



---

## Ein Metamodellansatz zum Aufbau dynamisch erweiterbarer Modellierungssysteme

### → Prozess- und Datenintegration in klinischen Anwendungen

**Stefan Jablonski**

**Sascha Müller**

- **Aufgabe und Lösungskonzept**
  - Beispielszenario: Medizin
  - Anwendungs- und Datenintegration
  - Aspektorientierte Prozessmodellierung
- **Anwendungen**
  - Prozessmodelle
  - Wissensmanagement
- **Umsetzung**
  - Metadatenverwaltung mit Repository

# Ein Projektkontext

Aufgabenstellung

- Sonderforschungsbereich 539:  
"Glaukome einschließlich Pseudoexfoliationssyndrom (PEX)"
  - Arbeitsgruppe InTech des Lehrstuhls für Datenbanksysteme
  - Augenklinik der Universität Erlangen-Nürnberg
- Glaukom: Degeneration des Augennervs

- Arbeitsschwerpunkte
  - Applikationsintegration
  - Datenintegration

Aufnahme der Papilla  
(Augenhintergrund)

aufgenommen mit einer  
speziellen Modalität



# Anforderungen an die Integration

---

- **Applikationsintegration**
  - Zusammenwirken der Applikationen (Modalitäten, Krankenhaus-Informationssystem (KIS), Befundarbeitsplatz, etc.)
  - Kompakte, vollständige, anwendungsspezifische Modellierung
  - Berücksichtigung anwendungsbezogener Besonderheiten
- **Datenintegration**
  - Viele sehr unterschiedliche Datenquellen (Modalitäten, Krankenhaus-Informationssystem (KIS), Befundarbeitsplatz, etc.)
  - Nicht nur Datensammlung sondern auch Datenaustausch
- **Allgemeine Anforderungen**
  - Qualität
  - Sicherheit

# Lösungsansatz

---

Aufgabenstellung

- Modellbasiert
  - Wiederverwendung, Generalisierung, Verifikation, Analyse
- Prozessorientiert
  - Vollständigkeit, Komplexität
- Datenbankbasiert
  - Konsistenz, Persistenz, Auswertbarkeit
- Berücksichtigung von Standards, Normen, Erfahrungen, spez. Sichtweisen
  - Klinische Pfade (evidenzbasiert) → "template"
  - Darstellung
    - 'Customized' (angepasst)
    - 'Compact' (kompakt)
    - 'Comprehensible' (verständlich)

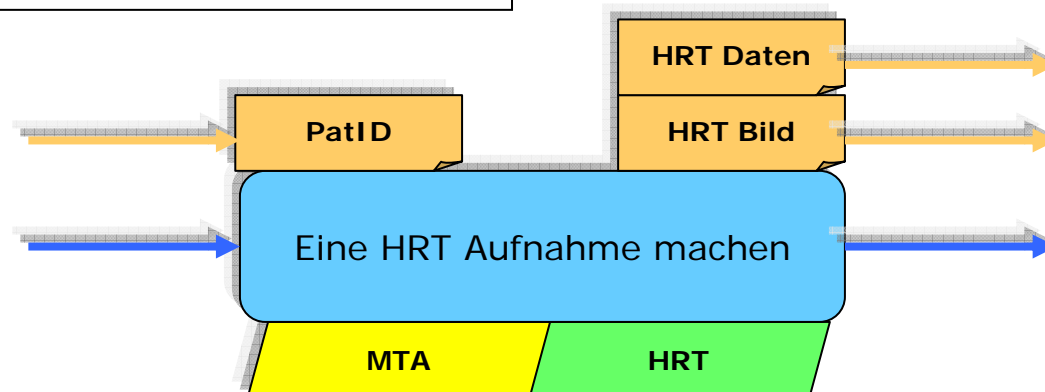
## Lösungsmethode

Domänen-spezifische Anpassung eines generischen Prozessmodells

## Framework: iPM4med

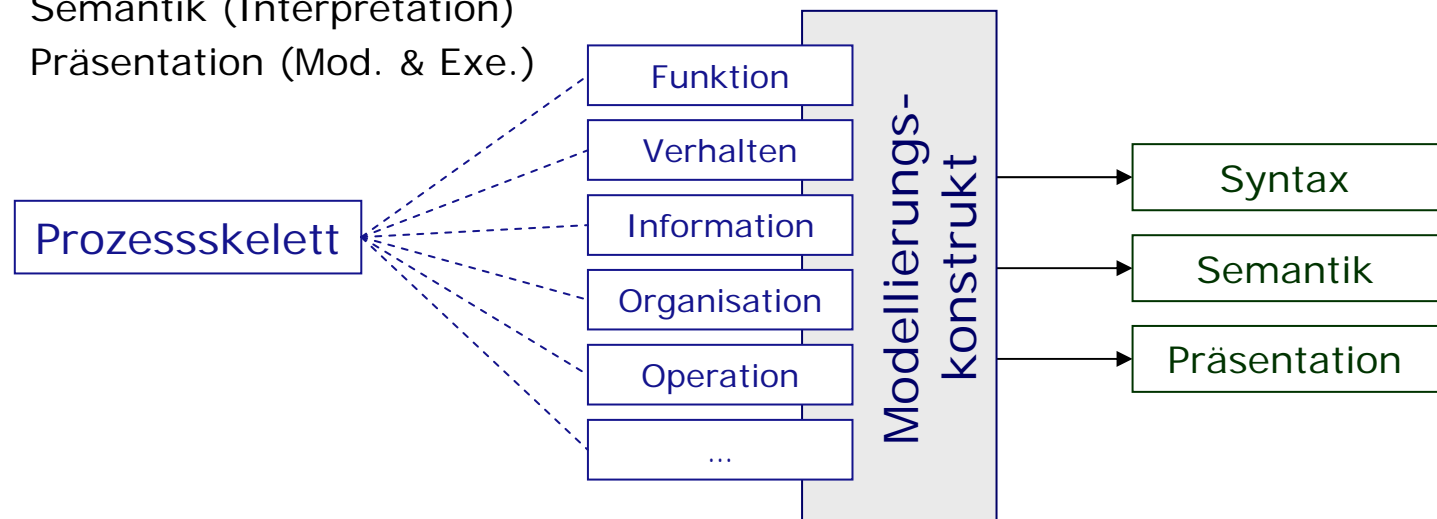
- iPM4med (integrated ProcessManager for medical applications)
  - Perspektiven-orientiertes Prozessmodell
  - Repositorium-basiertes Metamodell
- Perspektiven-orientiertes Prozessmodell
  - Funktionale Perspektive (Funktion, Inhalt)
  - Verhaltensorientierte Perspektive (Kontrollfluss)
  - Informationsbezogene Perspektive (Ein-/Ausgangsdaten, Datenfluss)
  - Organisatorische Perspektive (Agenten)
  - Operationale Perspektive (Applikationen, Systeme)
  - ...

Beispiel eines Modellierungskonstrukts: "Prozessschritt"



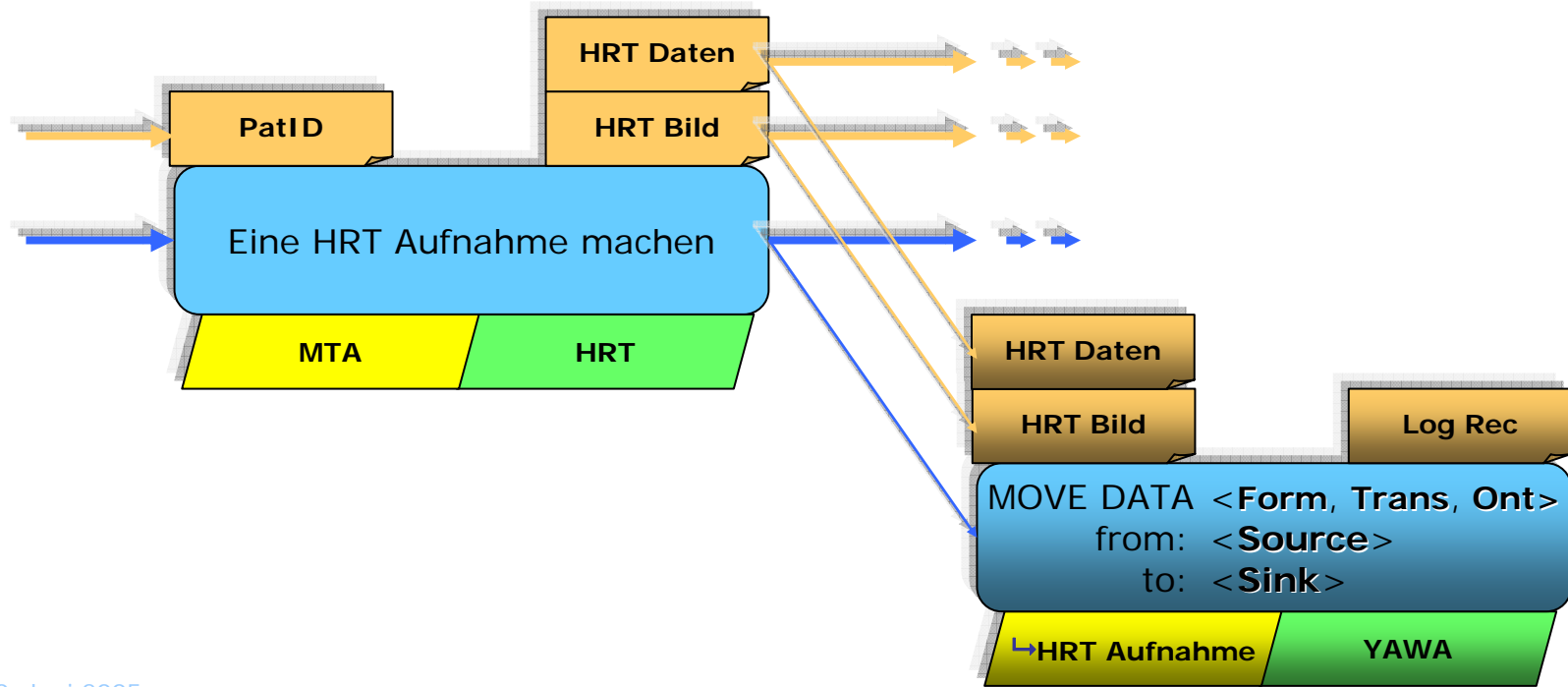
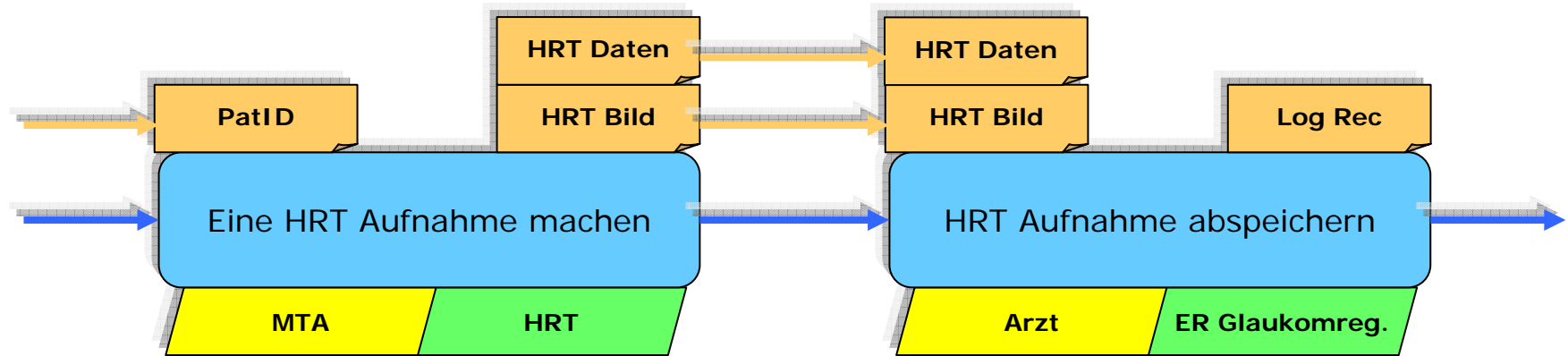
## Framework: iPM4med

- iPM4med (integrated ProcessManager for medical applications)
  - Perspektiven-orientiertes Prozessmodell
  - Repositorium-basiertes Metamodell
- Repositorium-basiertes Metamodell
  - Perspektiven sind nicht ins Prozessmodell "eingebrannt"
  - Prozessskelett interpretiert zugeordnete Perspektiven
  - Metamodelle für Perspektiven und Modellierungskonstrukte
  - Neue Modellierungskonstrukte werden eingeführt, indem – für alle betroffenen – Perspektiven definiert wird:
    - Syntax (Struktur, Attribute)
    - Semantik (Interpretation)
    - Präsentation (Mod. & Exe.)



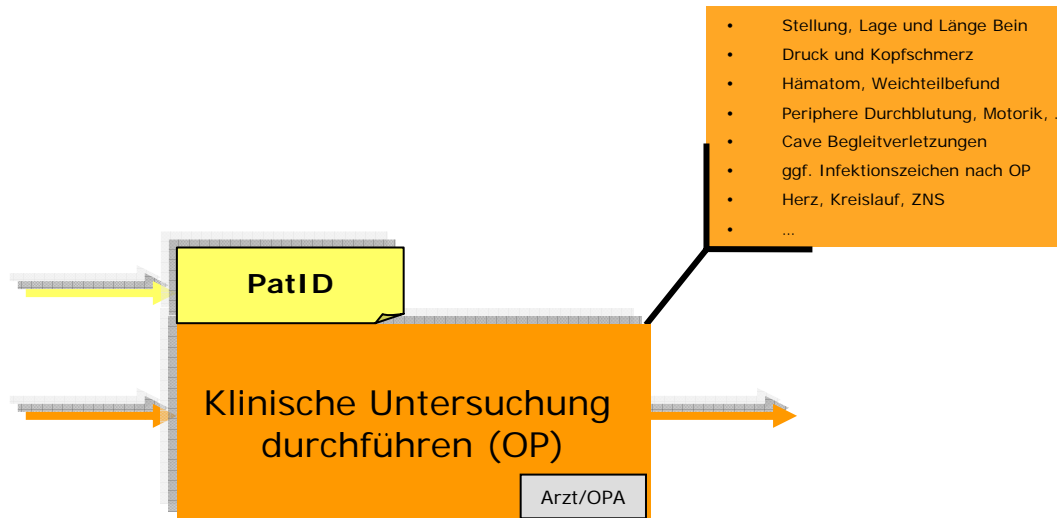
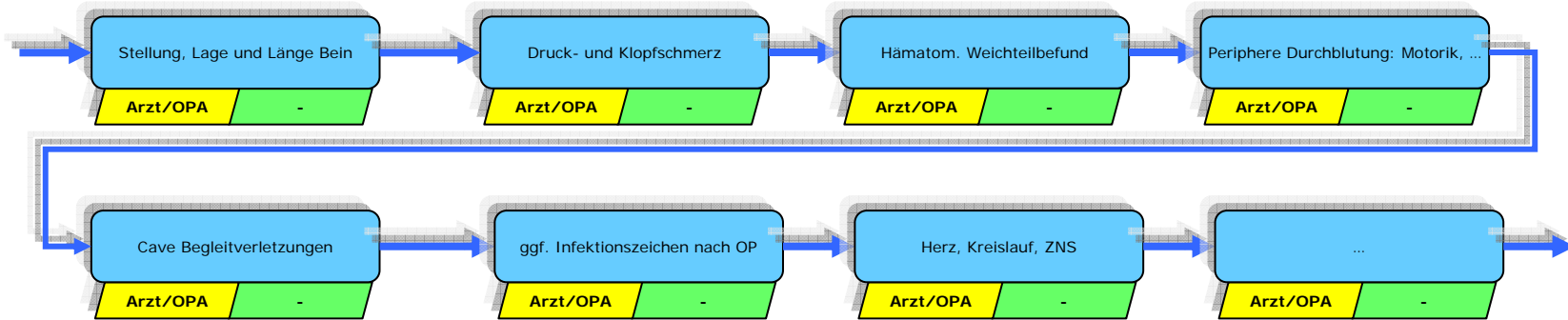
# iPM4med: Data Logistics

Beispielkonstrukt



# iPM4med: Checkliste

Beispielkonstrukt

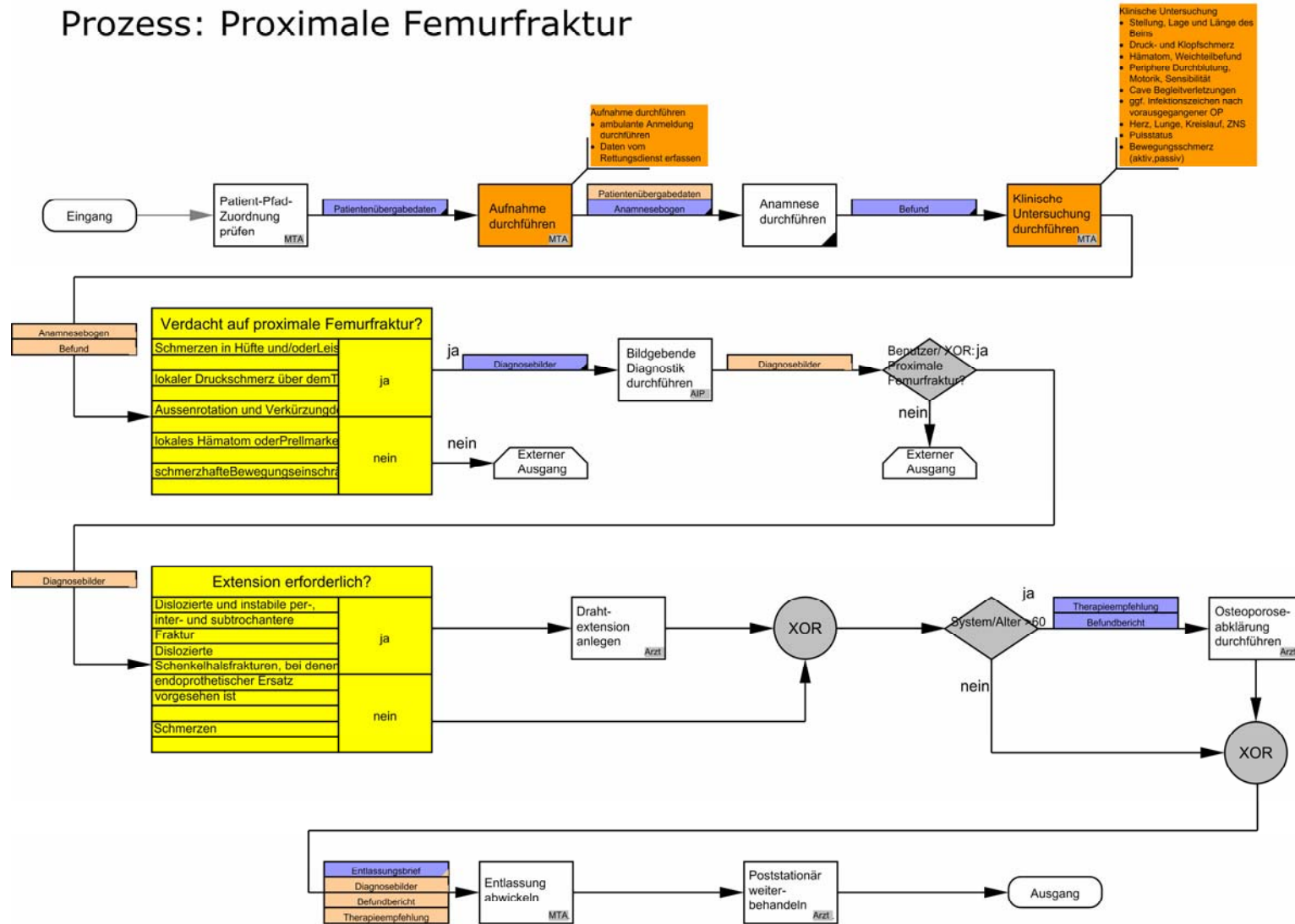




# iPM4med: Prozesse für medizinische Anwendungen

Anwendung

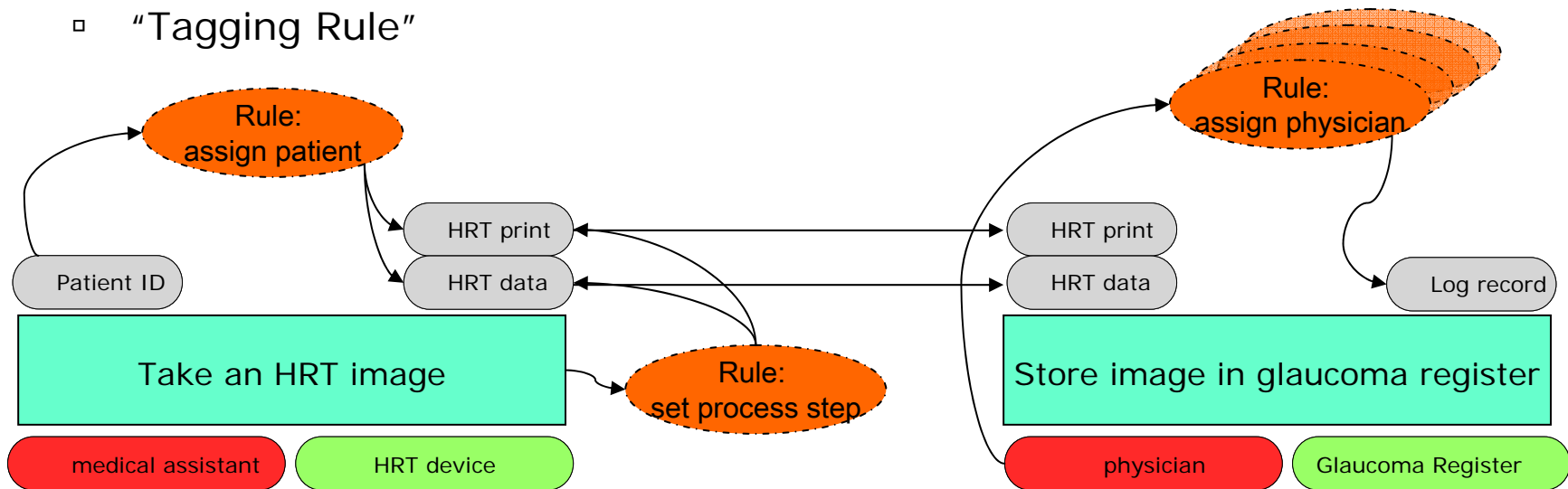
## Prozess: Proximale Femurfraktur



# iPM4med: Wissensmanagement

Beispielkonstrukt

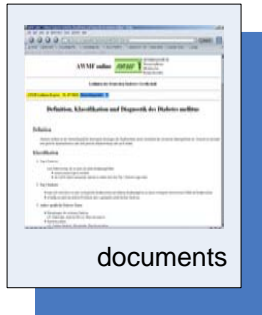
- **Goals**
  - (Re)use produced data as knowledge pool
  - Simple to use, but highly effective method to classify data
- **Method**
  - Introduce Meta Data
  - Define rules to specify what and how Meta Data is assigned
  - Integrate data into a central knowledge base (i>KM)
- **New Construct**
  - "Tagging Rule"



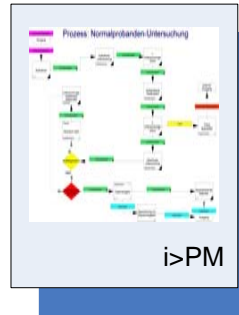
# Wissensmanagement - Überblick

Anwendung

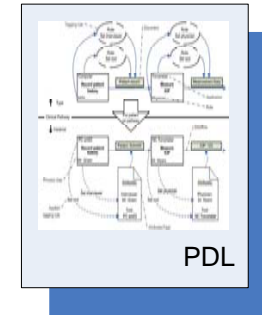
guideline/directive selection



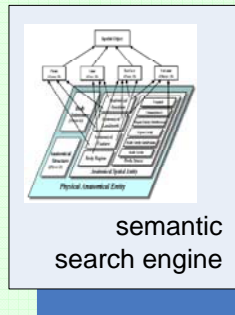
process modeling



process execution



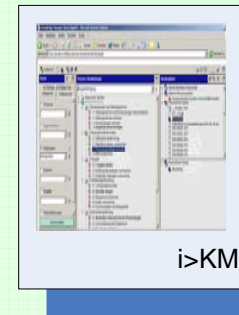
semantic search



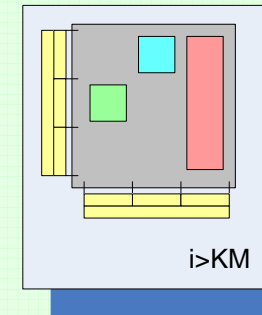
information export



information retrieval



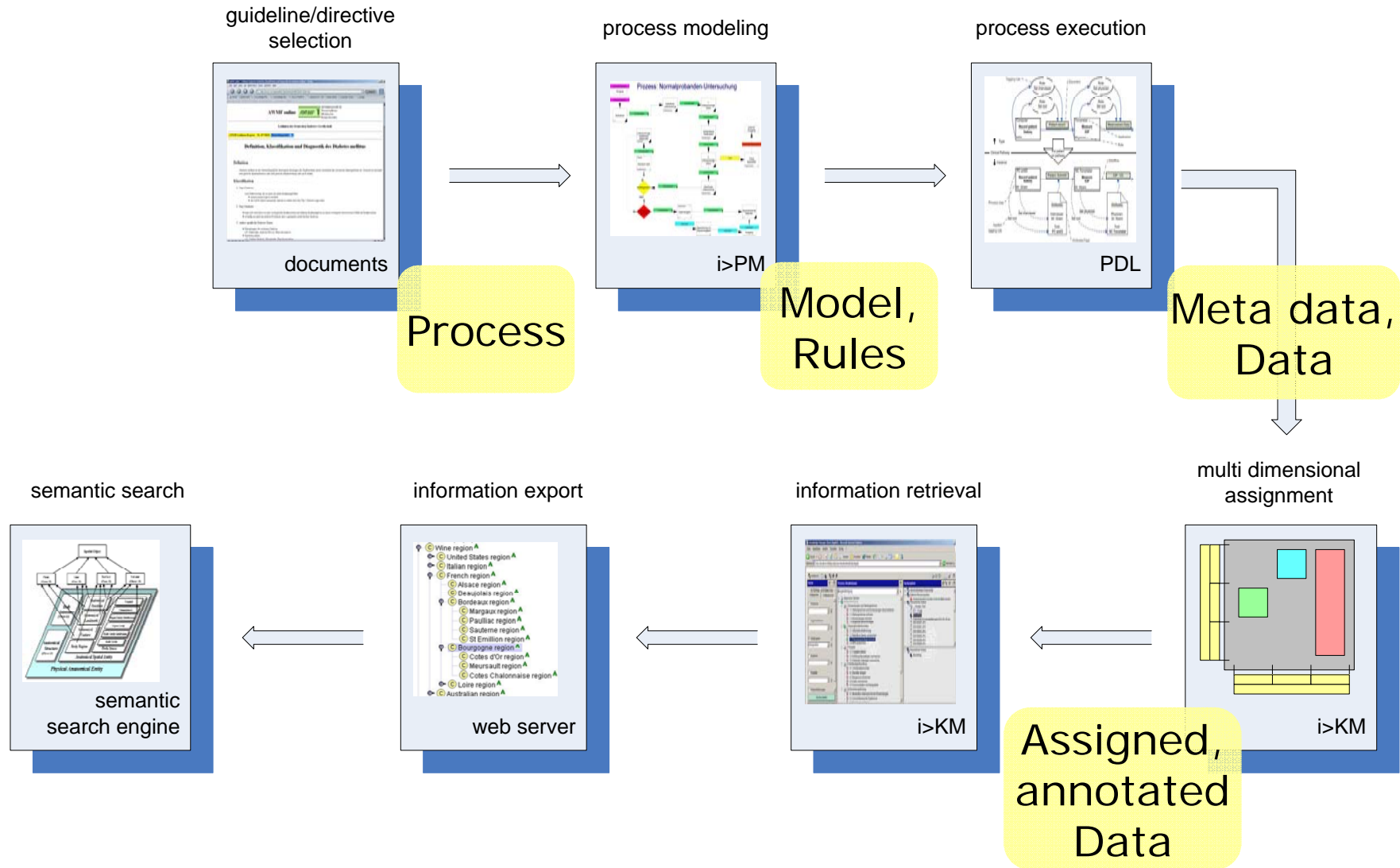
multi dimensional assignment



Knowledge Publishing Pipeline

# Wissensmanagement - Überblick

Anwendung



# Prozesse finden und auswählen

Anwendung

- Find basic process structures
- Typical sources
  - Medical guidelines and directives
  - Established processes (e.g. ISO certification documents)
  - Interviews with experts
- Often processes are already established
- Result:  
Decision which process to support

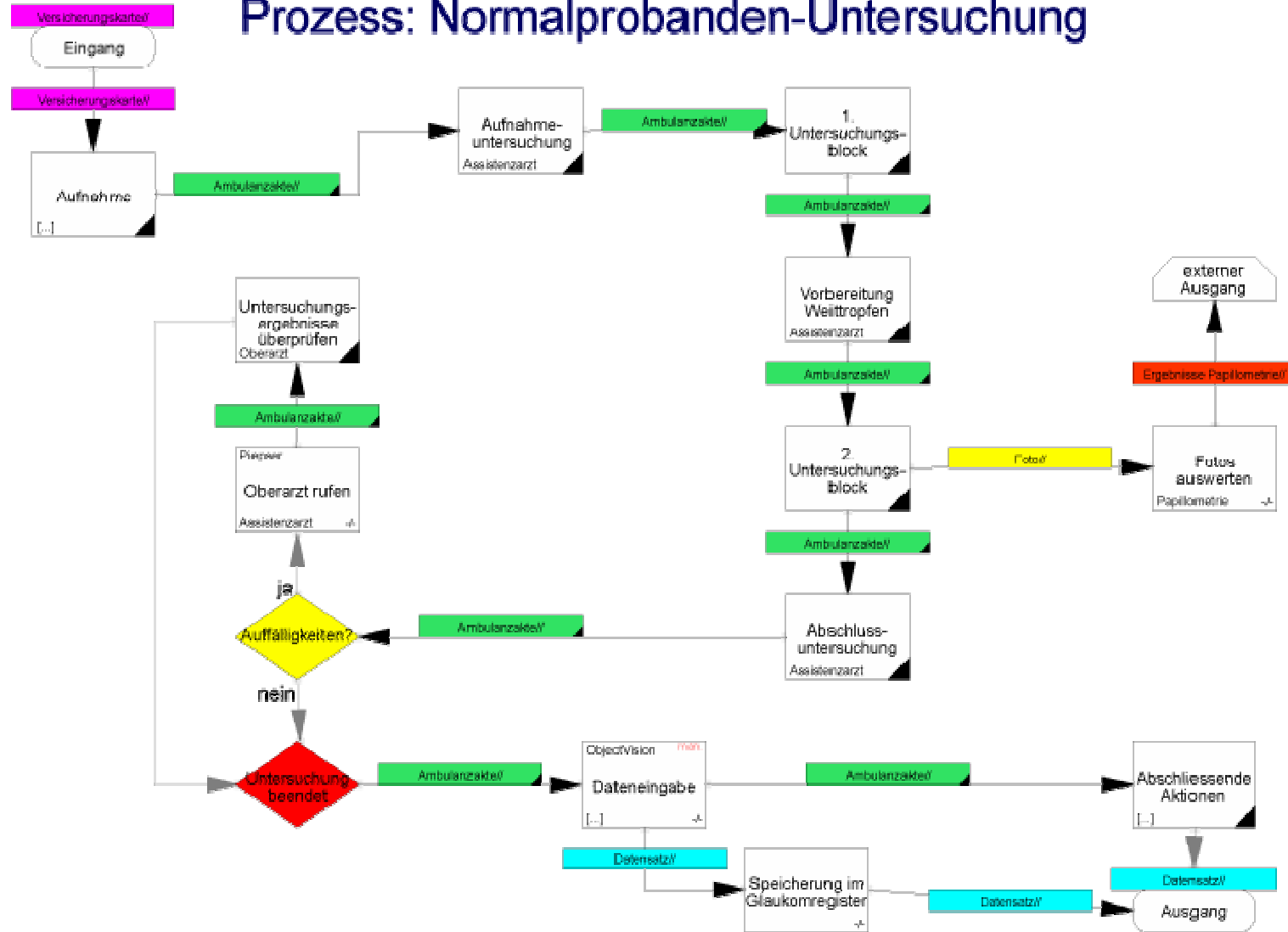
Typical problem:  
Find an agreement among  
the domain experts (physicians)



# Prozesse modellieren

Anwendung

## Prozess: Normalprobanden-Untersuchung

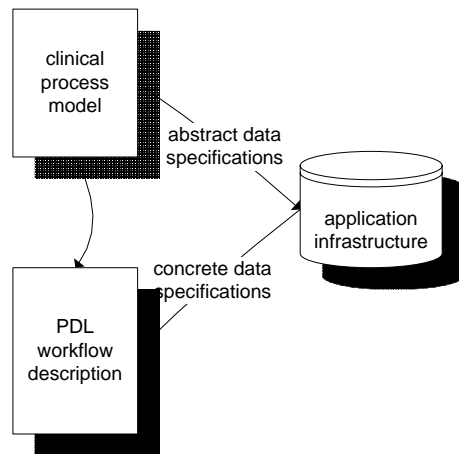


# Prozessbasierte Datenlogistik (PDL): Konzept

Anwendung

1. Derive data logistics processes
  - Clinical process as basis
  - Take “localized” configuration data out of repository
2. Configure data transport system
  - YAWA (wrapper framework)

Modeling phase

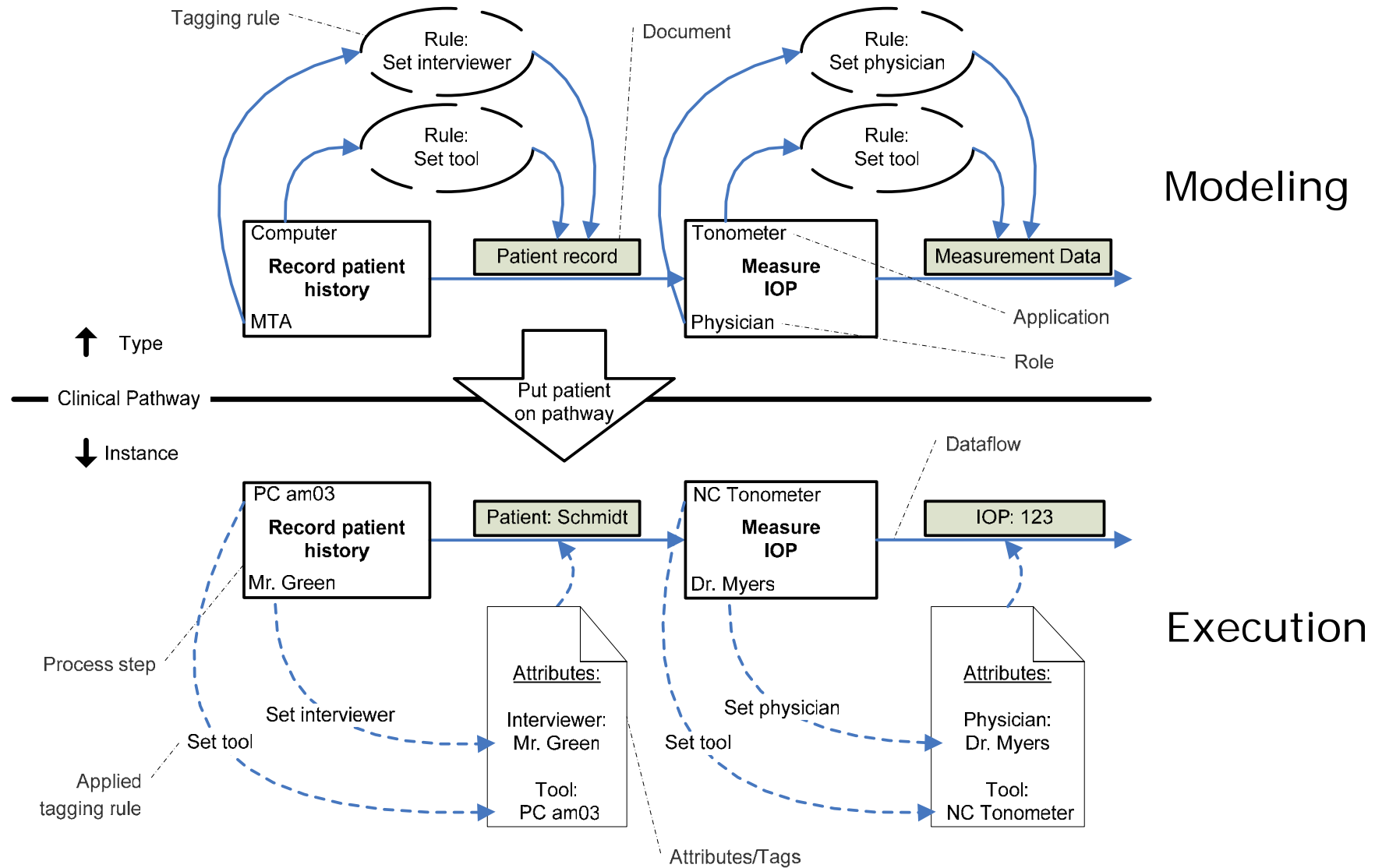


3. Handle application data
  - Data transportation/access
  - Data transformation

Execution phase

# Prozessbasierte Datenlogistik (PDL): Ausführung

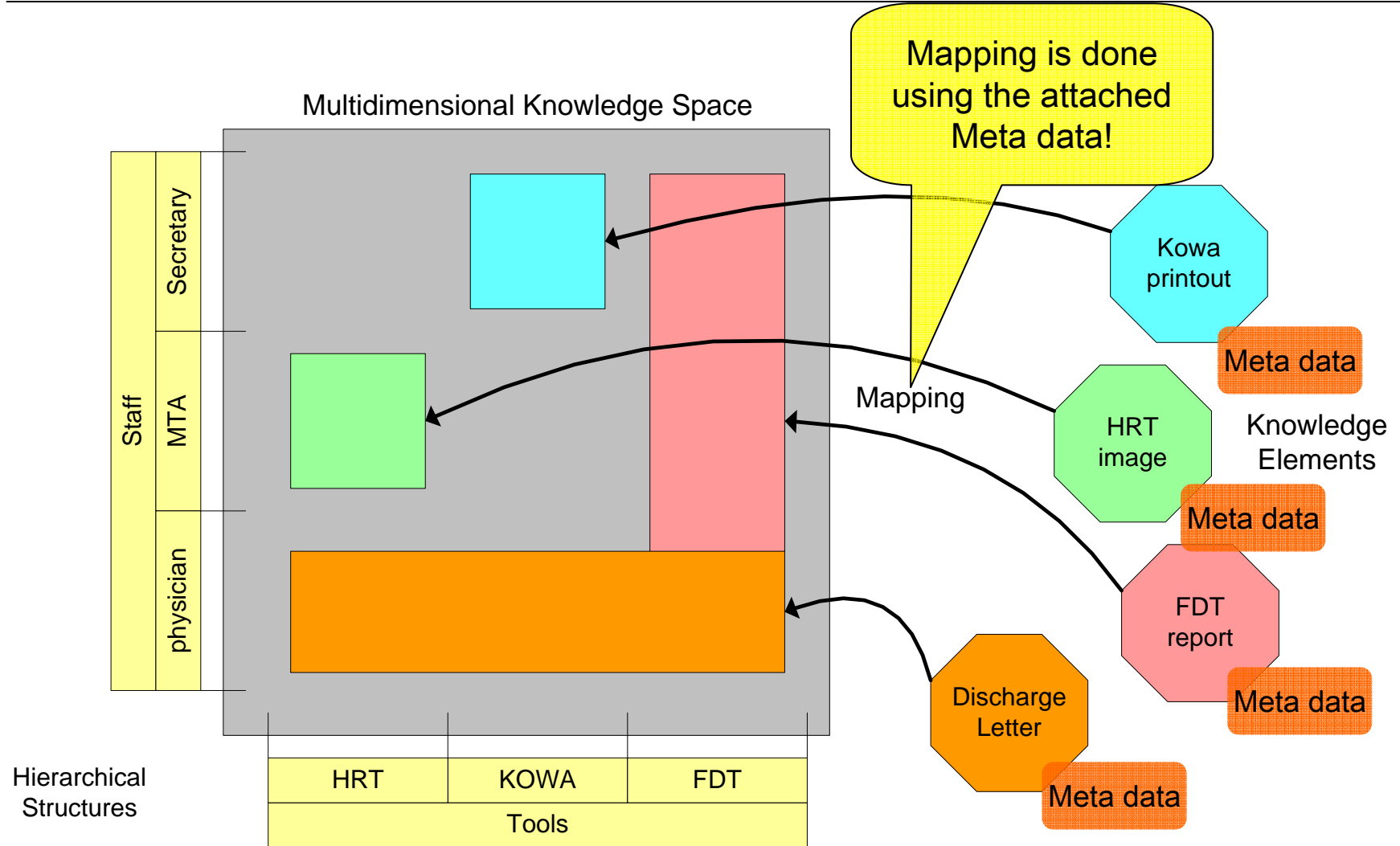
Anwendung





# Verknüpfung mit der Wissensbasis

Anwendung



# Werkzeuge für den Enterprise Knowledge Manager

Anwendung

Search panel:  
Category selection,  
full text search,  
external systems

Process tree:  
Central  
organization  
structure

Search results:  
Documents, multi  
media data,  
hyper-links

The screenshot displays the i-Knowledge Manager application interface. The browser address bar shows the URL: `http://localhost:8080/prodato/servlets/ilm/Html/Public/Apply`. The application is divided into three main sections:

- Suche (Search):** Located on the left, it features a search panel with options for "EXTERNAL SYSTEMS (1/4)", "Kategorien", "Volltextsuche", and "Prozesse". It includes a search term input field, a "Neu" button, and checkboxes for "Organisationen", "Zielgruppen" (checked), "Systeme", "Projekte", and "Praxiserfahrungen".
- Prozess-/Strukturbaum (Process/Structure Tree):** Located in the center, it displays a hierarchical tree structure for "Baugenehmigung". The tree includes categories like "Allgemeine Struktur", "Baugenehmigung", "1 Einwendungen und Stellungnahmen", "2 Planung/Vorabinformation", "3 Freigabe", "4 Vollständigkeitsprüfung", and "5 Entscheidungsfindung". The item "2.3 Planungsgrundlage einholen" is highlighted.
- Suchergebnis (Search Results):** Located on the right, it displays a list of search results, including "Administrierbare Dokumente", "Externe Wissensquellen", "Ansprechpartner konnten nicht ermittelt werden", "Personlicher Ordner", "017\_15.jpg", "Bauantrag", "Datenblatt iKnowledgeManager2002-05-29.doc", and several "DSC" files. A "Personlicher Ordner" named "Bauantrag" is also visible at the bottom.

# Prozessmodellierung und Wissensmanagement

---

Anwendung

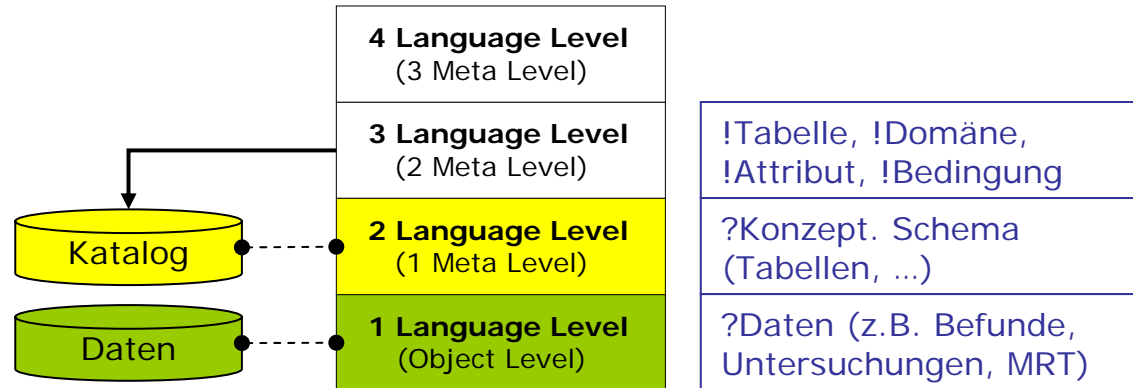
## Fusion: Knowledge Management at conceptual level

- New domain specific modeling construct: Tagging rules
- Specification within process model
- (Re)use of existing documents
  - Higher level of acceptance: well known
  - Increased productivity: no need to learn new formats/syntax
  - No effort to create “new” documents
- Availability during process pathway execution
  - Knowledge pool automatically filled
  - iKM central point of information
  - Intuitive search method

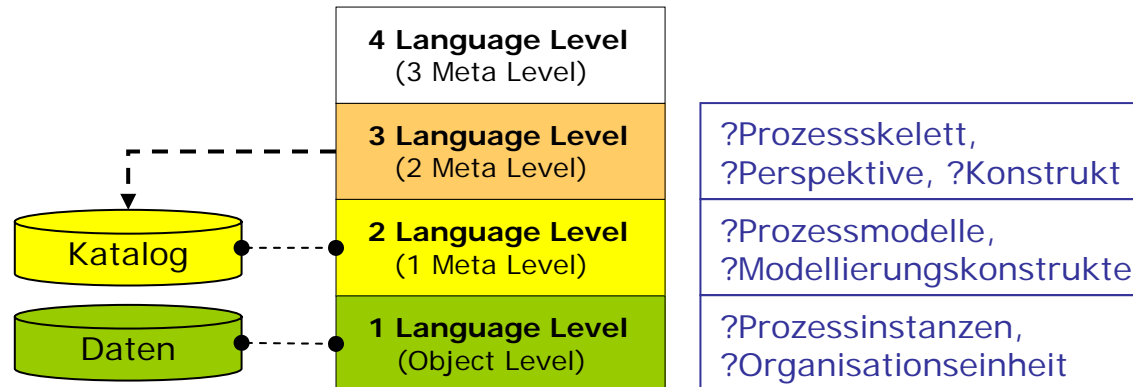
# Repository: Definition, Abgrenzung zu Datenbanken

Umsetzung

## ■ Datenbank



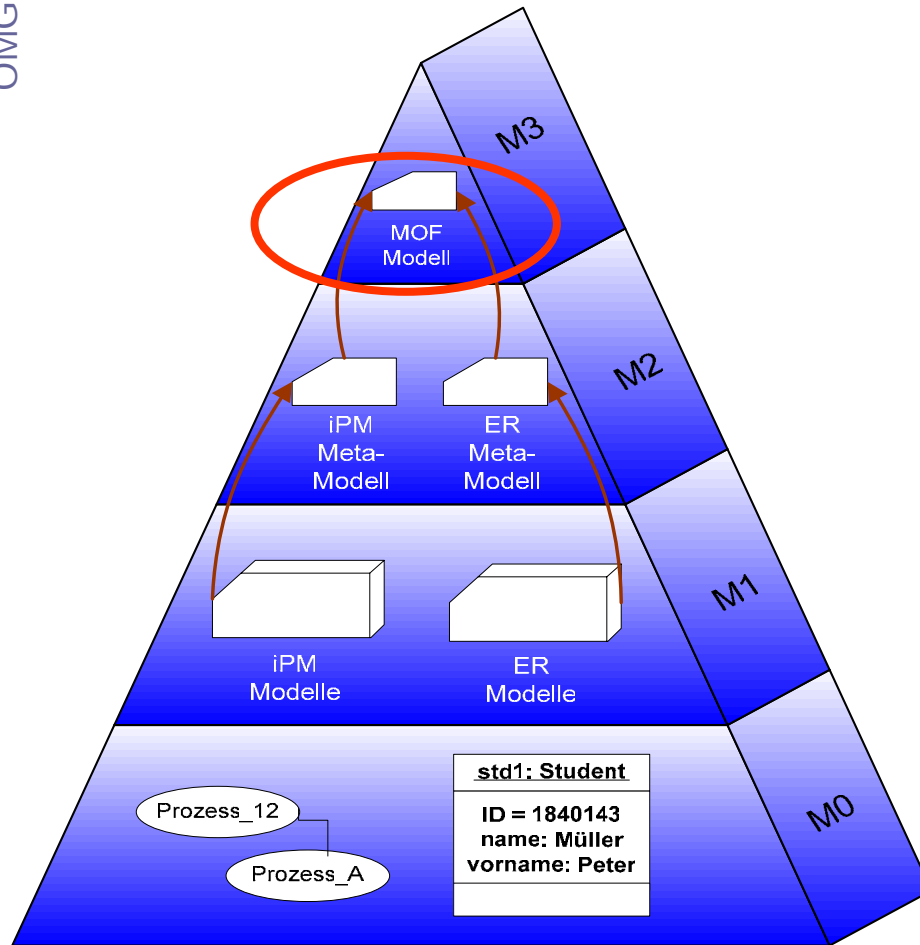
## ■ Repository



Datenspeicher mit anpassbarem Systemkatalog, in welchem Daten und Metadaten (2. LL) gleichberechtigt behandelt werden

# MOF Metadaten Architektur

OMG MOF

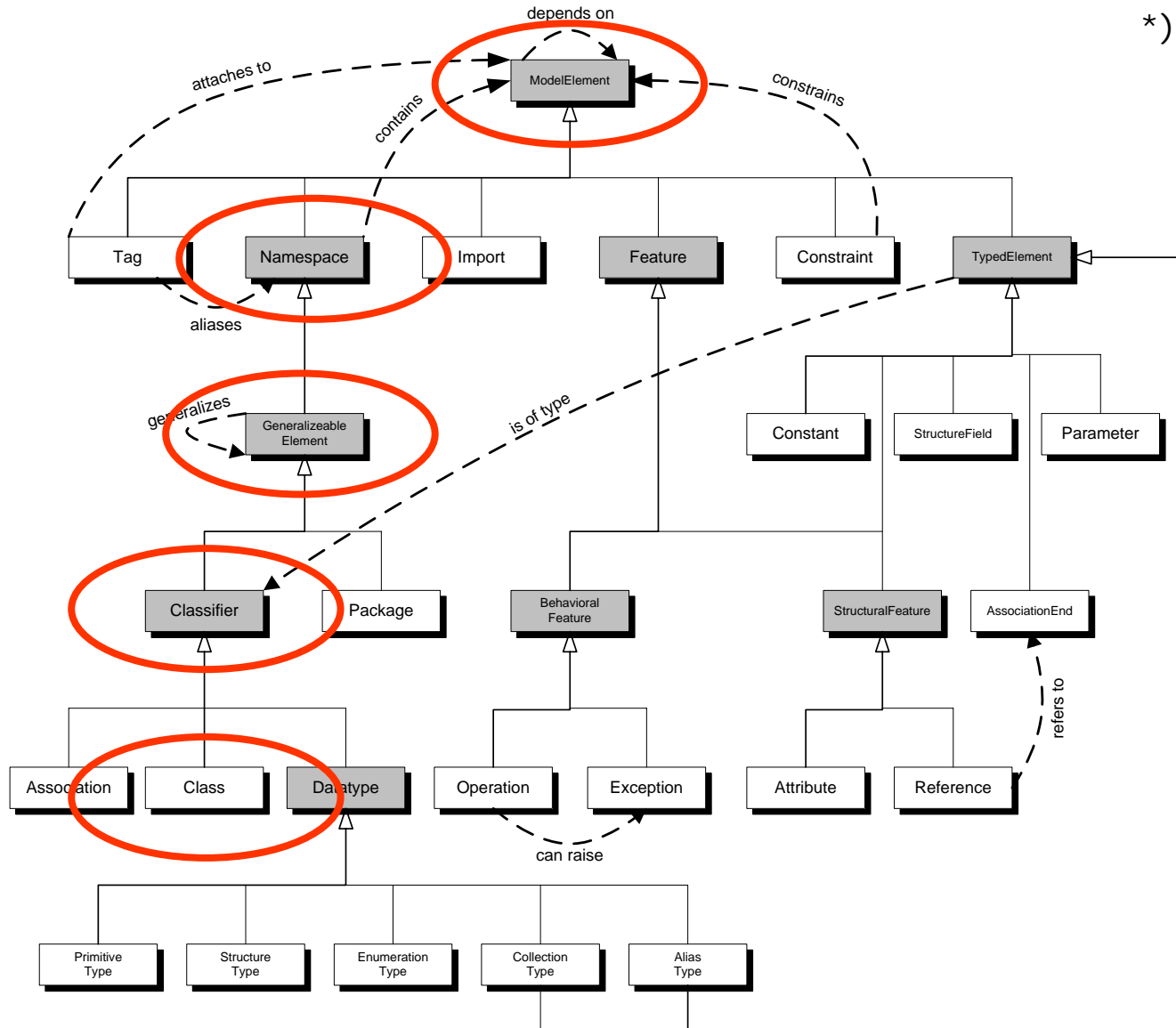


- **M3 (Meta-Meta Model / MOF layer)**
  - Umfasst Meta-Meta-Metadaten, welche die M2-Daten und sich selbst beschreiben (Selbst-Beschreibung)
- **M2 (Meta-Model layer)**
  - Umfasst Meta-Metadaten, welche die M1-Daten beschreiben
- **M1 (Model layer)**
  - Umfasst Metadaten, welche die M0-Daten beschreiben
- **M0 (Information layer)**
  - Umfasst die Rohdaten einer Anwendung (Prozesse, Tupel, Objekte, etc.)

# MOF Model Package\*

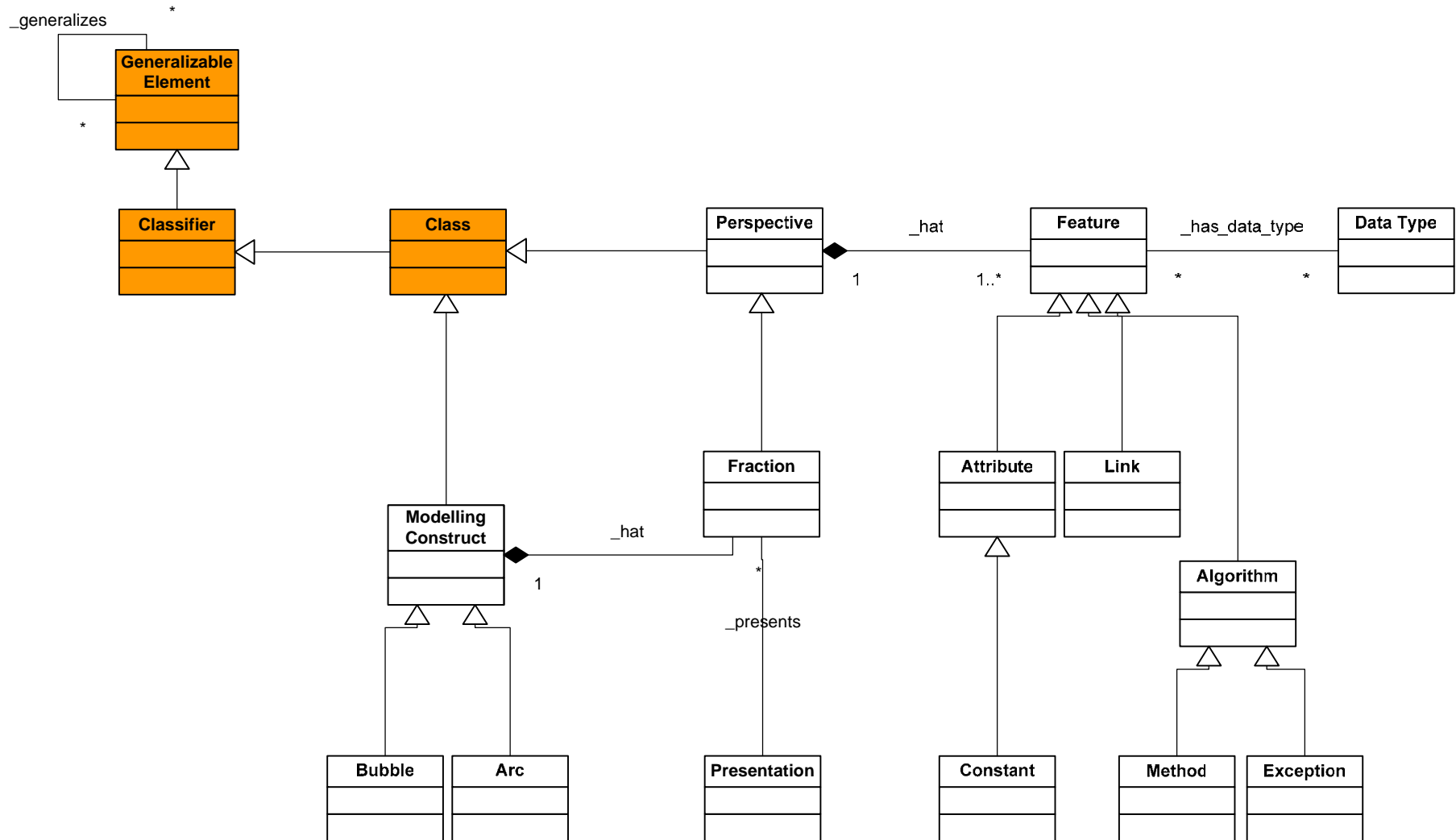
OMG MOF

\* ) vereinfacht



# M2-Modell: iPM Meta-Modell

Lösungskonzept

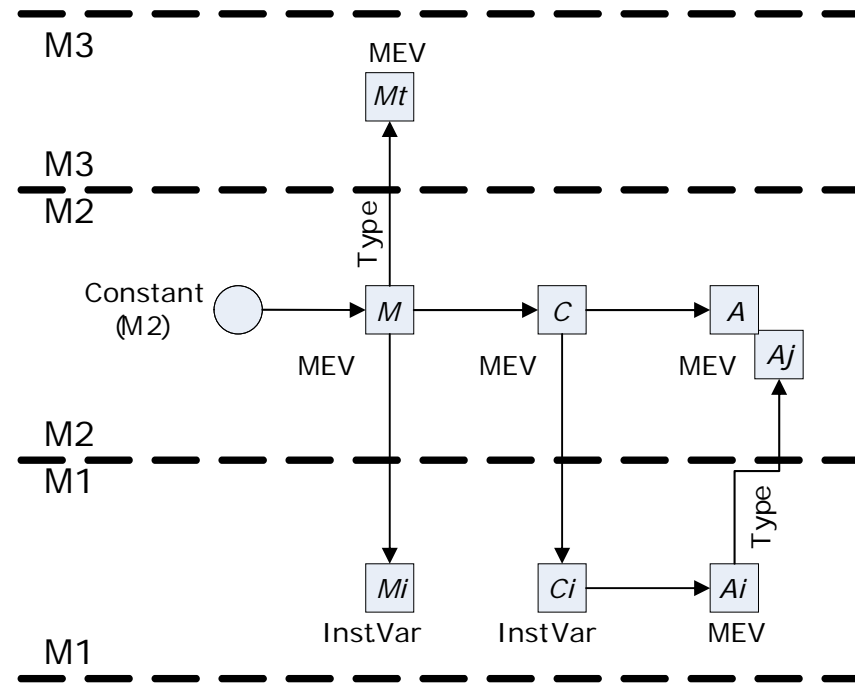


# iRM (integrated RepositoryManager): mSQL

Lösungskonzept

```

SELECT *
FROM M2::-->M,
    TYPEm(M) Mt,
    M->C, C->A^D,
    M Mi, C Ci,
    Ci->Ai^Di,
    TYPEm(Ai) At
    
```



- Schema-Exploration (horizontal), Schema-Discovery (vertikal) (reflection, introspection, intercession)
- Durchgängige, gleichartige Behandlung von Daten und Metadaten  
→ Aufheben der "schematic heterogeneity"
- Deklarative Anfrageformulierung → mSQL  
→ Schema-Unabhängigkeit



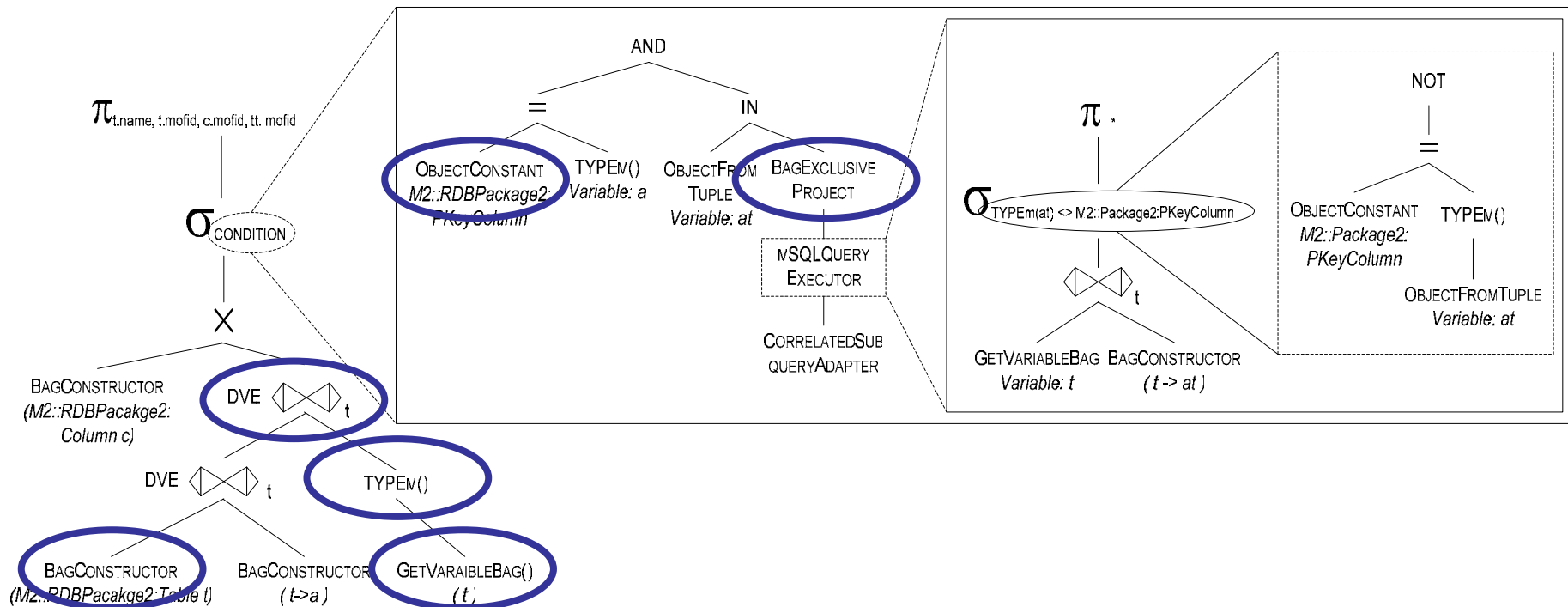
# mSQL: Beispielanfragen

- **Gib mir alle Prozesse im System**
  - ```
SELECT *
FROM M2::IPMPackage:Bubble bubbleInstances
WHERE bubbleInstances.name = 'process';
```
- **Gib mir den Namen und die Namen der Attribute aller Elemente des IPMPackages , sowie die Elemente in denen diese Attribute definiert worden sind, sofern diese Elemente Instanzen im Extent M1::MedPack haben**
  - ```
SELECT class.name, attr.name, defined_in.name
FROM M2::IPMPackage -> class, class -> attr, M2::IPMPackage -> defined_in
WHERE class IN ( SELECT * FROM M1::MedPack -> elements, TYPEm(elements) elementType )
AND attr IN defined_in;
```
- **Gib mir die Namen aller "Bubble"-instanzen, sowie die Namen und die Anzahl der Parameter ihrer Methoden**
  - ```
SELECT constructInstances.name, attributes.name, attributes.numberOfParameters
FROM M2::IPMPackage:ModelConstruct -> constructs constructInstances, constructInstances -> attributes,
TYPE(attributes) attributeTypes
WHERE M2::IPMPackage:Method IN attributeTypes;
```
- **Gib mir den Namen, die Farbe und die Form aller darzustellender Instanzen von ModelingConstruct im Extent M1::MedPack**
  - ```
SELECT constructInstances.name, presentationInstances.color, presentationInstance.shape
FROM M2::IPMPackage:ModelingConstruct -> constructs constructInstances,
M2::IPMPackage:Component -> components componentsInstances,
M2::IPMPackage:Presentation -> presentations presentationInstances
WHERE constructInstances._hat componentsInstances AND
componentsInstances._presents presentationInstances AND
constructInstances IN M1::MedPack AND presentationInstances.isVisible = true;
```

# mSQL: Anfragebearbeitung, Ausführungsplan

Lösungskonzept

- **Anfrage**
  - SELECT \*  
FROM M2::->M, TYPEm(M) Mt, M->C, C->A, M Mi, C Ci, Ci->Ai^Di, TYPEm(Ai) At
- **Neue Typen an Variablen (high order query language)**
  - Instanz-Variable (von SQL), Klassen-Variable, Package-Variable, Attribut-Variable, Datentyp-Variable, ...
  - ➔ Erweiterung der Relationenalgebra
- **Ausführungsplan**



## mSQL: Konsistenz

### ■ Strukturelle Konsistenz

- Sind Objekte konform zur Definition ("well-formed")
  - Komposition, Generalisierung, Assoziation (Kardinalität, Partizipation, etc.)
- Bei einer Modifikation
  - Ist die Operation erlaubt?
  - Können alle Instanzen an die modifizierte Objekt-Definition angepasst werden?

Manipulation ( $M_2$ )	Model Element	Required Action ( $M_1$ )
Deletion	Package	Delete the package proxy (including all contents)
Deletion	Class	Delete the class proxy (including all instances)
Deletion	Association	Delete the association proxy (including all links)
Deletion	Attribute (classifier-level)	Remove the attribute from the class proxy
Deletion	Attribute (instance-level)	Remove the attribute from all instances
Change	Containment Hierarchy	Roll back the RT
Change	Generalization Hierarchy	Roll back the RT
Change	Attribute Type	Roll back the RT if values are incompatible
Change	Association End Type	Roll back the RT, if the new type is not a super-type of the old one
Creation	Attribute (classifier-level)	Add a new attribute to the class proxy
Creation	Attribute (instance-level)	Add a new attribute to all class instances (initialized with default values)

### ■ Operationale Konsistenz

- Nebenläufiger Zugriff durch Repository-Transaktionen

# mSQL: Isolation

Lösungskonzept

- Repository soll zentrale Komponente einer Anwendung sein
  - ➔ Mehrbenutzerbetrieb
  
- Sperren auf mehreren Schichten (anders als bei DBMS)
  - ➔ komplexere Sperrverfahren
  
- Kumulative Sperren – "instance lattice"
  - Sperren von Instanz-Objekten auf allen darunter liegenden Ebenen
  - Sperren von abhängigen Objekten
  
- Sperren von Modell-Elementen
  - Vererbungshierarchie {gS, gX}
  - Kompositionshierarchie {cS, cX}
  - Assoziationen
  - Meta-Hierarchie {iS, iX}

		Requested		
		X	S	
M <sub>2</sub>	X	-	-	Present
	S	-	+	
M <sub>1</sub>	X	-	-	Present
	S	-	+	
M <sub>0</sub>	X	-	-	Present
	S	-	+	

# Beispiel zur (Anwendungs- und) Datenintegration

Lösungskonzept

Modalität_1		
Untersuchungen		
Patient	Durchgang	Wert
Pat11	D1	27.0
Pat11	D2	24.5
Pat13	D1	28.0
Pat13	D2	26.0
Pat12	D1	26.5
Pat12	D2	27.0

Modalität_2		
Untersuchungen		
Patient	D1	D2
Pat11	15.5	18.0
Pat13	19.0	19.0
Pat12	16.0	16.5

Modalität_3	
D2	
Patient	Wert
Pat11	25.5
Pat13	23.0
Pat12	20.0

D1	
Patient	Wert
Pat11	26.5
Pat13	22.0
Pat12	19.5

Modalität_4			
Untersuchungen			
Durchgang	Pat11	Pat13	Pat12
D1	6.0	7.5	5.5
D2	6.5	7.0	5.5

mSQL-Anfrage: "Alle Untersuchungsergebnisse"

```

SELECT      a.patient, a.durchgang, a.wert, b.attr_b, c.wert, d.attr_d
FROM        M1::Modalitaet_1:Untersuchungen a,
            M1::Modalitaet_2:Untersuchungen b, b -> attr_b,
            M1::Modalitaet_3 -> elem_c c,
            M1::Modalitaet_4:Untersuchungen d, d -> attr_d
WHERE       a.patient = b.patient AND a.patient = c.patient AND
            a.patient = attr_d AND a.durchgang = attr_b AND
            a.durchgang = elem_c AND a.durchgang = d.durchgang;
    
```

# Resümee

---

Zusammenfassung & Ausblick

- Framework-Architektur

