

# Service Systems Modelling

Prof. Dr. Klaus-Peter Fähnrich  
in Vertretung Martin Böttcher

05.05.2009

## Ziel der Vorlesung

Beantwortung der Frage:

# Was bedarf es für die Modellierung von Servicesystemen?

domänenrelevante  
Konzepte

Stand der Forschung

Produktmodell

Ressourcenmodell

Komponentenmodell

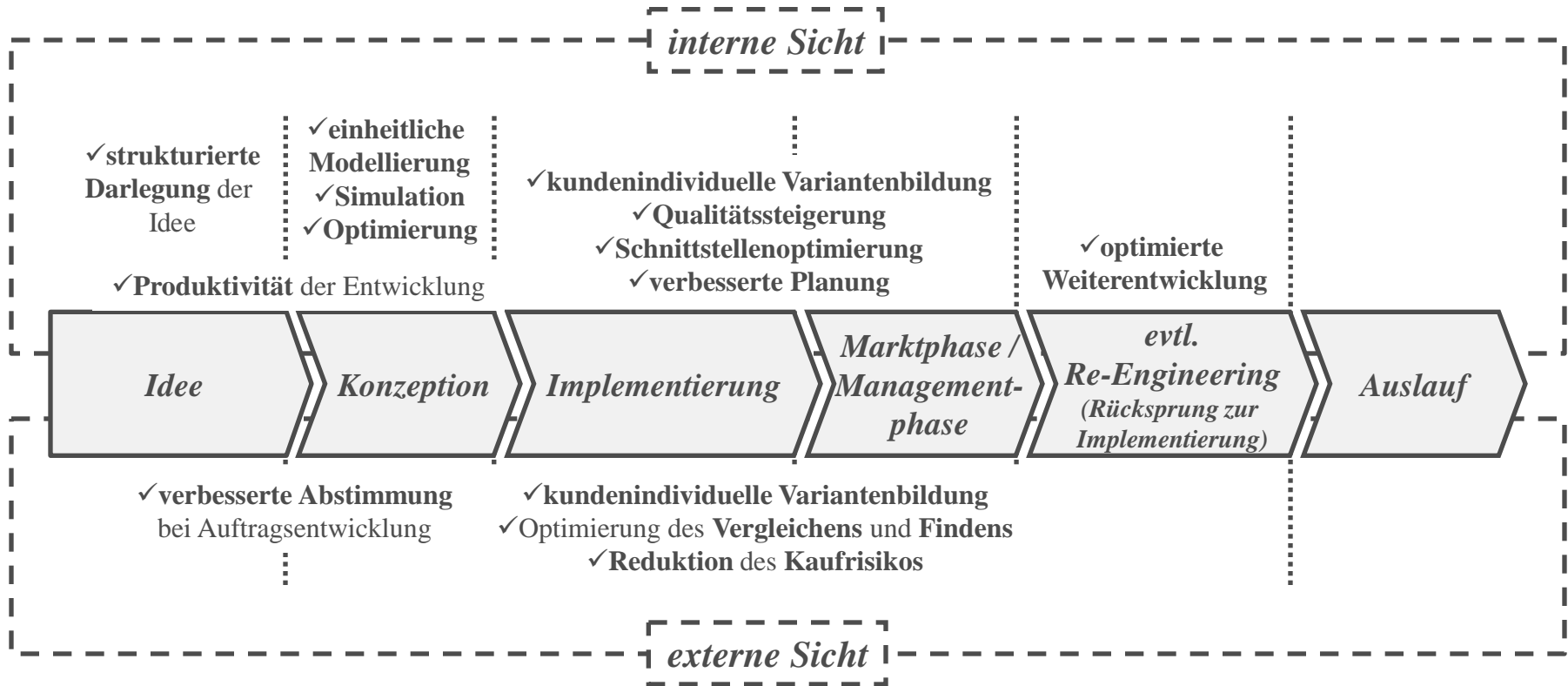
Metamodell

Prozessmodell

## Frage I

Warum beschäftigen wir uns mit Service Systems Modelling?

# Motivation des Service Systems Modelling (1)



## Motivation des Service Systems Modelling (2)

Welche Key-Performance-Dimensionen sind betroffen?

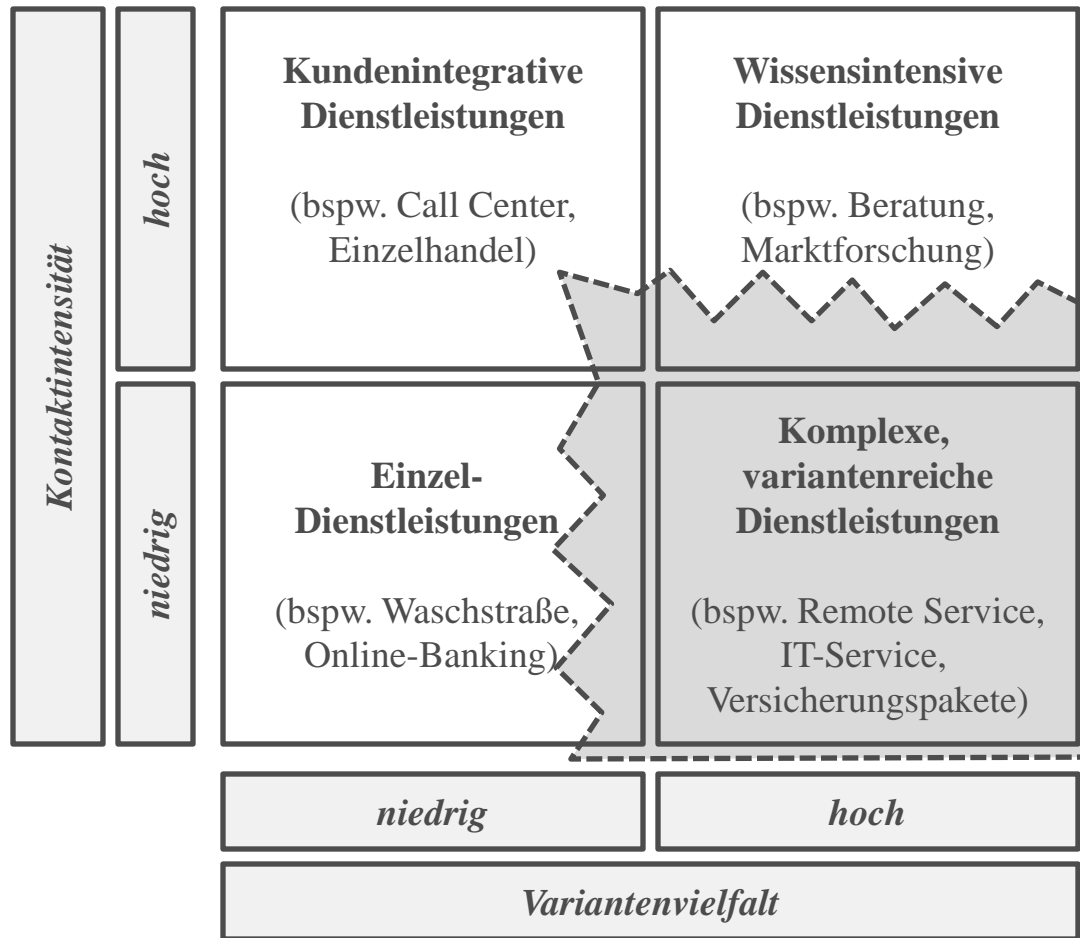
*Servicequalität*

*Serviceproduktivität (Entwicklung)*

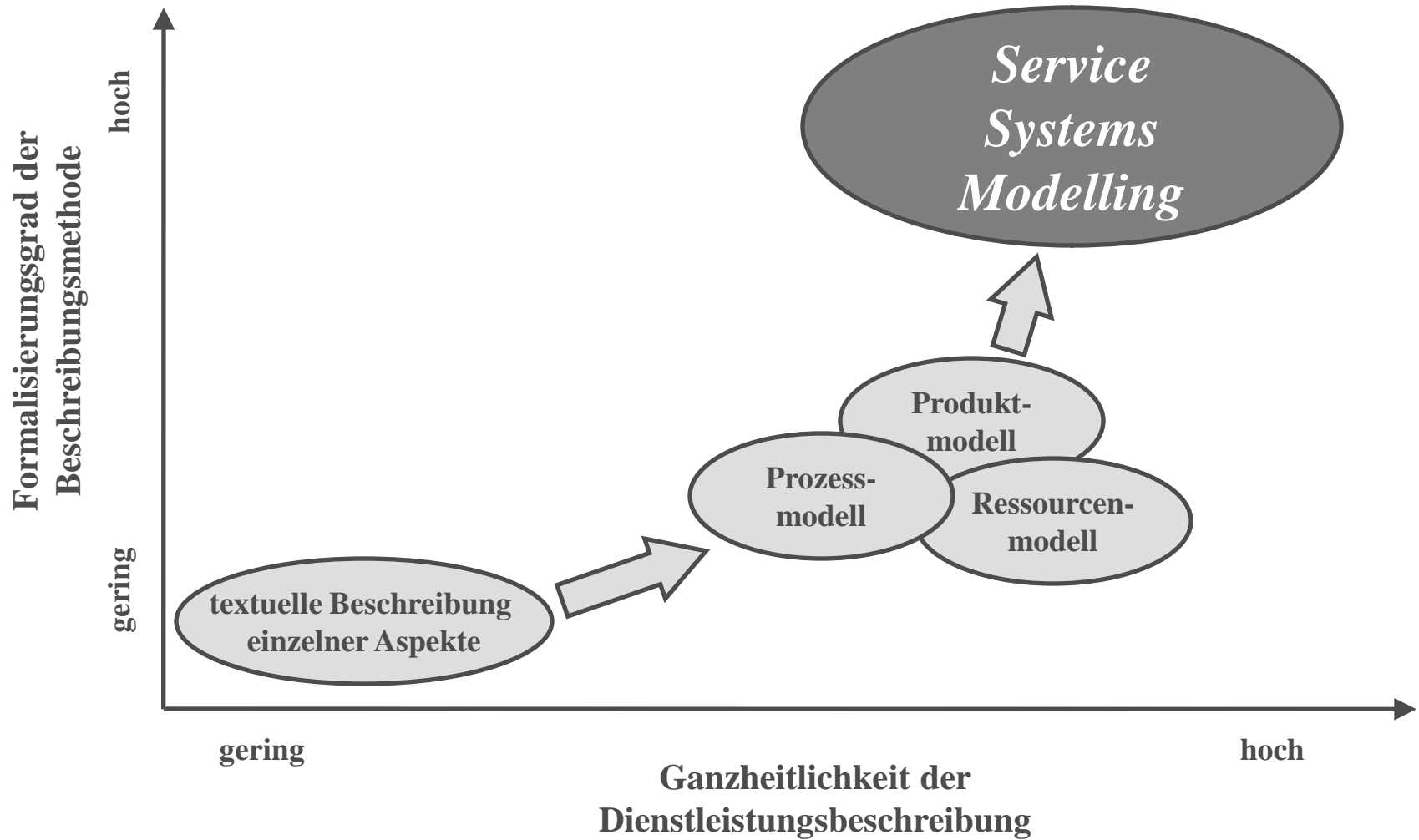
*Serviceproduktivität (Bereitstellung)*

*Serviceserviceinnovationen*

# Relevanz der Service Systems Modelling



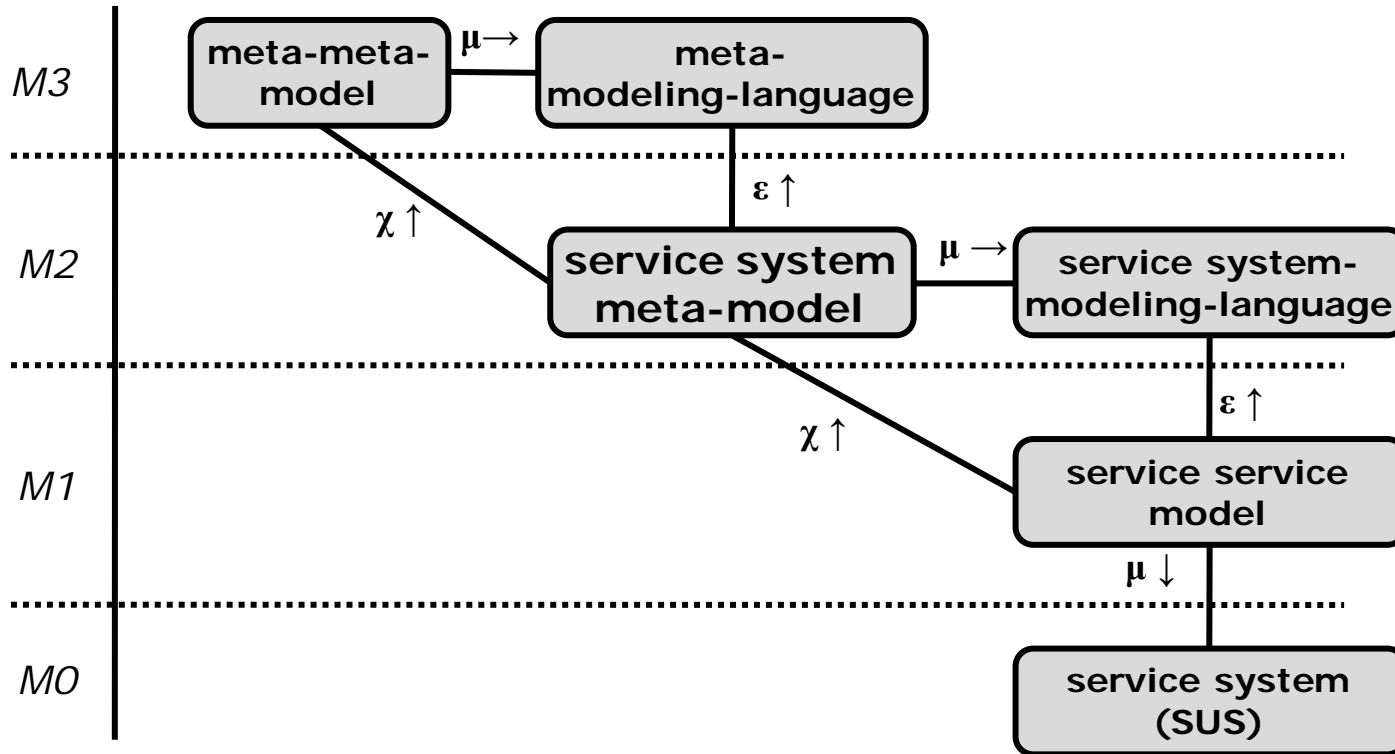
State of the Art und Entwicklung



## Frage II

Welche theoretischen Grundlagen bedarf es für die Modellierung von Service Systemen?





$\mu$  : RepresentationOf -  $\epsilon$  : ElementOf -  $\chi$  : ConformsTo

# Definition Service Systems (1)

Typologisierung

UNIVERSITÄT LEIPZIG

Institut für Informatik  
Betriebliche Informationssysteme

## Arbeitsdefinition des Begriffs Dienstleistung

- Vorherrschend werden Dienstleistungen als Leistungen definiert,
  - bei denen ein Leistungspotenzial existiert, welches die Fähigkeit und Bereitschaft zur Erbringung einer Leistung umfasst,
  - in deren Erstellungsprozesse externe Faktoren integriert werden, an denen oder mit denen die Leistung erbracht wird, und
  - deren Ergebnisse bestimmte materielle und immaterielle Wirkungen an den externen Faktoren darstellen.

Quelle: Fraunhofer IAO

# Was sind Service Systems und wie können diese Definiert werden?

**Definition Service Systems (2)**

service system  $(R, S, I, F)$

set of resources  $R_C \cup R_P = R.$

set of states  $S = \{s_1, \dots, s_o\}$

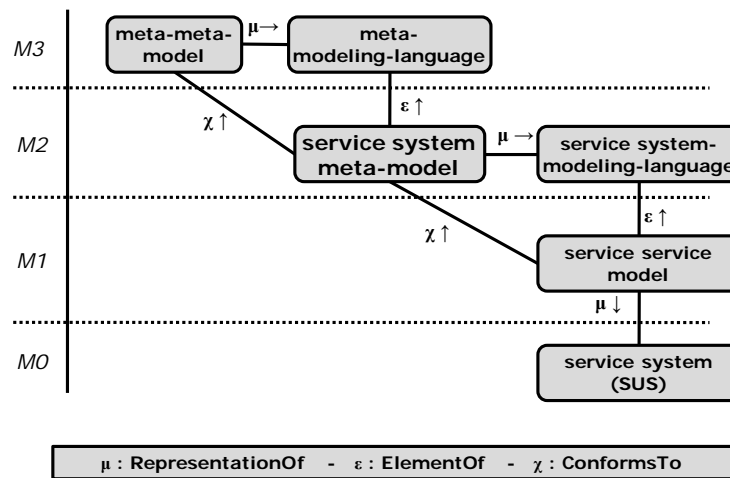
initial state  $I : R \rightarrow \mathcal{P}(S).$

final state  $F : R \rightarrow \mathcal{P}(S).$

constraints  $\exists r \in R_C : I(r) \neq F(r).$

# Entwicklung domänenspez. Sprachen (1)

Was bedarf es für die Erstellung domänenspezifischer Sprachen?



## Entwicklung domänenspez. Sprachen (2)

### 1. Domänenverständnis

manuelle Analyse von:

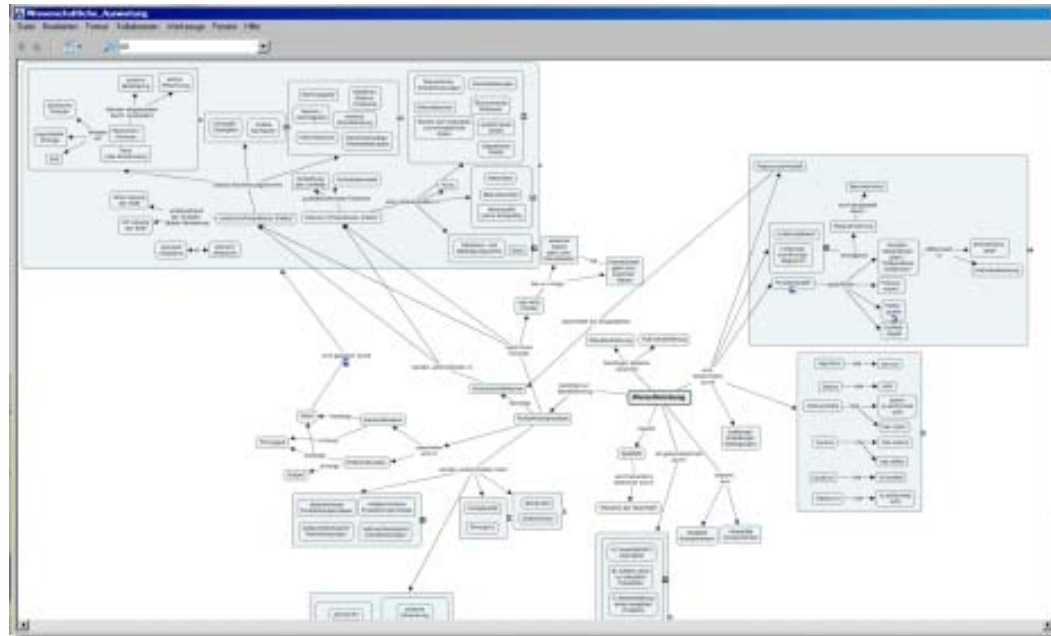
- 20 Standardwerken
- 12 existierenden Modellierungsansätze
- verschiedenen Anwendungsfällen

# Entwicklung domänenspez. Sprachen (2)

## 1. Domänenverständnis

## 2. Konzeptextraktion und Repräsentation

- Konzeptaggregation (Synonyme)
- Definition von Beziehungen
- Visualisierung



**Entwicklung domänenspez. Sprachen (2)**

1. Domänenverständnis
2. Konzeptextraktion und Repräsentation
- 3. Formalisierung**

- Anwendung mathematischer und aussagenlogischer Ausdrücke

*LocationPointAttributeAllocation : LocationPoints* →

$\mathbb{P}(\text{LocationPointAttributes} \times \text{LocationPointAttributeValue} \times \text{LocationPointAttributeValue})$ ,  
*LocationPointAttributeAllocation(locationPoint)* =  $\{(locationPointAttributes, min, max) : min \leq max\}$ .

*AvailabilityAllocation : Resources* →

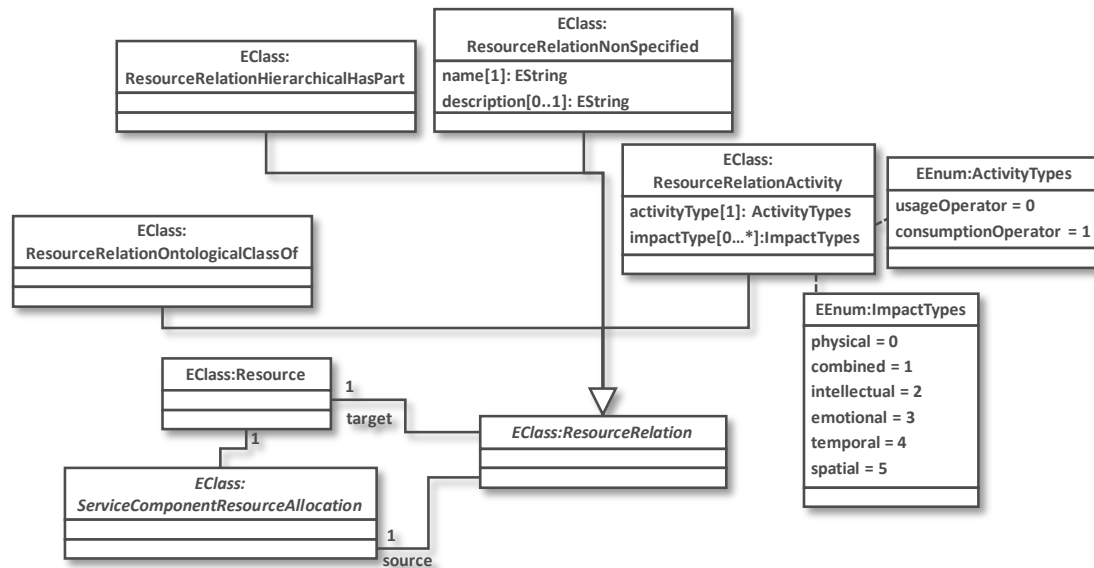
$\mathbb{P}(\mathbb{P} \text{ LocationPoints} \times \mathbb{P} \text{ LocationPoints} \times$   
 $\mathbb{P} \text{ TemporalPeriods} \times \mathbb{P} \text{ TemporalPeriods} \times$   
 $\mathbb{P} \text{ ServiceComponents} \times \text{ResourcePositions})$ .

*AvailabilityAllocation(resource)* =  $\{(LocativeAvailabilities, LocativeExceptions,$   
 $\text{TemporalAvailabilities}, \text{TemporalExceptions}, \text{ServiceComponents}', resourcePosition)\}$ .

# Entwicklung domänenspez. Sprachen (2)

1. Domänenverständnis
2. Konzeptextraktion und Repräsentation
3. Formalisierung
- 4. Technikraumspezifische Konkretisierung**

- Implementierung unter Nutzung des Eclipse Modeling Framework








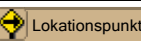



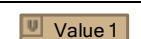






**Entwicklung domänenspez. Sprachen (2)**

1. Domänenverständnis
2. Konzeptextraktion und Repräsentation
3. Formalisierung
- 4. Technikraumspezifische Konkretisierung**

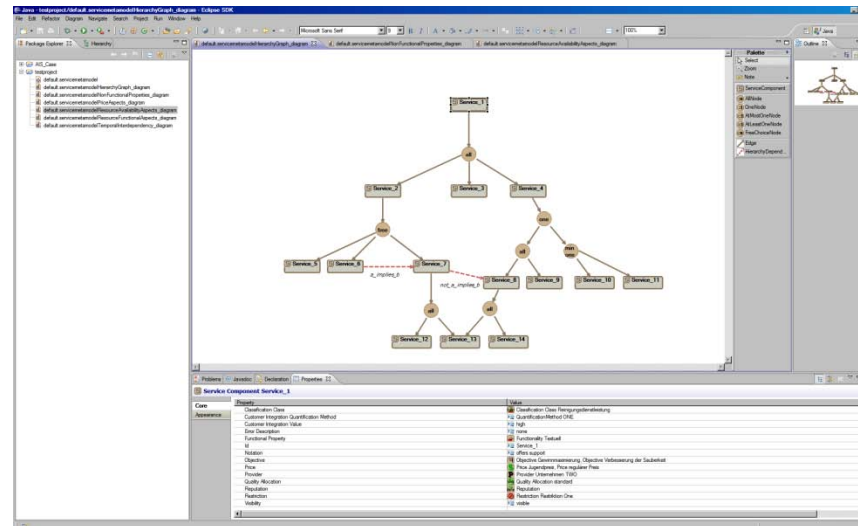
- Implementierung unter Nutzung des Eclipse Modeling Framework
- Spezifikation einer konkreten Syntax

<i>concrete syntax</i>	
.gmftool	.gmfgraph
	
	
	
	
	
	
	
	

## Entwicklung domänenspez. Sprachen (2)

1. Domänenverständnis
2. Konzeptextraktion und Repräsentation
3. Formalisierung
- 4. Technikraumspezifische Konkretisierung**

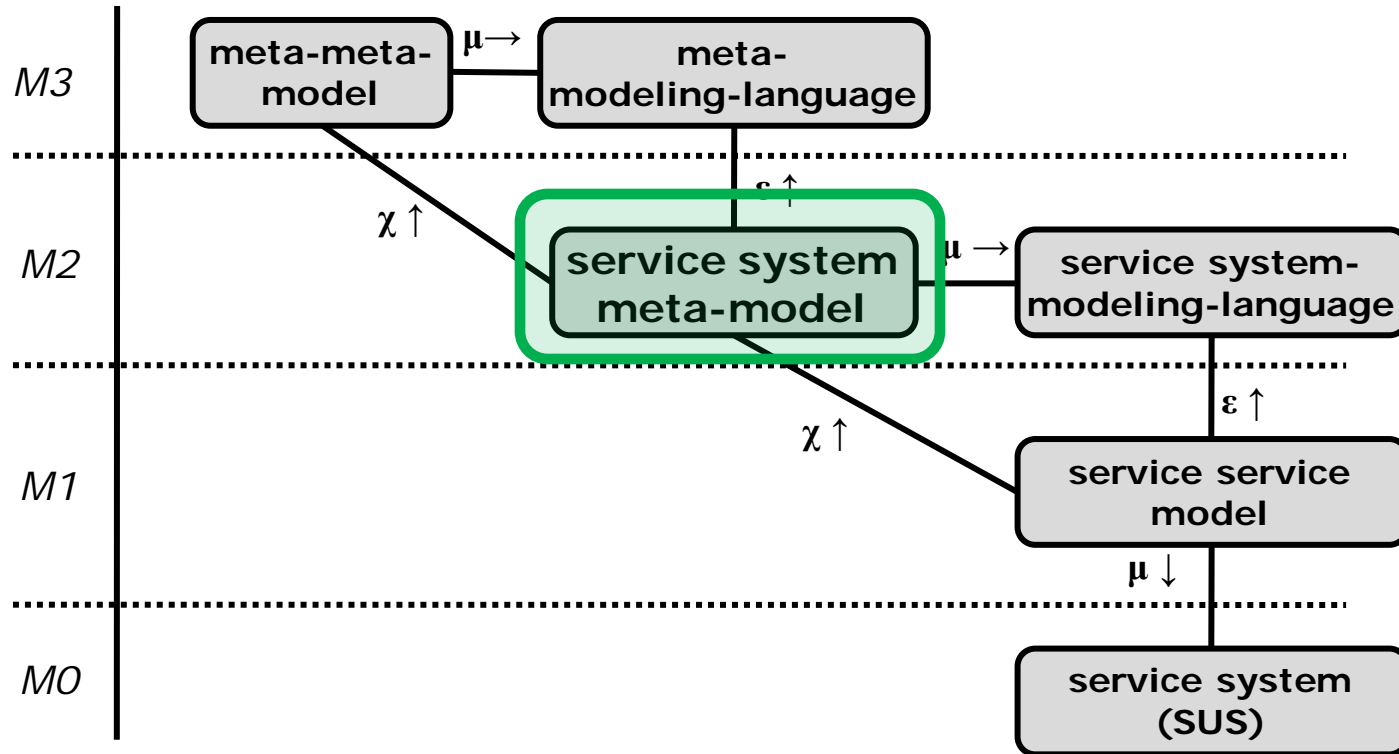
- Implementierung unter Nutzung des Eclipse Modeling Framework
  - Spezifikation einer konkreten Syntax
- Implementierung eines unterstützenden Werkzeugs



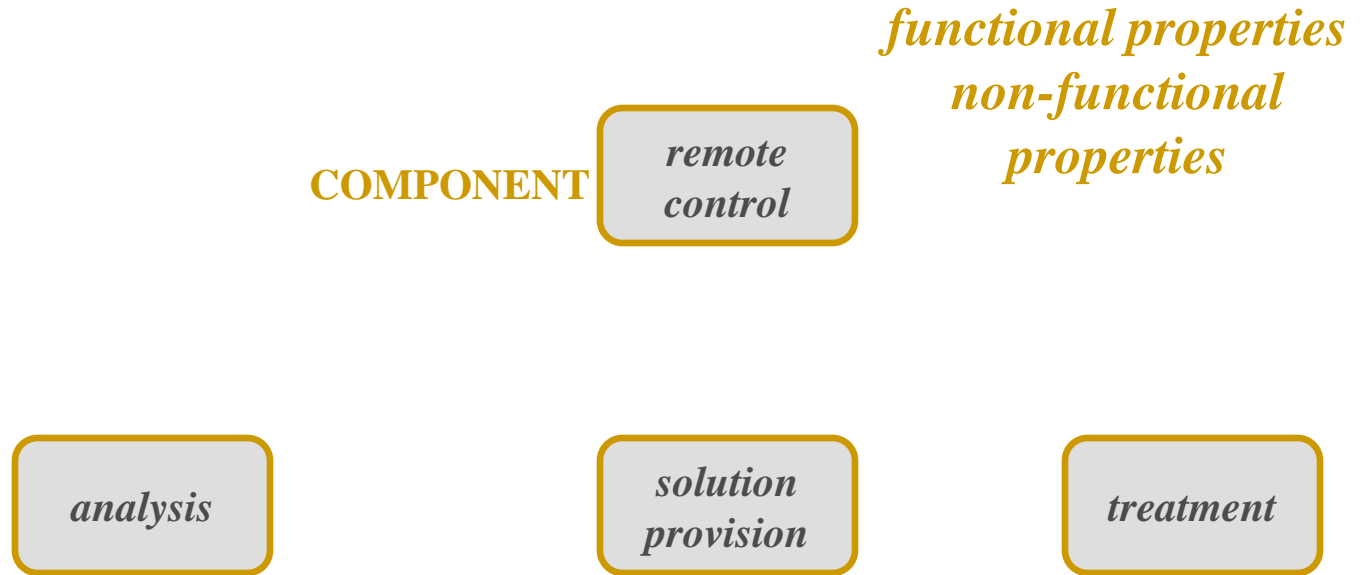
- 1) Modellierungstheorie (Metamodell-Ebenen)
- 2) Definition des Service Systems
- 3) Vorgehen zur Entwicklung domänenspezifischer Sprachen

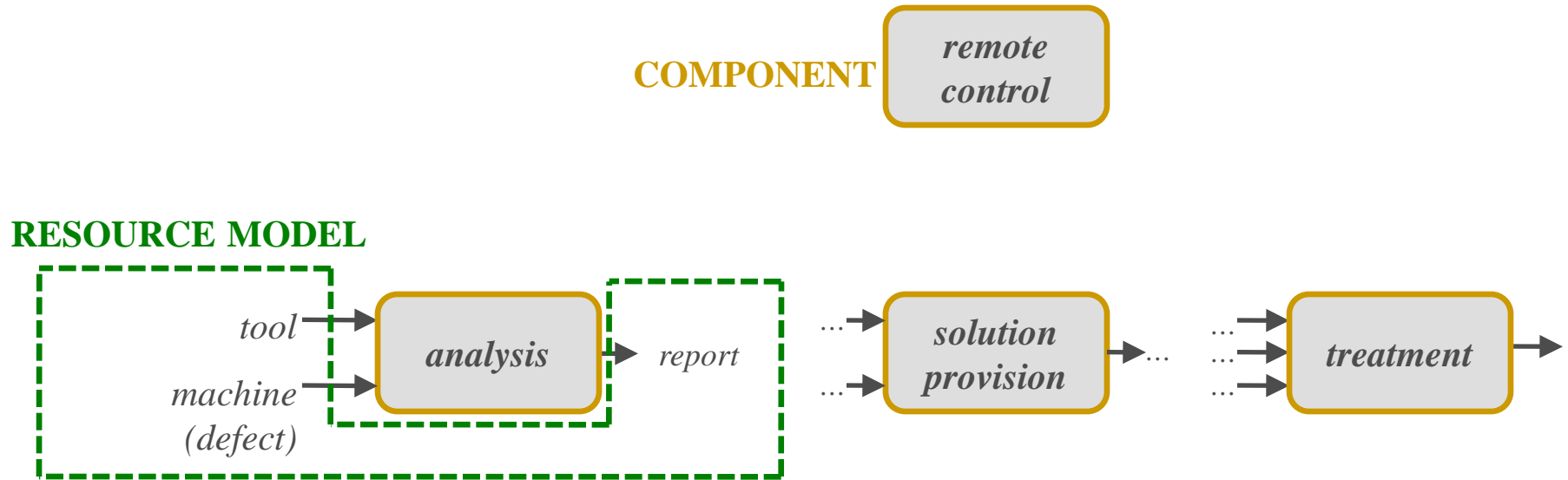
## Frage IV

Welche Konzepte werden für die Modellierung von  
Dienstleistungen benötigt?



$\mu$  : RepresentationOf -  $\varepsilon$  : ElementOf -  $\chi$  : ConformsTo





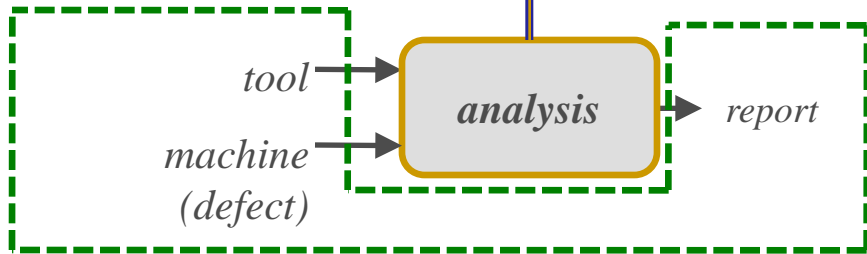
*attributes*  
*quantities*  
*change of states (7)*

...

*hierarchal dependencies*  
*configuration graph*  
*three levels of modelling*

...

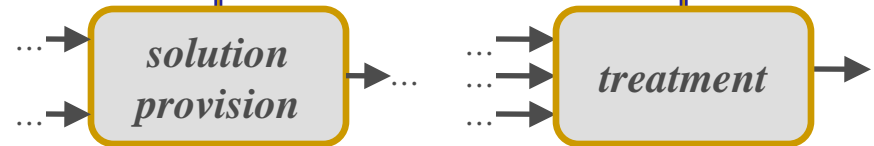
**RESOURCE MODEL**



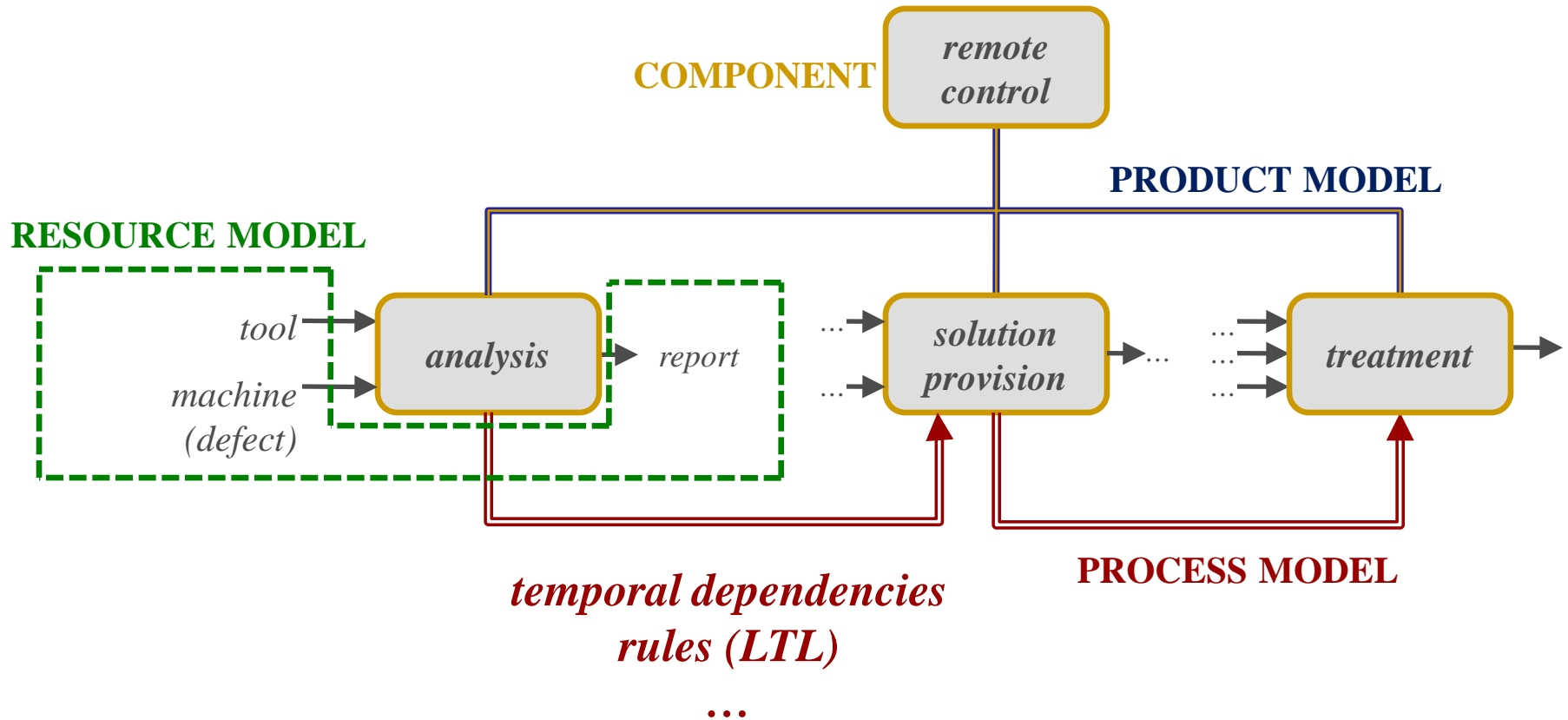
**COMPONENT**



**PRODUCT MODEL**

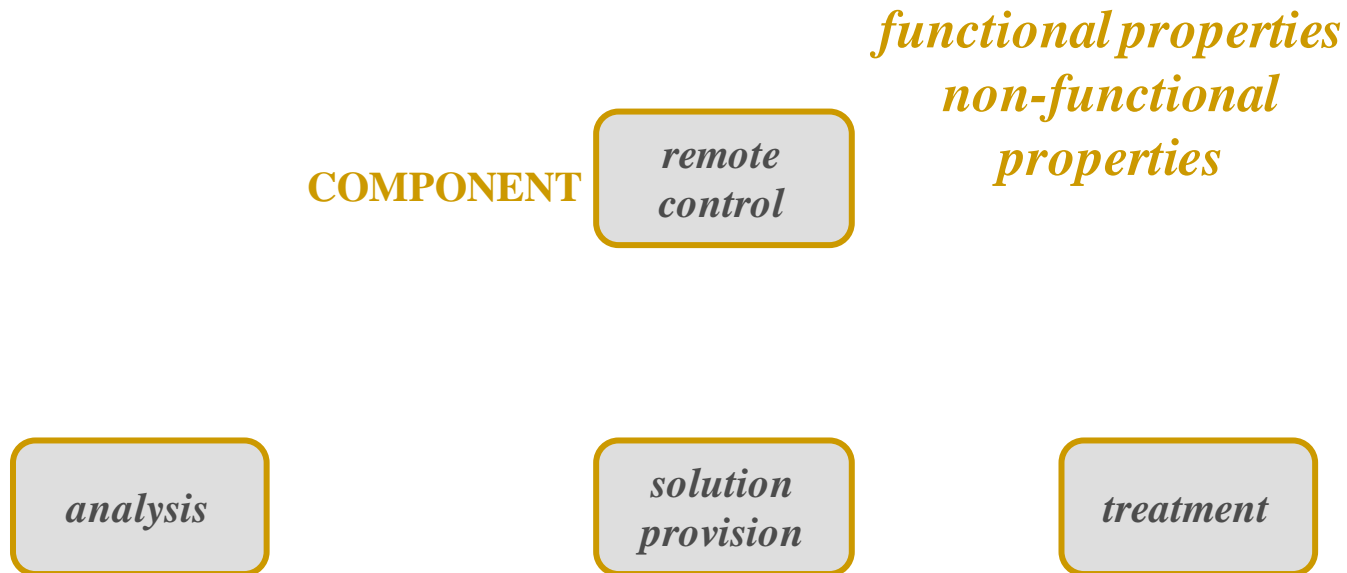






## Frage V

Was ist das Komponentenmodell und wie wird eine Dienstleistungskomponente beschrieben?



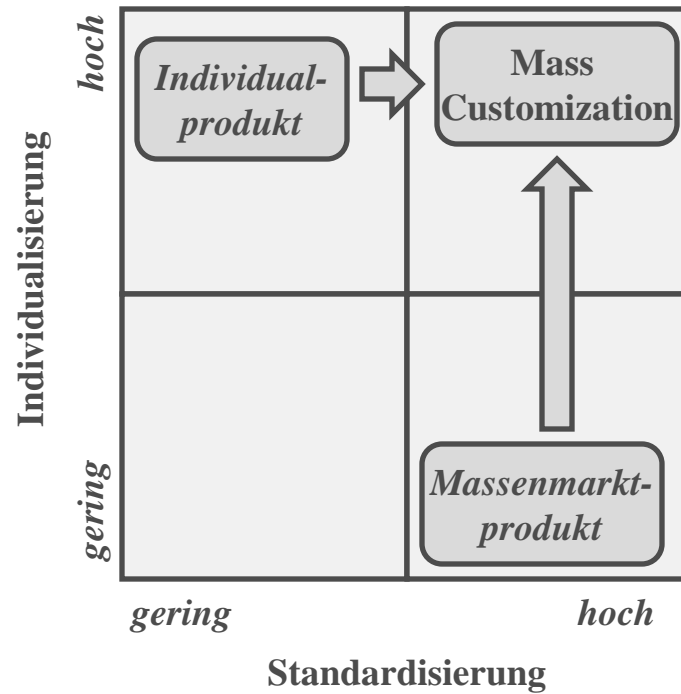
# Ziel der Komponentenorientierung

COMPONENT *remote control*

*analysis*

*solution provision*

*treatment*



## Definition Dienstleistungskomponente

COMPONENT

remote  
control

analysis

solution  
provision

treatment

*Eine **Dienstleistungskomponente** ist eine*

- *definierbare Funktionalität,*
- *welche eine abgrenzbare Menge von Systemelementen von einem Zustand in einen anderen Zustand überführt.*

*Zu den Systemelementen gehören*

- *Anbieter,*
- *Nachfrager und*
- *deren Ressourcen, wobei Anbieter und Nachfrager im engeren Sinne dieser Arbeit ebenfalls Ressourcen sind.*

*Dienstleistungskomponenten können bezüglich ihrer*

- ***Eigenschaften** beschrieben werden und*
- *in **hierarchische, nicht-hierarchische** sowie*
- ***zeitliche Abhängigkeiten** zueinander gesetzt werden.*

## Komponente - Ziele

*Bei einem **Ziel** handelt es sich um Sollgrößen, die darlegen, wie der Ist-Zustand zu verändern ist.*

Es gibt:

- interne Ziele
- externe Ziele

Zur Darlegung der Ziele werden benötigt:

- Skalen
- Attribute
- Werte

oder

- eine textuelle Darlegung



**Komponente - Klassifikation**COMPONENT  remote control analysis solution provision treatment

Zur Klassifikation von Dienstleistungen existieren verschiedene Standards:

<i>Bezeichner</i>	<i>Kurzbeschreibung</i>
CPV	Das gemeinsame Vokabular für öffentliche Ausschreibungen (CPV) ist für die Vergabe öffentlicher Aufträge innerhalb der Europäischen Union zwingend vorgeschrieben.
eCl@ss	Dieses Klassifikationsverfahren wurde von einem deutschen Industriekonsortium entwickelt und findet hauptsächlich in nationalen Projekten Verwendung.
ETIM	Diese Initiative verfolgt die Standardisierung des elektronischen Austausches von Produktdaten im Fachbereich Elektrotechnik. Hierbei erlaubt ETIM die Zuordnung zu einer Produktklasse.
proficlass	Diese Klassifikation dient insbesondere der Zuordnung von Materialklassen und Warengruppen innerhalb der Bau- und Gebäudetechnik.
UN/SPSC	Der „United Nations Standard for Products and Services Code“ ist ein branchenübergreifendes Klassifikationsverfahren, welches international eingesetzt wird.

## Komponente - Eigenschaften

Zur Beschreibung von Dienstleistungen werden

- funktionale und
- nicht-funktionale Eigenschaften

benötigt.

COMPONENT

remote control

analysis

solution provision

treatment

*Die **funktionalen Eigenschaften** einer Dienstleistungskomponente spezifizieren die von der Komponente vorgenommenen Zustandsänderungen.*

***Nicht-funktionale Eigenschaften** sind jene Attribute einer Dienstleistungskomponente, welche als Restriktionen bzw. Grenzen auf die Funktionalität von Dienstleistungen einwirken, ohne selbst die Funktionalität darzulegen.*

## Komponente - nicht-funkt. Eigenschaften (1)

*Das Konzept des **Anbieters** repräsentiert jene Geschäftseinheit, welche eine Dienstleistungskomponente offeriert.*



Zur Beschreibung des Anbieters sind notwendig:

- *Identifizierung* (bspw. Umsatzsteuernummer)
- *Bezeichner*
- *Kontaktinformationen*
  - Kontaktperson
  - postalische Adresse
  - Email
  - Telefonnummer
  - Webseite



## Komponente - nicht-funkt. Eigenschaften (2)

*Mit der zeitlichen Verfügbarkeit wird für eine Dienstleistungskomponente definiert, wann (respektive zu welchem Zeitpunkt) eine Ressource in das Dienstleistungssystem zu integrieren ist und wann eine Ressource als Ergebnis einer Dienstleistungserbringung zur Verfügung steht. Hierfür können singuläre und wiederholende Zeitpunkte und Zeiträume definiert werden.*



Zur Beschreibung der zeitlichen Verfügbarkeit sind notwendig (ISO 8601):

- *Datum*
  - *Kalendarische Daten*
  - *Ordinaldaten*
  - *Wochendaten*
- *Zeit*
  - *Stunden, Minuten, Sekunden, Zeitzone*

## Komponente - nicht-funkt. Eigenschaften (3)

*Mit der zeitlichen Verfügbarkeit ...*

- *wiederkehrende Zeitpunkte*
  - Zeittypen mit „verringerte Genauigkeit“  
*weekDate(0,0,5)*
- *Zeiträume*
  - Setzen von Start- und Endpunkten
- *Ausnahmen*

COMPONENT

remote control

analysis

solution provision

treatment

## Komponente - nicht-funkt. Eigenschaften (4)

*Die **Lokationsverfügbarkeit** definiert für eine Dienstleistungskomponente an welchem Punkt bzw. an welcher geordneten Ansammlung von Punkten (Routen) die Ressourcen als Potenzialfaktor zu integrieren sind bzw. als Ergebnis entnommen werden können.*



- (geografische) Punkte,
- Regionen,
- Adressen,
- Telefonnummern,
- URIs,
- IP-Adressen,
- Ethernetadressen,
- Spektren von elektromagnetischen Wellen

# Komponente - nicht-funkt. Eigenschaften (5)

## *Die Lokationsverfügbarkeit ...*

COMPONENT

remote  
control

analysis

solution  
provision

treatment

Zur Darlegung bedarf es:

- *Lokationspunktattribut*
- *Lokationspunktwert (Min)*
- *Lokationspunktwert (Max)*
  
- *Ausnahmen*

## Komponente - nicht-funkt. Eigenschaften (6)

*Mit der zeitlichen Dauer wird für eine Dienstleistungskomponente der Zeitraum von der Integration einer Kundenressource bis zu deren gewünschter Zustandsänderung angegeben.*



Zur Darlegung bedarf es:

- *Minimum*
- *Maximum*
- *Abhängigkeit von der zeitlichen und lokativen Verfügbarkeit*

## Komponente - nicht-funkt. Eigenschaften (7)

*Im Zusammenhang mit der Dienstleistungsmodellierung wird der **Preis** definiert als der Gegenwert, welcher für die Dienstleistung seitens des Leistungsempfängers zu zahlen ist.*



Der Preis besteht aus

- *Preiskomponenten (bspw. Courtage, Grundpreis, etc.)*

Der Preis lässt sich darstellen als:

- *Absolutpreis (bspw. 100 Euro)*
- *Anteiliger Preis (bspw. 4% des Kaufwertes)*
- *Preisfunktion*
- *Dynamischer Preis (Auktionen)*

Es bedarf weiterhin

- *Währung*
- *Steuern*
- *Gültigkeit (Studenten, etc.)*

## Komponente - nicht-funkt. Eigenschaften (8)

Die **Bezahlung** definiert, wie und unter welchen Restriktionen ein Kunde den angegebenen Preis dem Dienstleistungsanbieter zu überlassen hat.

**Konsequenzen** definieren alle Auswirkungen durch die Nichteinhaltung von Vereinbarungen, die zwischen Kunde und Anbieter getroffen wurden.

Unter den **Evaluationskriterien** werden Aspekte verstanden, welche die Bewertung der Qualität eines Dienstleistungsanbieters seitens des Kunden beeinflussen.



COMPONENT

remote  
control

analysis

solution  
provision

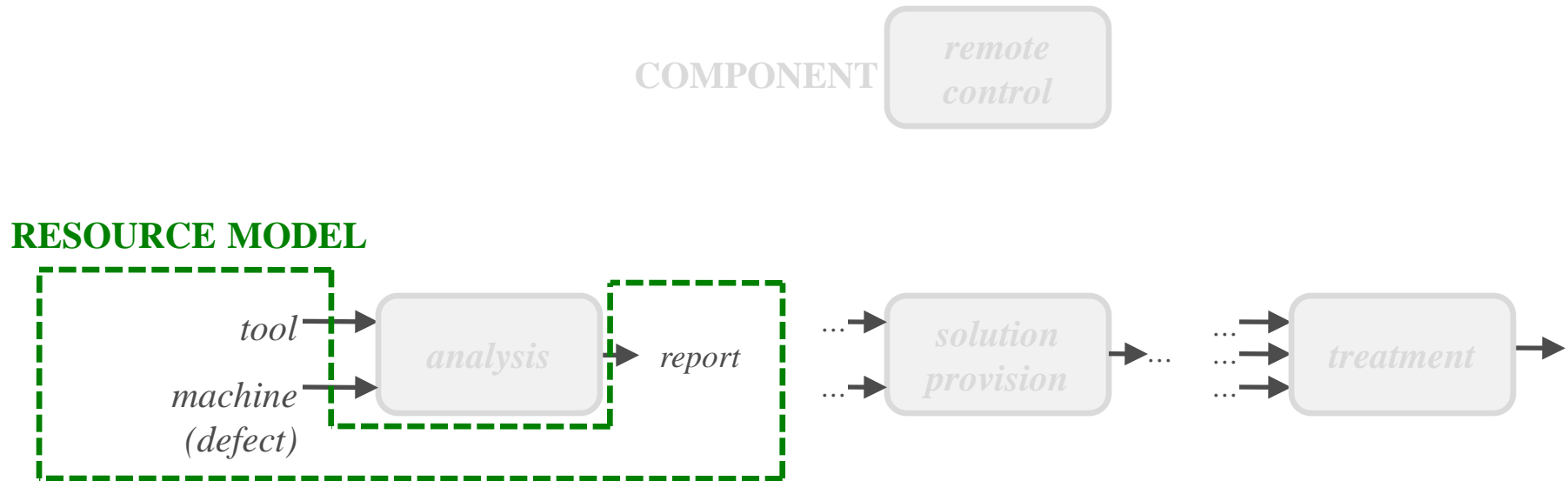
treatment

- 1) Definition von Dienstleistungskomponenten
  - 2) Zielspezifikation und Klassifikation
- 3) funktionale und nicht-funktionale Eigenschaften



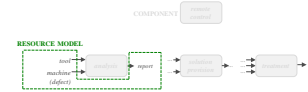
## Frage VI

Was ist das Ressourcenmodell und was umfasst es?



*attributes*  
*quantities*  
*change of states (7)*

...



*Ressourcen sind eine Menge von*

- *Objekten und*
- *Subjekten,*

*die Teil des Dienstleistungssystems sind. Sie*

- *interagieren miteinander und*
- *erzeugen die von einer Dienstleistung angebotene Funktionalität.*

*Der Zustand aller Ressourcen vor der Erbringung einer Dienstleistung wird durch den Begriff des*

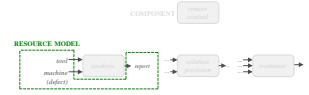
- *Potenzialfaktors und*

*der Zustand der durch die Dienstleistungserbringung veränderten Ressourcen durch den Begriff des*

- *Ergebnisses*

*näher spezifiziert.*

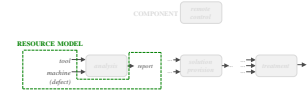
# Ressourcen - Position



$ResourcePositions = \{ "input", "output" \}.$

## Ressourcen – Attribute (1)

Mit der **Herkunft** einer Ressource wird definiert, wessen Eigentum diese Ressource ist und wessen Disponierbarkeit sie somit unterliegt.

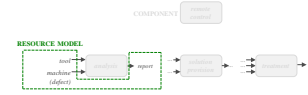


Zu den **Herkunftstypen** gehören:

- *intern*: innerhalb der Disponierbarkeit des Anbieters
- *extern*: von anderen Unternehmen; entziehen sich der Disponierbarkeit des Anbieters
- *Kunden*: besondere Ausprägung der externen Ressourcen; unterliegen der Disponierbarkeit des Kunden

## Ressourcen – Attribute (2)

Die **Tangibilität** definiert für eine Ressource deren „Berührbarkeit“ und somit deren Stofflichkeit im weiteren Sinne.

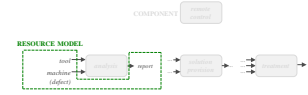


Zu den **Tangibilitätstypen** gehören:

- *tangible*: materielle „greifbare“ Ressourcen
- *human*: besondere Ausprägung materieller Ressourcen
- *intangible*: „nicht greifbare“ Ressourcen; bspw. Rechte, Patente, Organisationseinheiten, etc.

## Ressourcen – Attribute (3)

Die **Mobilität** definiert die technische Fähigkeit, aufwandsgerechte Umsetzbarkeit und individuelle Bereitschaft einer Ressource, sich einer lokalen Veränderung zu unterziehen, um als Potenzialfaktor für eine Dienstleistung genutzt werden zu können.

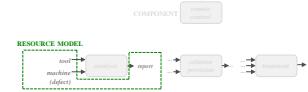


Zu den **Mobilitätstypen** gehören:

- *mobil*: können an den Ort der Dienstleistung gebracht werden
- *immobil*: bestimmen den Ort der Dienstleistungserbringung

## Ressourcen – Attribute (4)

*Die **Aktivität** einer Ressource definiert für diese, ob sie im Zusammenhang mit der Dienstleistungserbringung einer Zustandsveränderung unterzogen oder aktiv für die Durchführung einer Zustandsveränderung einer anderen Ressource genutzt wird.*

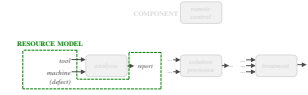


Zu den **Aktivitätstypen** gehören:

- *Operator*: wirkt auf Ressourcen ein und verändert diese
  - *Gebrauchs-Operator*: Weiterverwendbarkeit
  - *Verbrauchs-Operator*: stehen nicht mehr zur Verfügung
- *Operand*: unterliegen einer Veränderung

## Ressourcen – Attribute (5)

*Mittels der **Ressourcenrelationen** können Beziehungen zwischen einzelnen Ressourcen näher spezifiziert werden.*



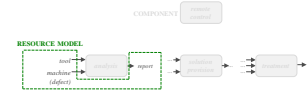
„**Ressourcenrelationen**“ stellt ein wesentliches Attribut zur Betrachtung von Systemen dar (bspw. „verleimt“, „benachbart“)

**Ressourcenhierarchien** („ist Teil von“) stellen eine besondere Form der Ressourcenrelationen dar:



## Ressourcen – Attribute (6)

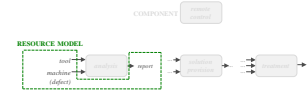
*Sonstige Ressourcenattribute beschreiben weitere Merkmale von Ressourcen und sind, wie auch die Attributwerte mit denen sie belegt werden, individuell zu spezifizieren.*



Terme der Ressourcenattribute und Ressourcenattributwerte sind domänenspezifisch und dienen der Darlegung der Zustandsänderung der Ressourcen und somit der Darlegung der Funktionalität einer Dienstleistung.

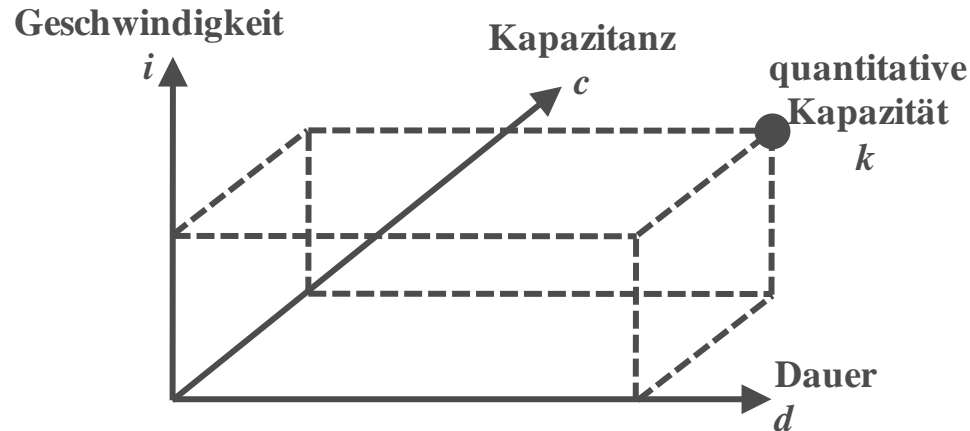
## Ressourcen – Attribute (7)

Weitere Attribute sind Kapazität ...



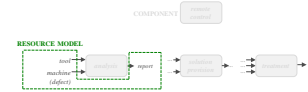
Die **qualitative Kapazität** kennzeichnet die Art und Güte des Leistungsvermögens einer Ressource und determiniert die Brauchbarkeit für bestimmte Verwendungszwecke und somit auch die potenziellen Möglichkeiten hinsichtlich der Erbringung alternativer Leistungen.

Die **quantitative Kapazität** definiert die Ausbringungsmenge je betrachteter Zeiteinheit und wird für Menschen und materielle Ressourcen definiert.



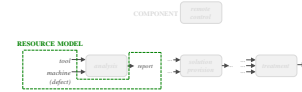
## Ressourcen – Attribute (8)

... und Flexibilität



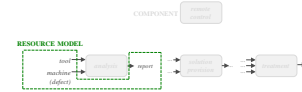
*Mit der **Flexibilität** wird die Anpassungs- und Umstellungsfähigkeit bezogen auf ein Anpassungserfordernis (bspw. Störungen etc.) definiert*

# Ressourcen – implizite Funktionalität (1)



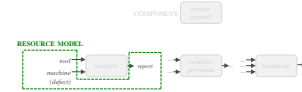
<i>Änderungsgegenstand</i>	<i>Visualisierung</i>	<i>Beispiel</i>	<i>Ressourcenverwertungstyp</i>
Änderung der <i>Attributausprägung</i>	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p><b>Potenzial</b></p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; width: 100px;"> <p>Ressource A</p> <p>Attribut_1=Wert_1</p> </div> </div> <div style="font-size: 2em;">→</div> <div style="text-align: center;"> <p><b>Ergebnis</b></p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; width: 100px;"> <p>Ressource A</p> <p>Attribut_1=Wert_2</p> </div> </div> </div>	Für die Wäsche eines Gegenstandes wird die Attributausprägung bzgl. des Attributs „sauber“ von „nein“ auf „ja“ geändert.	„durchgängige Ressourcenverwertung“

# Ressourcen – implizite Funktionalität (2)

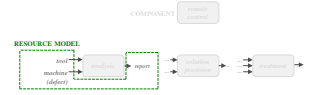


<i>Änderungsgegenstand</i>	<i>Visualisierung</i>	<i>Beispiel</i>	<i>Ressourcenverwertungstyp</i>
Änderung der <b>Beziehung</b> zu anderen Ressourcen	<p style="text-align: center;"><b>Potenzial</b></p> <p style="text-align: center;">→</p> <p style="text-align: center;"><b>Ergebnis</b></p>	Für den Zusammenbau zweier Einzelteile wird eine Beziehung „verschweißt“ definiert.	„austauschende oder umgruppierende Ressourcenverwertung“
	<p style="text-align: center;"><b>Potenzial</b></p> <p style="text-align: center;">→</p> <p style="text-align: center;"><b>Ergebnis</b></p>	Für die Trennung zweier Teile (bspw. durch absägen) wird die Beziehung „verbunden“ aufgelöst.	

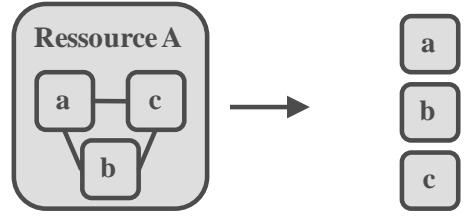
Ressourcen – implizite Funktionalität (3)



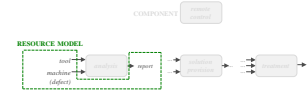
<i>Änderungsgegenstand</i>	<i>Visualisierung</i>	<i>Beispiel</i>	<i>Ressourcenverwertungstyp</i>
<p>Änderung bzgl. der <i>Subsysteme</i> einer Ressource (als Sonderform der Beziehungsveränderung)</p>	<p>The diagram illustrates the transformation of three separate components, labeled 'a', 'b', and 'c', into a single resource named 'Ressource A'. On the left, under the heading 'Potenzial', the three components are shown as individual boxes. An arrow points to the right, where 'Ressource A' is shown as a larger rounded rectangle containing the three components. Inside 'Ressource A', component 'a' is connected to 'c' by a horizontal line, and component 'b' is connected to both 'a' and 'c' by lines that meet at a central point below 'b'.</p>	<p>Definition einer Ressource, welche nur als Ergebnis existiert</p> <p>Bspw. Zusammenbau eines Stuhls aus einzelnen Leisten.</p>	<p>„synthetische Ressourcenverwertung“</p>



# Ressourcen – implizite Funktionalität (4)

<i>Änderungsgegenstand</i>	<i>Visualisierung</i>	<i>Beispiel</i>	<i>Ressourcenverwertungstyp</i>
<p>Änderung bzgl. der <i>Subsysteme</i> einer Ressource (als Sonderform der Beziehungsveränderung)</p>	<p><b>Potenzial</b></p>  <p><b>Ergebnis</b></p>	<p>Definition einer Ressource, welche nur als Potenzialfaktor existiert</p> <p>Bspw. Zerteilung eines Gegenstandes in seine Einzelteile.</p>	<p>„analytische Ressourcenverwertung“</p>

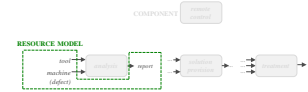
Ressourcen – implizite Funktionalität (5)



<i>Änderungsgegenstand</i>	<i>Visualisierung</i>	<i>Beispiel</i>	<i>Ressourcenverwertungstyp</i>
<p>Änderung bzgl. der <i>Subsysteme</i> einer Ressource (als Sonderform der Beziehungsveränderung)</p>		<p>Definition von Ressourcen als Potenzialfaktoren und Ergebnisse. Bspw.: Reifenwechsel am Auto.</p>	<p>„austauschende oder umgruppierende Ressourcenverwertung“</p>

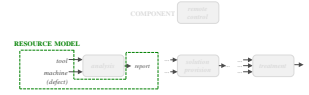


# Ressourcen – implizite Funktionalität (6)

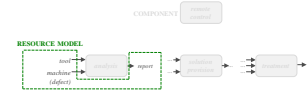


Änderungsgegenstand	Visualisierung	Beispiel	Ressourcenverwertungstyp
<p>Änderung bzgl. der <b>Subsysteme</b> einer Ressource (als Sonderform der Beziehungsveränderung)</p>		<p>Definition von Ressourcen als Potenzialfaktoren und Ergebnisse.</p> <p>Bspw. Veränderung eines Stuhls (vierbeinig) in einen Hocker (dreibeinig).</p>	<p>„austauschende oder umgruppierende Ressourcenverwertung“</p>

Ressourcen – Einwirkungsart



		<i>Einwirkungsart</i>						
		<i>Einwirkung auf die Objeksubstanz</i>				<i>Einwirkung auf die Objektzuordnung</i>		
		<i>physisch</i>	<i>kombiniert</i>	<i>nicht-physisch</i>		<i>zeitlich</i>	<i>räumlich</i>	<i>sonstig sachlich</i>
				<i>intellektuell</i>	<i>emotional</i>			
<i>Operanden-Ressource</i>	<i>human</i>	Körperpflege, Gesundheitswesen	Sicherung und Rettung	Weiterbildung	Unterhaltung, „Theme-Parks“	Beherbergung	Personentransport	Personenvermittlung
	<i>materiell</i>	Baugewerbe, Reparatur, Energieerzeugung	Werkschutz			Lager	Gütertransport	Einzelhandel, Großhandel
	<i>immateriell</i>	Informationsveränderung, Forschung	Archive, Datenschutzorganisation			Datenbanken, Bibliotheken	Post- und Fernmeldeamt	Informationsvermittlung, Beratung, Börse



## 1) Definition des Ressourcenbegriffs

### 2) Attributtypen

- Herkunft -
- Tangibilität -
- Mobilität -
- Aktivität -
- Kapazität und Flexibilität –
- Ressourcenrelationen -

## 3) Darlegung impliziter Funktionalität

Frage VII

Was ermöglicht das Produktmodell?

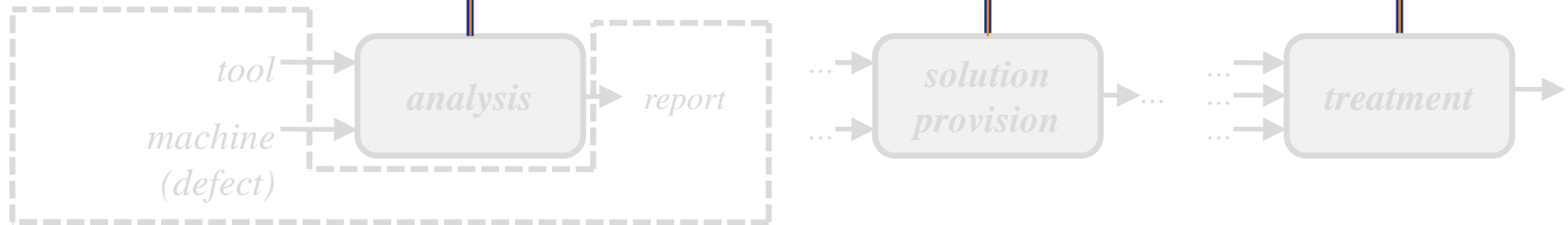
*hierarchal dependencies*  
*configuration graph*  
*three levels of modelling*

...

RESOURCE MODEL

COMPONENT

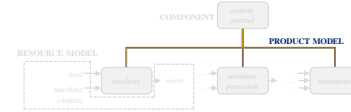
PRODUCT MODEL



## Produktmodell - Motivation

Grundlegend dient das Produktmodell:

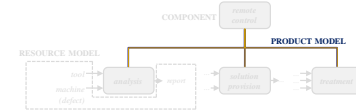
- Übersicht über die Komponenten
- Darlegung der Abhängigkeit der Komponenten
- Vorbereitung der Konfiguration von Dienstleistungen



## Produktmodell - Modellierungsstufen

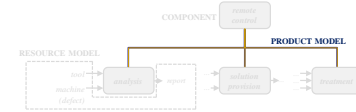
Es existieren drei Modellierungsstufen mit unterschiedlicher Mächtigkeit:

- Stufe 1: Basisgraph
- Stufe 2: Kardinalitäten
- Stufe 3: nicht-hierarchische Abhängigkeiten



## Produktmodell – Netzgraph (1)

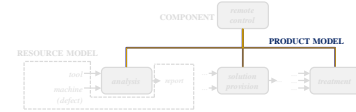
Zunächst wird ein einfacher Netzgraph mit Knoten und gerichteten Kanten definiert.



Der Graph wird um zusätzliche Knotentypen erweitert, um die Semantik für spätere Konfigurationen definieren zu können.

# Produktmodell – Netzgraph (2)

Es existieren verschiedene Konnektor-Knoten (*Connectors*):

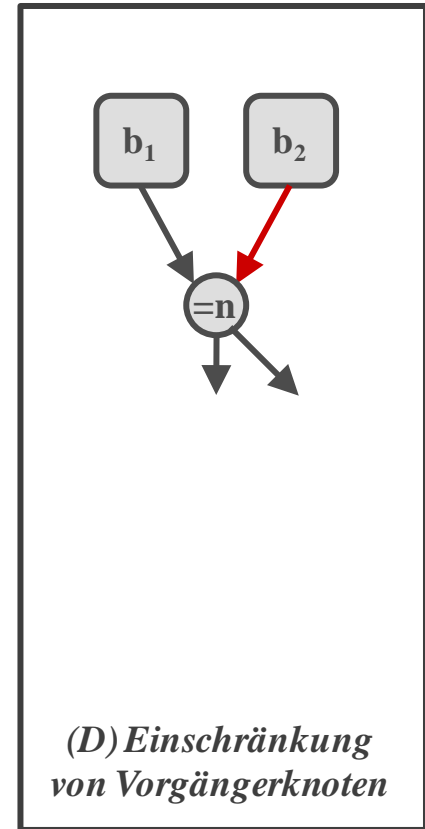
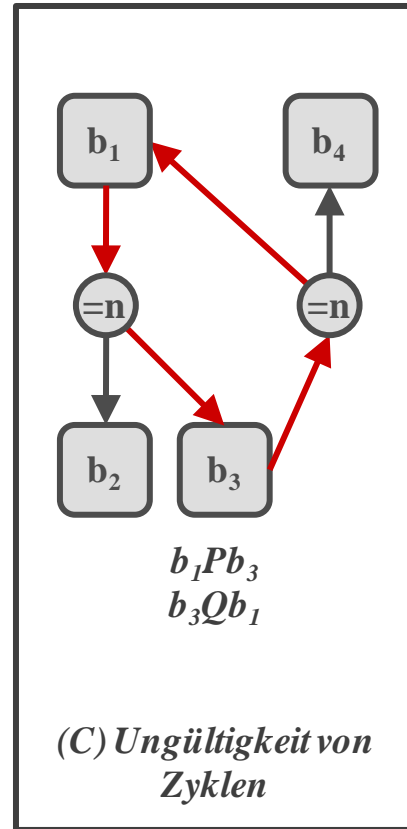
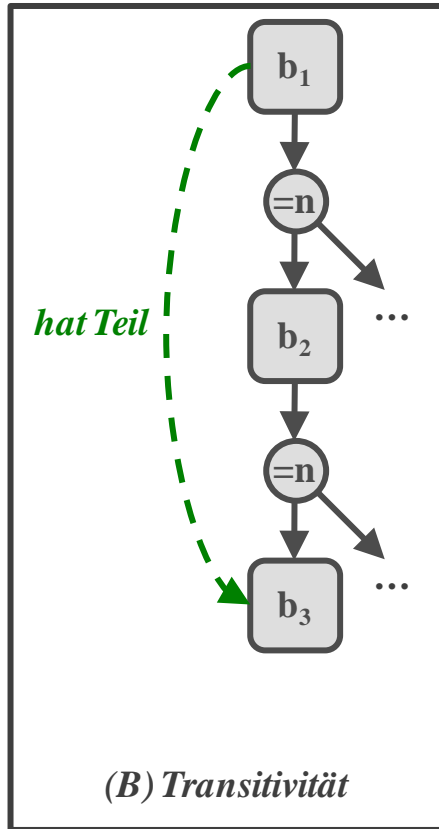
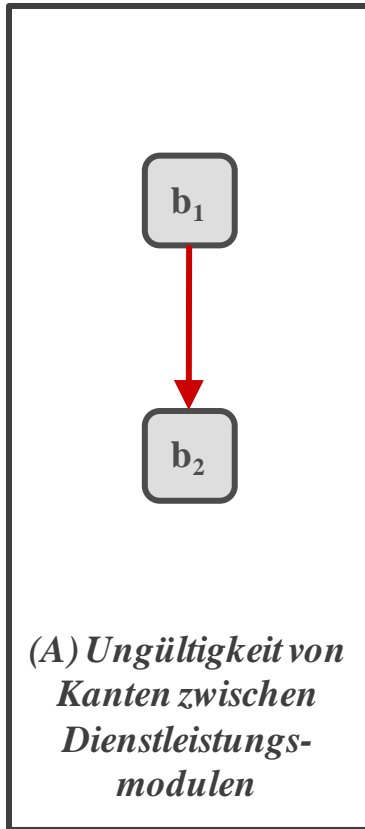
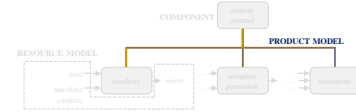


Knotentyp	Symbol	Beispiel von Kombinationsmöglichkeiten
<b>Konjunktivknoten</b> (all-node)	$\textcircled{=n}$	
<b>Exklusives-oder-Knoten</b> (one-node)	$\textcircled{=1}$	
<b>Optional-exklusives-oder-Knoten</b> (at-most-one-node)	$\textcircled{\leq 1}$	
<b>Disjunktiv-obligatorischer-Knoten</b> (at-least-one-node)	$\textcircled{\geq 1}$	
<b>Disjunktivknoten</b> (free-choice-node)	$\textcircled{\leq n}$	
<p><input type="checkbox"/> Dienstleistungsmodul    <input type="radio"/> Verbindungsknoten    <math>\vdots</math> Verbindung im Graphen         Verbindung durch Konfiguration</p>		



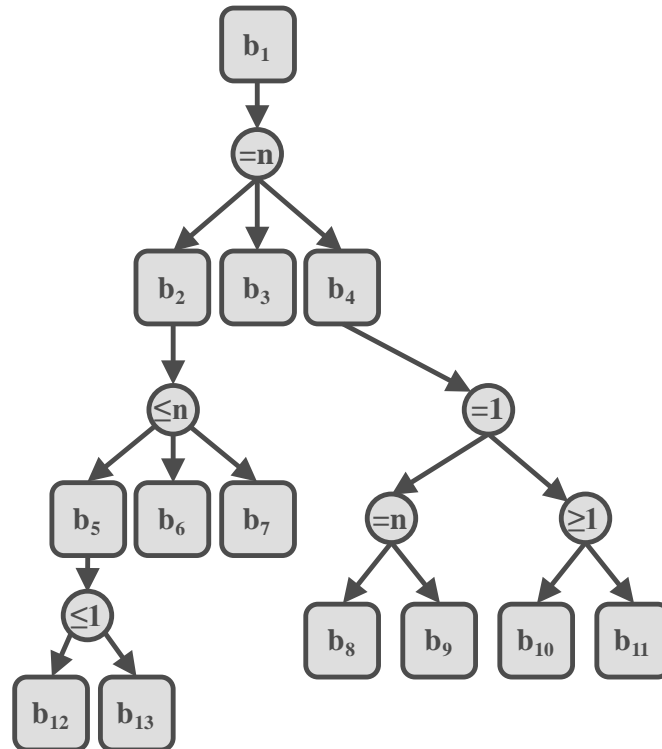
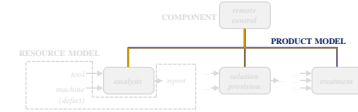
# Produktmodell – Netzgraph (3)

Restriktionen innerhalb eines Netzgraphen:



## Produktmodell – Netzgraph (4)

Beispiel eines Netzgraphen:

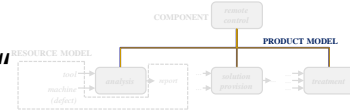


- $b_i$**  Dienstleistungskomponente
- $=n$**  Konjunktivknoten
- $=1$**  Exklusives-Oder-Knoten
- $\leq 1$**  Optional-exklusives-oder-Knoten
- $\geq 1$**  Disjunktiv-obligatorischer Knoten
- $\leq n$**  Disjunktivknoten

Eine Aussage „2 bis 3 der nachfolgenden Komponenten sind zu wählen“ ist mit dem Graphen nicht möglich → Modellierungsstufe 2.

## Produktmodell – Kardinalitäten (1)

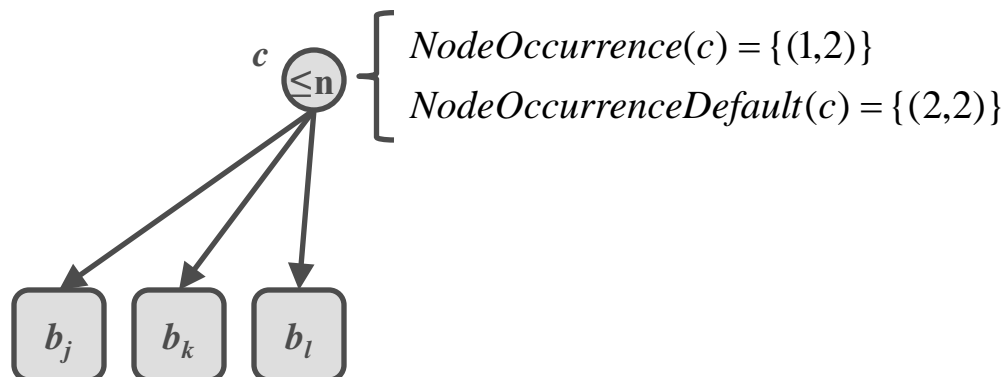
Die wichtigste Kardinalität ist die „besteht-aus-Kardinalität“ (*NodeOccurrence*)



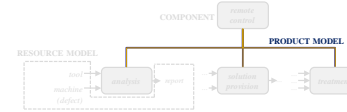
Aussagen wie: „Komponente A setzt sich aus 4 bis 6 der nachfolgenden Komponenten zusammen“ sind durch diese Kardinalität möglich.

Für die Kardinalität können angegeben werden:

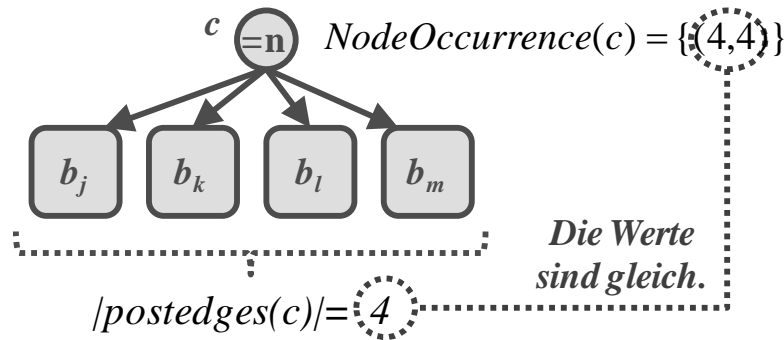
- Minimalwert,
- Maximalwert und
- Defaultwert.



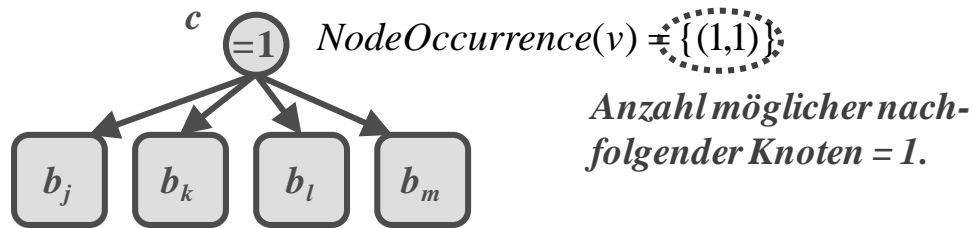
# Produktmodell – Kardinalitäten (2)



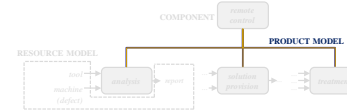
Abhängigkeit zwischen Knotentypen und Kardinalitäten:  
*Konjunktivknoten*



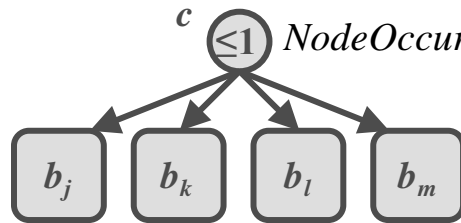
*Exklusives-oder-Knoten*



# Produktmodell – Kardinalitäten (3)



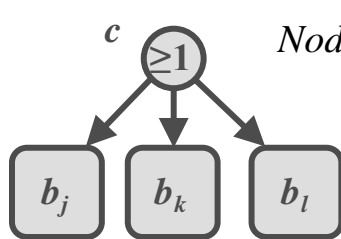
Abhängigkeit zwischen Knotentypen und Kardinalitäten:  
*Optional-exklusives-oder-Knoten*



$NodeOccurrence(c) = \{(0,1)\}$

*Anzahl erlaubter nachfolgender Knoten ist null oder eins.*

*Disjunktiv-obligatorischer-Knoten*

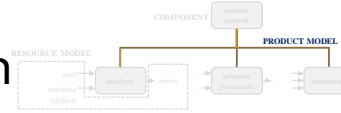


$NodeOccurrence(c) = \{1,2\}$

*Der Minimalwert ist größer als null.*

## Produktmodell – nicht-hier. Abhängigkeiten

Es gibt Abhängigkeiten, die sich nicht durch den definierten Graphen darstellen lassen.



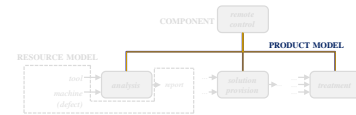
Es lassen sich

- Constraints (ungerichtete Restriktionen) und
- Regeln (gerichtete Restriktionen)

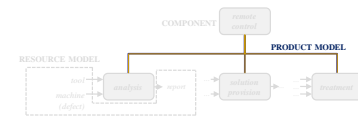
spezifizieren.

# Produktmodell – nicht-hier. Abhängigkeiten

Folgende „typische“ Regeln können definiert werden:



<i>Tupelnotation</i>	<i>Erläuterung</i>
$b\_implies\_b'$	Wenn die Komponente $b$ <b>gewählt</b> wird, muss die Komponente $b'$ <b>ebenfalls gewählt</b> werden.
$not\_b\_implies\_b'$	Wenn die Komponente $b$ <b>nicht gewählt</b> wird, muss die Komponente $b'$ <b>gewählt</b> werden.
$b\_excludes\_b'$	Wenn die Komponente $b$ <b>gewählt</b> wird, darf die Komponente $b'$ <b>nicht gewählt</b> werden (gilt vice versa).
$not\_b\_implies\_not\_b'$	Wenn die Komponente $b$ <b>nicht gewählt</b> wird, darf die Komponente $b'$ <b>ebenfalls nicht gewählt</b> werden.

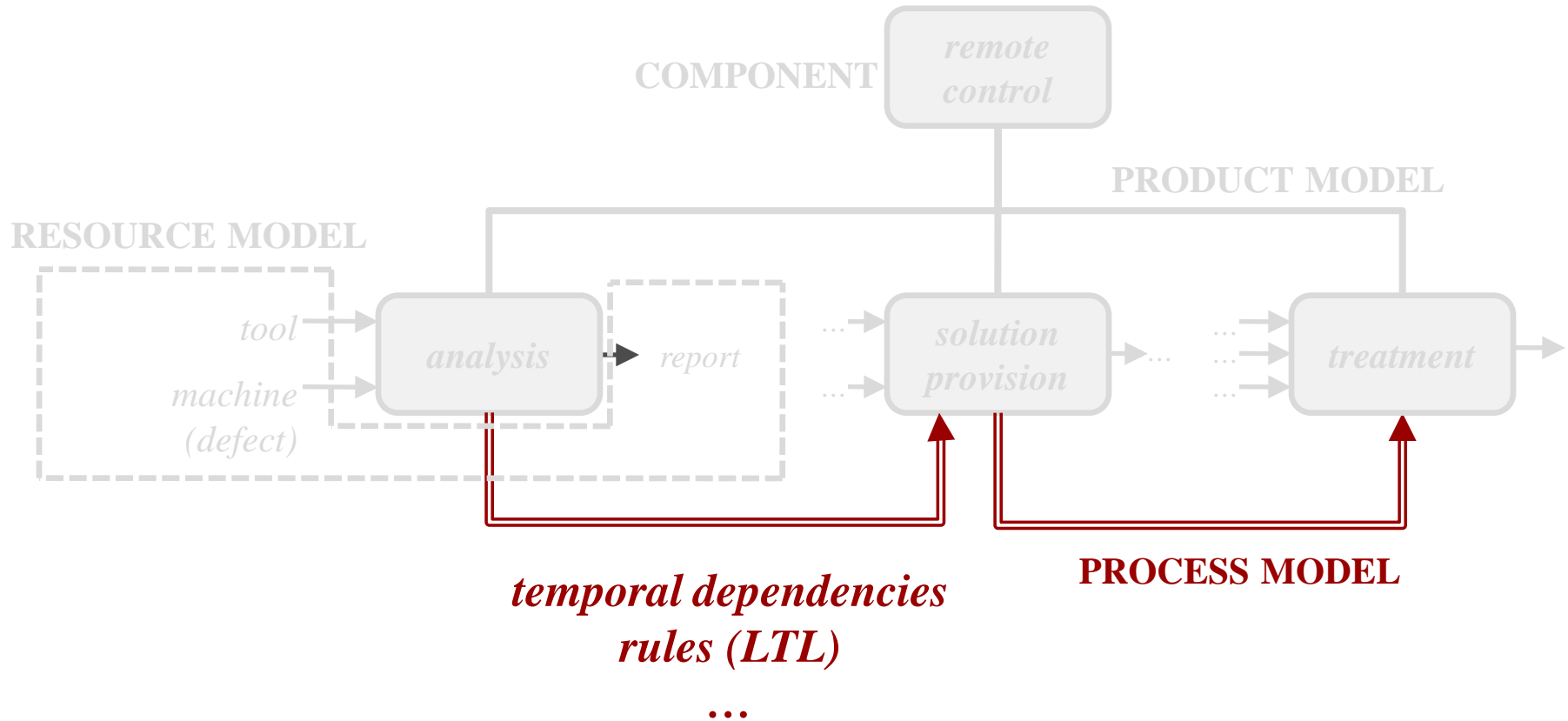


- 1) Spezifikation des Netzgraphen mit Verbindungsknoten
- 2) Spezifikation von Kardinalitäten
- 3) Spezifikation von Constraints und Regeln



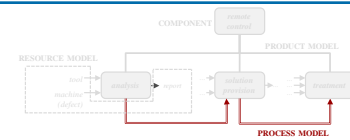
Frage VIII

Wofür wird das Prozessmodell benötigt?

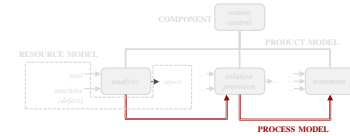


## Prozessmodell - Modelltyp

Die Flexibilität des Konfigurationsgraphen verlangt einen **deklarativen Beschreibungsansatz**. Dieser definiert Regeln und Grenzen, in denen letztlich der Ablauf stattfinden muss.

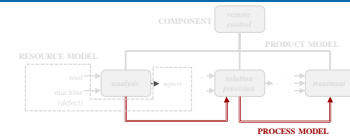


## Prozessmodell - Notwendigkeitsregeln



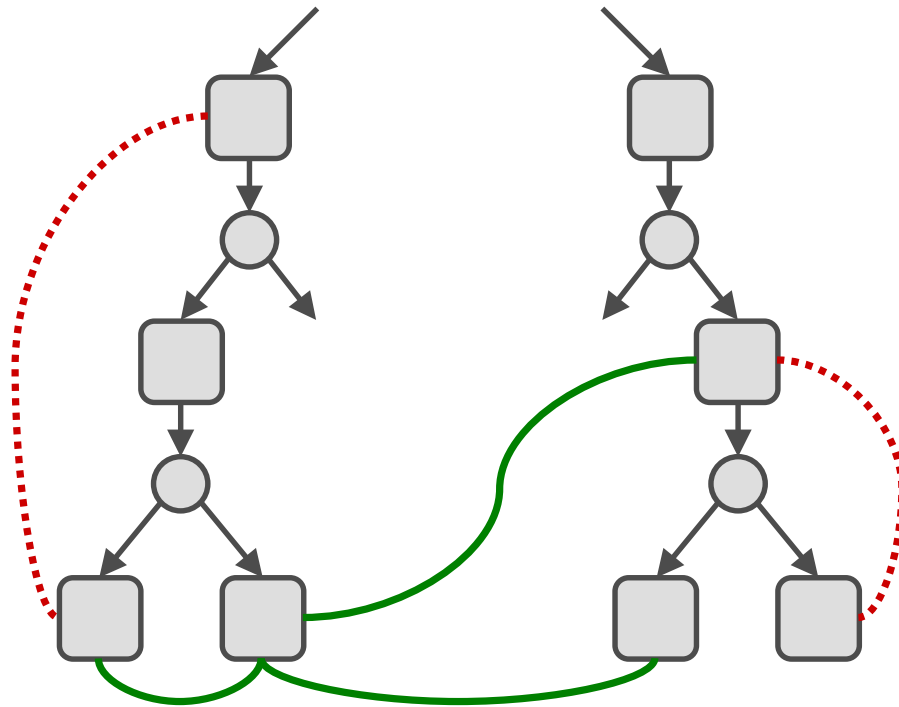
Regel	Notation in LTL	Erläuterung
$(b, b') \in b\_needs\_succeeding\_b'$	$\Box(b \Rightarrow \Diamond(b'))$	Wenn die Komponente $b$ ausgeführt wird, muss <b>irgendwann später</b> die Komponente $b'$ ebenfalls ausgeführt werden.
$(b, b') \in b'\_needs\_preceding\_b$	$\neg(\neg b U b')$	Die Komponente $b'$ kann nur ausgeführt werden, wenn <b>irgendwann zuvor</b> Komponente $b$ ausgeführt wurde.
$(b, b') \in b\_needs\_immediate\_succeeding\_b'$	$\Box(b \Rightarrow (\circ(b')) \vee (\circ(b)))$	Wenn die Komponente $b$ ausgeführt wird, muss sofort die Komponente $b'$ ausgeführt werden. Es kann zuvor jedoch Komponente $b$ nochmals ausgeführt werden.
$(b, b') \in b'\_needs\_immediate\_preceding\_b$	$\neg\Diamond(\neg b \wedge \circ b')$	Die Komponente $b'$ kann nur ausgeführt werden, wenn unmittelbar zuvor Komponente $b$ ausgeführt wurde.

## Prozessmodell - Ausschlussregeln

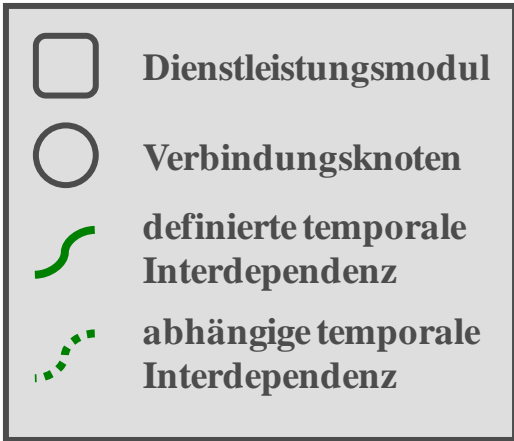
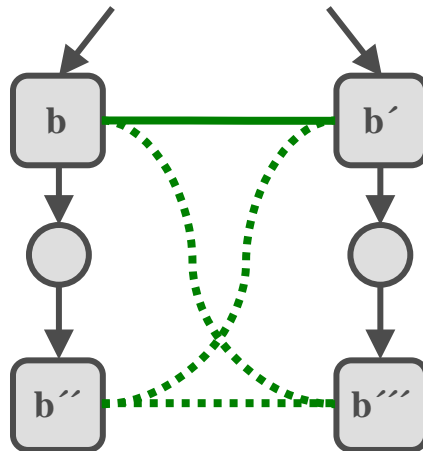


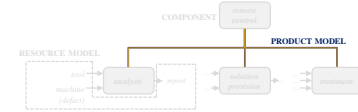
<i>Regel</i>	<i>Notation in LTL</i>	<i>Erläuterung</i>
$(b, b') \in b\_never\_before\_b'$	$\neg \diamond (b \wedge \diamond b')$	Der Ausführung von Komponente $b$ darf die Ausführung von Komponente $b'$ zu <b>keinem Zeitpunkt</b> folgen.
$(b, b') \in b\_never\_immediately\_before\_b'$	$\neg \diamond (b \wedge \circ b')$	Die Komponente $b'$ darf <b>nicht zum nächsten Zeitpunkt</b> nach der Ausführung von Komponente $b$ ausgeführt werden. Zu späteren Zeitpunkten darf eine Ausführung erfolgen.

# Prozessmodell – Restriktionen



# Prozessmodell – Implikationen





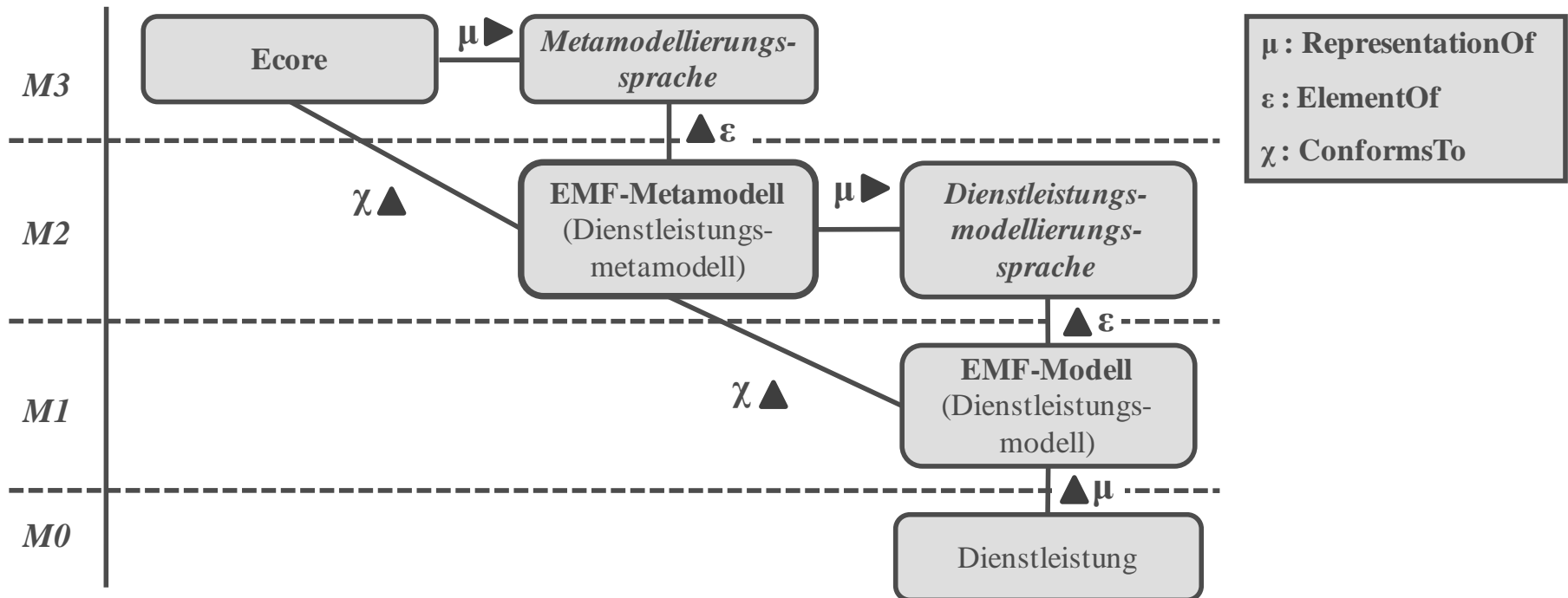
- 1) Deklarativer Prozessbeschreibungsansatz
- 2) Notwendigkeitsregeln
- 3) Ausschlussregeln
- 4) Restriktionen und Implikationen

Wie kann eine technikraumspezifische Konkretisierung  
erfolgen?

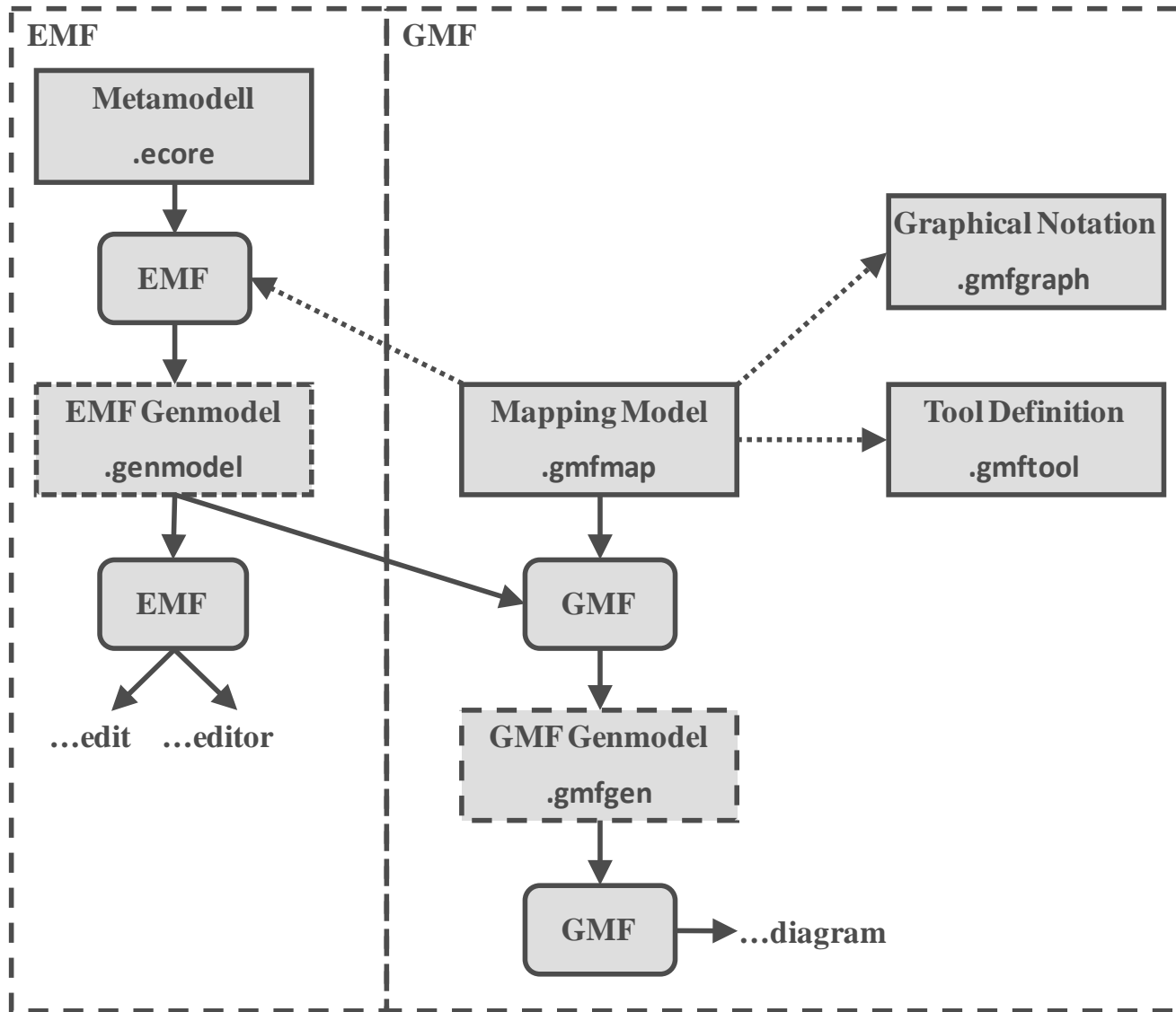


## Eclipse Modeling Framework (1)

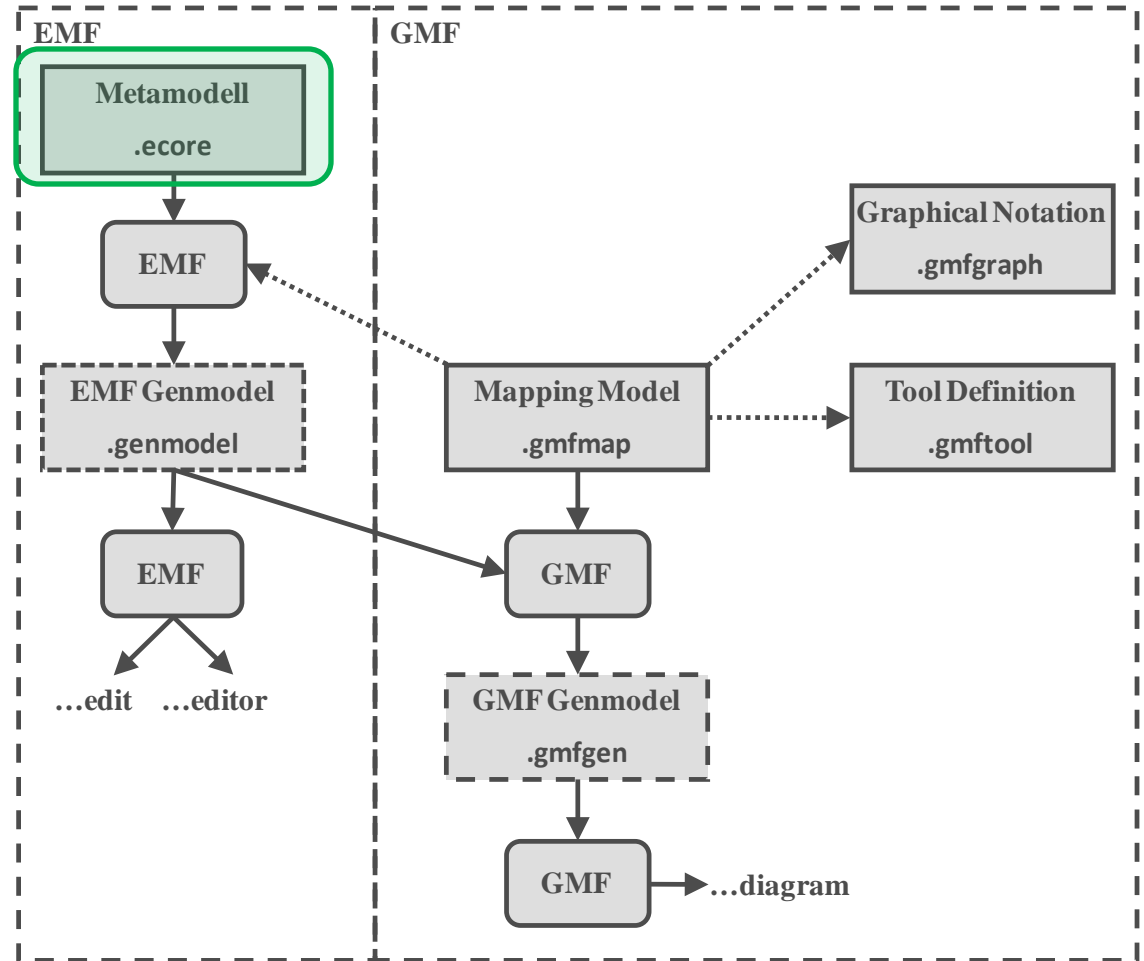
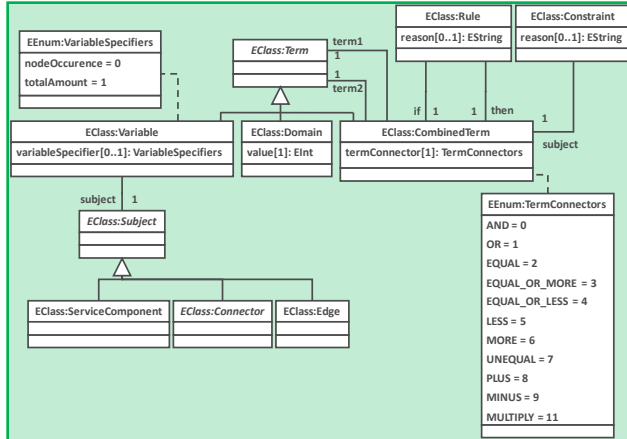
Metaebenen unter Nutzung von „ecore“



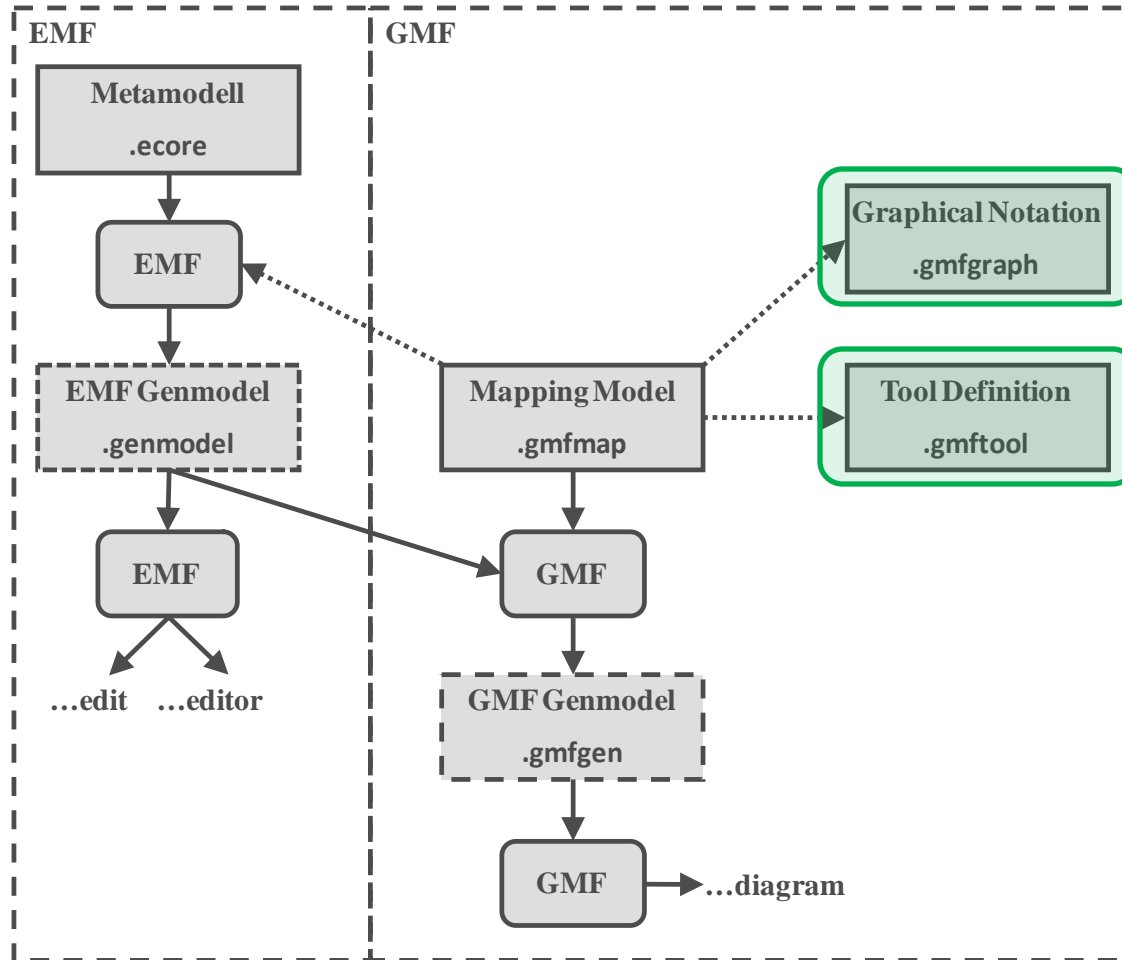
## Eclipse Modeling Framework (2)



# Eclipse Modeling Framework (3)



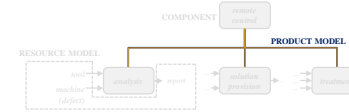
# Eclipse Modeling Framework (4)



Element	Konkrete Syntax	
	Palette .gmftool	Diagramm .gmfgraph
Dienstleistungskomponente entsprechend der Menge <i>ServiceComponents</i> .		
Ressource entsprechend der Menge <i>Resources</i> .		
Allokation der Ressource zu einer Dienstleistungskomponente entsprechend der Ressourcenposition „input“. Dies beinhaltet ebenfalls die Quantität entsprechend der Funktion <i>ResourceQuantityAllocation</i> .		
Allokation der Ressource zu einer Dienstleistungskomponente entsprechend der Ressourcenposition „output“. Dies beinhaltet ebenfalls die Quantität entsprechend der Funktion <i>ResourceQuantityAllocation</i> .		

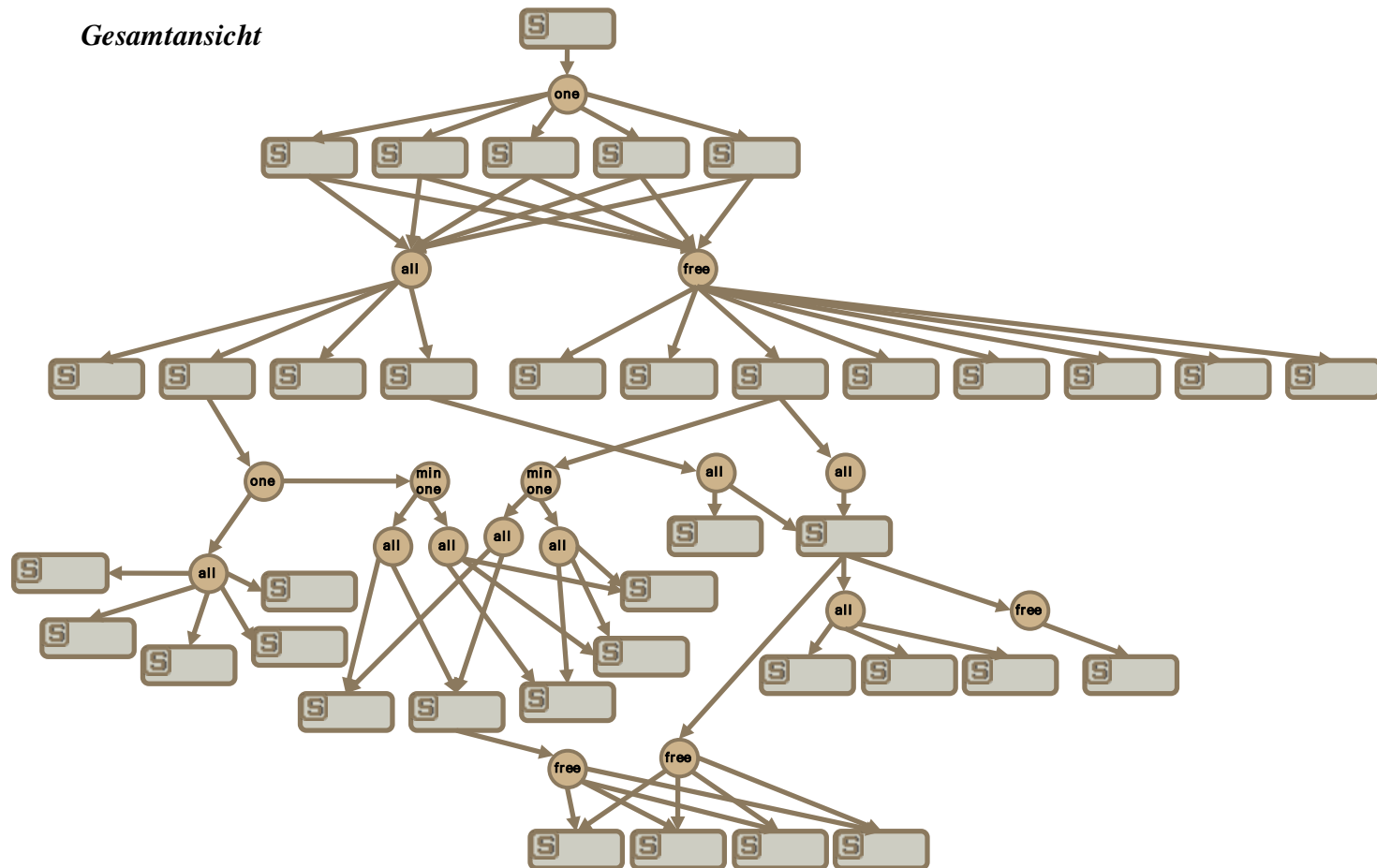
The screenshot displays the Eclipse IDE interface for modeling a service hierarchy. The main workspace (B) contains a hierarchy graph starting with 'Service\_1' at the root. It branches into 'all' and 'Service\_2', 'Service\_3', and 'Service\_4'. 'Service\_2' further branches into 'free', 'Service\_5', and 'Service\_6'. 'Service\_4' branches into 'one' and 'min one'. 'one' branches into 'all' and 'Service\_9'. 'min one' branches into 'Service\_10' and 'Service\_11'. 'all' (under 'one') branches into 'Service\_7' and 'Service\_8'. 'Service\_7' branches into 'all' and 'Service\_12'. 'all' (under 'Service\_7') branches into 'Service\_13' and 'Service\_14'. A red dashed arrow labeled 'a\_implies\_b' points from 'Service\_6' to 'Service\_7', and another labeled 'not\_a\_implies\_b' points from 'Service\_7' to 'Service\_8'. The Package Explorer (A) on the left shows the project structure. The Palette (E) on the right lists modeling elements like 'ServiceComponent', 'AllNode', 'OneNode', etc. The Outline (C) shows a small tree view. The Properties (F) window at the bottom shows the details for 'Service Component Service\_1'.

Property	Value
Core	
Classification Class	Classification Class Reinigungsdienstleistung
Appearance	
Customer Integration Quantification Method	QuantificationMethod ONE
Customer Integration Value	high
Error Description	none
Functional Property	Functionality Textuell
Id	Service_1
Notation	offers support
Objective	Objective Gewinnmaximierung, Objective Verbesserung der Sauberkeit
Price	Price Jugendpreis, Price regulärer Preis
Provider	Provider Unternehmen TWO
Quality Allocation	Quality Allocation standard
Reputation	Reputation
Restriction	Restriction Restriktion One
Visibility	visible

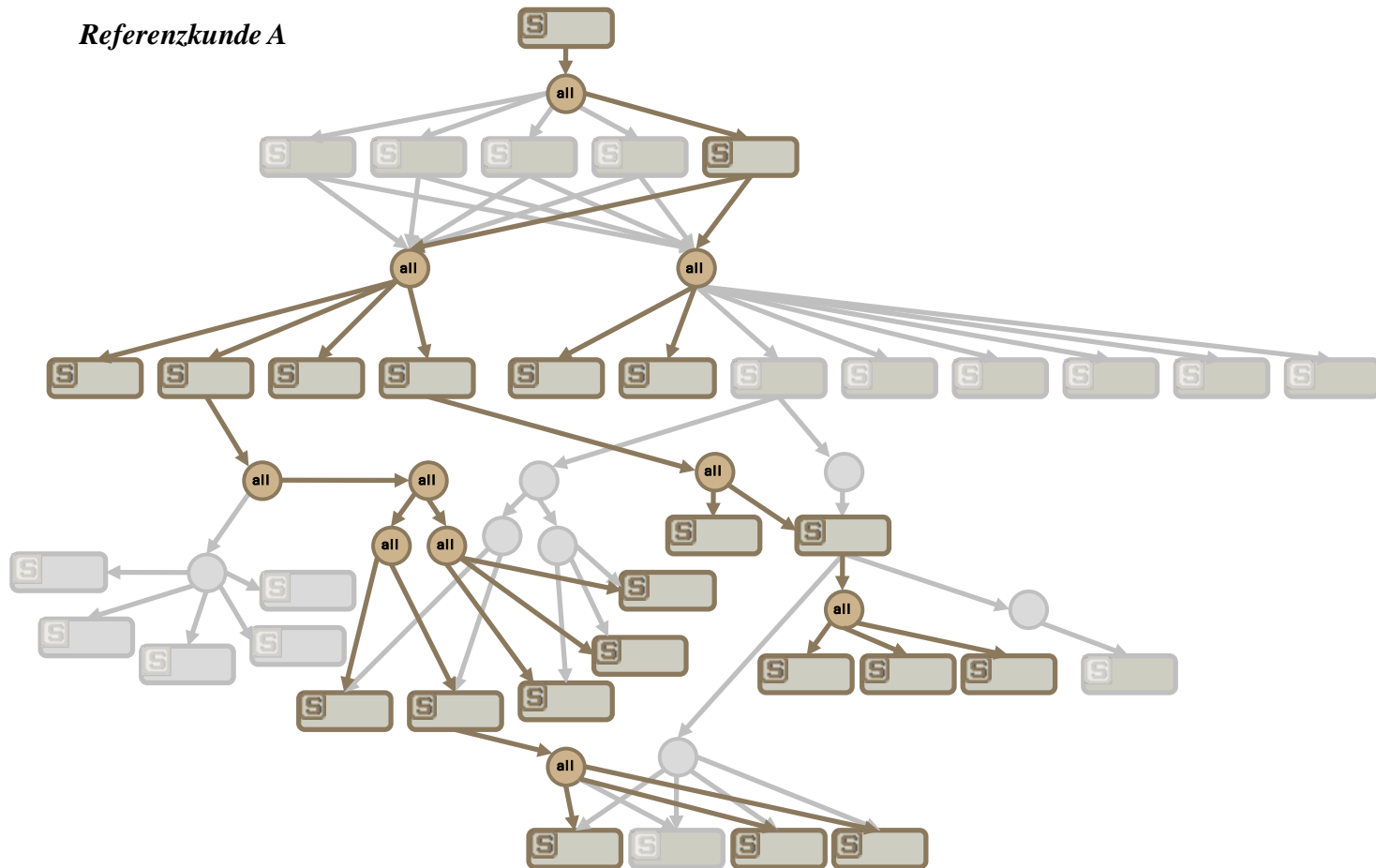


- 1) Eclipse Modeling Framework als möglicher Technikraum
- 2) Ecore
- 3) konkrete Syntax
- 4) Editor

Produktmodell (Ausschnitt)



Produktmodell (Ausschnitt) – mögliche Konfiguration





Anwendungsfall (3)

Prozessmodellanalyse

	In...	In...(2)	Dia...	Rep...	Fa...	Tr...	Er...	Re...(2)	Fe...	Fe...(2)	An...	Ma...	En...	Di...(2)	Fi...	In...(3)
In...		u.V.														
In...(2)	u.N.															
Di...				m.V.			u.N.	m.N.						u.V.		m.V.
Re...			m.N.				m.N.	m.N.						u.V.		u.V.
Fa...						u.N.			m.N.							
Tr...					u.V.				u.N.							
Er...			u.V.	m.V.				m.N.						u.V.		m.V.
Re...(2)			m.V.	m.V.			m.V.							m.V.		m.V.
Fe...					m.V.	u.V.										
Fe...(2)											u.N.	m.N.	m.N.		m.N.	
An...										u.V.		u.N.	u.N.		m.N.	
Ma...										m.V.	u.V.		u.N.		m.N.	
En...										m.V.	u.V.	u.V.			m.N.	
Di...(2)			u.N.	u.N.			u.N.	m.N.								u.V.
Fi...										m.V.	m.V.	m.V.	m.V.			
In...(3)			m.N.	u.N.			m.N.	m.N.						u.N.		

# Prozessmodell Editor – Ausschnitt

