Large-Scale Integration of Assessment Form Data With RDF Networks

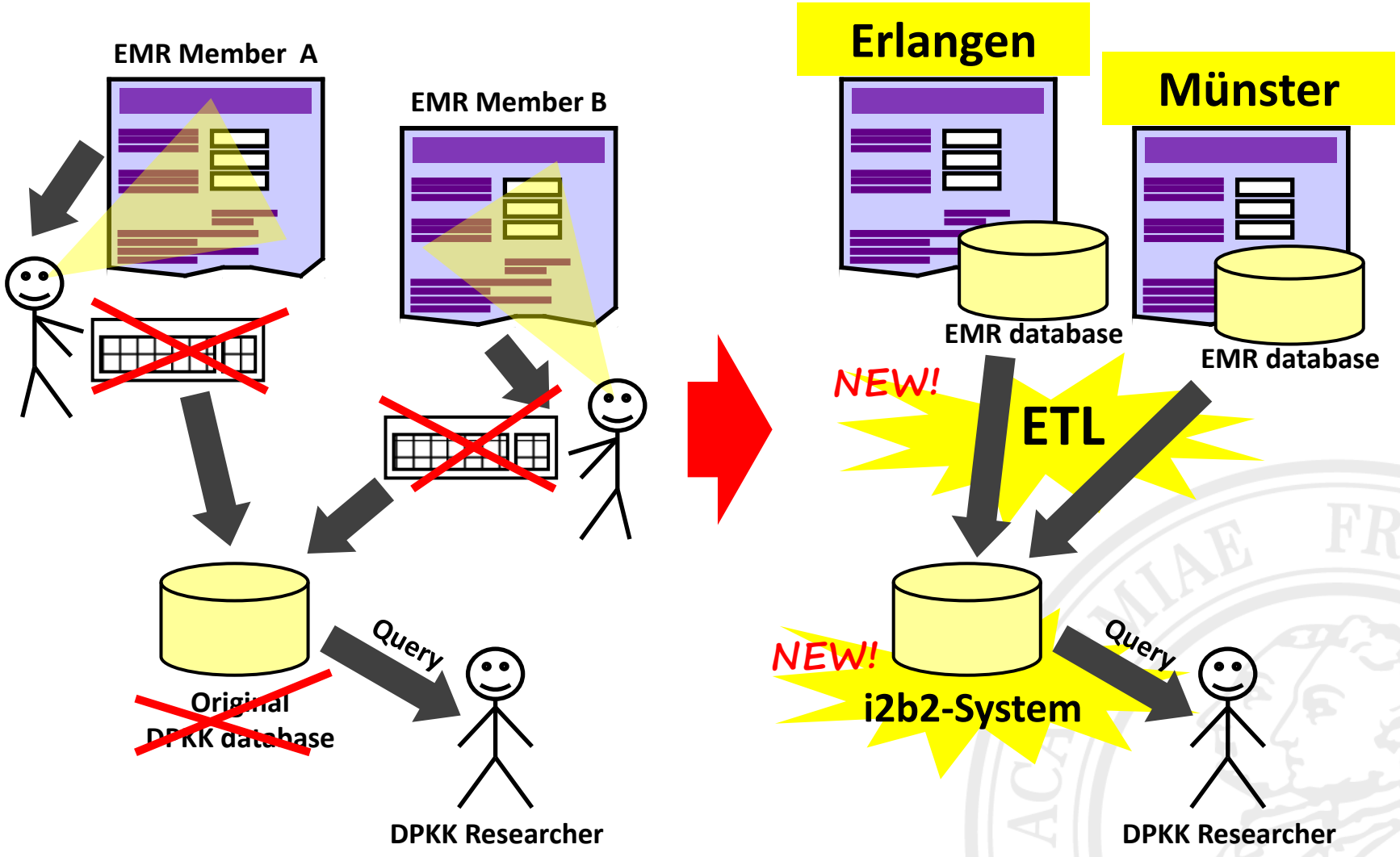
Sebastian Mate
12.12.2012

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Universitätsklinikum
Erlangen

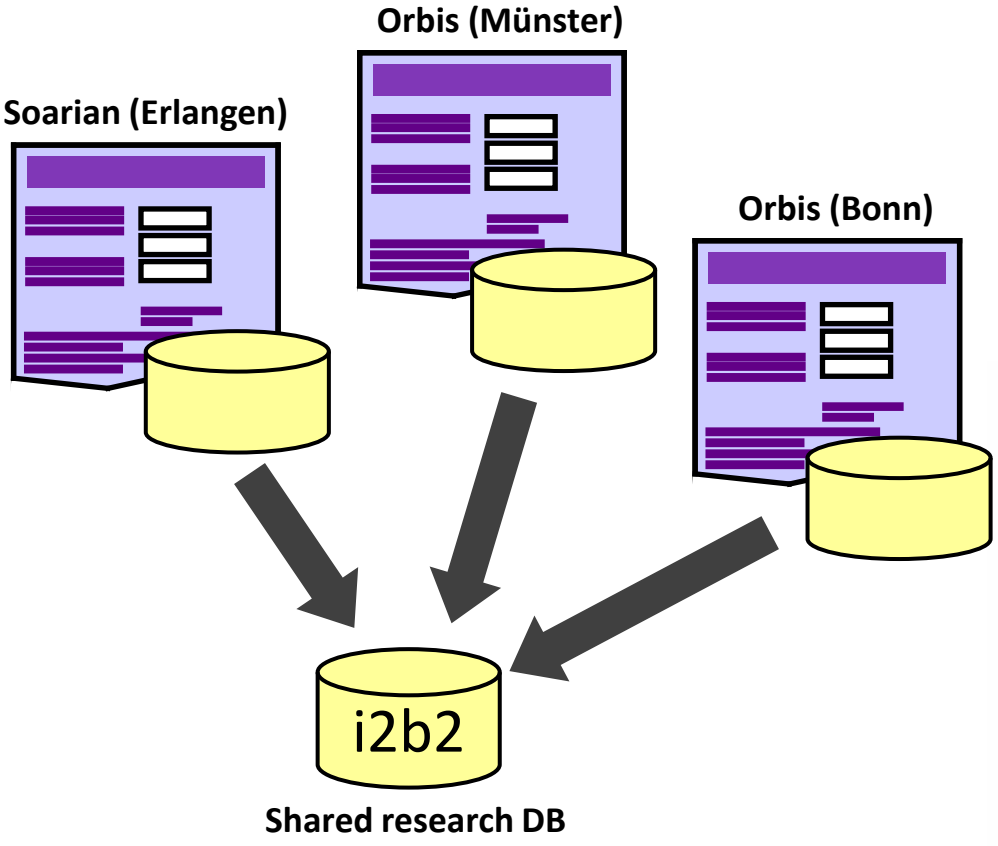
Background: The DPKK i2b2 Project

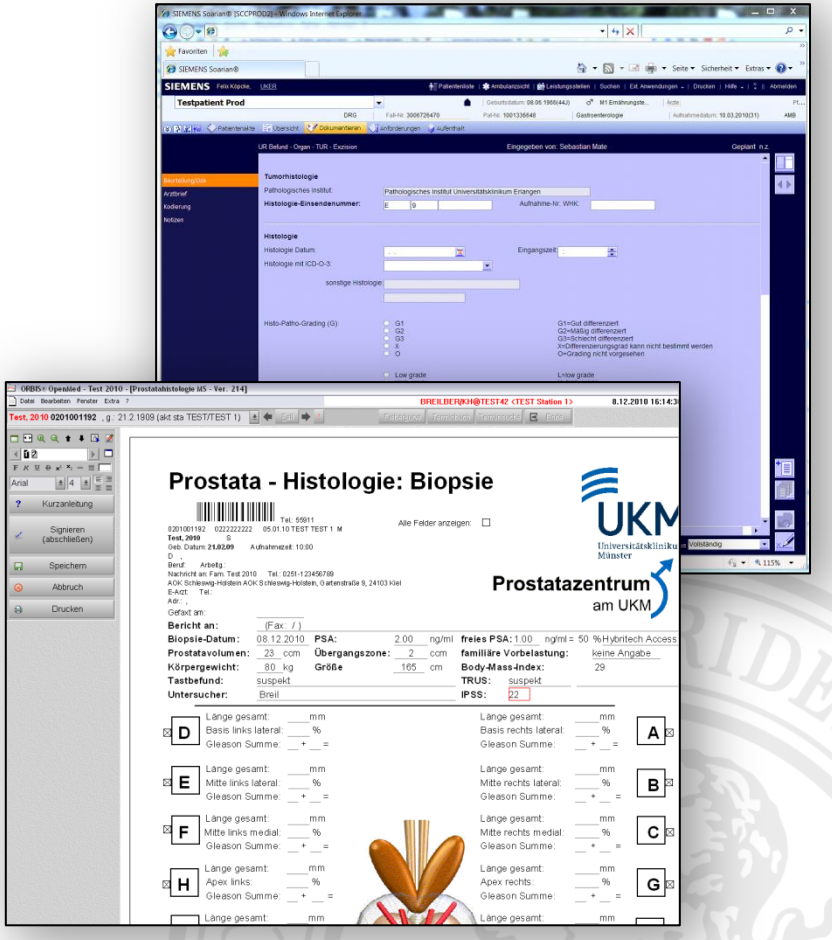
Reuse of EMR Data in a Shared Research Database



Background: The DPKK i2b2 Project

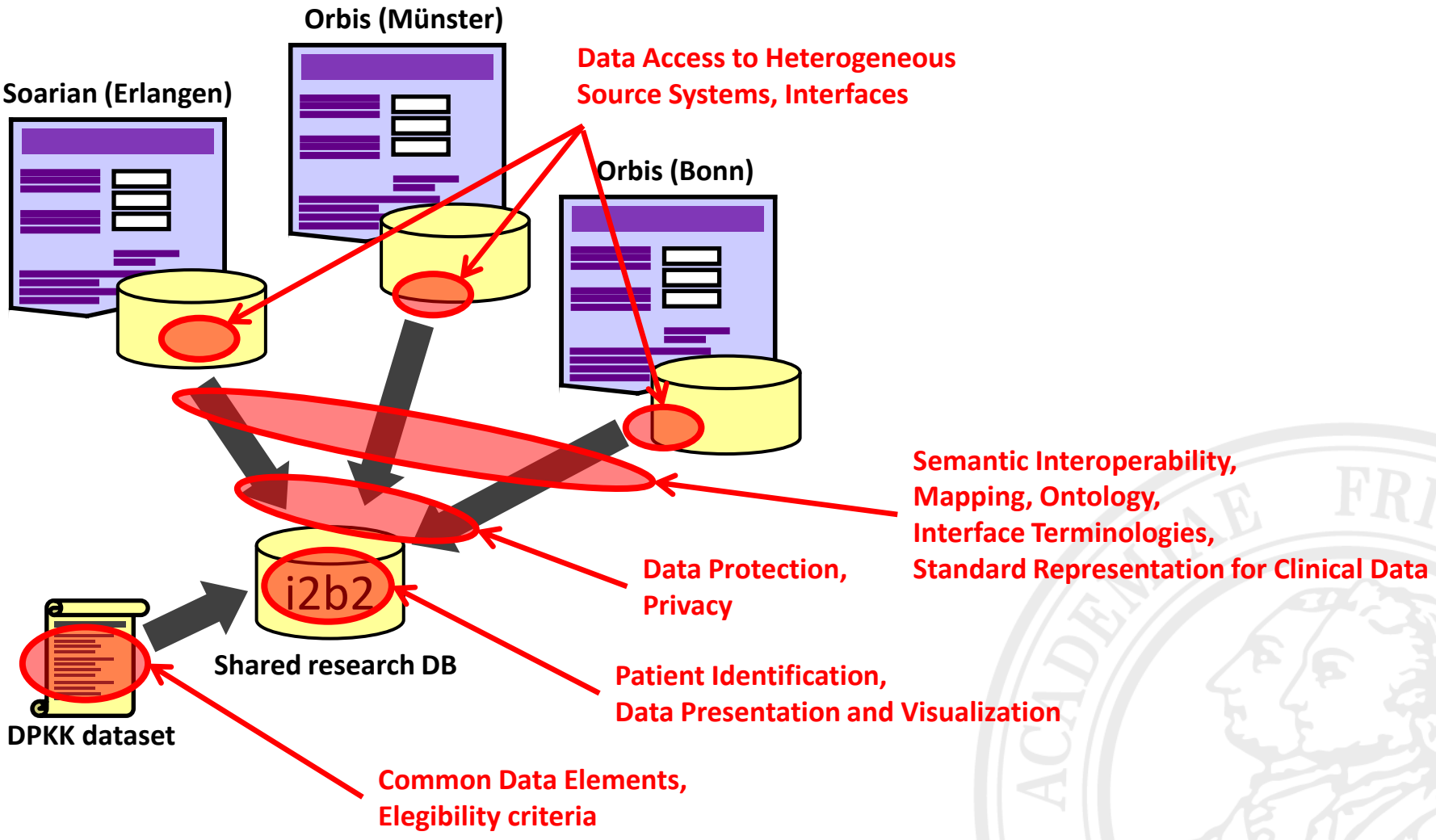
Reuse of EMR Data in a Shared Research Database



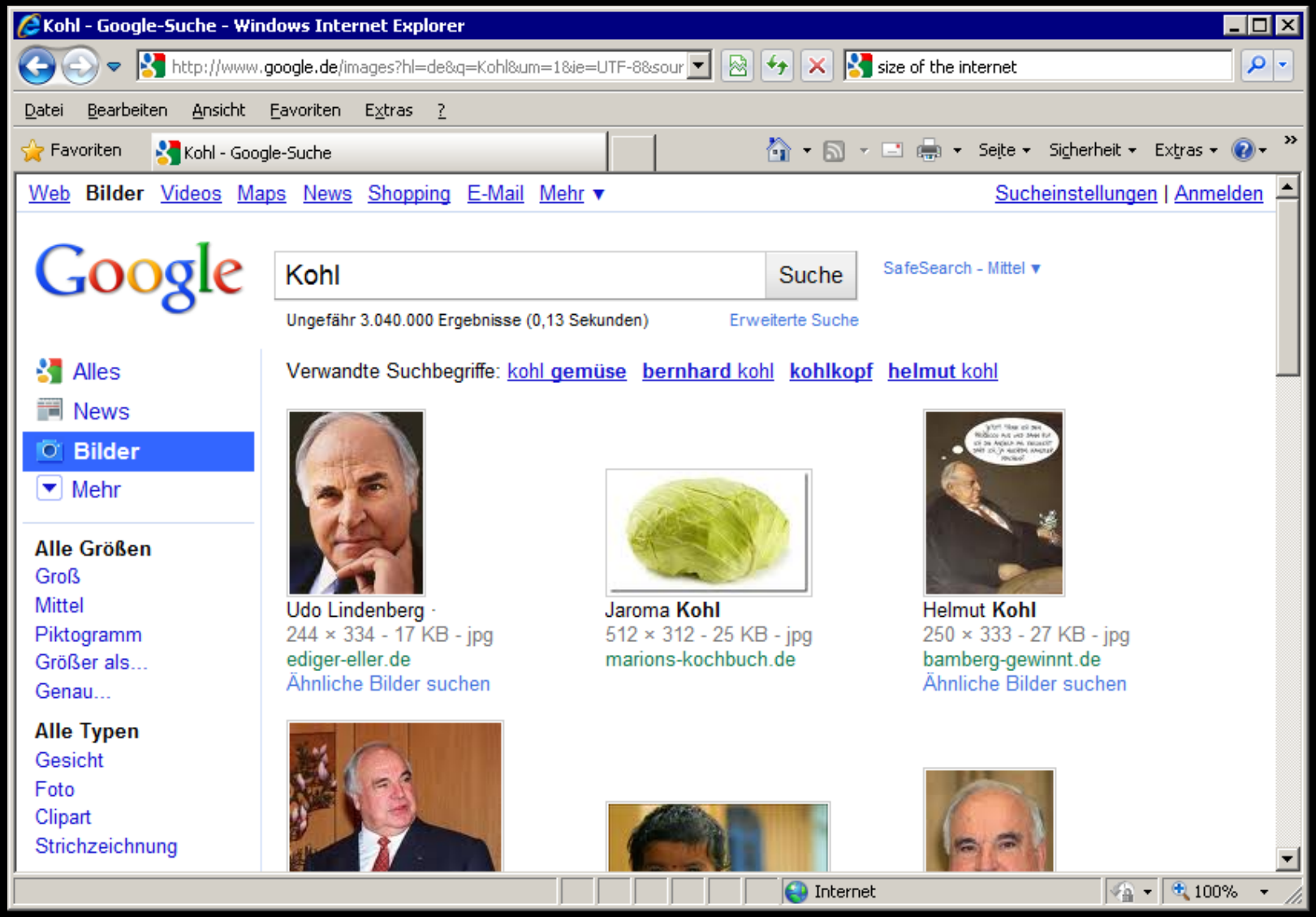


Background: The DPKK i2b2 Project

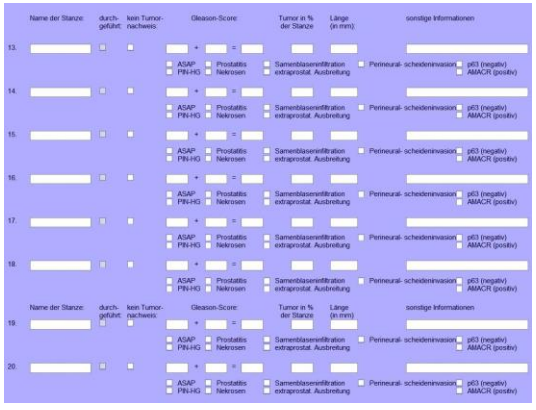
Reuse of EMR Data in a Shared Research Database



EMR vs. WWW


Kohl - Google-Suche - Windows Internet Explorer
http://www.google.de/images?hl=de&q=Kohl&um=1&ie=UTF-8&sour
size of the internet
Datei Bearbeiten Ansicht Favoriten Extras ?
Favoriten
Kohl - Google-Suche
Seite Sicherheit Extras
Web Bilder Videos Maps News Shopping E-Mail Mehr
Sucheinstellungen | Anmelden
Google
Kohl
Suche
SafeSearch - Mittel
Ungefähr 3.040.000 Ergebnisse (0,13 Sekunden)
Erweiterte Suche
Alles
News
Bilder
Mehr
Alle Größen
Groß
Mittel
Piktogramm
Größer als...
Genau...
Alle Typen
Gesicht
Foto
Clipart
Strichzeichnung
Verwandte Suchbegriffe: kohl gemüse bernhard kohl kohlkopf helmut kohl
Udo Lindenberg ·
244 × 334 - 17 KB - jpg
ediger-eller.de
Ähnliche Bilder suchen
Jaroma Kohl
512 × 312 - 25 KB - jpg
marions-kochbuch.de
Helmut Kohl
250 × 333 - 27 KB - jpg
bamberg-gewinnt.de
Ähnliche Bilder suchen
Internet
100%

Properties of Heterogeneous Data in Medical Documentation And The World Wide Web



Paper-based documentation

- Sometimes ;-) human-readable
- Understandable for humans

- Not machine-readable
- Not machine-processible

Electronic documentation (EMR)

- Human-readable
- Understandable for humans

- Machine-readable
- Not efficiently machine-processible

- Heterogeneous data
- Distributed storage

World Wide Web

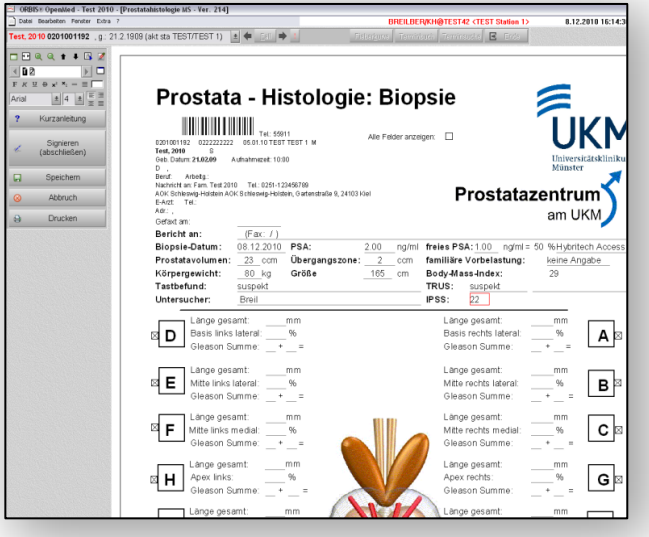
- Human-readable
- Understandable for humans

- Machine-readable
- Not efficiently machine-processible

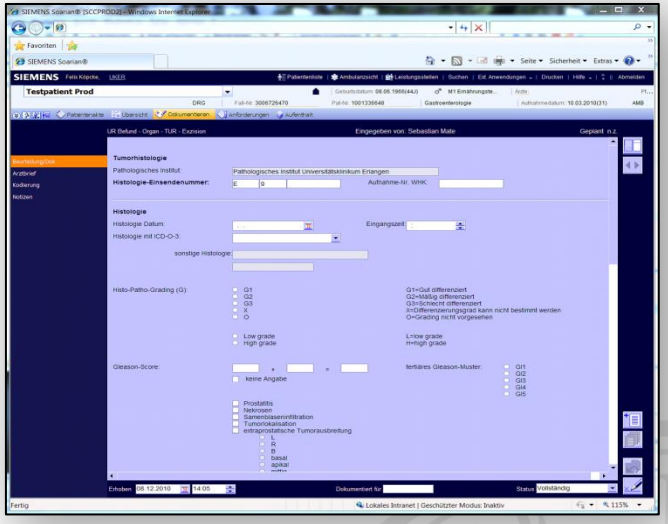
- Heterogeneous data
- Distributed storage

Developing a New Data Integration Approach Challenges

Problem 1: How to align and merge data from independent sources?



EMR in Münster: AGFA ORBIS



EMR in Erlangen: Siemens Soarian

- Different data structures (different EMR vendors)
- Similar information (full prostate cancer documentation in EMR)

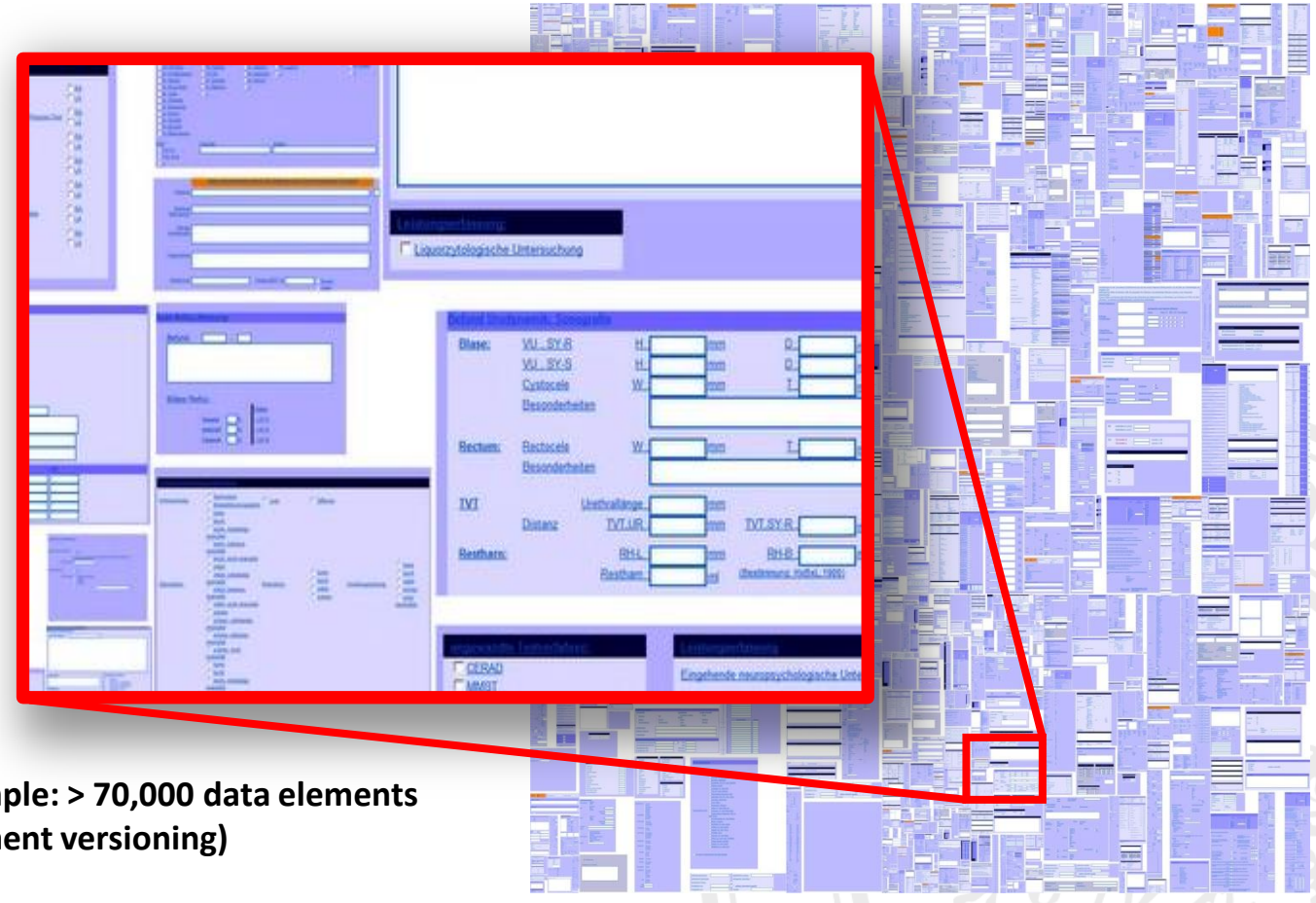
Developing a New Data Integration Approach

Challenges

Problem 2: How to efficiently access and process the huge piles of data elements?

Data:

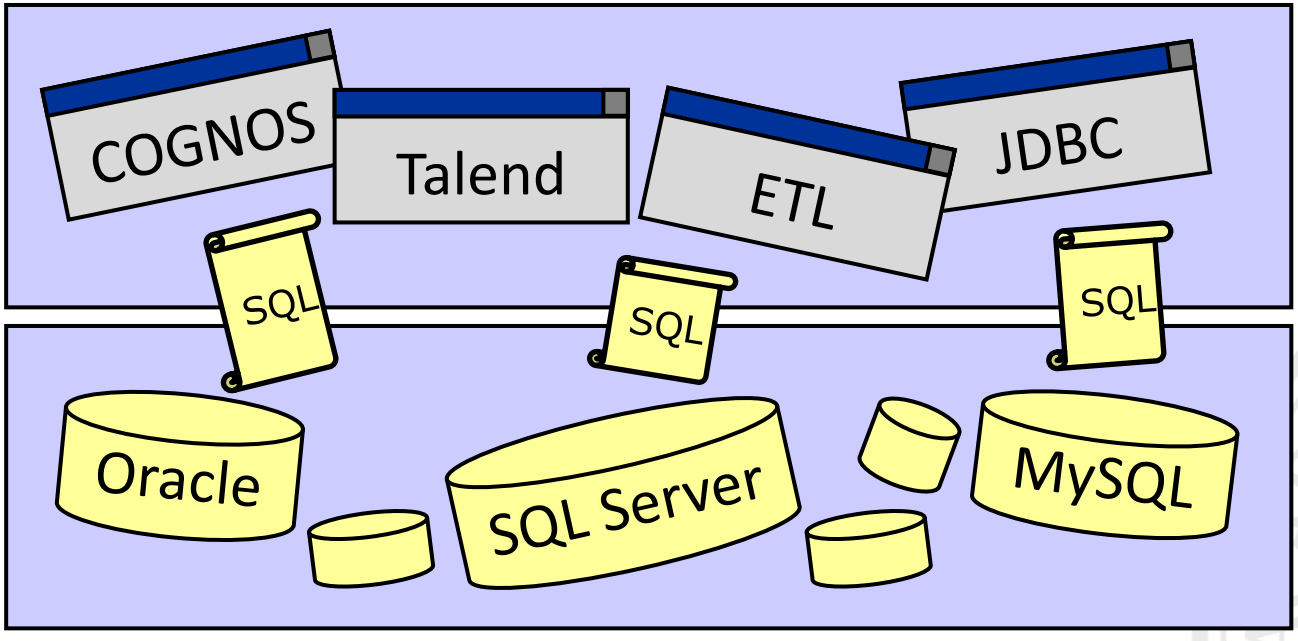
- Structured
- Uncoded



Erlangen as an example: > 70,000 data elements (including data element versioning)

Background: Digging for Gold

How to Access Clinical Databases?

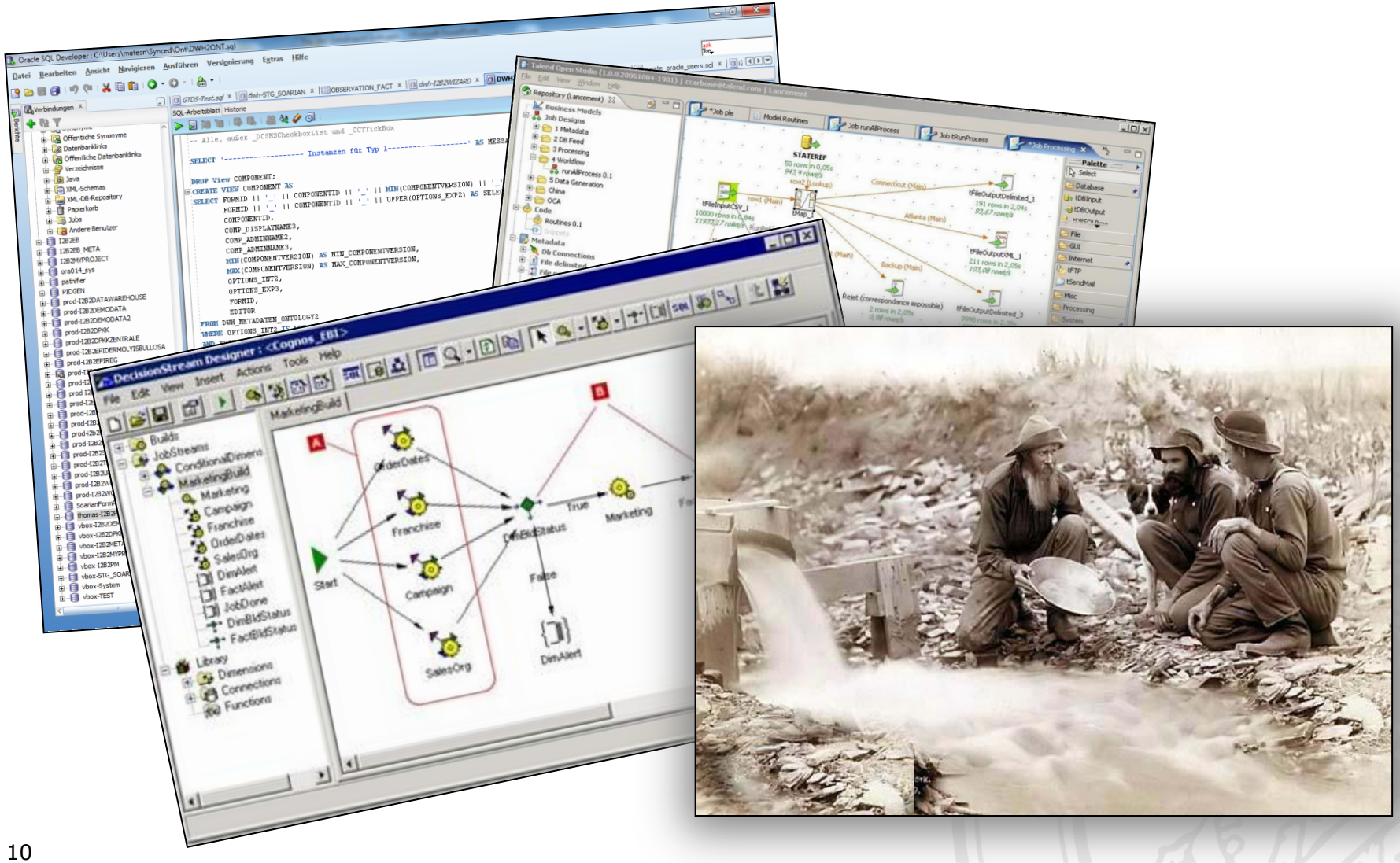


Tools / Data Access Layer

Data / Storage Layer

Background: Digging for Gold

… is very tedious at the Tools/Software/ETL Layer!



Background: Digging for Gold

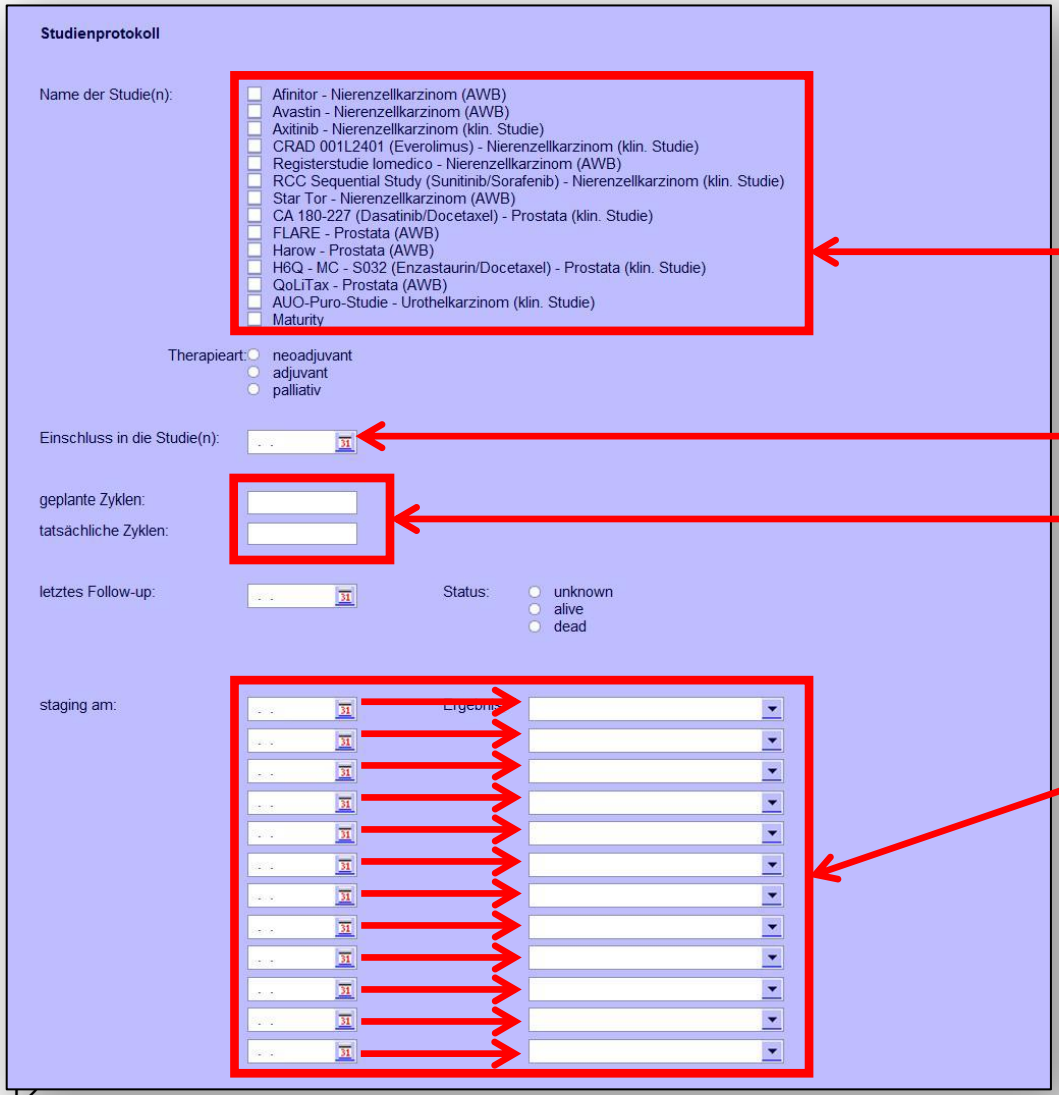
Why is it so tedious?

- **While technical access to the databases is easy, semantically correct reuse of the data is a very difficult, tedious task**
  - Little attention regarding the reuseability has been paid when the forms have been created
    - Primary goal was to replace paper-based documentation, not to support data reuse
    - EMRs do not offer functions to allow semantic annotations, only simple forms
  - Giant pile of data elements (> 70,000 in the Erlangen EMR)
  - Forms are designed for the human to be read but not to be processed by a machine
  - Clinicians know how to interpret relationships between different data elements. These are, however, documented nowhere!

- **In a cross-institutional scenario: Disparate Source Systems**
  - Different system architectures, database systems
  - Electronic documentation has evolved independently for each source system
    - Different clinical background
    - Available data elements differ

Background: Digging for Gold

Why is it so tedious? => Example from Erlangen



studies a patient can be enrolled in

date of enrollement

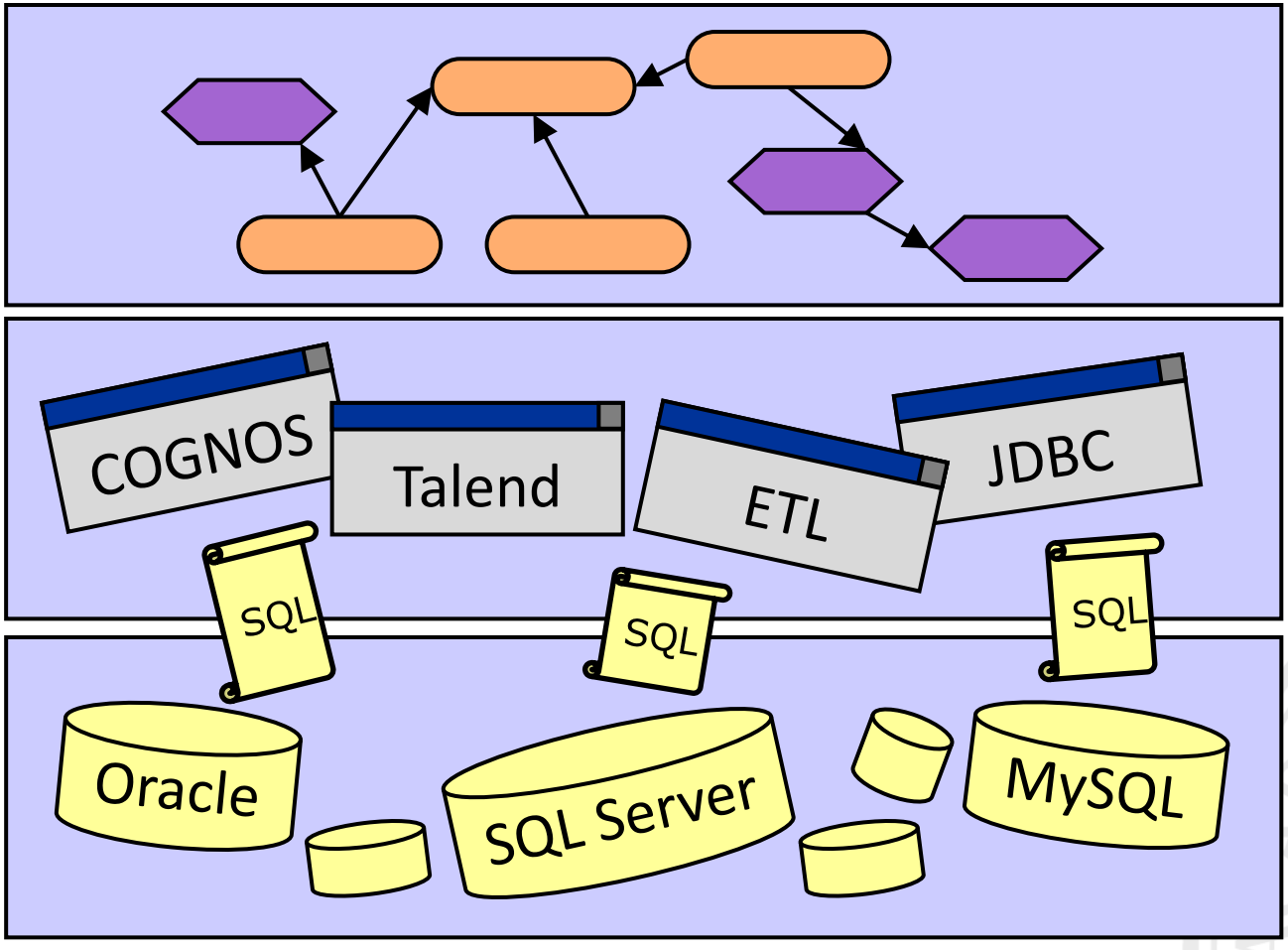
cycles

stagings & results

Semantic relations between CRF elements

Background: Digging for Gold

Need for Knowledge Management



Ontology / Knowledge Base Layer

Tools / Data Access Layer

Data / Storage Layer

Working at the Ontology layer:

… if done right, it can be very efficient!

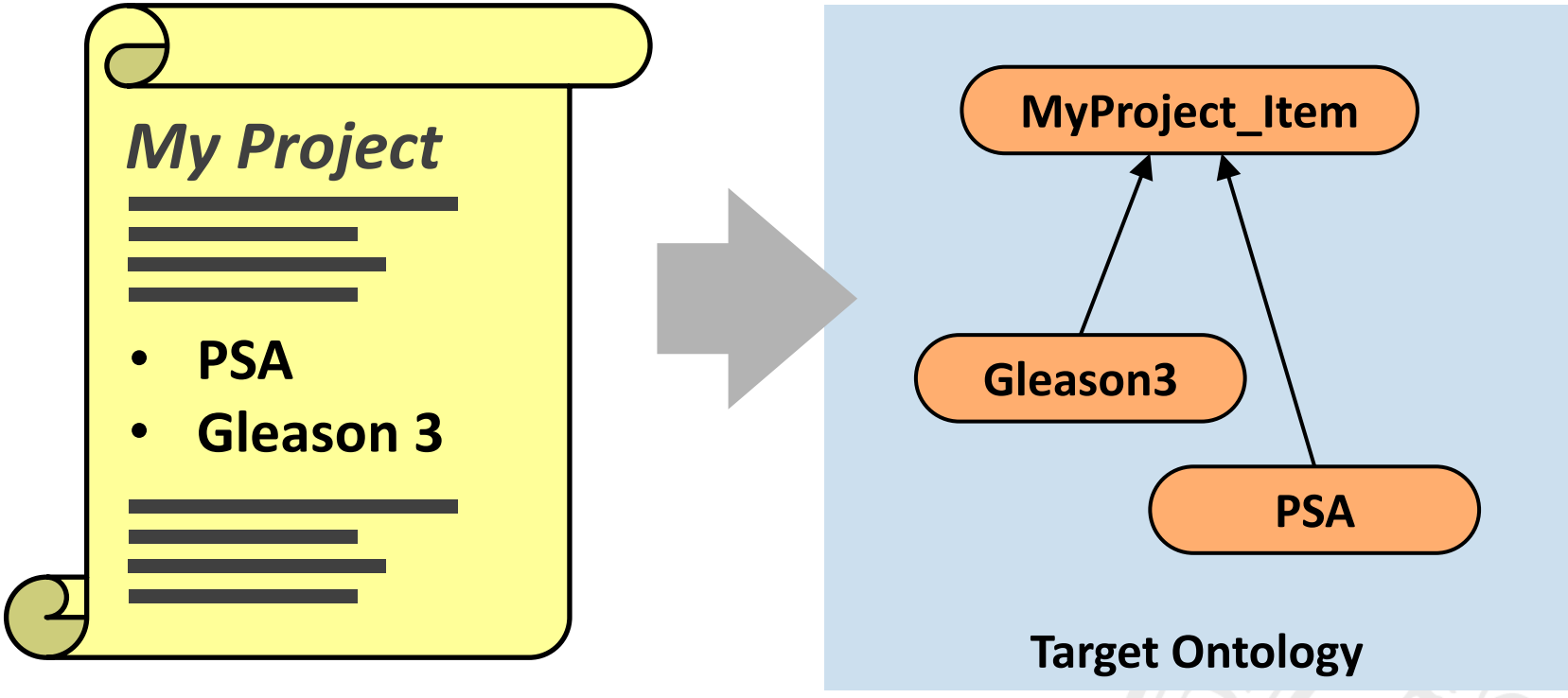
Our Ontolgy-Based Approach

A system for accessing heterogenious EMR data

- **Uses OWL ontologies to describe data providing source systems (EMR), target systems (i2b2) and the mappings in between**
  - Declarative, machine processible description of all information
  - Vision: future linkage with other biomedical ontologies / Semantic Web
- **Supports the annotation of typical relationships between CRF data elements in order to post-construct a hospital-wide ontology/MDR**
  - For Erlangen: post-annotation of relationships between 42,000 data elements
  - Operates on a generic EAV-like data model to support the easy processing of these relationships
- **Mappings can be „simple“ or „complex“ including support for full data transformations and filtering**
  - Theoretical support for all relational database systems and database operations
  - Mappings are transformed into SQL scripts to extract the data from the source systems (EMR) and export the data into the target system (here: i2b2)

Methods: Ontology Mapping

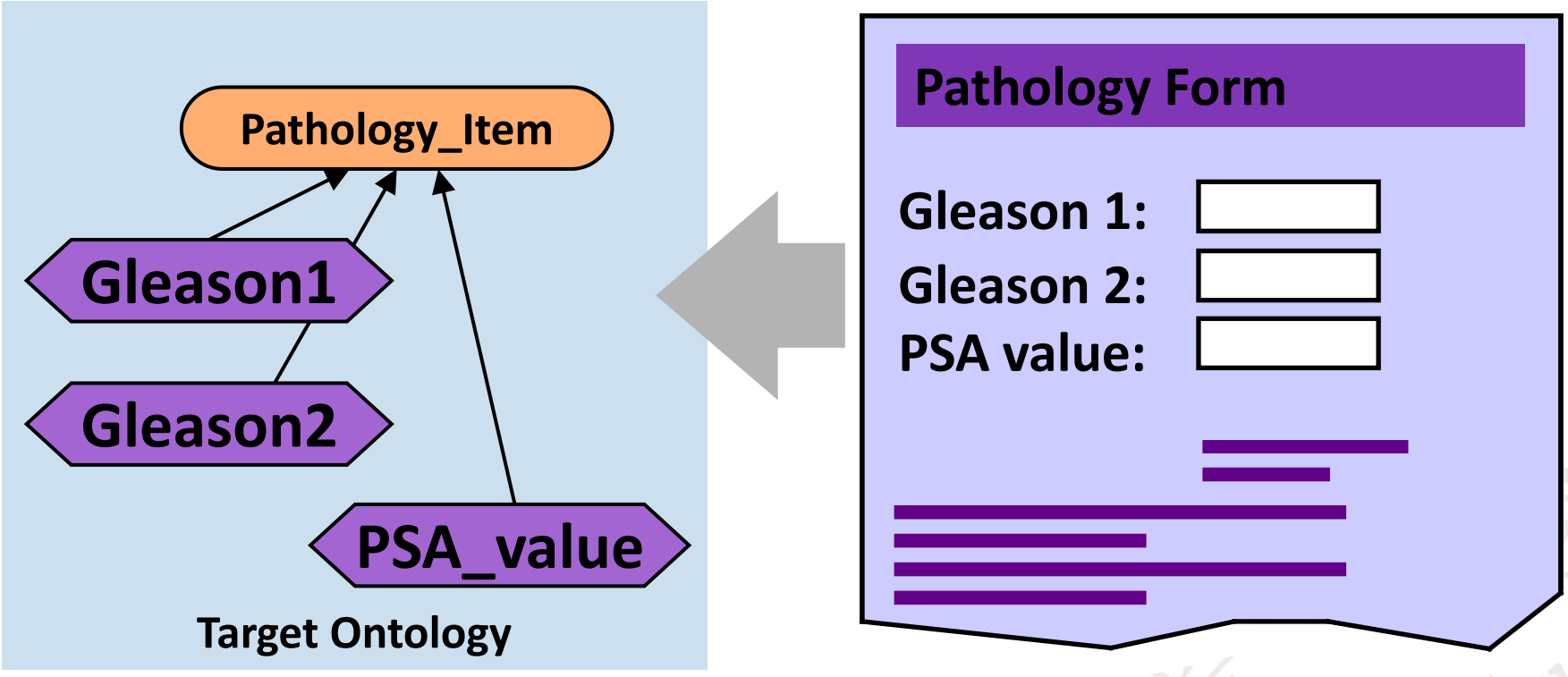
… with Semantic Web techniques



The target dataset (the data we want to use for research purposes) is expressed by a *target ontology*.

Methods: Ontology Mapping

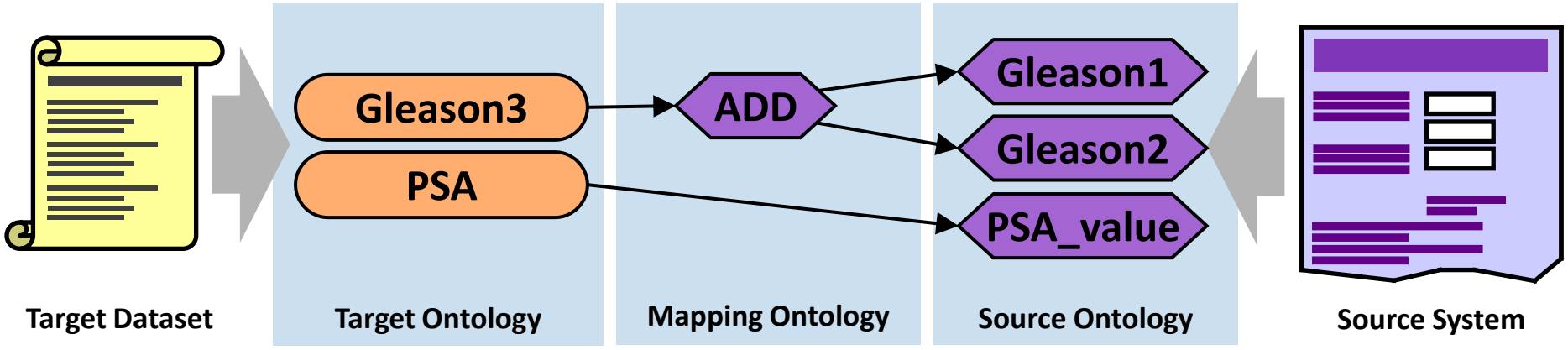
… with Semantic Web techniques



The source system (the data we have), i.e. the EMR, is expressed by a *source ontology*.

Methods: Ontology Mapping

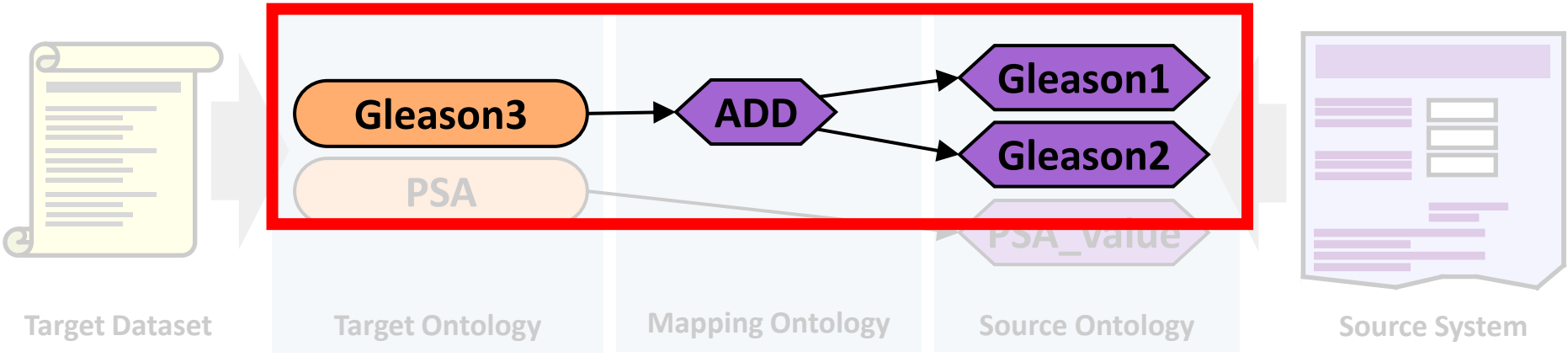
… with Semantic Web techniques



- We can then create a *mapping ontology* to connect concepts from the target ontology to concepts from the source ontology.
- Some concepts can be mapped directly,
- others might require transformations and/or data filtering, which can be expressed with intermediate nodes.

Methods: Ontology Mapping

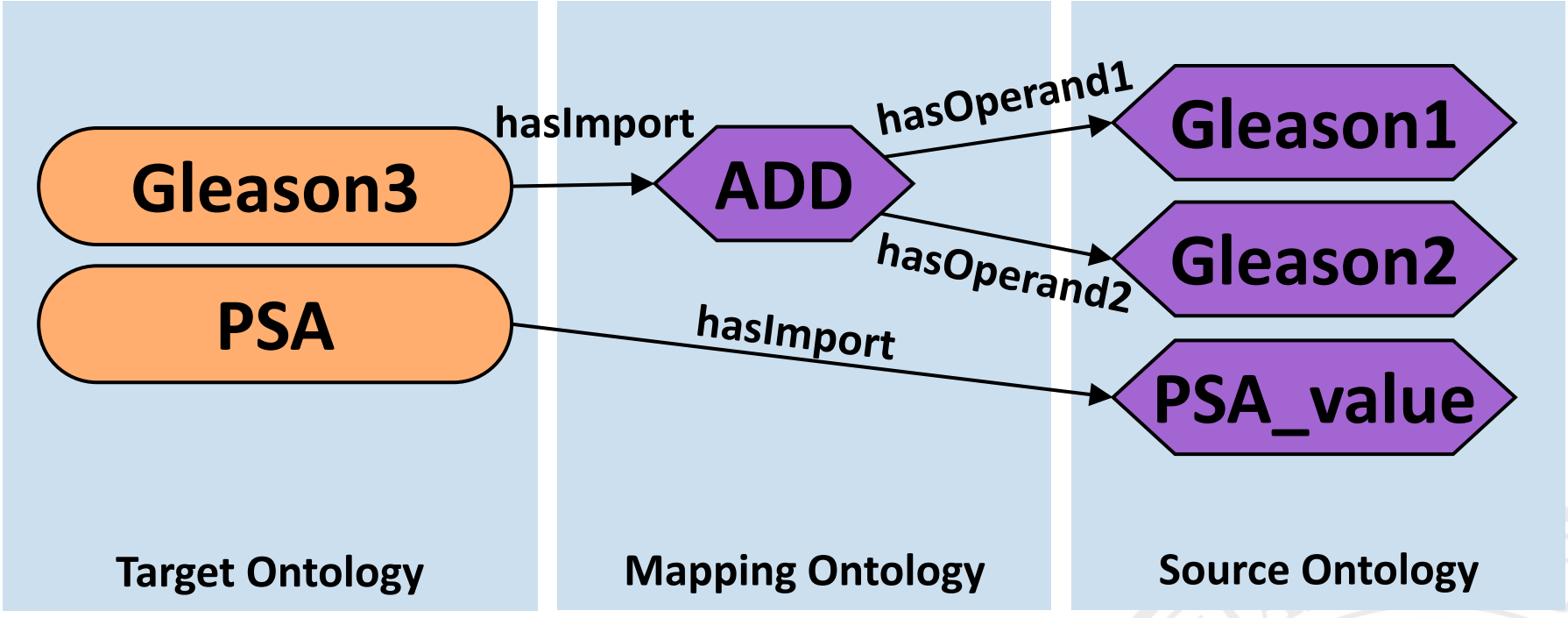
… with Semantic Web techniques



- **Administrative knowledge:** The data element „Gleason3“ is not captured in the source system (EMR), but it is needed to answer research questions.
- **Medical knowledge:** The Gleason Score 3 is the sum of Gleason Score 1 and 2.
- **Technical knowledge:** To export the data element „Gleason3“, you need to add the data elements „Gleason1“ and „Gleason2“.

Methods: Ontology Mapping

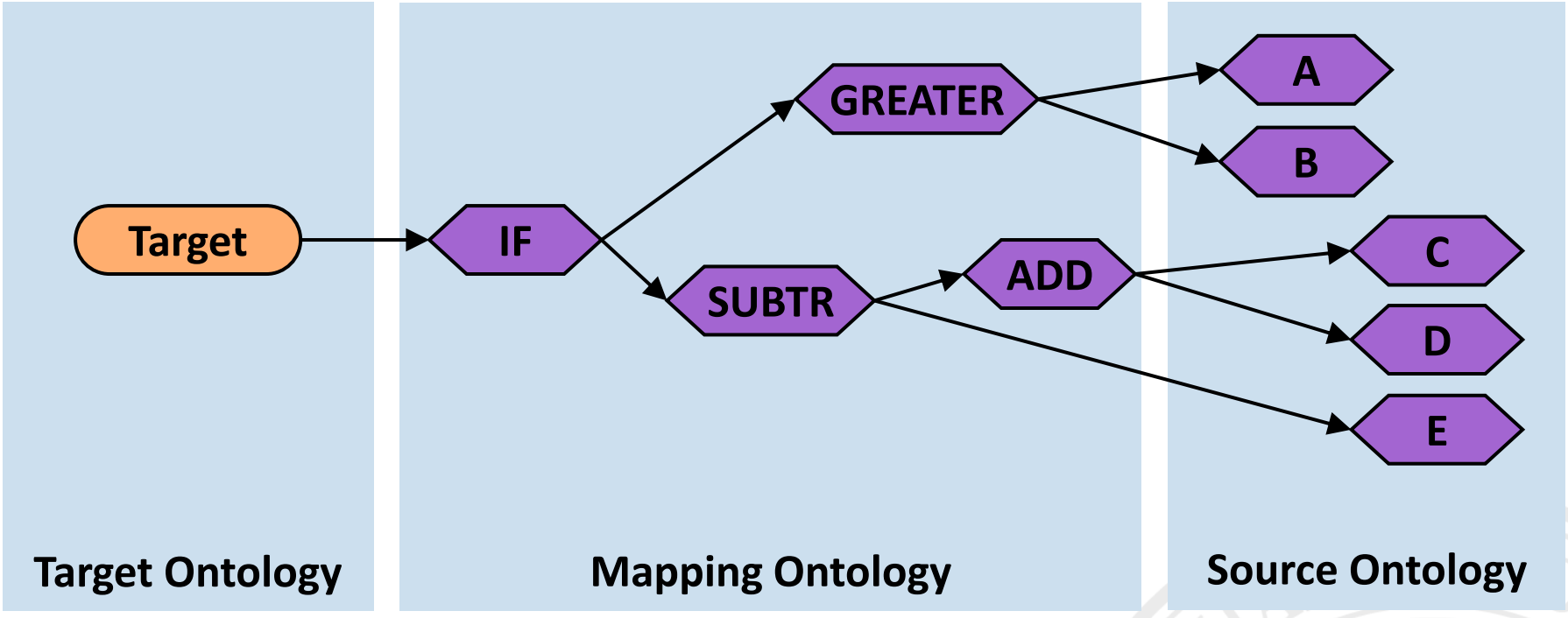
… with Semantic Web techniques



- Intermediate nodes are limited to two operands in order to keep the operations simple and „semantically atomic“.
- The distinction between the two relations *hasOperand1* and *hasOperand2* enables the correct processing order of non-commutative operations.

Methods: Ontology Mapping

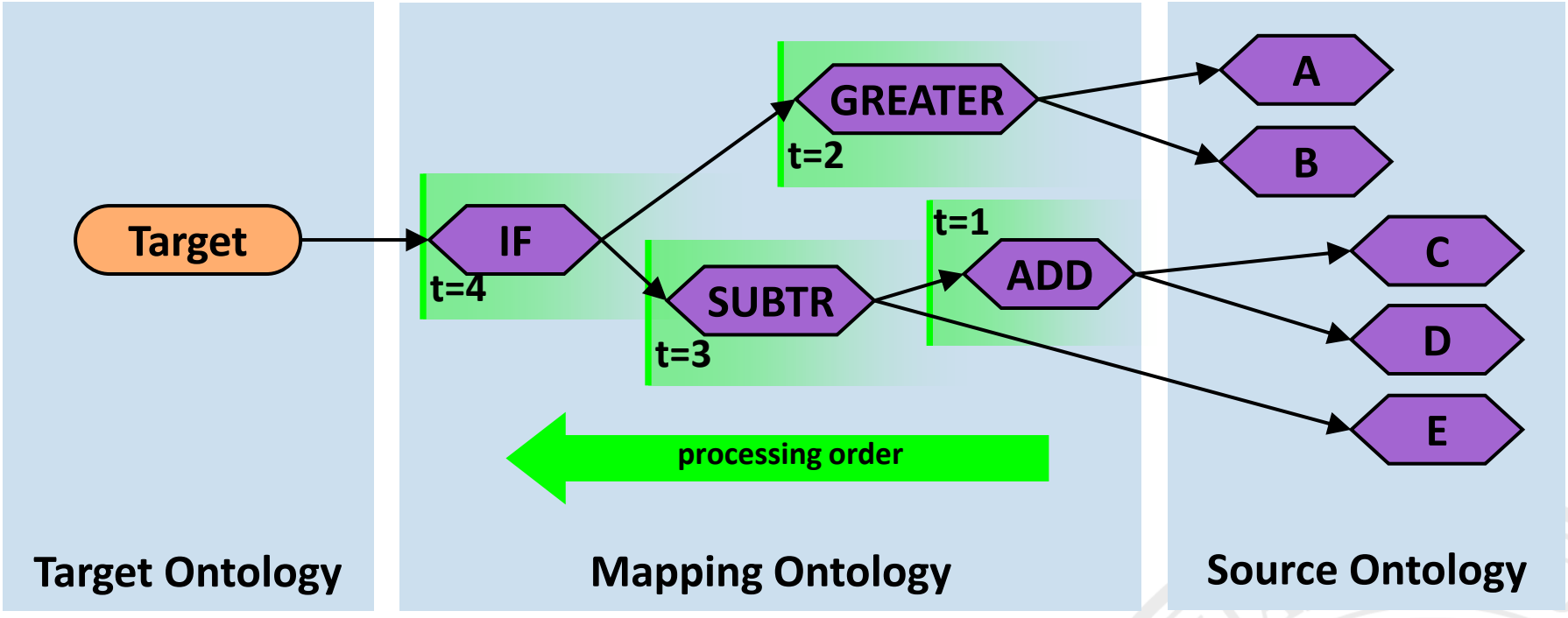
… with Semantic Web techniques



- This means: if A is greater than B, export (C + D) – E.
- QuickMapp prefix notation:

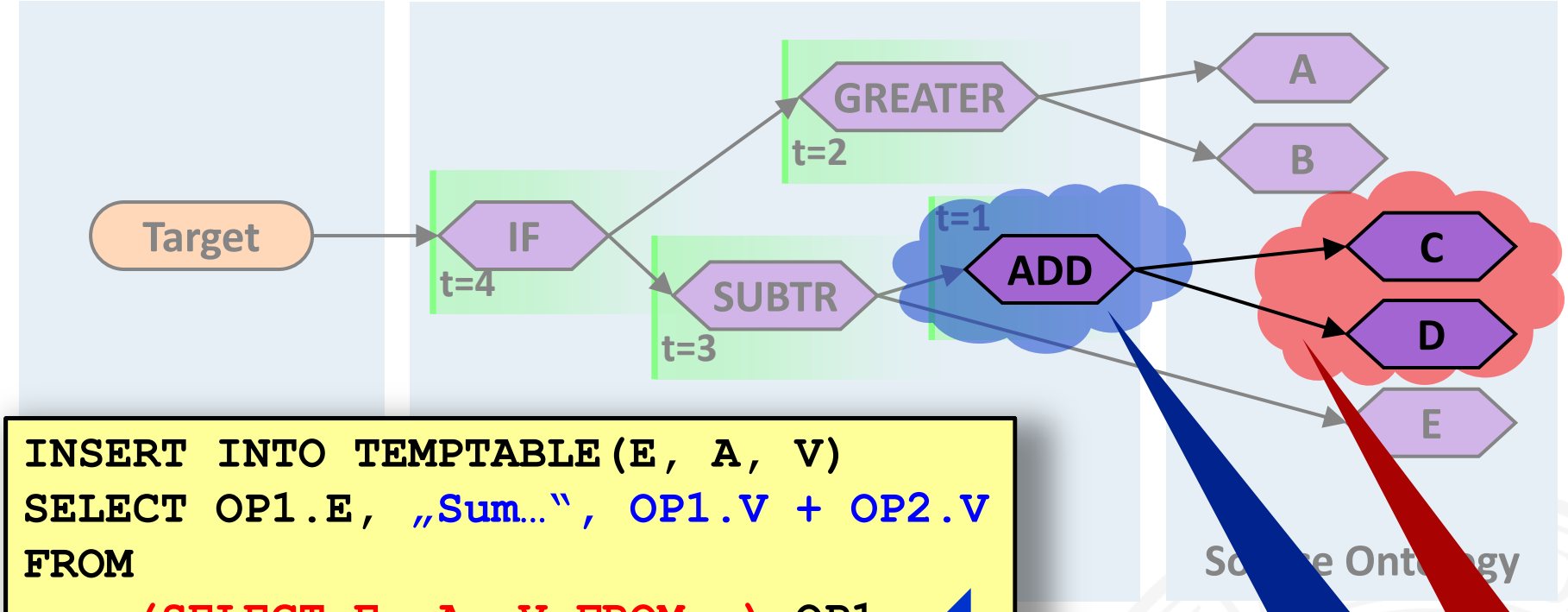
```
Target : IF (GREATER A B) (SUBTR (ADD C D) E)
```

Methods: Approach Overview



- To allow complex data transformations, nodes can be cascaded into „expression trees“.
- OWL-to-SQL translation: For each mapping node, an SQL statement, which performs the operation expressed by the node, can be constructed automatically .

Methods: OWL-2-SQL Translation



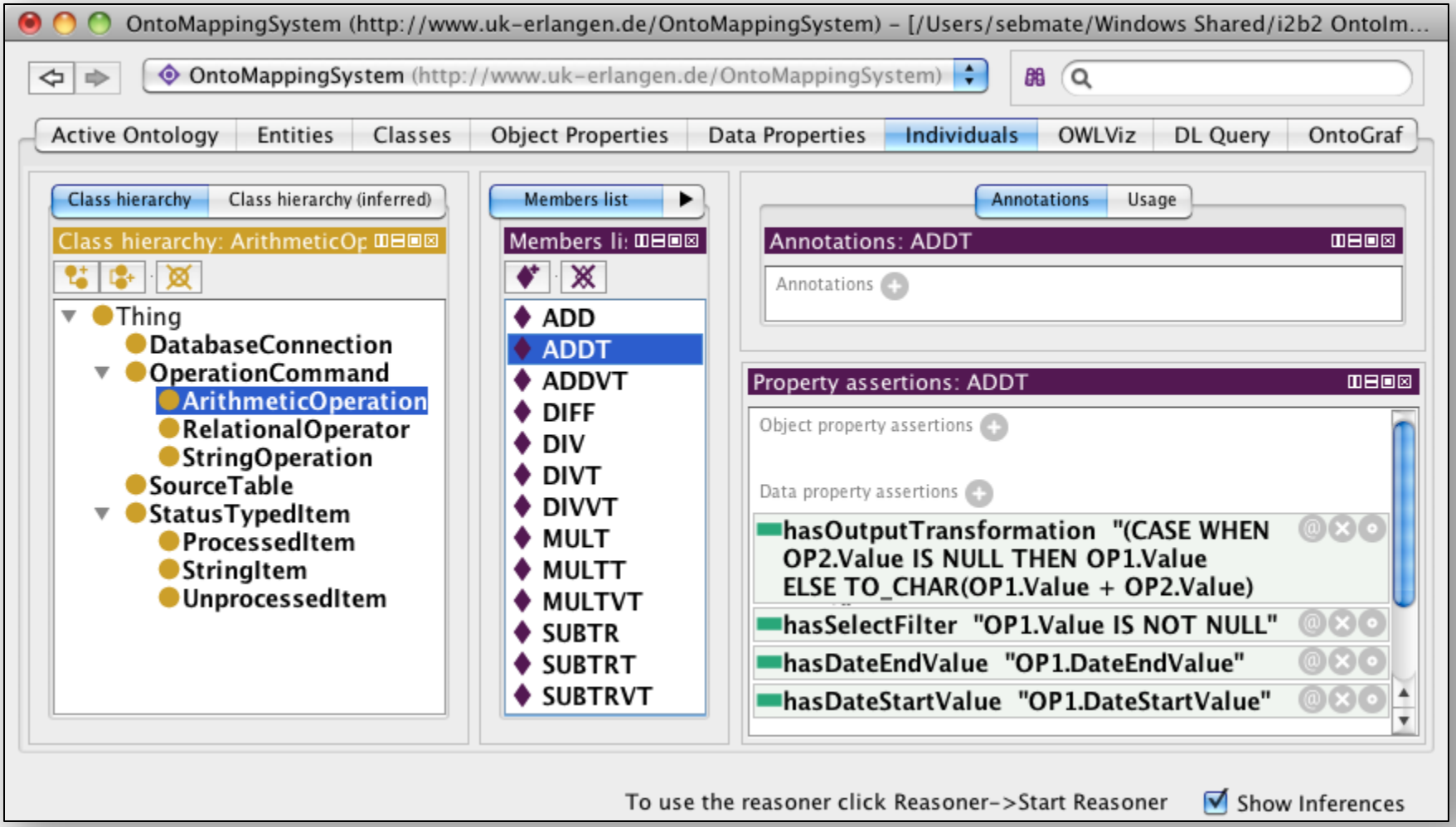

```
INSERT INTO TEMPTABLE(E, A, V)
SELECT OP1.E, „Sum…“, OP1.V + OP2.V
FROM
     (SELECT E, A, V FROM …) OP1
     FULL OUTER JOIN
     (SELECT E, A, V FROM …) OP2
     ON OP1.E = OP2.E
WHERE OP1.V IS NOT NULL AND …
```

How to process nodes

How to access data

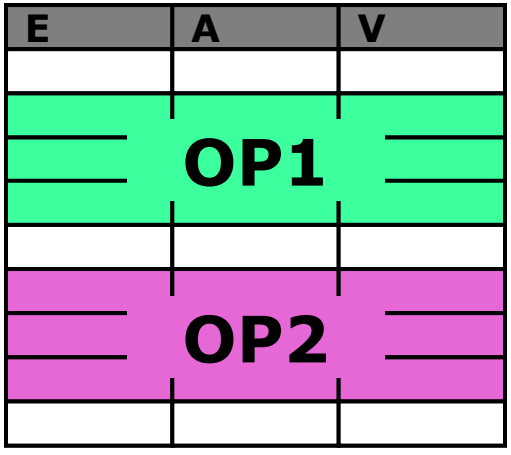
Data Export

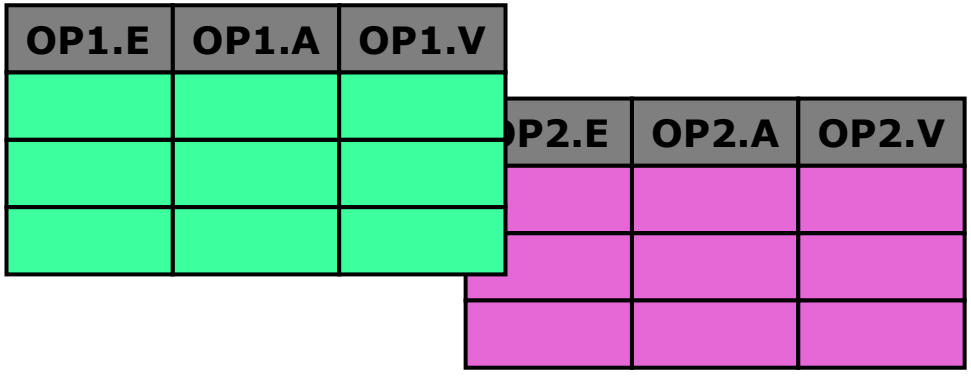
Information to construct SQL statements



Data Export

SQL statement construction





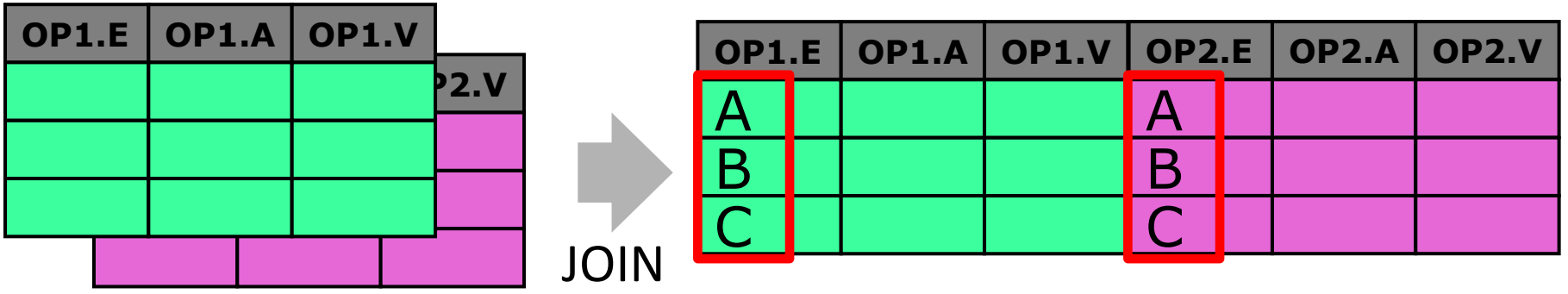

```
(SELECT E, A, V FROM …) OP1
(SELECT E, A, V FROM …) OP2
```

Fetch Operand 1 data

Fetch Operand 2 data

Data Export

SQL statement construction

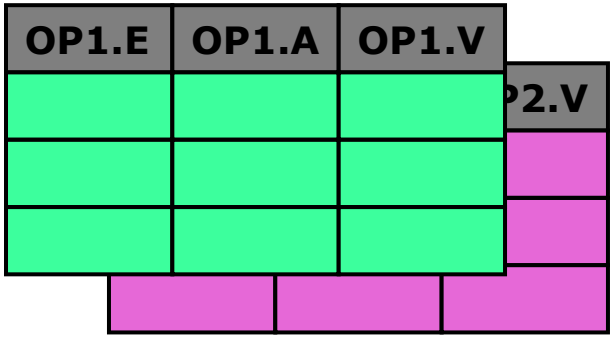



```
(SELECT E, A, V FROM …) OP1
FULL OUTER JOIN
(SELECT E, A, V FROM …) OP2
ON OP1.E = OP2.E
```

Join on entity (Patient)

Data Export

SQL statement construction



JOIN

OP1.E	OP1.A	OP1.V	OP2.E	OP2.A	OP2.V
A		2	A		2
B		4	B		5
C			C		4

```
SELECT OP1.E, „Sum…“, OP1.V + OP2.V
FROM
     (SELECT E, A, V FROM …) OP1
     FULL OUTER JOIN
     (SELECT E, A, V FROM …) OP2
     ON OP1.E = OP2.E
WHERE OP1.V IS NOT NULL AND …
```

Transformation

Filter criteria

Data Export

SQL statement construction

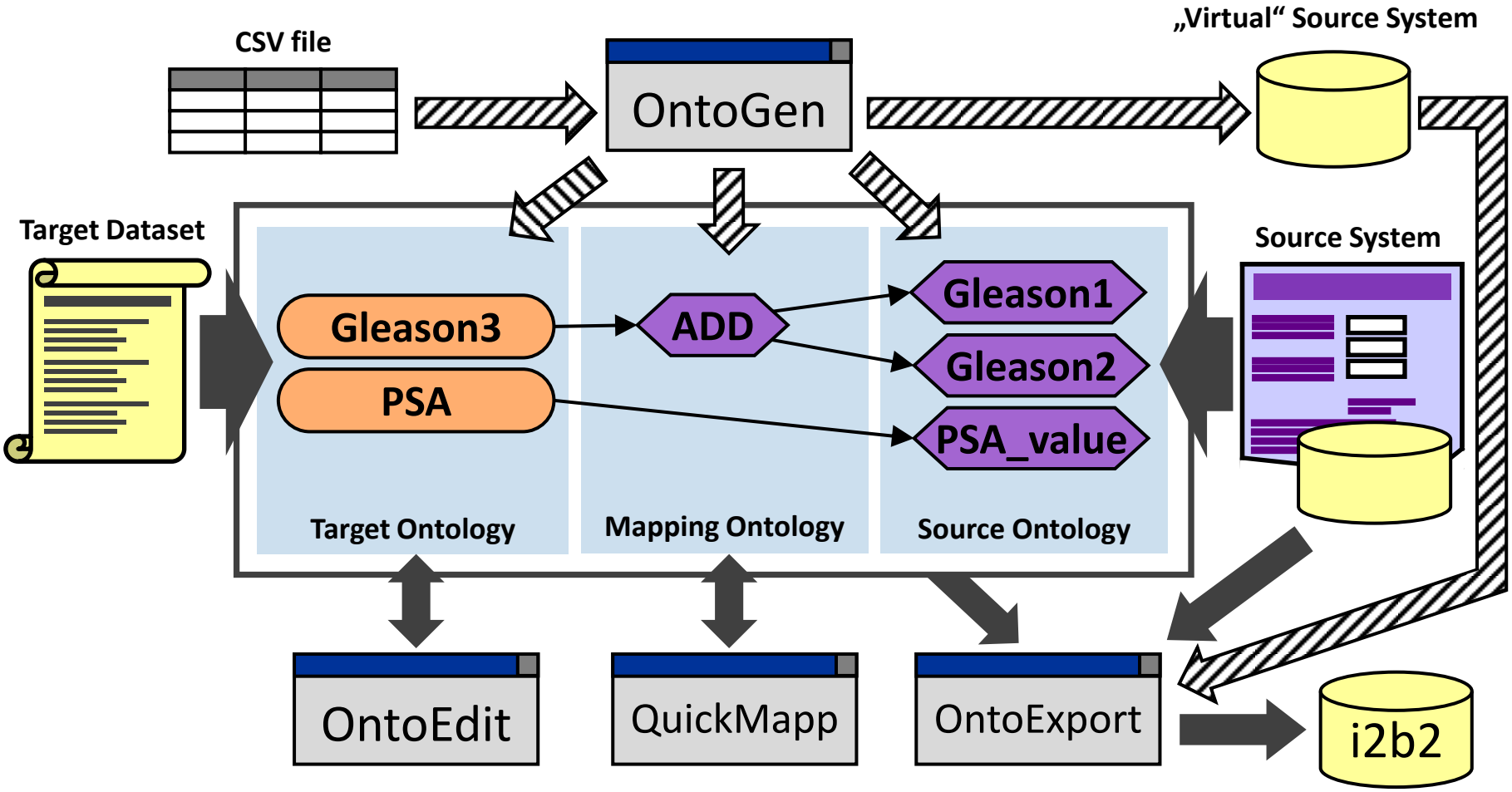
OP1.E	OP1.A	OP1.V	OP2.E	OP2.A	OP2.V
A		2	A		2
B		4	B		5
C			C		4

E	A	V
A	…	4
B	…	9

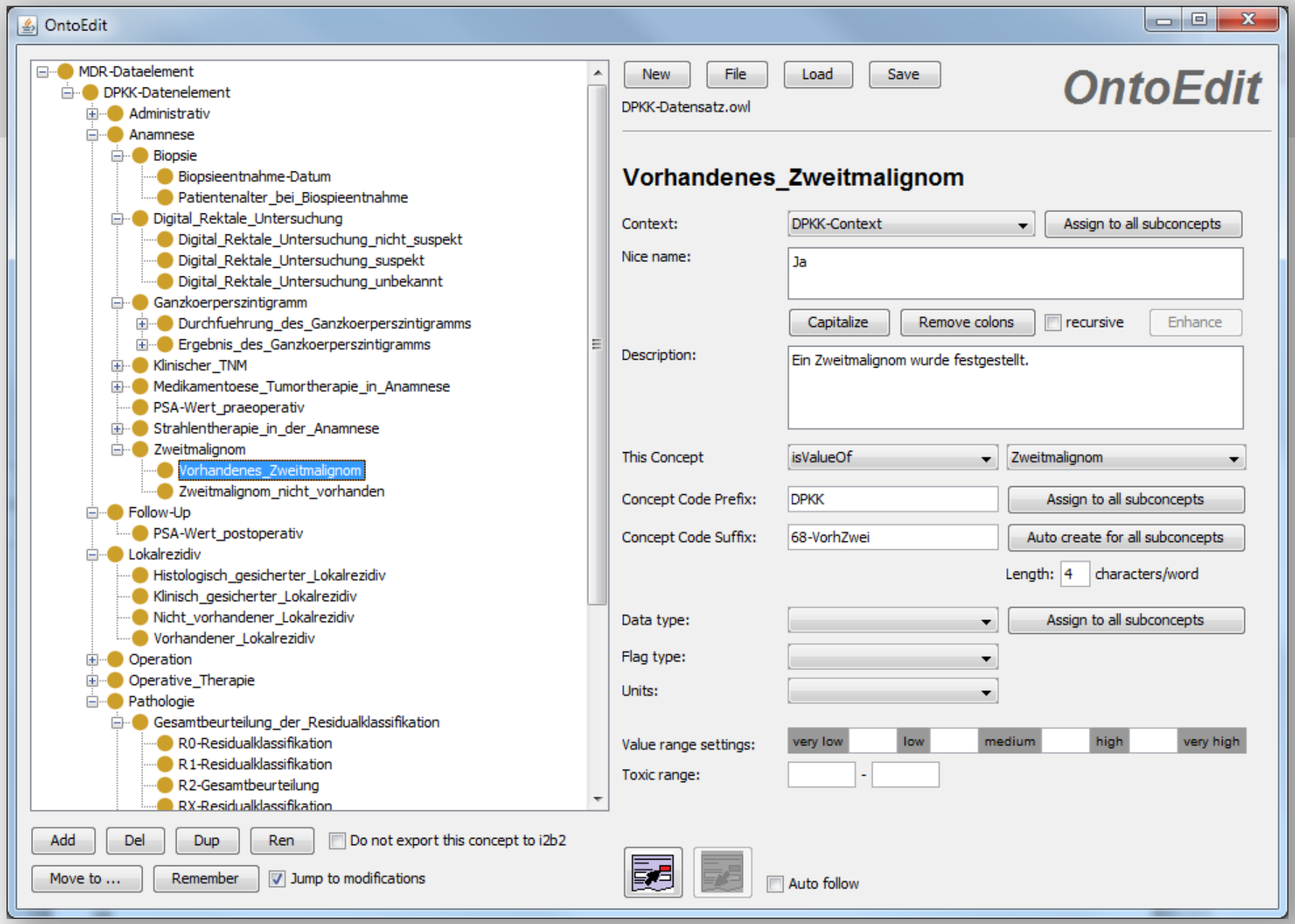
```
INSERT INTO TEMPTABLE(E, A, V)
SELECT OP1.E, „Sum…“, OP1.V + OP2.V
FROM
     (SELECT E, A, V FROM …) OP1
     FULL OUTER JOIN
     (SELECT E, A, V FROM …) OP2
     ON OP1.E = OP2.E
WHERE OP1.V IS NOT NULL AND …
```

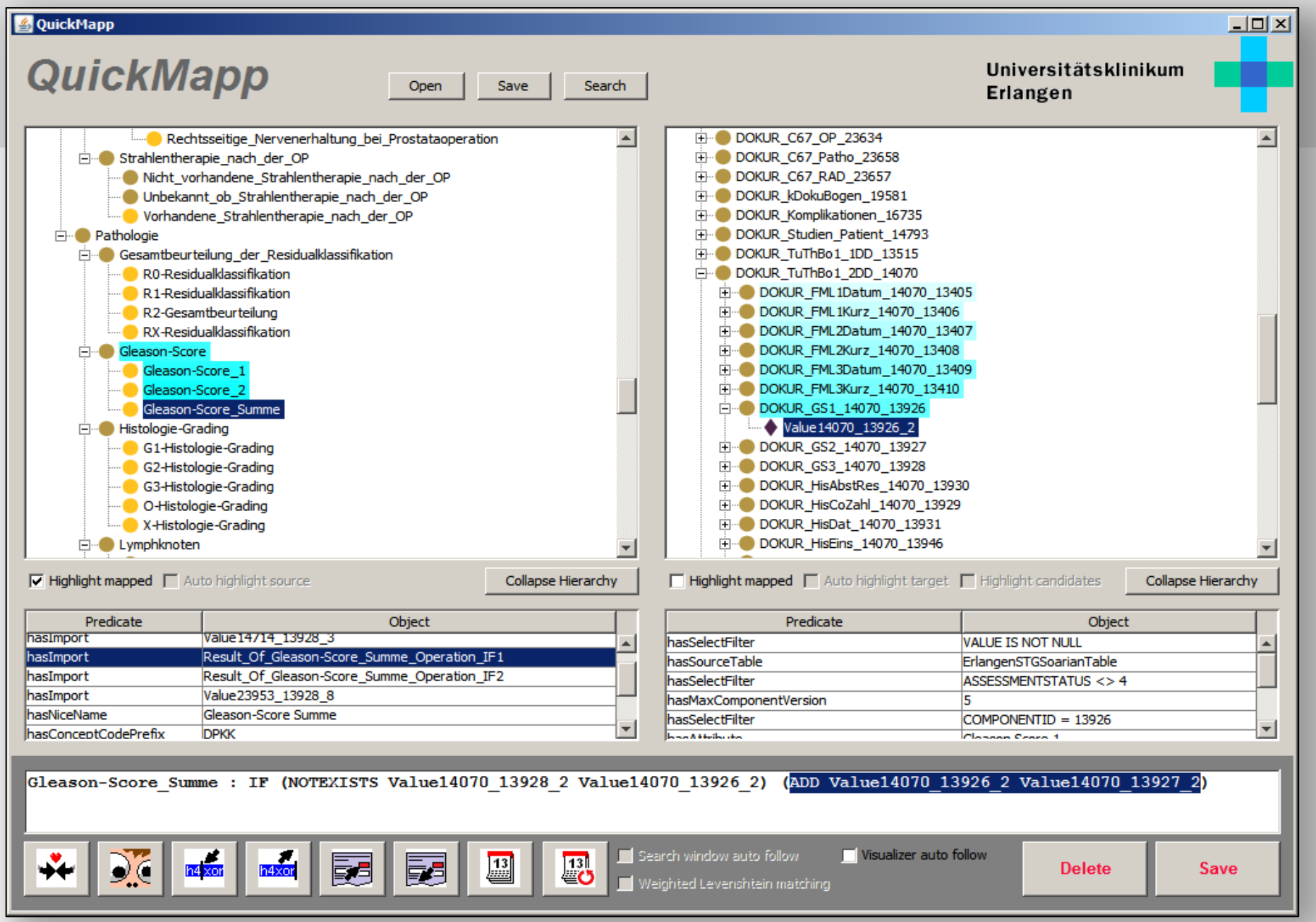
Write result into temporary table

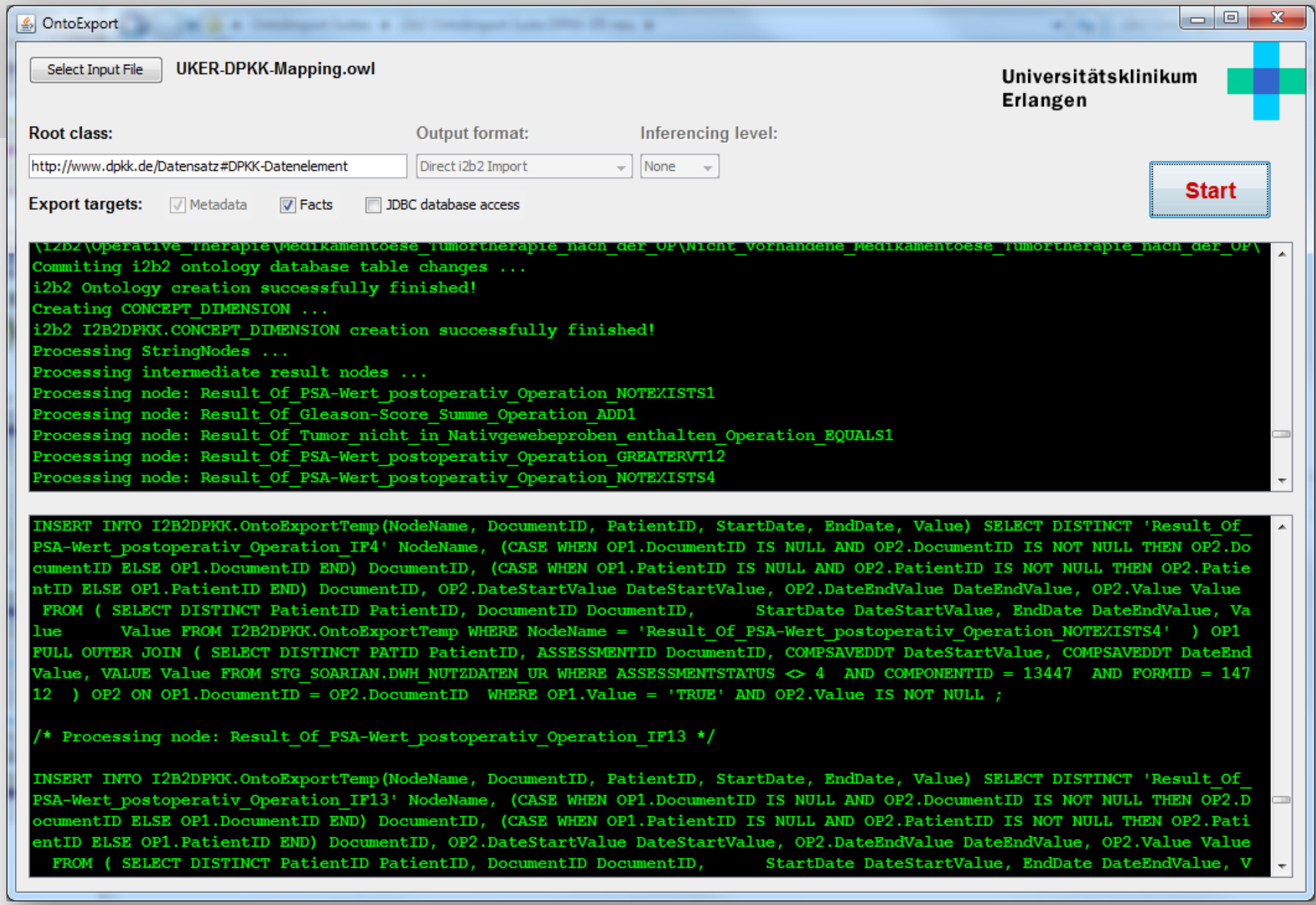
Summary and Tools

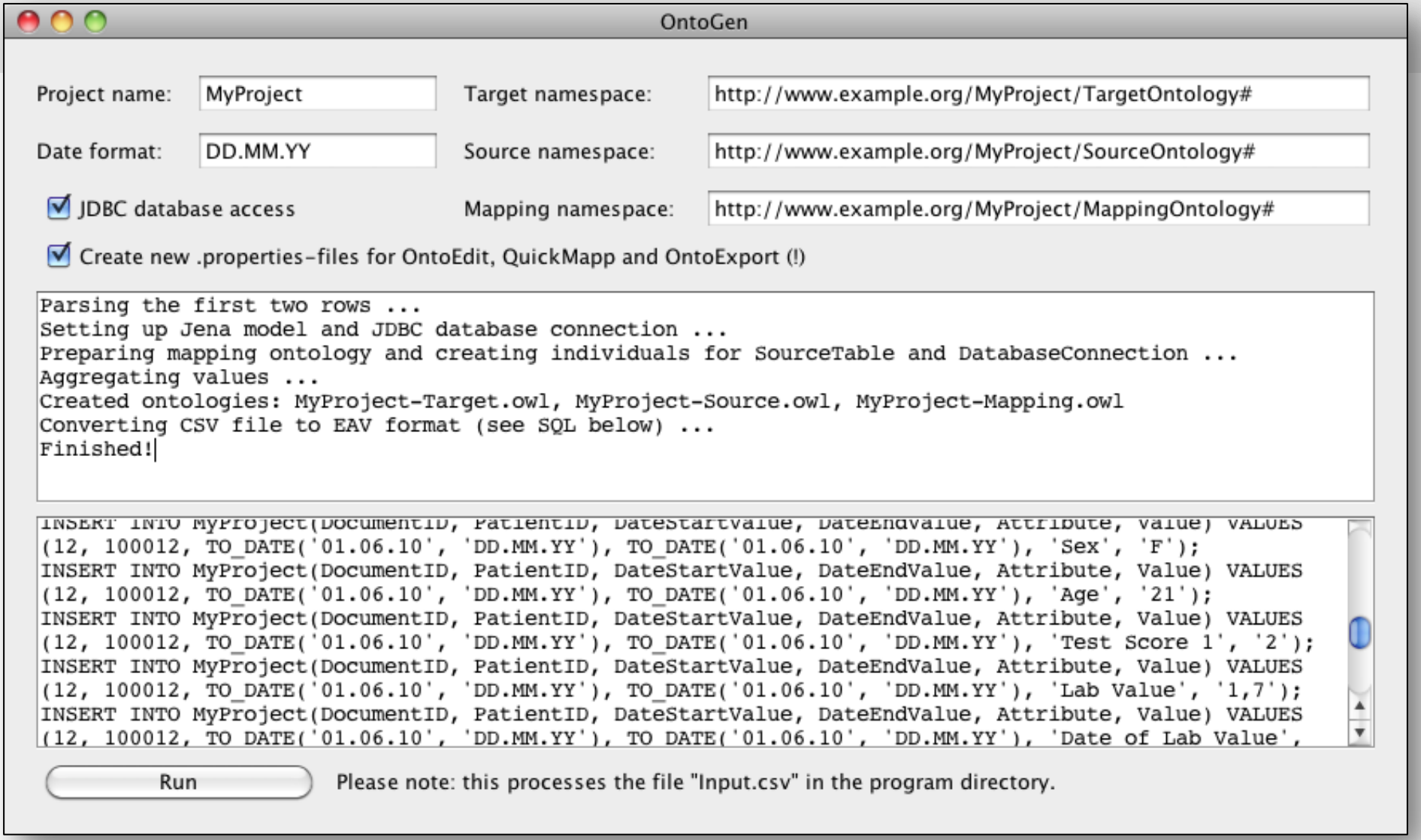


Alternative workflow to integrate CSV files if access to the source system is not available.


OntoEdit
MDR-Dataelement
DPKK-Datenelement
Administrativ
Anamnese
Biopsie
Biopsieentnahme-Datum
Patientenalter_bei_Biospieentnahme
Digital_Rektale_Untersuchung
Digital_Rektale_Untersuchung_nicht_suspekt
Digital_Rektale_Untersuchung_suspekt
Digital_Rektale_Untersuchung_unbekannt
Ganzkoerperszintigramm
Durchfuehrung_des_Ganzkoerperszintigramms
Ergebnis_des_Ganzkoerperszintigramms
Klinischer_TNM
Medikamentoese_Tumortherapie_in_Anamnese
PSA-Wert_praeoperativ
Strahlentherapie_in_der_Anamnese
Zweitmalignom
Vorhandenes_Zweitmalignom
Zweitmalignom_nicht_vorhanden
Follow-Up
PSA-Wert_postoperativ
Lokalrezidiv
Histologisch_gesicherter_Lokalrezidiv
Klinisch_gesicherter_Lokalrezidiv
Nicht_vorhandener_Lokalrezidiv
Vorhandener_Lokalrezidiv
Operation
Operative_Therapie
Pathologie
Gesamtbeurteilung_der_Residualklassifikation
R0-Residualklassifikation
R1-Residualklassifikation
R2-Gesamtbeurteilung
RX-Residualklassifikation
Add
Del
Dup
Ren
Do not export this concept to i2b2
Move to ...
Remember
Jump to modifications
New
File
Load
Save
DPKK-Datensatz.owl
OntoEdit
Vorhandenes_Zweitmalignom
Context:
DPKK-Context
Assign to all subconcepts
Nice name:
Ja
Capitalize
Remove colons
recursive
Enhance
Description:
Ein Zweitmalignom wurde festgestellt.
This Concept
isValueOf
Zweitmalignom
Concept Code Prefix:
DPKK
Assign to all subconcepts
Concept Code Suffix:
68-VorhZwei
Auto create for all subconcepts
Length:
4
characters/word
Data type:
Assign to all subconcepts
Flag type:
Units:
Value range settings:
very low
low
medium
high
very high
Toxic range:
Auto follow

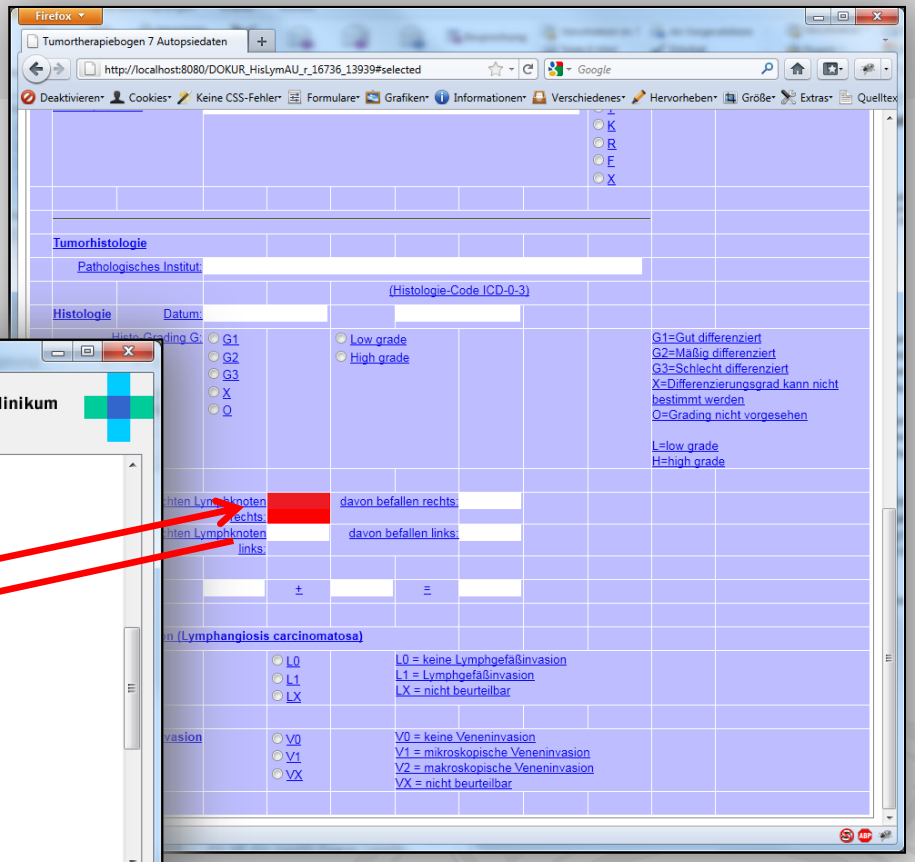

QuickMapp
QuickMapp
Open
Save
Search
Universitätsklinikum Erlangen
Rechtsseitige_Nervenerhaltung_bei_Prostataoperation
Strahlentherapie_nach_der_OP
Nicht_vorhandene_Strahlentherapie_nach_der_OP
Unbekannt_ob_Strahlentherapie_nach_der_OP
Vorhandene_Strahlentherapie_nach_der_OP
Pathologie
Gesamtbeurteilung_der_Residualklassifikation
R0-Residualklassifikation
R1-Residualklassifikation
R2-Gesamtbeurteilung
RX-Residualklassifikation
Gleason-Score
Gleason-Score_1
Gleason-Score_2
Gleason-Score_Summe
Histologie-Grading
G1-Histologie-Grading
G2-Histologie-Grading
G3-Histologie-Grading
O-Histologie-Grading
X-Histologie-Grading
Lymphknoten
DOKUR_C67_OP_23634
DOKUR_C67_Patho_23658
DOKUR_C67_RAD_23657
DOKUR_kDokuBogen_19581
DOKUR_Komplikationen_16735
DOKUR_Studien_Patient_14793
DOKUR_TuThBo1_1DD_13515
DOKUR_TuThBo1_2DD_14070
DOKUR_FML1Datum_14070_13405
DOKUR_FML1Kurz_14070_13406
DOKUR_FML2Datum_14070_13407
DOKUR_FML2Kurz_14070_13408
DOKUR_FML3Datum_14070_13409
DOKUR_FML3Kurz_14070_13410
DOKUR_GS1_14070_13926
Value14070_13926_2
DOKUR_GS2_14070_13927
DOKUR_GS3_14070_13928
DOKUR_HisAbstRes_14070_13930
DOKUR_HisCoZahl_14070_13929
DOKUR_HisDat_14070_13931
DOKUR_HisEins_14070_13946
Highlight mapped
Auto highlight source
Collapse Hierarchy
Highlight mapped
Auto highlight target
Highlight candidates
Collapse Hierarchy
Predicate
Object
hasImport
Value14714_13928_3
hasImport
Result_Of_Gleason-Score_Summe_Operation_IF1
hasImport
Result_Of_Gleason-Score_Summe_Operation_IF2
hasImport
Value23953_13928_8
hasNiceName
Gleason-Score Summe
hasConceptCodePrefix
DPKK
Predicate
Object
hasSelectFilter
VALUE IS NOT NULL
hasSourceTable
ErlangenSTGSoarianTable
hasSelectFilter
ASSESSMENTSTATUS <> 4
hasMaxComponentVersion
5
hasSelectFilter
COMPONENTID = 13926
Gleason-Score_Summe : IF (NOTEXISTS Value14070_13928_2 Value14070_13926_2) (ADD Value14070_13926_2 Value14070_13927_2)
h4xor
h4xor
Search window auto follow
Visualizer auto follow
Weighted Levenshtein matching
Delete
Save

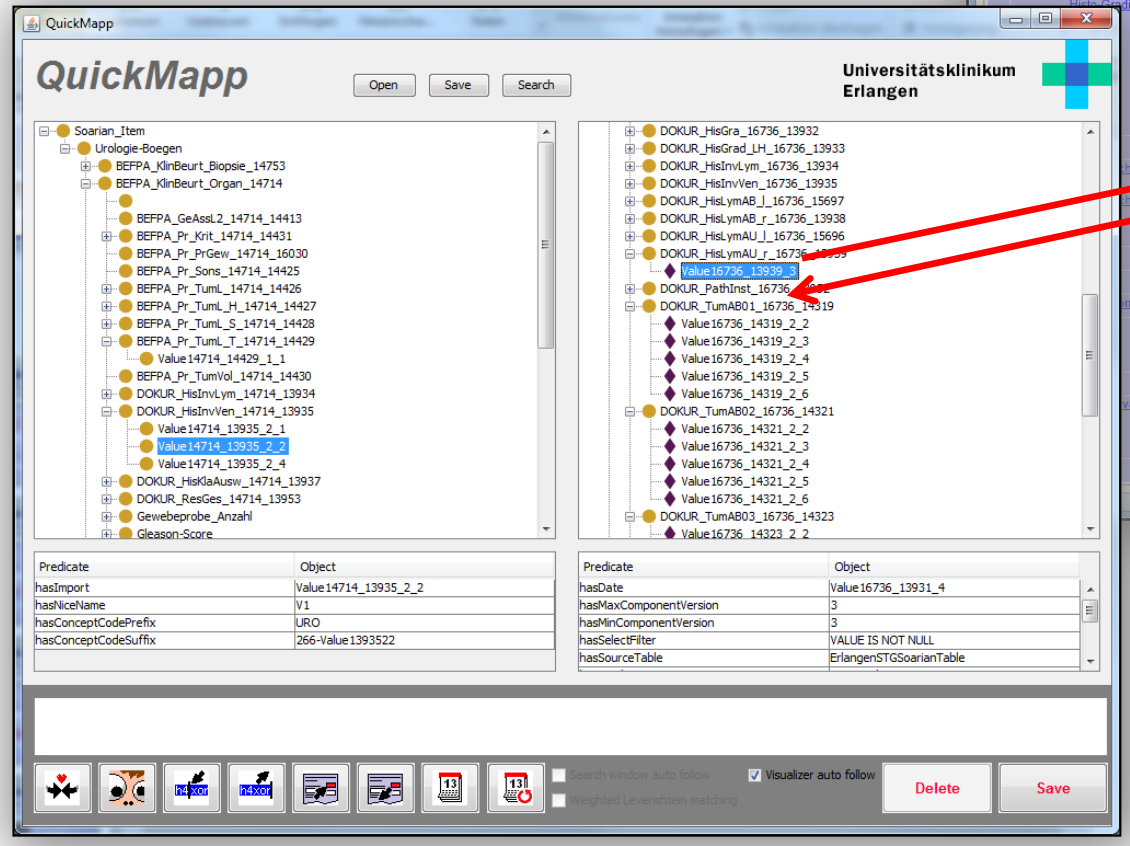

OntoExport
Select Input File
UKER-DPKK-Mapping.owl
Universitätsklinikum Erlangen
Root class:
Output format:
Inferencing level:
http://www.dpkk.de/Datensatz#DPKK-Datenelement
Direct i2b2 Import
None
Start
Export targets:
Metadata
Facts
JDBC database access
Commiting i2b2 ontology database table changes ...
i2b2 Ontology creation successfully finished!
Creating CONCEPT_DIMENSION ...
i2b2 I2B2DPKK.CONCEPT_DIMENSION creation successfully finished!
Processing StringNodes ...
Processing intermediate result nodes ...
Processing node: Result_Of_PSA-Wert_postoperativ_Operation_NOTEXISTS1
Processing node: Result_Of_Gleason-Score_Summe_Operation_ADD1
Processing node: Result_Of_Tumor_nicht_in_Nativgewebeproben_enthalten_Operation_EQUALS1
Processing node: Result_Of_PSA-Wert_postoperativ_Operation_GREATERVT12
Processing node: Result_Of_PSA-Wert_postoperativ_Operation_NOTEXISTS4
INSERT INTO I2B2DPKK.OntoExportTemp(NodeName, DocumentID, PatientID, StartDate, EndDate, Value) SELECT DISTINCT 'Result_Of_PSA-Wert_postoperativ_Operation_IF4' NodeName, (CASE WHEN OP1.DocumentID IS NULL AND OP2.DocumentID IS NOT NULL THEN OP2.DocumentID ELSE OP1.DocumentID END) DocumentID, (CASE WHEN OP1.PatientID IS NULL AND OP2.PatientID IS NOT NULL THEN OP2.PatientID ELSE OP1.PatientID END) DocumentID, OP2.DateStartValue DateStartValue, OP2.DateEndValue DateEndValue, OP2.Value Value FROM (SELECT DISTINCT PatientID PatientID, DocumentID DocumentID, StartDate DateStartValue, EndDate DateEndValue, Value Value FROM I2B2DPKK.OntoExportTemp WHERE NodeName = 'Result_Of_PSA-Wert_postoperativ_Operation_NOTEXISTS4') OP1 FULL OUTER JOIN (SELECT DISTINCT PATID PatientID, ASSESSMENTID DocumentID, COMPSAVEDDT DateStartValue, COMPSAVEDDT DateEndValue, VALUE Value FROM STG_SOARIAN.DWH_NUTZDATEN_UR WHERE ASSESSMENTSTATUS <> 4 AND COMPONENTID = 13447 AND FORMID = 14712) OP2 ON OP1.DocumentID = OP2.DocumentID WHERE OP1.Value = 'TRUE' AND OP2.Value IS NOT NULL ;
/* Processing node: Result_Of_PSA-Wert_postoperativ_Operation_IF13 */
INSERT INTO I2B2DPKK.OntoExportTemp(NodeName, DocumentID, PatientID, StartDate, EndDate, Value) SELECT DISTINCT 'Result_Of_PSA-Wert_postoperativ_Operation_IF13' NodeName, (CASE WHEN OP1.DocumentID IS NULL AND OP2.DocumentID IS NOT NULL THEN OP2.DocumentID ELSE OP1.DocumentID END) DocumentID, (CASE WHEN OP1.PatientID IS NULL AND OP2.PatientID IS NOT NULL THEN OP2.PatientID ELSE OP1.PatientID END) DocumentID, OP2.DateStartValue DateStartValue, OP2.DateEndValue DateEndValue, OP2.Value Value FROM (SELECT DISTINCT PatientID PatientID, DocumentID DocumentID, StartDate DateStartValue, EndDate DateEndValue, V

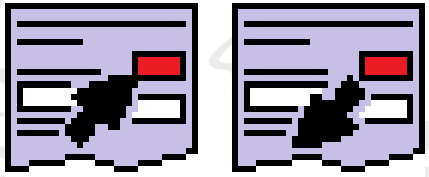

OntoGen
Project name:
MyProject
Target namespace:
http://www.example.org/MyProject/TargetOntology#
Date format:
DD.MM.YY
Source namespace:
http://www.example.org/MyProject/SourceOntology#
JDBC database access
Mapping namespace:
http://www.example.org/MyProject/MappingOntology#
Create new .properties-files for OntoEdit, QuickMapp and OntoExport (!)
Parsing the first two rows ...
Setting up Jena model and JDBC database connection ...
Preparing mapping ontology and creating individuals for SourceTable and DatabaseConnection ...
Aggregating values ...
Created ontologies: MyProject-Target.owl, MyProject-Source.owl, MyProject-Mapping.owl
Converting CSV file to EAV format (see SQL below) ...
Finished!
INSERT INTO MyProject(DocumentID, PatientID, DateStartValue, DateEndValue, Attribute, Value) VALUES
(12, 100012, TO_DATE('01.06.10', 'DD.MM.YY'), TO_DATE('01.06.10', 'DD.MM.YY'), 'Sex', 'F');
INSERT INTO MyProject(DocumentID, PatientID, DateStartValue, DateEndValue, Attribute, Value) VALUES
(12, 100012, TO_DATE('01.06.10', 'DD.MM.YY'), TO_DATE('01.06.10', 'DD.MM.YY'), 'Age', '21');
INSERT INTO MyProject(DocumentID, PatientID, DateStartValue, DateEndValue, Attribute, Value) VALUES
(12, 100012, TO_DATE('01.06.10', 'DD.MM.YY'), TO_DATE('01.06.10', 'DD.MM.YY'), 'Test Score 1', '2');
INSERT INTO MyProject(DocumentID, PatientID, DateStartValue, DateEndValue, Attribute, Value) VALUES
(12, 100012, TO_DATE('01.06.10', 'DD.MM.YY'), TO_DATE('01.06.10', 'DD.MM.YY'), 'Lab Value', '1,7');
INSERT INTO MyProject(DocumentID, PatientID, DateStartValue, DateEndValue, Attribute, Value) VALUES
(12, 100012, TO_DATE('01.06.10', 'DD.MM.YY'), TO_DATE('01.06.10', 'DD.MM.YY'), 'Date of Lab Value',
Run
Please note: this processes the file "Input.csv" in the program directory.

Soarian Form Rendering Service

Ontology & Form Navigation

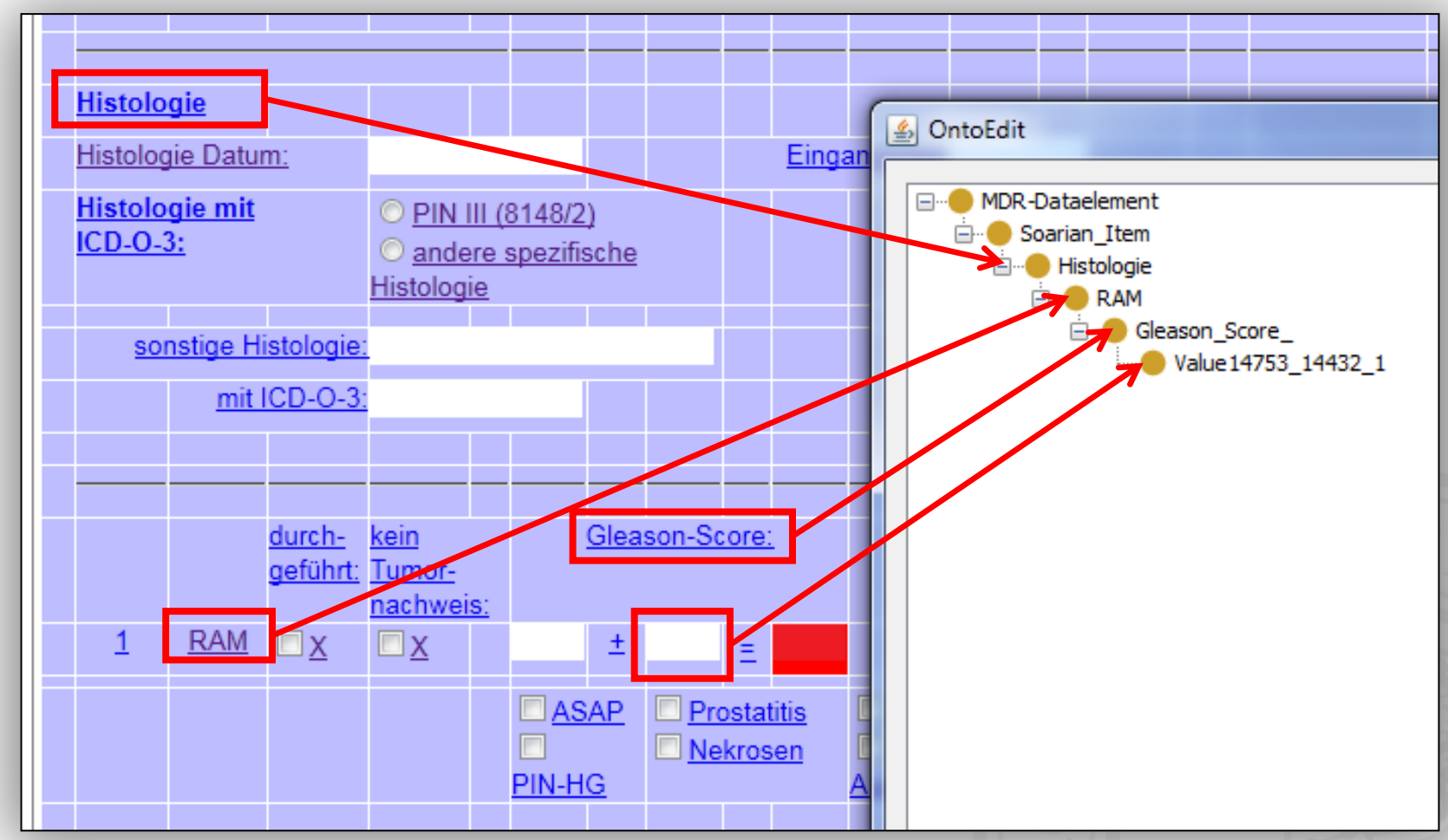






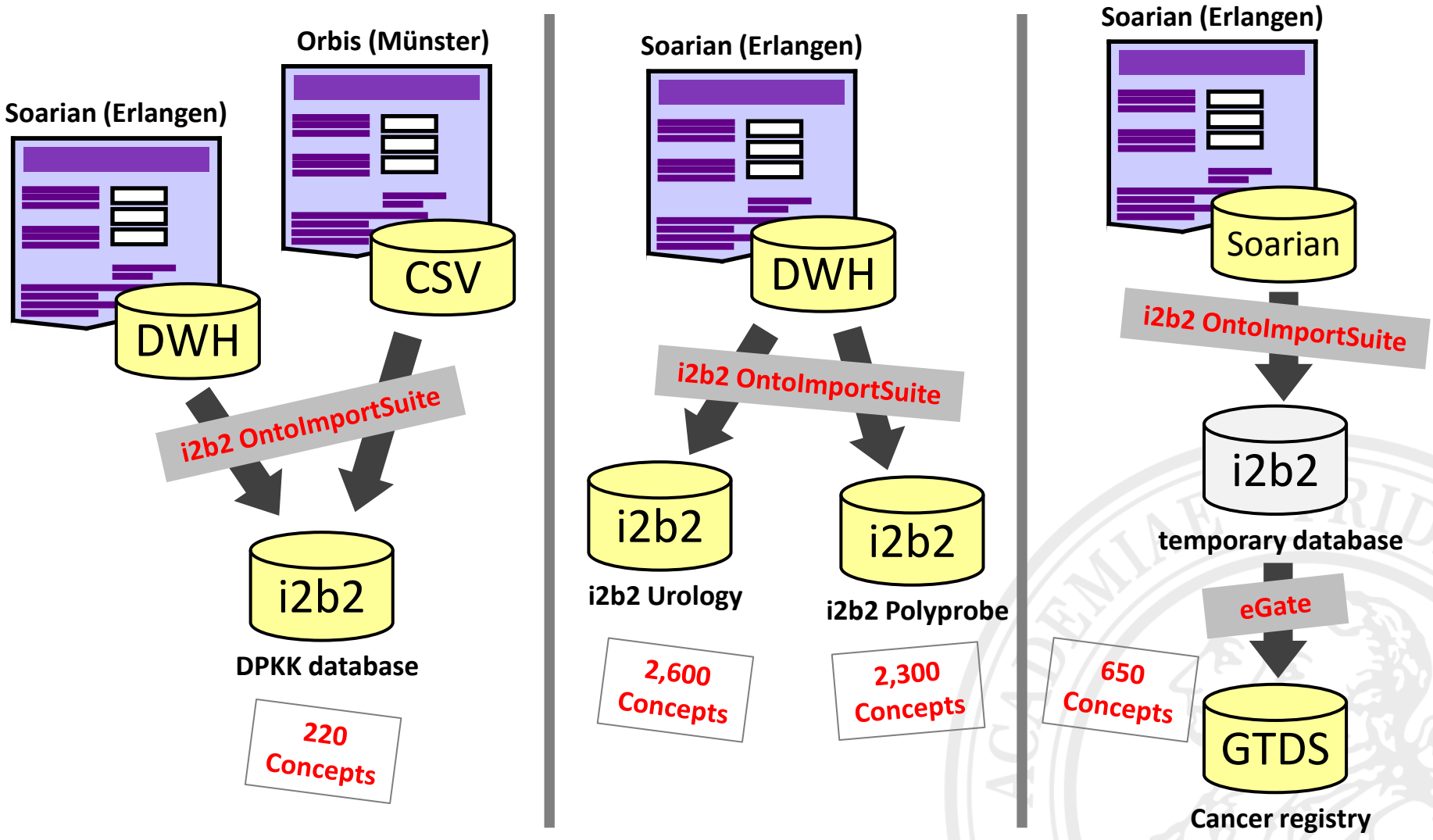
Soarian Form Rendering Service

Drag&drop ontology construction



Results

Three different scenarios (numbers are outdated)



Results: Ontology Mapping for DPKK

Results between Münster and Erlangen

- Preliminary results for Münster and Erlangen (166 DPKK concepts, also see MIE2011 paper):

	Erlangen Hospital	Münster Hospital
Directly mapped (*hasImport* only) concepts:	138	127
Through transformations mapped concepts:	10 (4 required a small trick)	1
Concept is not documented in source system:	15	36
Currently impossible / impractical mappings:	1 / 2	0 / 2
Generated SQL statements / execution time:	548 / ~15 seconds	284 / ~ 3 seconds
Number of facts / patients in source table:	29,721,416 / 161,512	5,100 / 500 (test data)
Obtained facts / patients for DPKK i2b2:	3,686 / 155	2,585 / 487 (test data)

Discussion

The Future of this Project

- **TODO/Vision: split into Client/Server architecture**
  - Server stores only one large ontology for central knowledge management
  - Simplifies administrative aspects (who is allowed to edit the ontology?)

- **Optimizations**
  - Improve processing order of intermediate nodes (similar to compiler construction)
  - Integration with Talend OpenStudio to simplify support for multiple database systems? (Project with University of Göttingen)
  - Use various tricks to let the DMBS do its own SQL optimizations
  - Evaluate „super nodes“ concept (graph-based optimizations)

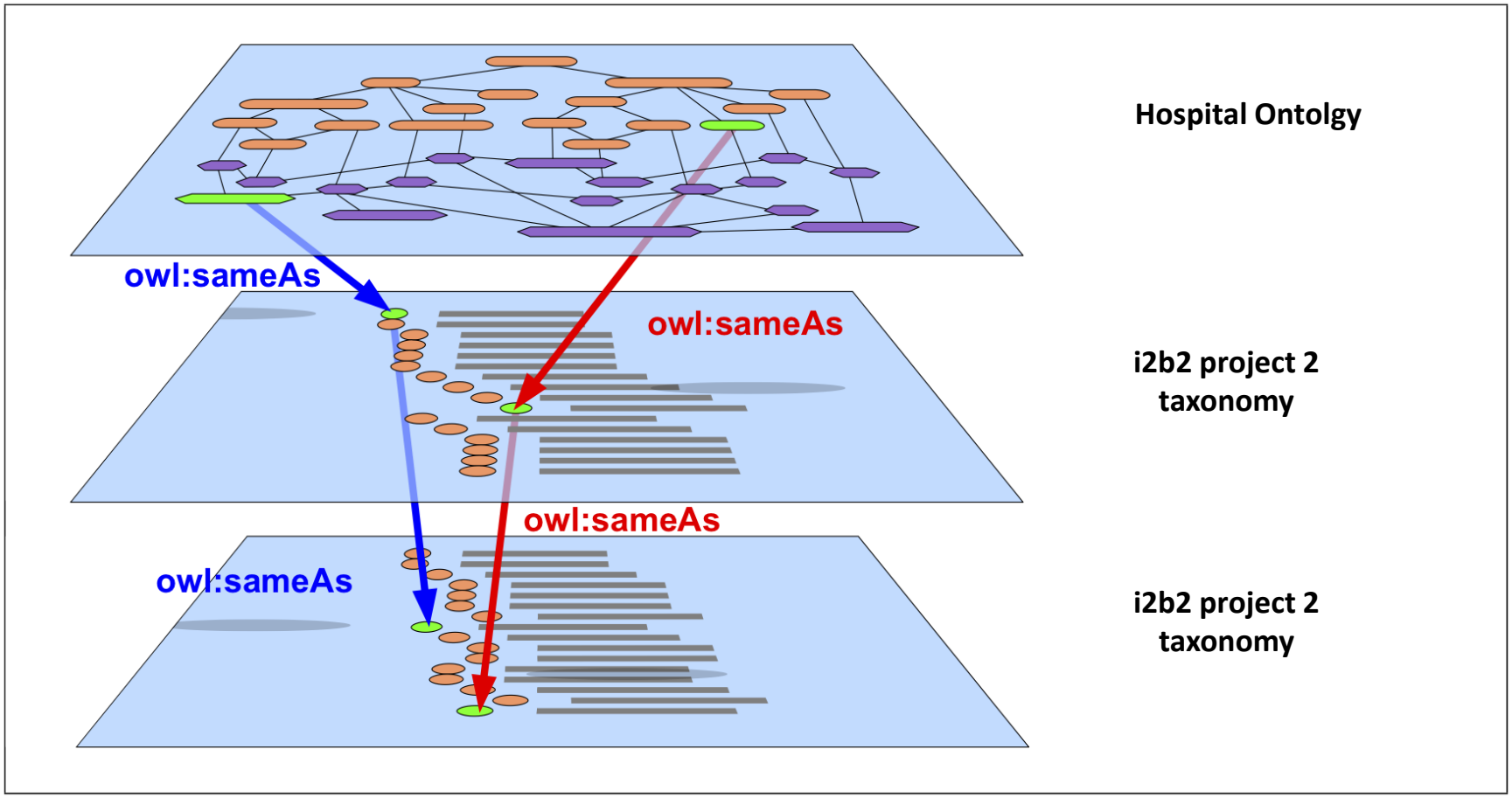
Discussion

The Future of this Project

- **Further developments**
  - Model the target ontology in a generic form (not only i2b2-specific)
  - Add further relationships to the target ontology (isAttributeOf, isValueOf, describesNonexistanceOf, describesExistanceOf, …)
  - Enable modeling of complex relationships between data elements
  - Allow mappings at different hierarchy levels
  - Allow abstract mappings without useing tricks (e.g. check for the existance of a form)

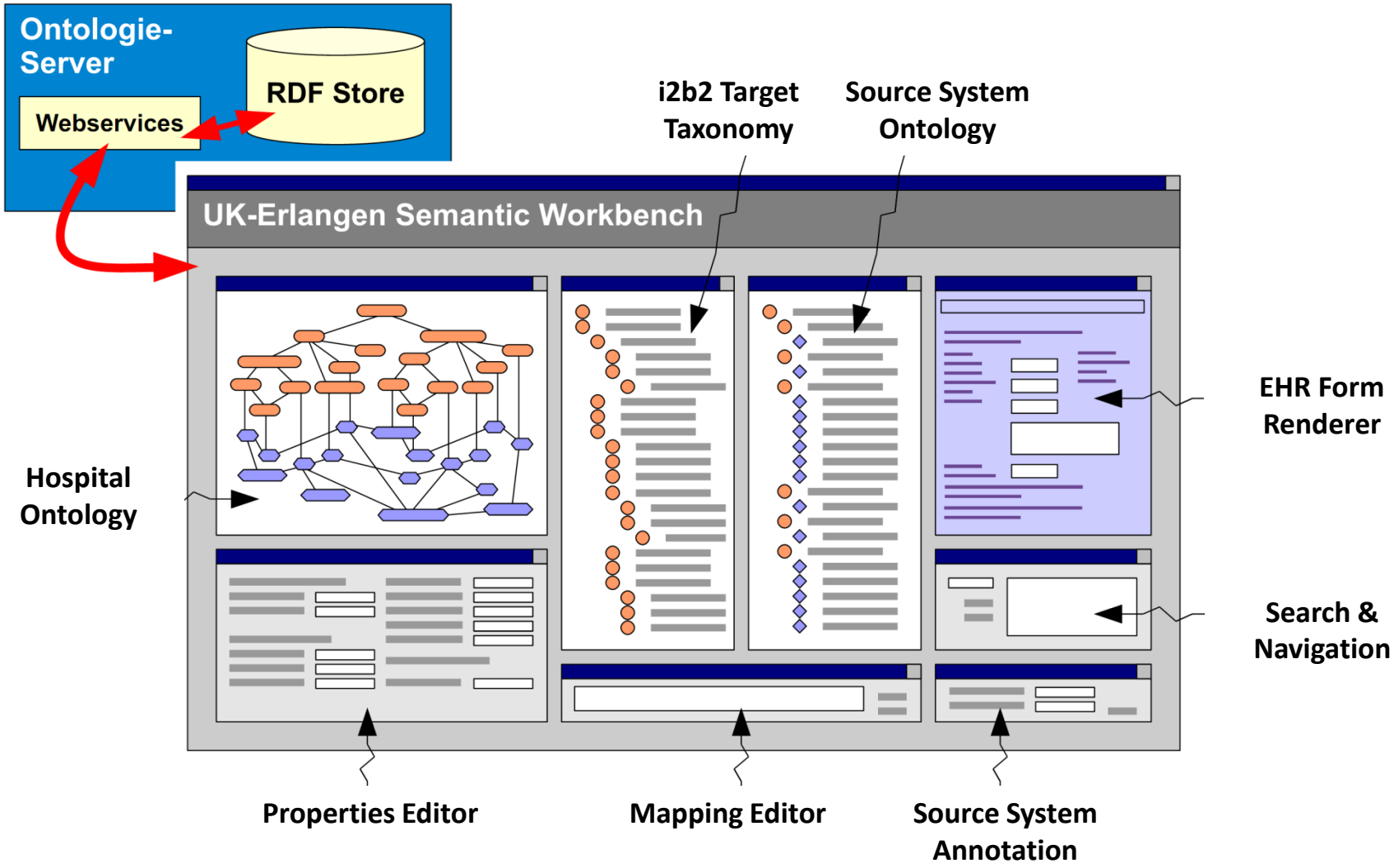
Discussion

Vision 1: linkage to other ontologies



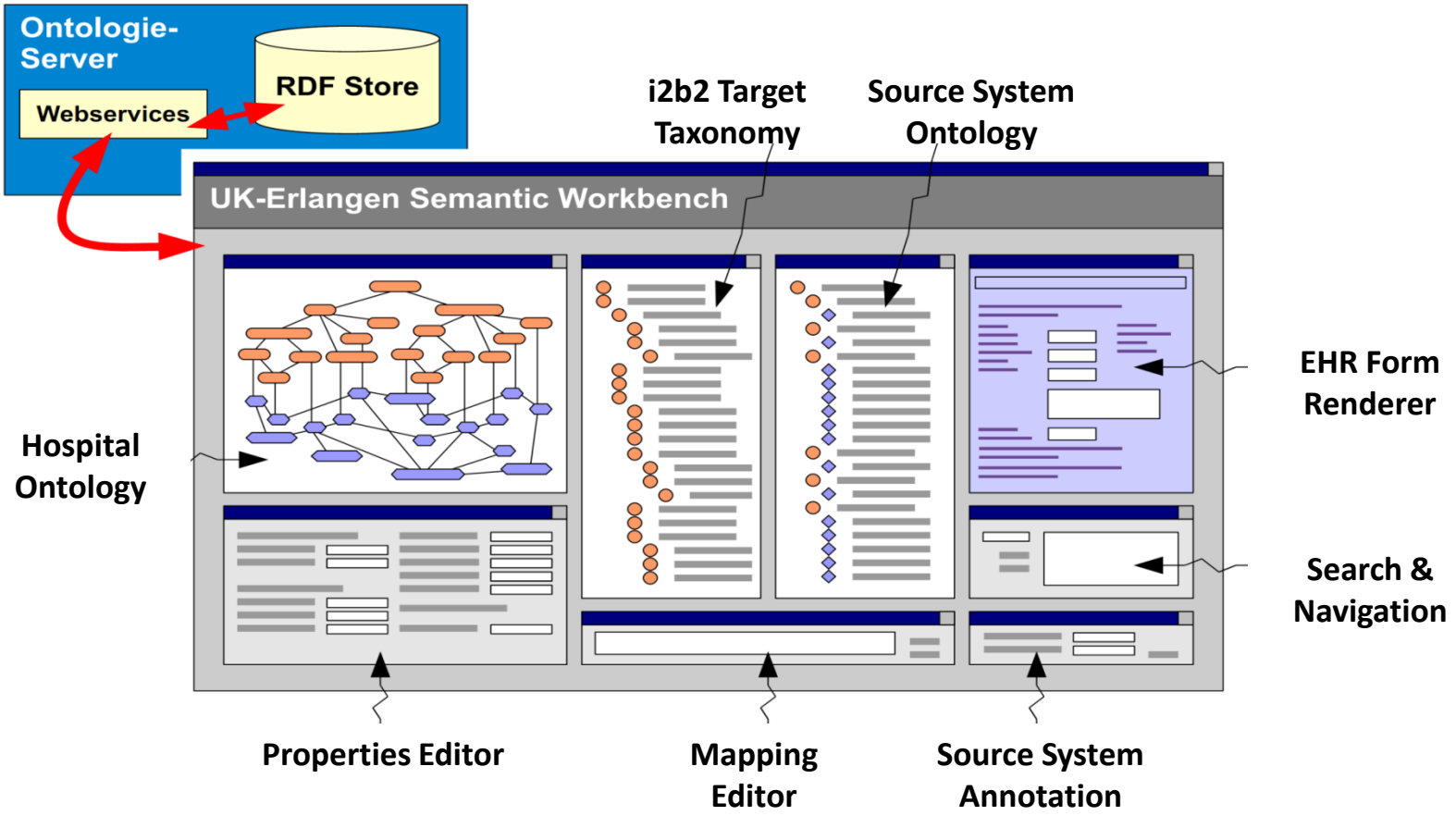
Discussion

Vision 2: A Client/Server Architecture



Thanks for Listening!

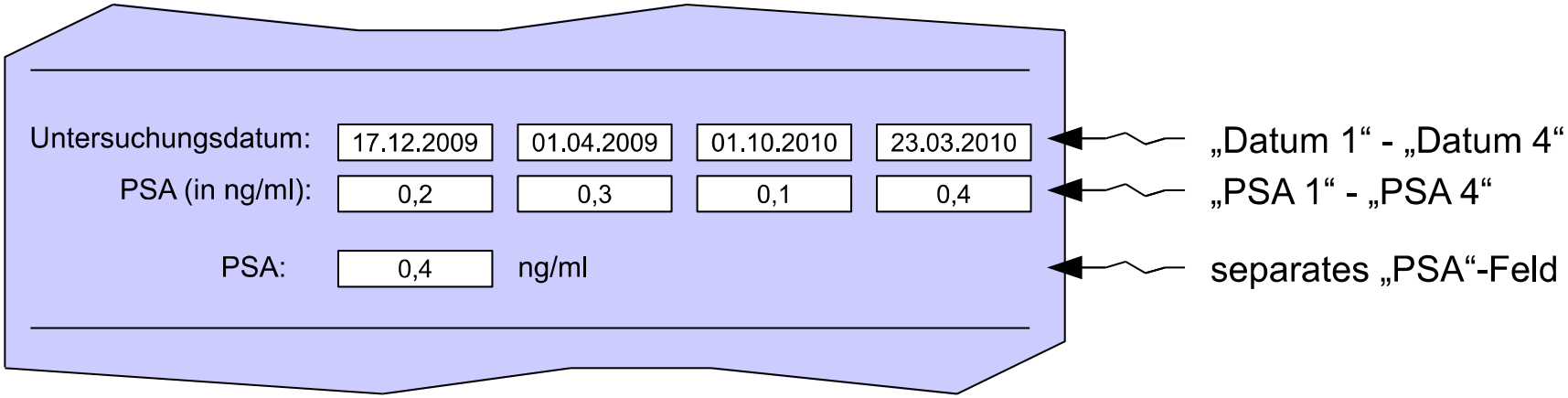
sebastian.mate@imi.med.uni-erlangen.de



Backup / Additional Slides

Ontology Mapping: PSA Example

=> see MIE2011 paper!



Task: export the last PSA value. If none of the above fields has been filled out, export the separate field.

4x

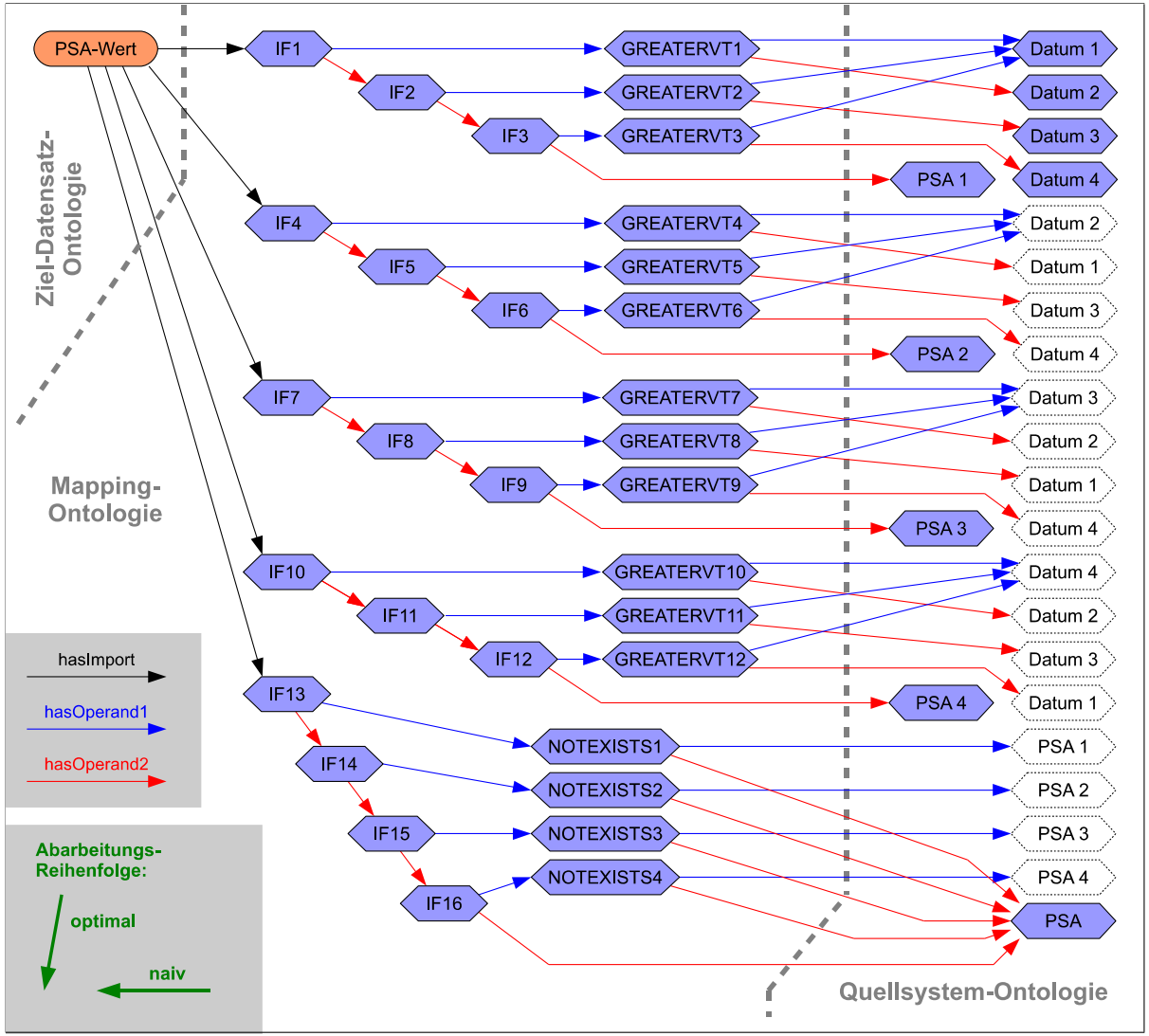
```
1  PSA-Wert : IF (GREATERVT Datum1 Datum2)
2             (IF (GREATERVT Datum1 Datum3)
3             (IF (GREATERVT Datum1 Datum4) PSA1))
```

1x

```
1      PSA-Wert: IF (NOTEXISTS PSA1 PSA)
2                (IF (NOTEXISTS PSA2 PSA)
3                (IF (NOTEXISTS PSA3 PSA)
4                (IF (NOTEXISTS PSA4 PSA) PSA)))
```

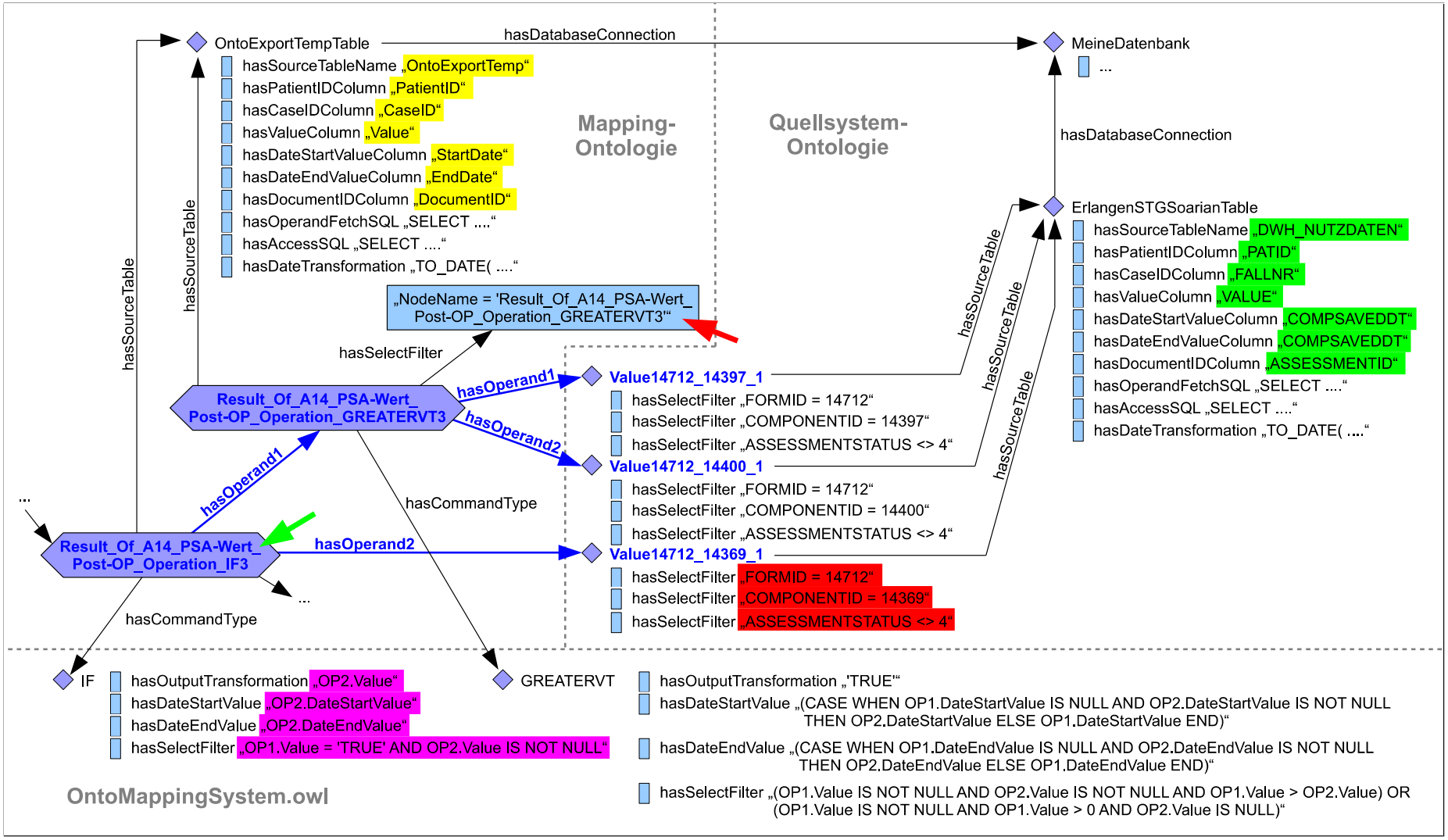
Ontology Mapping: PSA Example

=> see MIE2011 paper!



Data Export

Real world SQL code construction example



Data Export

Real world SQL code construction example

```
1  INSERT INTO OntoExportTemp (NodeName, DocumentID, PatientID, CaseID, StartDate, EndDate, Value)
2
3  SELECT DISTINCT
4
5      'Result_Of_A14_PSA-Wert_Operation_IF3' NodeName,
6
7      (CASE WHEN OP1.DocumentID IS NULL AND OP2.DocumentID IS NOT NULL THEN OP2.DocumentID ELSE OP1.DocumentID END) DocumentID,
8      (CASE WHEN OP1.PatientID IS NULL AND OP2.PatientID IS NOT NULL THEN OP2.PatientID ELSE OP1.PatientID END) PatientID,
9      (CASE WHEN OP1.CaseID IS NULL AND OP2.CaseID IS NOT NULL THEN OP2.CaseID ELSE OP1.CaseID END) CaseID,
10
11     OP2.DateStartValue DateStartValue,
12     OP2.DateEndValue DateEndValue,
13     OP2.Value Value
14
15 FROM
16
17     (SELECT DISTINCT DocumentID DocumentID, PatientID PatientID, CaseID CaseID, StartDate DateStartValue, EndDate DateEndValue,
       Value Value FROM OntoExportTemp WHERE NodeName = 'Result_Of_A14_PSA-Wert_Post-OP_Operation_GREATERVT3') OP1
18
19     FULL OUTER JOIN
20
21     (SELECT DISTINCT ASSESSMENTID DocumentID, PATID PatientID, FALLNR CaseID, COMPSAVEDDT DateStartValue, COMPSAVEDDT
       DateEndValue, VALUE Value FROM DWH_NUTZDATEN WHERE FORMID = 14712 AND COMPONENTID = 14369 AND ASSESSMENTSTATUS <> 4) OP2
22
23     ON OP1.DocumentID = OP2.DocumentID
24
25 WHERE OP1.Value = 'TRUE' AND OP2.Value IS NOT NULL
```

A few mapping examples

The source table

Sheet ID	MySheet Date	PatID	Sex	Age	Test Score 1	Test Score 2	Lab Value	Date of Lab Value	Patient Notes
DocumentID	GlobalDate	PatientID		Integer	Integer	Integer	PosFloat	Date	String
1	01.06.10	100001	Male	10		4	0,2	01.01.10	Lorem ipsum dolor
2	01.06.10	100002	Female	11		5	0,5	02.01.10	sit amet, consete
3	01.06.10	100003	M	12		1	0,8	03.01.10	tur Morleys sadipsc
4	01.06.10	100004	F	13		2	1,1	04.01.10	e et lung cancer d
5	01.06.10	100005	M	14		3	1,4	05.01.10	umy smoking eirm
6	01.06.10	100006	F		3	4	1,7	06.01.10	invidunt ut labor
7	01.06.10	100007	M	16	2	5	0,2	07.01.10	itr, sed diam non
8	01.06.10	100008	F	17	3	1	0,5	08.01.10	aliquyam erat, se
9	01.06.10	100009	M	18	4	3	0,8	09.01.10	d diam voluptua.
10	01.06.10	100010	F	19	5	3	1,1	10.01.10	At [Hello World] eos
11	01.06.10	100011		20	1	4	1,4	11.01.10	cusam [Hallo DPKK]
12	01.06.10	100012	F	21	2		1,7	12.01.10	o [Huhu IMI] et ea
13	01.06.10	100013	M	22	3		0,2	13.01.10	ebum. Stet clita
14	01.06.10	100014	F		4		0,5	14.01.10	kasd gubergren,
15	01.06.10	100015	M	24	5		0,8	15.01.10	no sea takimata s
16	01.06.10	100016	F	25	1		1,1	16.01.10	anctus est Lorem
17	01.06.10	100017	M	26			1,4	17.01.10	Lorem ipsum dolor
18	01.06.10	100018	F	27			1,7	18.01.10	sit amet, consete

A few mapping examples

ADD: adding two values (both must not be NULL)

Sheet ID	MySheet Date	PatID	Sex	Age	Test Score 1	Test Score 2	Lab Value	Date of Lab Value	Patient Notes
DocumentID	GlobalDate	PatientID		Integer	Integer	Integer	PosFloat	Date	String
1	01.06.10	100001	Male	10		4	0,2	01.01.10	Lorem ipsum dolor
2	01.06.10	100002	Female	11		5	0,5	02.01.10	sit amet, consete
3	01.06.10	100003	M	12		1	0,8	03.01.10	tur Morleys sadipsc
4	01.06.10	100004	F	13		2	1,1	04.01.10	e et lung cancer d
5	01.06.10	100005	M	14		3	1,4	05.01.10	umy smoking eirm
6	01.06.10	100006	F		3	4	1,7	06.01.10	invidunt ut labor
7	01.06.10	100007	M	16	2	5	0,2	07.01.10	itr, sed diam non
8	01.06.10	100008	F	17	3	1	0,5	08.01.10	aliquyam erat, se
9	01.06.10	100009	M	18	4	3	0,8	09.01.10	d diam voluptua.
10	01.06.10	100010	F	19	5	3	1,1	10.01.10	At [Hello World] eos
11	01.06.10	100011		20	1	4	1,4	11.01.10	cusam [Hallo DPKK]
12	01.06.10	100012	F	21	2		1,7	12.01.10	o [Huhu IMI] et ea
13	01.06						0,2	13.01.10	ebum. Stet clita
14	01.06						0,5	14.01.10	kasd gubergren,
15	01.06						0,8	15.01.10	no sea takimata s
16	01.06.10	100016	F	25	1		1,1	16.01.10	anctus est Lorem
17	01.06.10	100017	M	26			1,4	17.01.10	Lorem ipsum dolor
18	01.06.10	100018	F	27			1,7	18.01.10	sit amet, consete

Sum : ADD TestScore1 TestScore2

A few mapping examples

"Tolerant" ADD (2nd operarand may be NULL)

Sheet ID	MySheet Date	PatID	Sex	Age	Test Score 1	Test Score 2	Lab Value	Date of Lab Value	Patient Notes
DocumentID	GlobalDate	PatientID		Integer	Integer	Integer	PosFloat	Date	String
1	01.06.10	100001	Male	10		4	0,2	01.01.10	Lorem ipsum dolor
2	01.06.10	100002	Female	11		5	0,5	02.01.10	sit amet, consete
3	01.06.10	100003	M	12		1	0,8	03.01.10	tur Morleys sadipsc
4	01.06.10	100004	F	13		2	1,1	04.01.10	e et lung cancer d
5	01.06.10	100005	M	14		3	1,4	05.01.10	umy smoking eirm
6	01.06.10	100006	F		3	4	1,7	06.01.10	invidunt ut labor
7	01.06.10	100007	M	16	2	5	0,2	07.01.10	itr, sed diam non
8	01.06.10	100008	F	17	3	1	0,5	08.01.10	aliquyam erat, se
9	01.06.10	100009	M	18	4	3	0,8	09.01.10	d diam voluptua.
10	01.06.10	100010	F	19	5	3	1,1	10.01.10	At [Hello World] eos
11	01.06.10	100011		20	1	4	1,4	11.01.10	cusam [Hallo DPKK]
12	01.06.10	100012	F	21	2		1,7	12.01.10	o [Huhu IMI] et ea
13	01.06.10	100013	M	22	3		0,2	13.01.10	ebum. Stet clita
14	01.06.10	100014	F		4		0,5	14.01.10	kasd gubergren,
15	01.06.10	100015	M	24	5		0,8	15.01.10	no sea takimata s
16	01.06.10	100016	F	25	1		1,1	16.01.10	anctus est Lorem
17	01.06.10	100017	M	26			1,4	17.01.10	Lorem ipsum dolor
18							1,7	18.01.10	sit amet, consete

Sum : ADDT TestScore1 TestScore2

A few mapping examples

"Very Tolerant" ADD (both values may be NULL)

Sheet ID	MySheet Date	PatID	Sex	Age	Test Score 1	Test Score 2	Lab Value	Date of Lab Value	Patient Notes
DocumentID	GlobalDate	PatientID		Integer	Integer	Integer	PosFloat	Date	String
1	01.06.10	100001	Male	10		4	0,2	01.01.10	Lorem ipsum dolor
2	01.06.10	100002	Female	11		5	0,5	02.01.10	sit amet, consete
3	01.06.10	100003	M	12		1	0,8	03.01.10	tur Morleys sadipsc
4	01.06.10	100004	F	13		2	1,1	04.01.10	e et lung cancer d
5	01.06.10	100005	M	14		3	1,4	05.01.10	umy smoking eirm
6	01.06.10	100006	F		3	4	1,7	06.01.10	invidunt ut labor
7	01.06.10	100007	M	16	2	5	0,2	07.01.10	itr, sed diam non
8	01.06.10	100008	F	17	3	1	0,5	08.01.10	aliquyam erat, se
9	01.06.10	100009	M	18	4	3	0,8	09.01.10	d diam voluptua.
10	01.06.10	100010	F	19	5	3	1,1	10.01.10	At [Hello World] eos
11	01.06.10	100011		20	1	4	1,4	11.01.10	cusam [Hallo DPKK]
12	01.06.10	100012	F	21	2		1,7	12.01.10	o [Huhu IMI] et ea
13	01.06.10	100013	M	22	3		0,2	13.01.10	ebum. Stet clita
14	01.06.10	100014	F		4		0,5	14.01.10	kasd gubergren,
15	01.06.10	100015	M	24	5		0,8	15.01.10	no sea takimata s
16	01.06.10	100016	F	25	1		1,1	16.01.10	anctus est Lorem
17	01.06.10	100017	M	26			1,4	17.01.10	Lorem ipsum dolor
18							1,7	18.01.10	sit amet, consete

Sum : ADDVT TestScore1 TestScore2

A few mapping examples

Accessing unstructured text (simple NLP)

Sheet ID	MySheet Date	PatID	Sex	Age	Test Score 1	Test Score 2	Lab Value	Date of Lab Value	Patient Notes
DocumentID	GlobalDate	PatientID		Integer	Integer	Integer	PosFloat	Date	String
1	01.06.10	100001	Male	10		4	0,2	01.01.10	Lorem ipsum dolor
2	01.06.10	100002	Female	11		5	0,5	02.01.10	sit amet, consete
3	01.06.10	100003	M	12		1	0,8	03.01.10	tur Morleys sadipsc
4	01.06.10	100004	F	13		2	1,1	04.01.10	e et lung cancer d
5	01.06.10	100005	M	14		3	1,4	05.01.10	umy smoking eirm
6	01.06.10	100006	F		3	4	1,7	06.01.10	nvidunt ut labor
7	01.06.10	100007	M	16	2				non
8	01.06.10	100008	F	17	3				at, se
9	01.06.10	100009	M	18	4				btua.
10	01.06.10	100010	F	19	5				orld] eos
11	01.06.10	100011		20	1				o DPKK]
12	01.06.10	100012	F	21	2				l] et ea
13	01.06.10	100013	M	22	3				clita
14	01.06.10	100014	F		4				ren,
15	01.06.10	100015	M	24	5				nata s
16	01.06.10	100016	F	25	1		1,1	16.01.10	anctus est Lorem
17	01.06.10	100017	M	26			1,4	17.01.10	Lorem ipsum dolor
18	01.06.10	100018	F	27			1,7	18.01.10	sit amet, consete

Smoker : INSTR PatientNotes „Morleys“
Smoker : INSTR PatientNotes „lung cancer“
Smoker : INSTR PatientNotes „smoking“
Smoker : INSTR PatientNotes „smoke“
Smoker : INSTR PatientNotes „cigarettes “
…

A few mapping examples

Accessing unstructured text (String manipulations)

Sheet ID	MySheet Date	PatID	Sex	Age	Test Score 1	Test Score 2	Lab Value	Date of Lab Value	Patient Notes
DocumentID	GlobalDate	PatientID		Integer	Integer	Integer	PosFloat	Date	String
1	01.06.10	100001	Male	10		4	0,2	01.01.10	Lorem ipsum dolor
2	01.06.10	100002	Female	11		5	0,5	02.01.10	sit amet, consete
3	01.06.10	100003	M	12		1	0,8	03.01.10	tur Morleys sadipsc
4	01.06.10	100004	F	13		2	1,1	04.01.10	e et lung cancer d
5	01.06.10	100005	M	14		3	1,4	05.01.10	umy smoking eirm
6	01.06.10	100006	F		3	4	1,7	06.01.10	invidunt ut labor
7	01.06.10	100007	M	16	2	5	0,2	07.01.10	itr, sed diam non
8	01.06.10	100008	F	17	3	1	0,5	08.01.10	aliquyam erat, se
9	01.06.10	100009	M	18	4	3	0,8	09.01.10	d diam voluptua.
10	01.06.10	100010	F	19	5	3	1,1	10.01.10	At [Hello World] eos
11	01.06.10	100011		20	1	4	1,4	11.01.10	cusam [Hallo DPKK]
12	01.06.10	100012	F	21	2		1,7	12.01.10	o [Huhu IMI] et ea
13	01.06.10	100013	M	22	3		0,2	13.01.10	um. Stet clita
14	01.06.10	100014	F		4				
15	01.06.10	100015	M	24	5				a s
16	01.06.10	100016	F	25	1				em
17	01.06.10	100017	M	26					olor
18	01.06.10	100018	F	27					ete

Greetings : TAIL (HEAD PatientNotes
SUBTR (STRPOS PatientNotes „]“) „1“))
(ADD (STRPOS PatientNotesValue „[“) „1“)

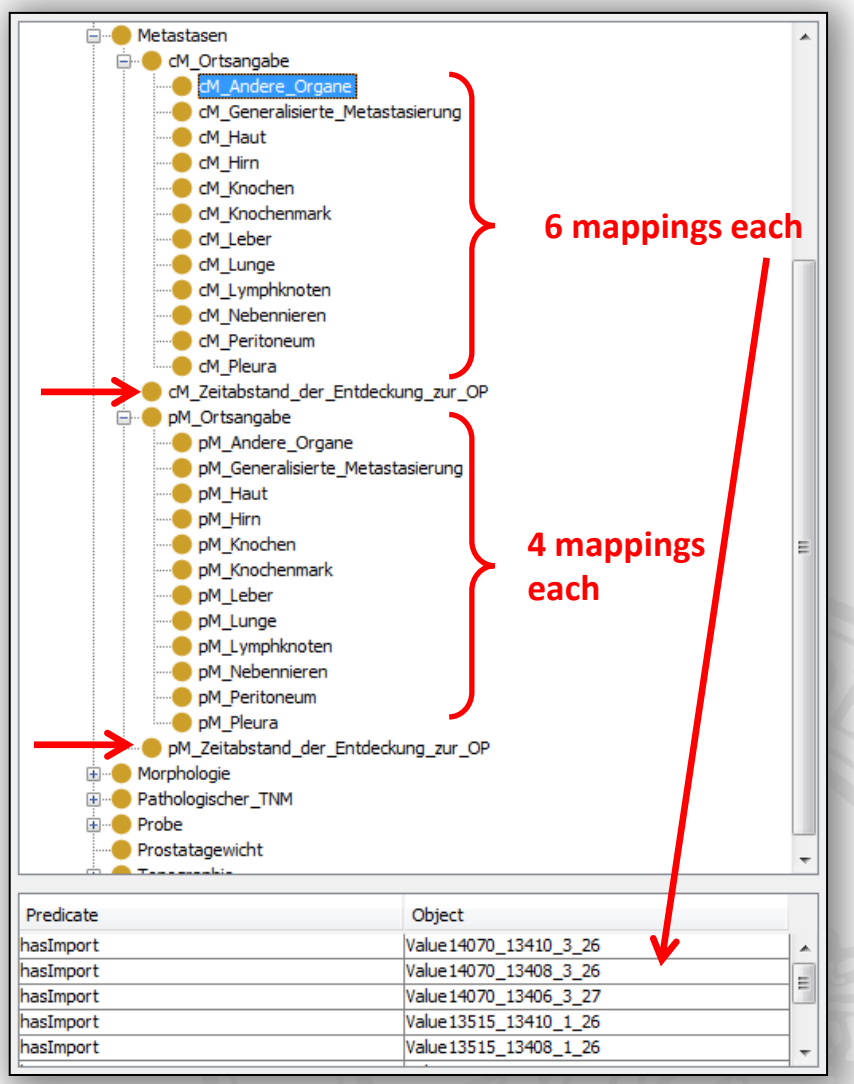
PatientID	Value
100010	Hello World
100011	Hallo DPKK
100012	Huhu IMI

Results: Ontology Mapping for DPKK

Results between Münster and Erlangen

- **Impossible mappings**
  - Erlangen: One mapping required access so SAP IS-H (to get the patient age), the system currently only support one source system at a time
- **Impractical mappings**
  - Erlangen/Münster: calculation of time intervals between two data elements (operation and detection of metastases)
    Problem: mapping at the values level (including versioning) would result in 108 distinct mappings for Erlangen!

We need to create mappings at various hierarchy levels



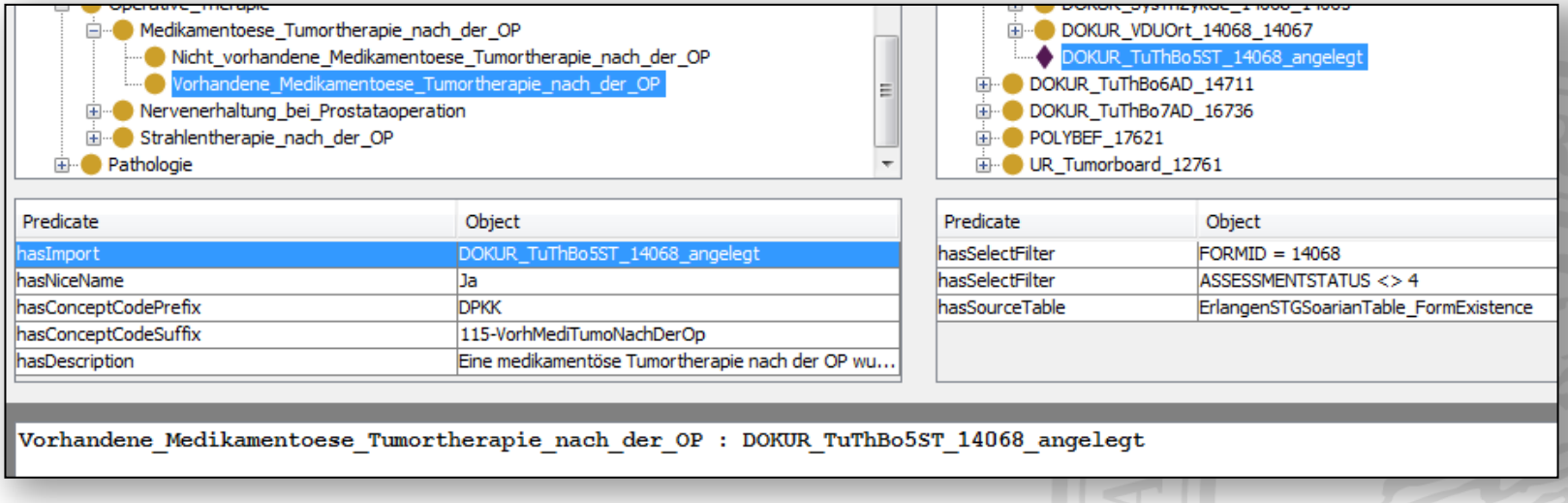
Results: Ontology Mapping for DPKK

Results between Münster and Erlangen

- **„A small trick“: cheating our own system**

  Four mappings required to check whether a form has been created or not (ignoring what data has been filled in). The mapping system does not support this. **Dirty solution:** „Simulate“ an other source system table (ErlangenSTGSoarianTable_FormExistence), which does not reurn real data, but „True“ if any data element from this form exists.

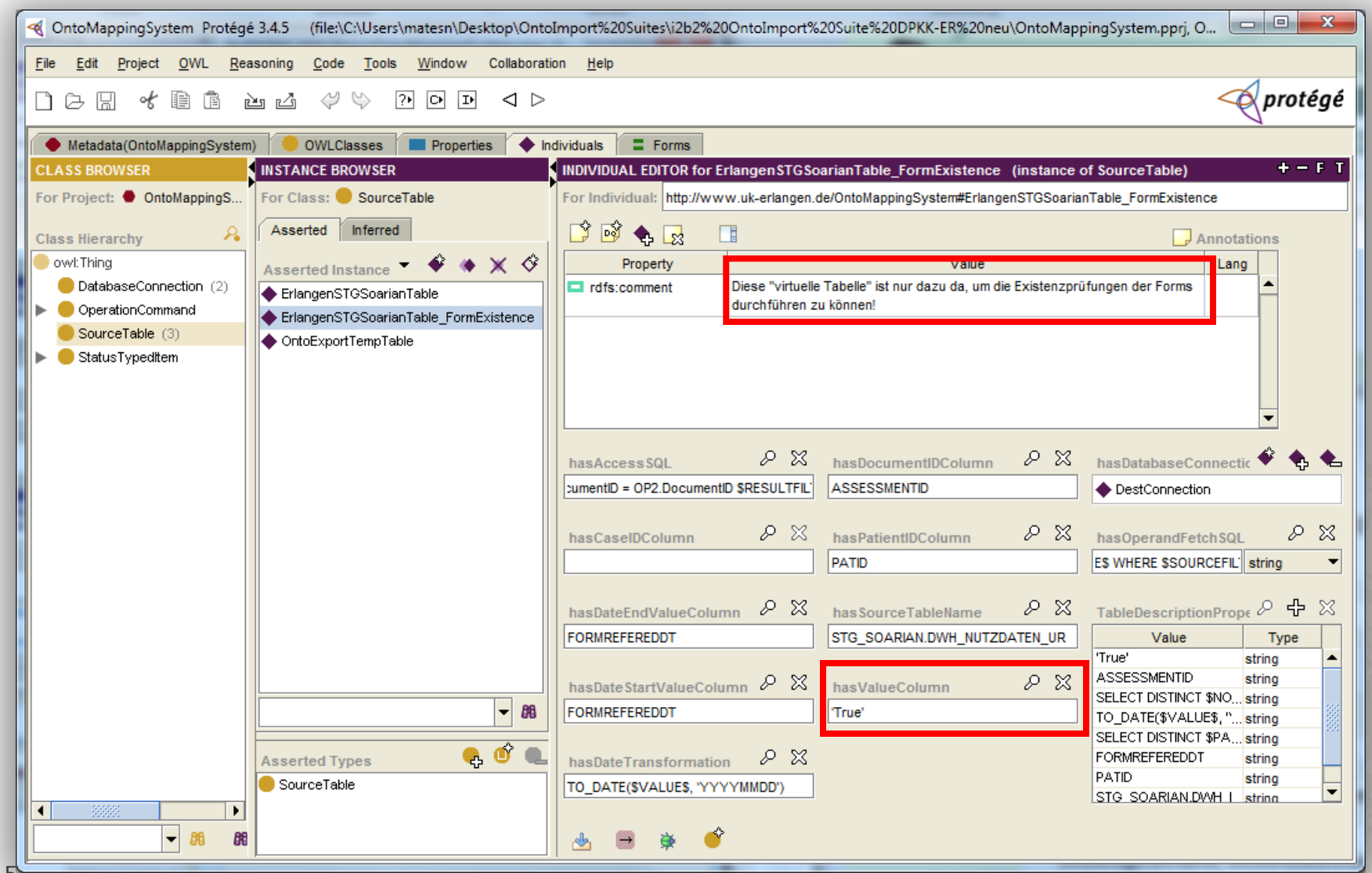
  => The ontology describes a table, which doesn‘t exist in reality!!!



We need to create „abstract“ mappings!

Results: Ontology Mapping for DPKK

Results between Münster and Erlangen



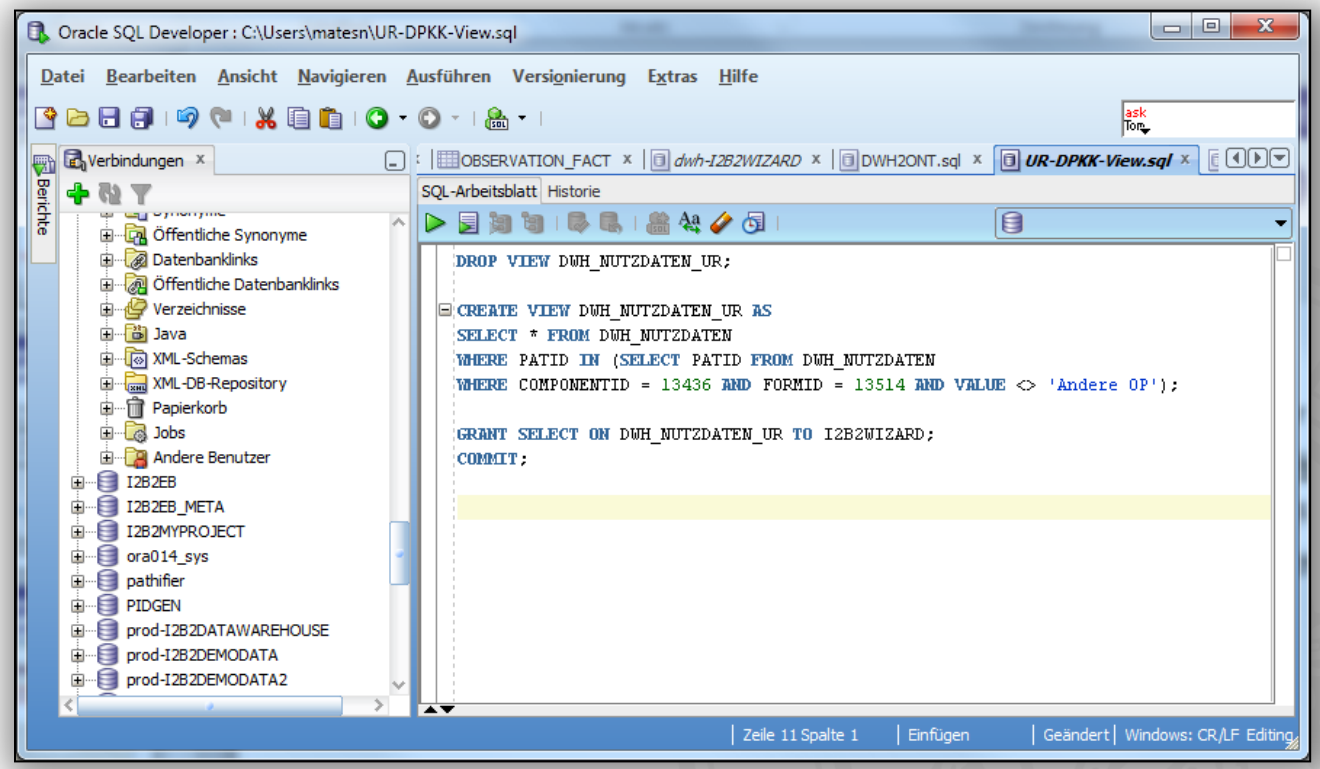
Results: Ontology Mapping for DPKK

Results between Münster and Erlangen

- **Another small trick:**

  In Erlangen, the DPKK project required prefiltering the patients‘ data with a view to model a relation between one concept and all other DPKK concepts.

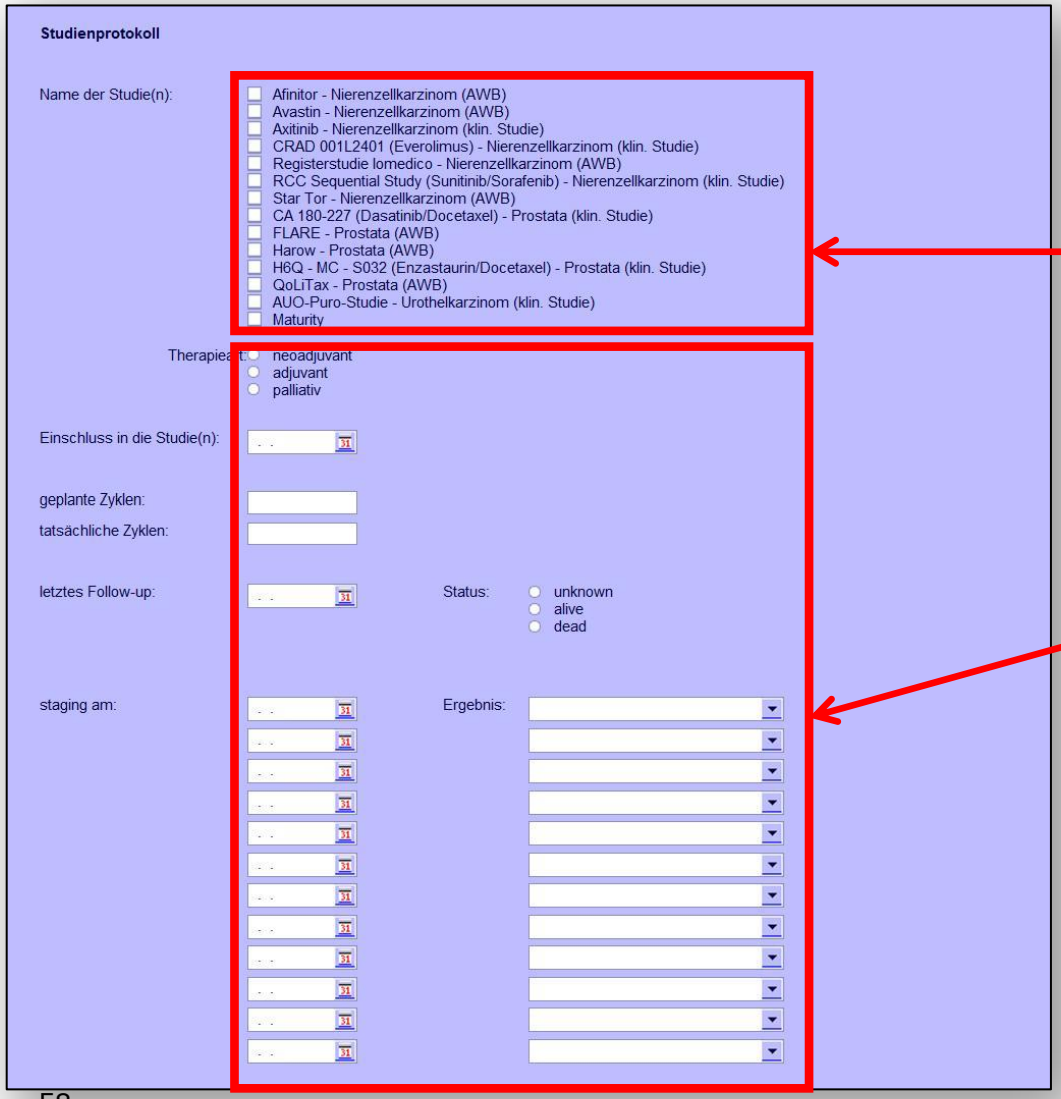
  => Important information is not stored inside the ontology!



Again: We need to allow mappings at various hierarchy levels

Results: Mapping for Urology

Impossible Mappings

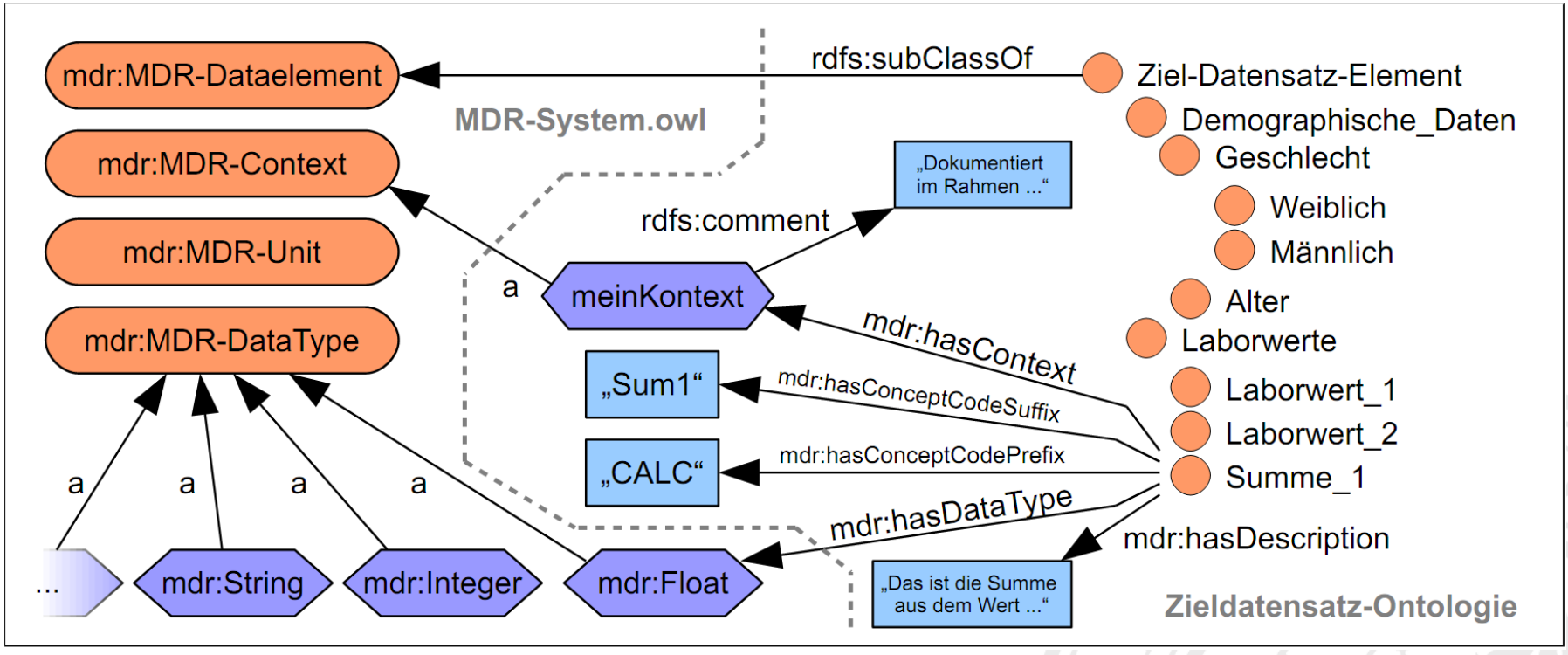


studies a patient can be enrolled in

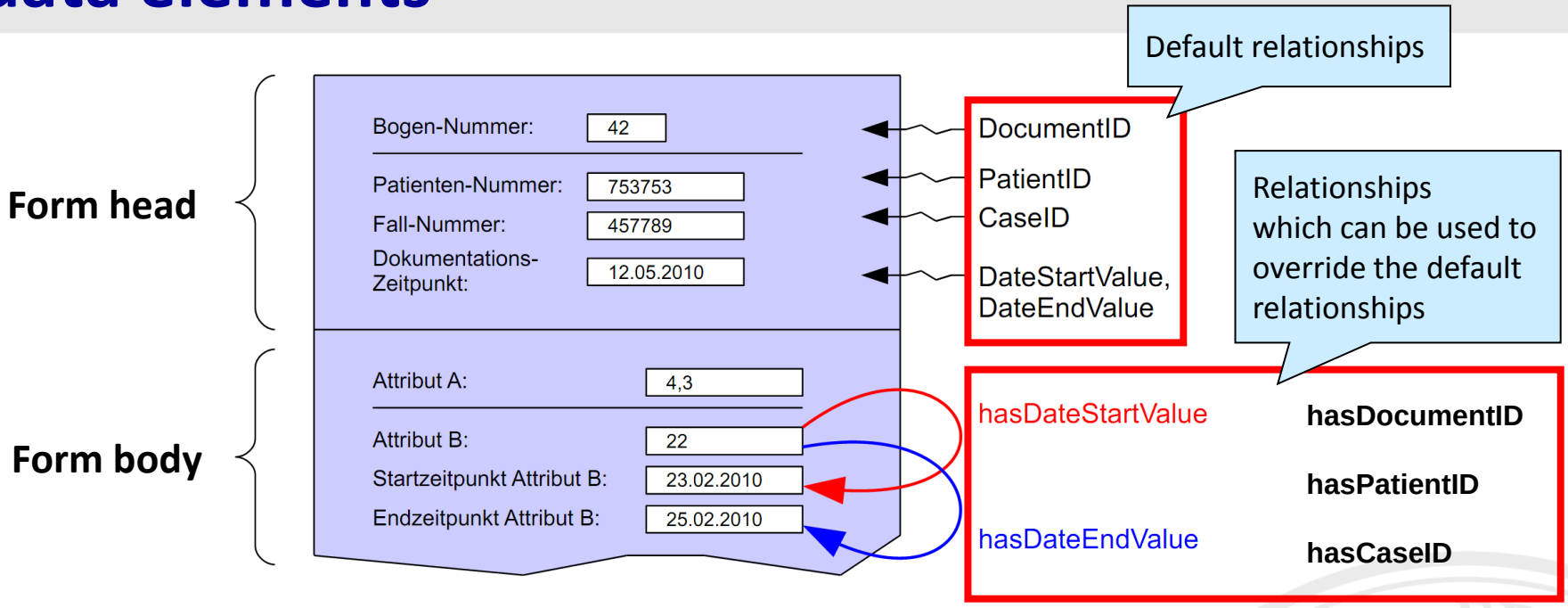
All other data elements

How can we model such complex relationships?

i2b2-specific Target Ontology

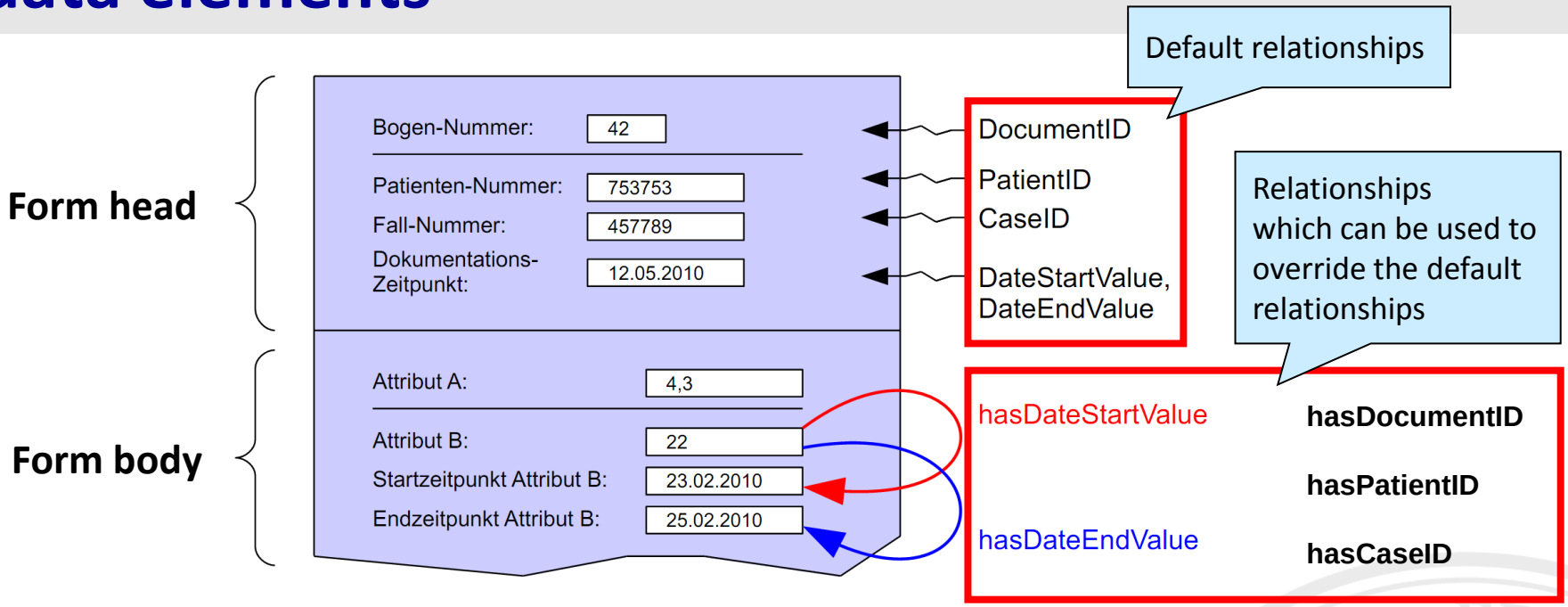


EAV data model to support relationships between data elements



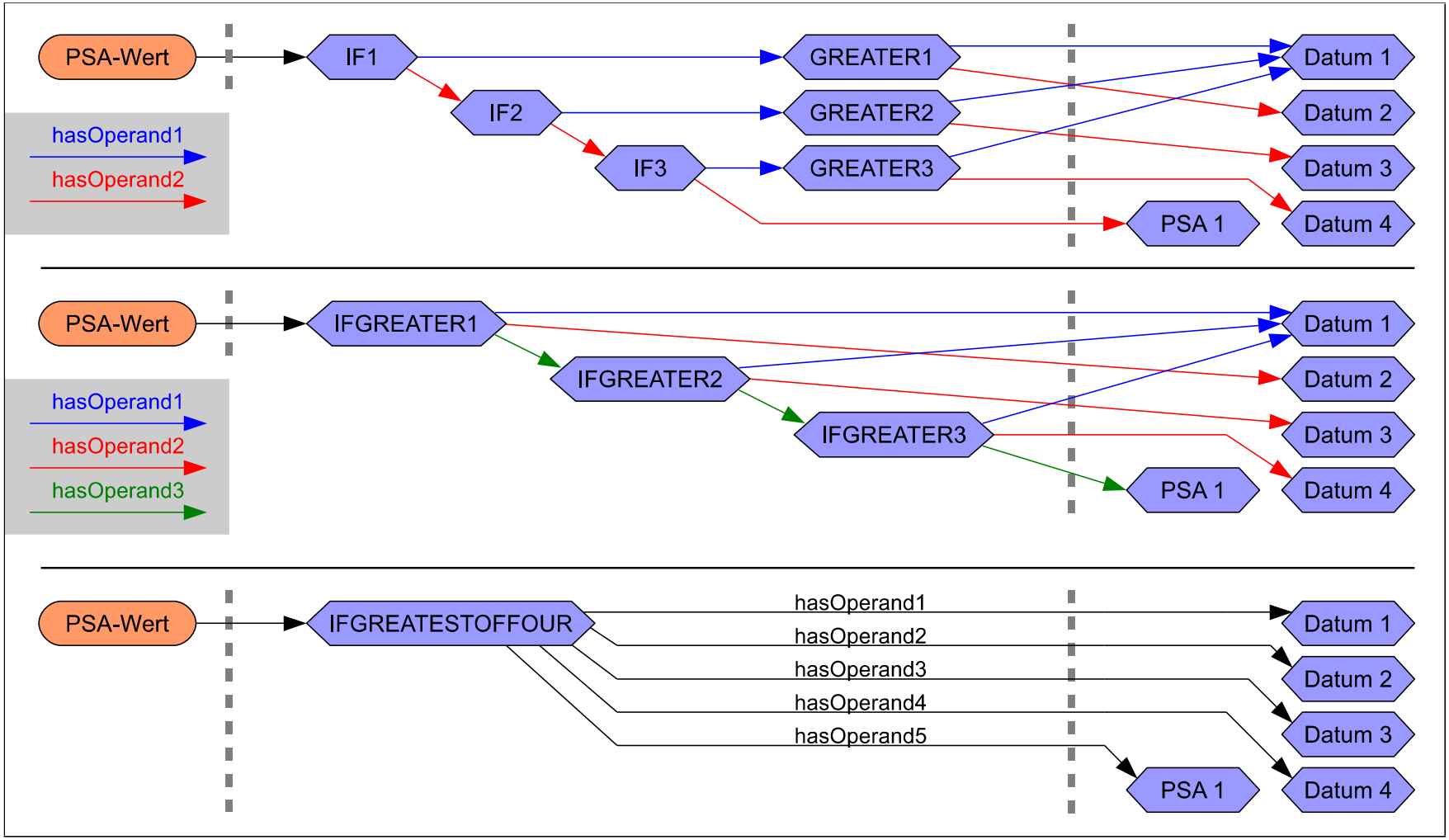
DocumentID	PatientID	CaseID	DateStartValue	DateEndValue	Attribute	Value
42	753753	457789	12.05.2010	12.05.2010	Attribut A:	4,3
42	753753	457789	12.05.2010	12.05.2010	Attribut B:	22
42	753753	457789	12.05.2010	12.05.2010	Startzeitpunkt Attribut B:	23.02.2010
42	753753	457789	12.05.2010	12.05.2010	Endzeitpunkt Attribut B:	25.02.2010

EAV data model to support relationships between data elements



DocumentID	PatientID	CaseID	DateStartValue	DateEndValue	Attribute	Value
42	753753	457789	12.05.2010	12.05.2010	Attribut A:	4,3
42	753753	457789	**23.02.2010**	**25.02.2010**	Attribut B:	22
42	753753	457789	12.05.2010	12.05.2010	Startzeitpunkt Attribut B:	23.02.2010
42	753753	457789	12.05.2010	12.05.2010	Endzeitpunkt Attribut B:	25.02.2010

„Super Nodes“ Concept



Discussion

Temporal Aspects