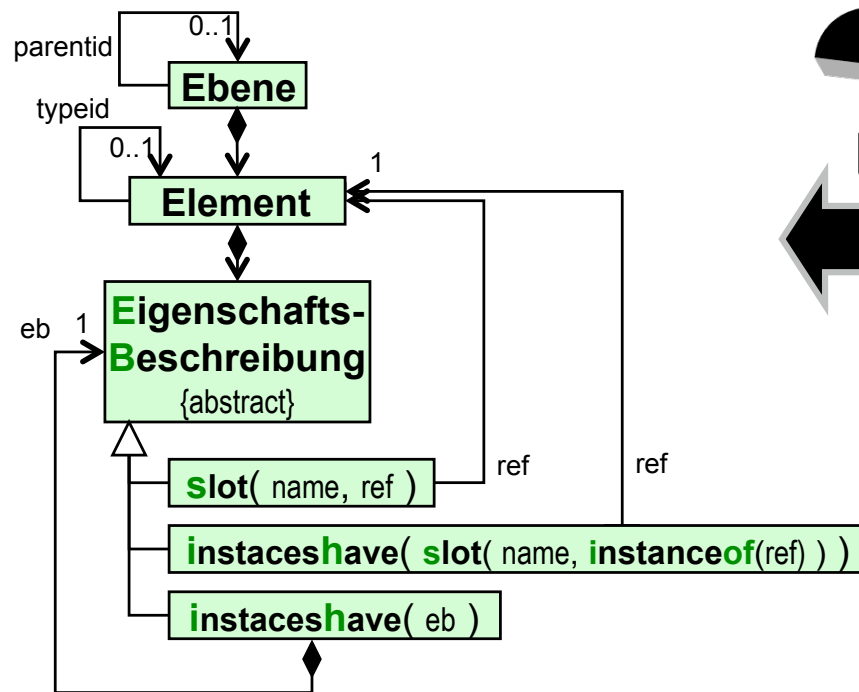


Lösung

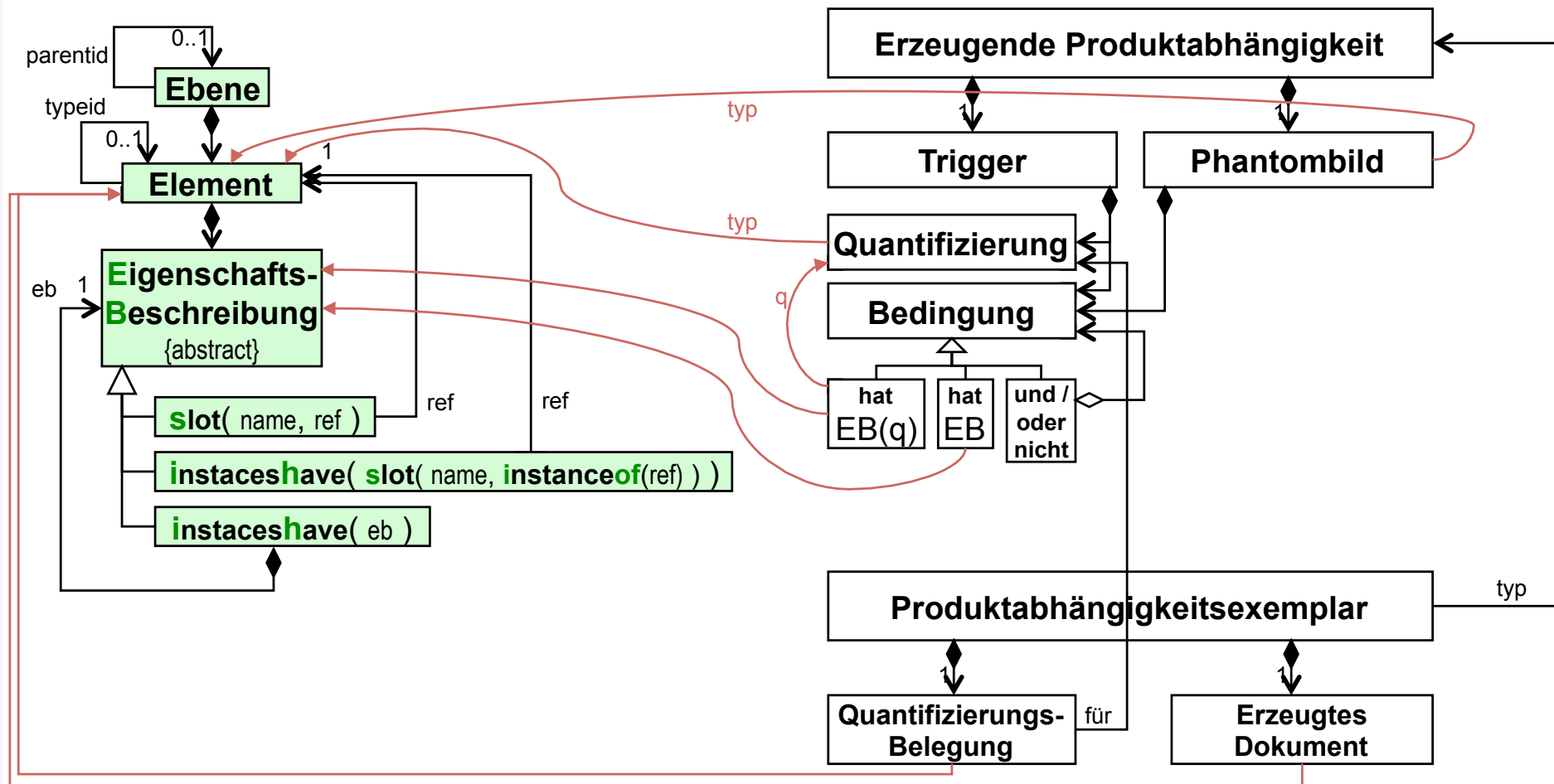
- Lösungsidee B&C – Umgang mit Änderungen
- Lösungsidee A – Beschreibung von Produktstrukturen
- Integration der Lösungsideen



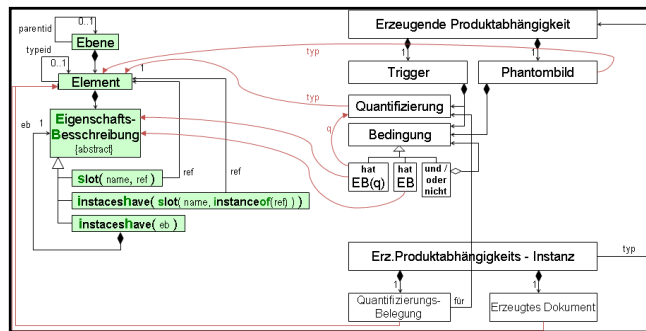
Lösungsintegration



Lösungsintegration



Lösungsintegration – Formalisierung



~>
Algebraische
Formalisierung

Definition Menge EB (alle Eigenschaftsbeschreibungen):
 $name \in String, ref \in ID \rightarrow slot(name, ref) \in EB$
 $name \in String, ref \in ID \rightarrow instanceshave(slot(name, instanceof(ref))) \in EB$
 $t \in EB \rightarrow instanceshave(t) \in EB$

Definition Menge EL (alle Elemente):
 $id \in ID, typeid \in (ID \cup \{none\}), T \subseteq EB \rightarrow (id, T, typeid) \in EL$
 mit
 $\forall name \in String: |\{slot(name, ref) \in T\}| \leq 1$

Definition Relation instanceOf_{EL} ⊆ EL x EL (ist ein e ∈ EL Instanz von e1 ∈ EL):
 $((id, T, typeid), (typeid, T1, id1)) \in instanceOf_{EL} \Leftrightarrow \forall t \in T: eval(T1, t)$
 wobei
 $eval(T1, ih(slot(n, iof(r)))) = \exists r' \in ID: slot(n, r') \in T1$
 $eval(T1, ih('t')) \text{ mit } t' \in EB = t' \in T1$
 $eval(T1, slot(n, r)) = true$

Definition Menge GM (alle Gesamtmodelle):
 $M \subseteq EL \rightarrow M \in GM$
 mit
 $\forall id \in ID: |\{(id, T, typeid) \in M\}| \leq 1$

Definition Relation wohlgetypt_{GM} ⊆ GM
 $M \in GM \rightarrow M \in wohlgetypt_{GM}$
 mit
 $(_ , _ , typeid) \in M \rightarrow \exists (typeid, _) \in M,$
 $(_ , T, _) \in M, slot(_ , r) \in T, r \in ID \rightarrow \exists (r, _) \in M,$
 $(_ , T, typeid) \in M, slot(n, r) \in T, r \in ID \rightarrow$
 $\exists (r, _) \in M: \exists (typeid, T1, _) \in M: \exists ih(s(n, iof(R))) \in T1$

Definition Menge Q (aller Quantifizierungen):
 $x \in Variable, typeid \in ID \rightarrow x : typeid \in Q$

Definition Menge SC (aller Bedingungen):
 $x \in Variable, p \in EB \rightarrow (x \text{ has } p) \in SC$
 $x, y \in Variable, n \in String \rightarrow (x \text{ has } slot(n, y)) \in SC$
 $t \in SC \rightarrow (\text{not } t) \in SC$
 $t1, t2 \in SC \rightarrow (t1 \text{ and } t2) \in SC$
 $t1, t2 \in SC \rightarrow (t1 \text{ or } t2) \in SC$
 $true \in SC$

Definition Funktion match: TRIG x wohlgetypt_{GM} → Bool
 $match(\{(x1:typeid1), \dots, (xn:typeidn)\}, sc, M) =$
 $\forall x1: \dots \forall xn:$
 $(x1_typeid1) \in M, \dots, (xn_typeidn) \in M,$
 $interpret(sc, M):$
 wobei
 $interpret(x \text{ has } slot(n, z)) = (x1, T, _) \in M \wedge slot(n, z) \in T$
 $interpret(t1 \text{ and } t2, M) = interpret(t1, M) \wedge interpret(t2, M)$
 $interpret(t1 \text{ or } t2, M) = interpret(t1, M) \vee interpret(t2, M)$
 $interpret(\text{not } t, M) = \neg interpret(t, M)$
 $interpret(true, M) = true$

Definition Menge PBLD (aller Phantombilder):
 $sc \subseteq (SC \cup EB), typeid \in ID \rightarrow (sc, typeid) \in PBLD$
 mit
 $\forall name \in String: |\{slot(name, ref) \in sc\}| \leq 1$

Definition Menge TRIG (aller Trigger):
 $q \subseteq Q, sc \in SC \rightarrow (q, sc) \in TRIG$
 mit
 $\forall x \in Variable: |\{(x : typeid) \in q\}| \leq 1$

Definition Menge IA (aller Induktionsanfänge):
 $b \in PBLD, pid \in PID \rightarrow (pid, b) \in IA$

Definition Menge IS (aller Induktionsschritte):
 $t \in TRIG, b \in PBLD, pid \in PID \rightarrow (pid, t, b) \in IS$

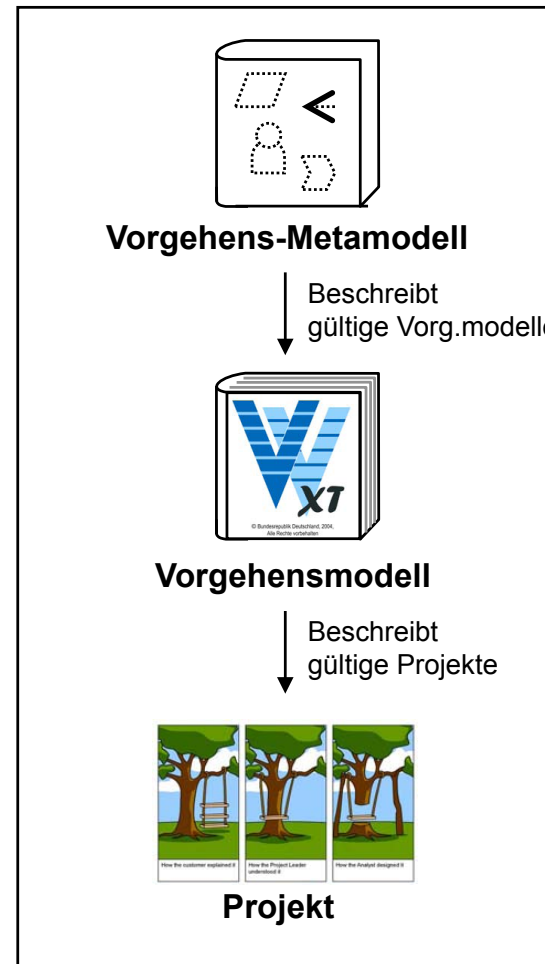


Lösungsintegration – das Gesamtbild

Definition Menge EB (alle Eigenschaftsbeschreibungen) $name \in String, ref \in ID \rightarrow slot(name, ref) \in EB$ $name \in String, ref \in ID \rightarrow instancehave(slot(name, instance(ref))) \in EB$ $t \in EB \rightarrow instancehave(t) \in EB$	Definition Menge TI (alle Instanztypen) $id \in ID, typeid \in (ID \cup \{none\}), T \subseteq EB \rightarrow (id, T, typeid) \in TI$ mit $\forall name \in String: \{ \{ slot(name, ref) \in T \} \} \Leftarrow 1$
Definition Funktion instanceOf_{EB, TI} (ist eine Instanz von einer EB) $((id, T, typeid), (typeid, T, id1)) \in instanceOf_{EB, TI} \Leftrightarrow \forall t \in T: eval(T, t)$ $woher$ $eval(T, \{ \{ slot(name, ref) \} \}) = \exists t \in ID: slot(name, ref) \in T$ $eval(T, \{ \{ t \} \}) \text{ mit } t \in EB = \{ t \in T$ $eval(T, \{ slot(name, ref) \}) = true$	Definition Funktion wohlgepflegt_{GM, CM} $M \in GM \rightarrow M \in CM$ mit $C_typeid \in M \rightarrow \exists typeid_jeM,$ $C_T_jeM, slot_jeT, refID \rightarrow \exists v_jeM,$ $C_T_typeid \in M, slot(name, ref) \in T, refID \rightarrow$ $\exists v_jeM: \exists typeid, T_jeM: \exists \{ \{ slot(name, ref) \} \in T$
Definition Menge Q (alle Quantifizierungen) $x \in Variable, typeid \in ID \rightarrow x: typeid \in Q$	Definition Funktion match_{TBI, x wohlgepflegt_GM} \rightarrow bool $match(\{ \{ \{ x: typeid \}, \dots, \{ x: typeid \} \}, sc, M) =$ $\forall t1: \forall t2: \dots \forall t_n: \{ \{ x: typeid \} \} \in M,$ $interpret(sc, M)$ $woher$ $interpret(x: has slot(n, z)) = \{ \{ T_jeM \} \} \text{ mit } \{ \{ slot(name, ref) \} \} \in T$ $interpret(t1 \wedge t2: M) = interpret(t1, M) \wedge interpret(t2, M)$ $interpret(t1 \vee t2: M) = interpret(t1, M) \vee interpret(t2, M)$ $interpret(not t: M) = \neg interpret(t, M)$ $interpret(trip: M) = true$
Definition Menge SC (alle Bedingungen) $x \in Variable, p \in EB \rightarrow (x \text{ has } p) \in SC$ $x \in Variable, n \in String \rightarrow (x \text{ has slot}(n, y)) \in SC$ $t \in SC \rightarrow (not t) \in SC$ $t1 \wedge t2 \in SC \rightarrow (t1 \wedge t2) \in SC$ $t1 \vee t2 \in SC \rightarrow (t1 \vee t2) \in SC$ $true \in SC$	Definition Menge PBLD (alle Projektionen) $sc \subseteq (SC \cup EB), typeid \in ID \rightarrow (sc, typeid) \in PBLD$ mit $\forall name \in String: \{ \{ slot(name, ref) \} \in sc \} \Leftarrow 1$
Definition Menge IA (alle Instanzaktionen) $b \in PBLD, pid \in PID \rightarrow (pid, b) \in IA$	Definition Menge TRIG (alle Trigger) $q \subseteq Q, sc \in SC \rightarrow (q, sc) \in TRIG$ mit $\forall x \in Variable: \{ \{ \{ x: typeid \} \in q \} \} \Leftarrow 1$
Definition Menge IS (alle Instanzaktionen) $t \in TRIG, b \in PBLD, pid \in PID \rightarrow (pid, t, b) \in IS$	



Integriertes Lösungskonzept
 - zur Beschreibung aller Ebenen
 - zum Umgang mit Änderungen



Agenda

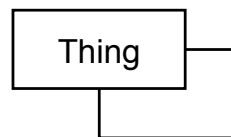
- Motivation & Zielsetzung
- Problemstellung
- Lösung
- **Was offen bleibt**
- Zusammenfassung



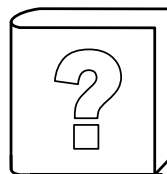
Was offen bleibt...

Neben Erzeugenden Produktabhängigkeiten noch weitere Konzepte vorhanden
(basierend auf Versionen und inhaltlichen Produktabhängigkeiten bspw)

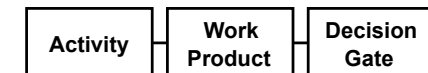
- **Frage:** wie sind diese in die vorgestellte Lösung einzubauen?
- Aber: „natürliches Wachstum“ des Konzeptes sollte vermieden werden
- Daher: konkrete Bedarfsanalyse Durchführung + Anforderungsfestlegung
- Dazu: Ein geeignetes Metamodell bestimmen



Allgemeinstmöglichstes MM



gesuchtes
Vorgehens-Metamodell



Z.B. VMXT-Metamodell

Agenda

- Motivation & Zielsetzung
- Problemstellung
- Lösung
- Was offen bleibt
- **Zusammenfassung**



Zusammenfassung

- **Was ist erreicht?**
 - Allgemeines Lösungskonzept zur
 - Beschreibung von Produktinhalten und Erzeugender Abhängigkeiten
 - Umgang mit Veränderungen von Produkten & -inhalten
- **Was sind die nächsten Schritte?**
 - Bedarfsanalyse und Festlegung eines Metamodells
 - Erweiterung des Lösungskonzeptes um noch fehlende Konzepte



Vielen Dank für die Aufmerksamkeit!

Fragen

