Detecting and Eliminating Bacteria UsinG Information Technology

Rémy Choquet, PhD in biomedical informatics
APHP, INSERM U872 EQ20
and the DebugIT consortium…

The debugIT Project in short

- Funded by the European Community's Seventh Framework Program under grant agreement n° FP7–217139 (7M€)
- Project period: from Jan 1st, 2008 to December 31st, 2011 – extended until mid-2012.
- 14 Partners (next slide)

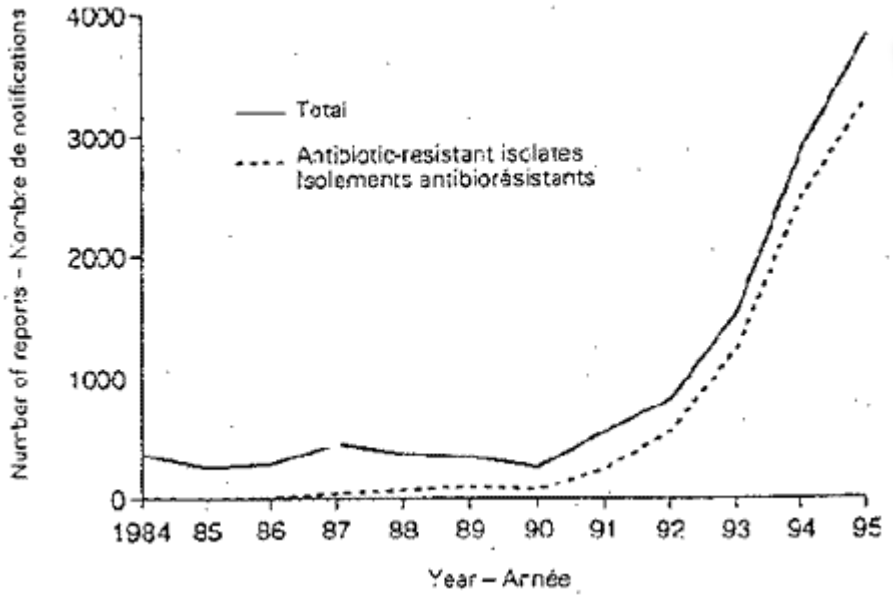
The Partners

- **Agfa HealthCare, Belgium (coordinator)**
- empirica Gesellschaft für Kommunikations- und Technologieforschung mbH, Germany
- Gama Sofia Ltd., Bulgaria
- Institut National de la Santé et de la Recherche Médicale, France
- Internetový Pristup Ke Zdravotním Informacím Pacienta (IZIP), Czech Republic
- Linköpings Universitetet, Sweden
- Technologiko Expedeftiko Idrima Lamias, Greece
- University College London, United Kingdom
- Les Hôpitaux Universitaires de Genève, Switzerland
- Universitätsklinikum Freiburg, Germany
- Université de Genève, Switzerland
- Averbis, Freiburg, Germany
- MDA, Czech Republic
- HEG, Geneva, Switzerland

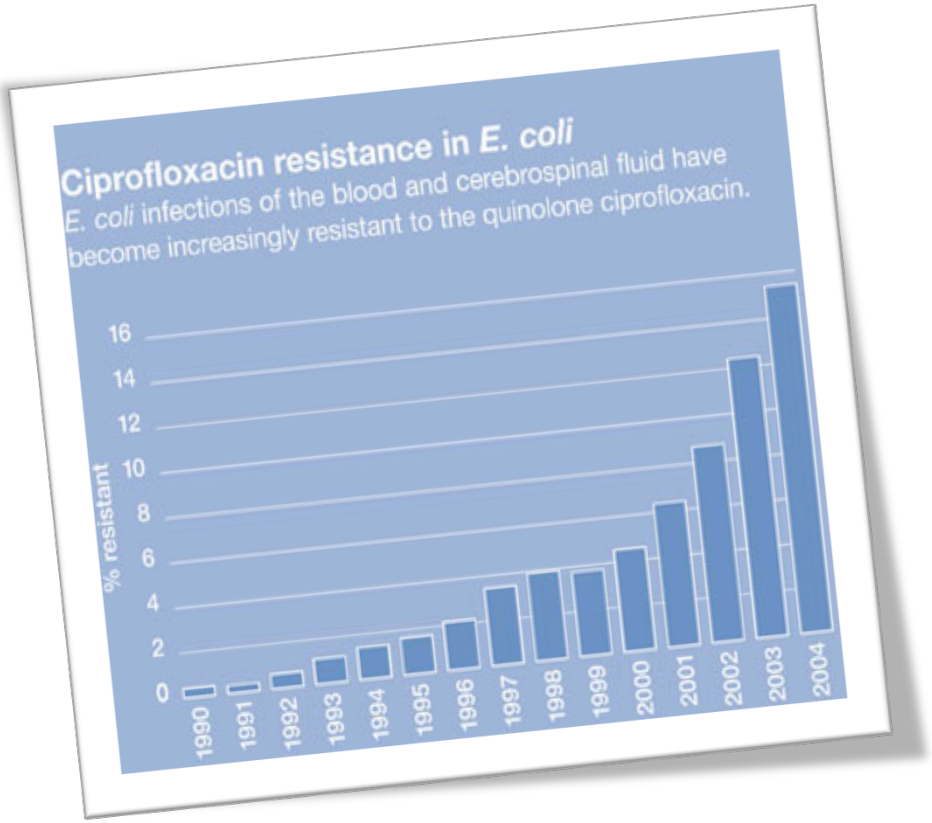
Why ?

The Problem

antibiotic resistance in Salmonella typhimurium DT104, England and Wales, 1984-1995



WHO Weekly Epidemiological Record, Vol 71, No 18, 1996



The debugIT Response

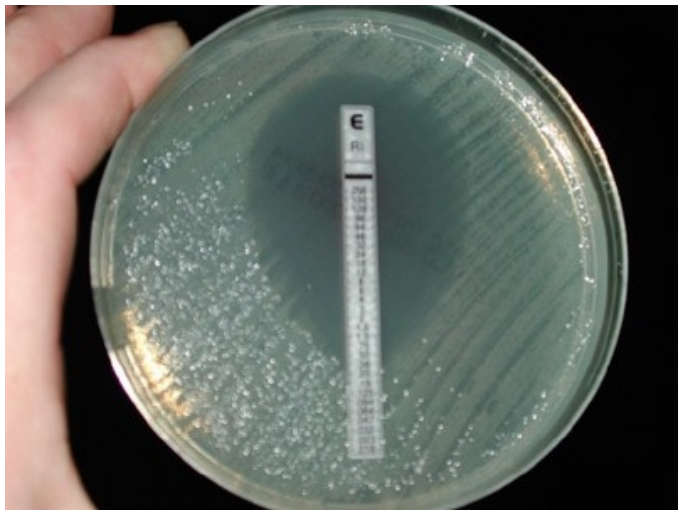
- If new antibiotics can not keep up with the bacterial resistance and a race against evolution can only be lost,

  → we need new solutions

- If this is a war and current weapons don't work anymore,

  → we need a new weapon

  → **ITbiotics** to help antibiotics



How ?

The debugIT Response

- the debugIT project
  - collects routinely stored data from clinical systems
  - learns by applying advanced data mining techniques
  - stores the extracted knowledge in repositories
  - applies the knowledge for decision support and monitoring

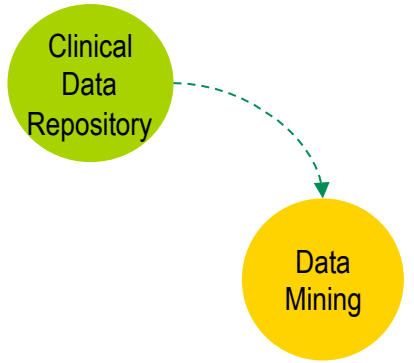
Collect Data

- Routinely stored clinical data is collected and aggregated across different
  - hospitals
  - countries
  - languages
  - information models
  - legislations
- via
  - commonly agreed data models (minimal data sets)
  - standards
  - mapping algorithms
  - unified and enhanced ontologies
- organized in a virtualized, Clinical Data Repository (CDR).

Learn

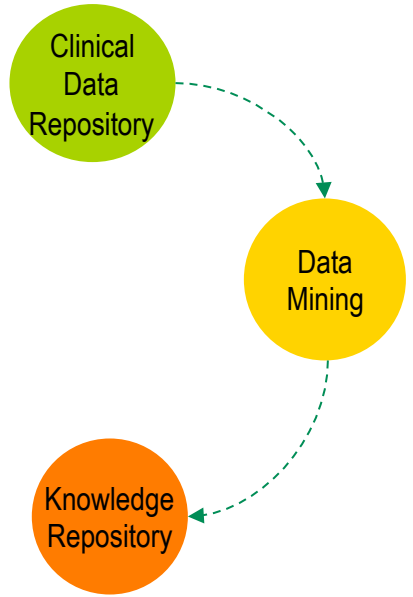
- debugIT learns by detecting patterns, relevant for patient safety and the better treatment of infectious diseases
- Advanced data mining techniques are used on
  - multimodal & multi-source data
    - structured data mining
    - text mining
    - image mining

  to create new knowledge



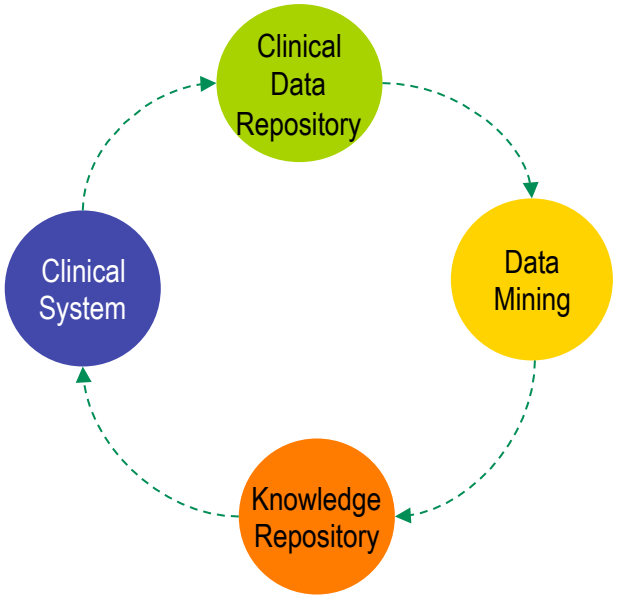
Store and Author Knowledge

- This knowledge is
  - stored in a distributed repository
  - validated by clinicians
  - visualized and
  - aggregated together with pre-existing medical and biological knowledge (guidelines, regulations)
- to achieve a consolidated view on the required knowledge



Apply

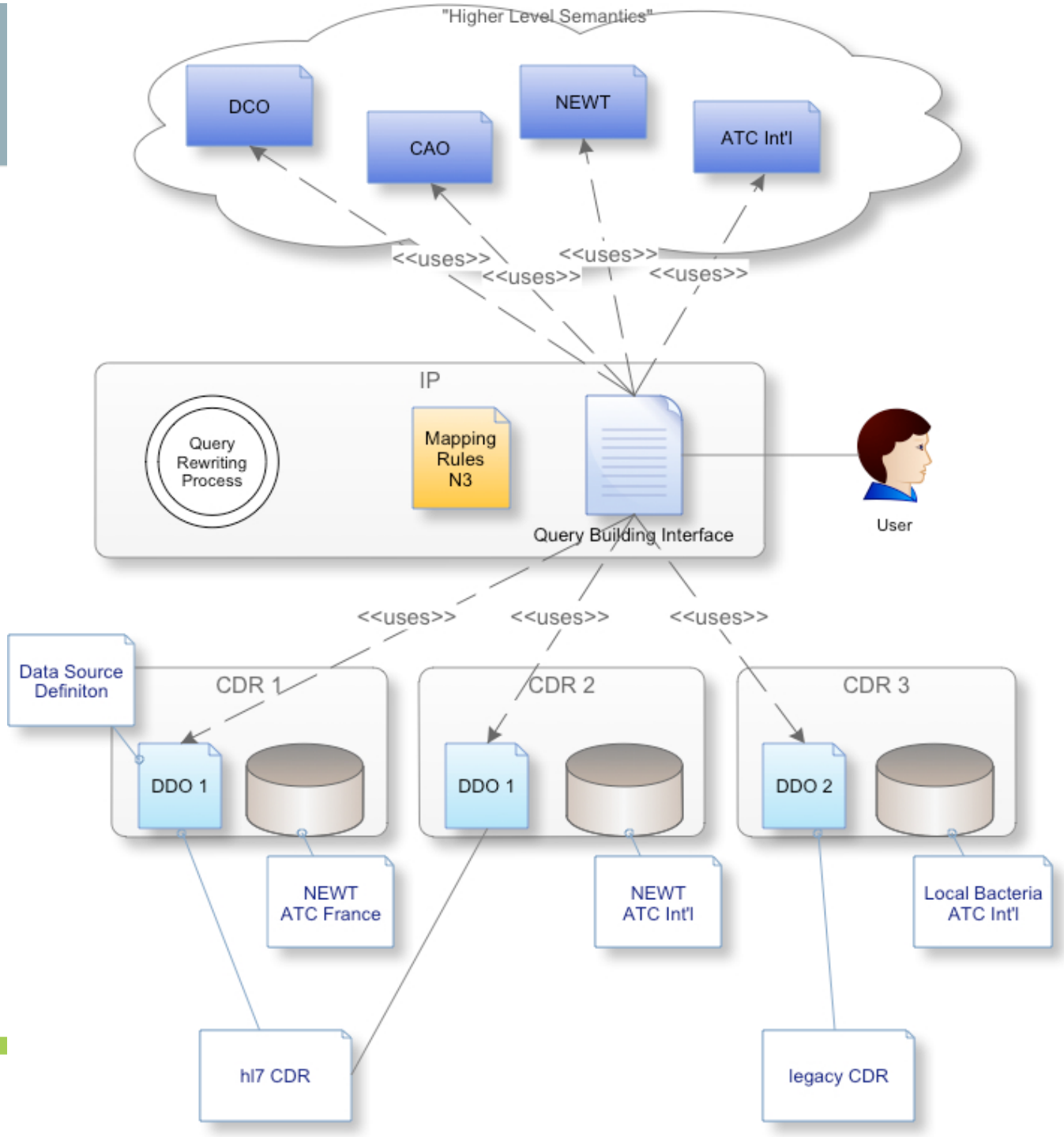
- Tools are available as web applications and can also be integrated in in-house applications.
  - Population driven monitoring tools analyze ongoing care activities and outcomes. It provides data to review and steer policies.
  - Patient centered decision support tools apply the newly generated knowledge and help the clinician to provide improved clinical care (choice, dose and administration of antibiotics for example).
- Integration in existing CIS enable the recording of activities and results and thus make sure the necessary data are generated for a next cycle of learning.



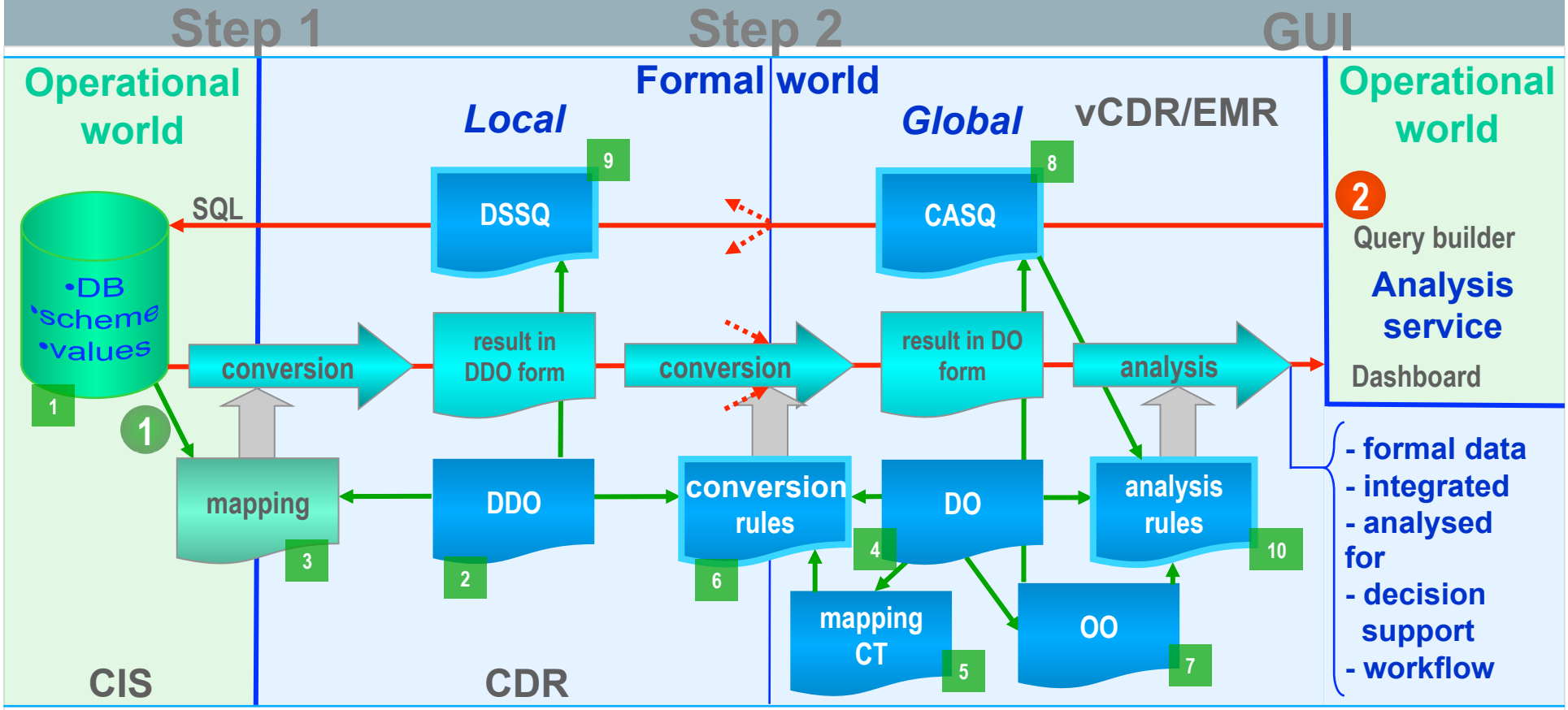
DEBUGIT : ONTOLOGY-MEDIATED DATA INTEGRATION FOR REAL-TIME ANTIBIOTICS RESISTANCE SURVEILLANCE

Constraints

- Heterogenous databases that can’t be modified/adapted
- Not possible to impose any model nor vocab
- Mediation architecture must scale both horizontally (adding new endpoints) or vertically (adding new concepts)
- Make domain knowledge independant with database raw data
- Make knowledge independent from each other (IM – DO)
- Re-usable architecture…


"Higher Level Semantics"
DCO
CAO
NEWT
ATC Int'l
<<uses>>
<<uses>>
<<uses>>
<<uses>>
IP
Query Rewriting Process
Mapping Rules N3
Query Building Interface
User
<<uses>>
<<uses>>
<<uses>>
Data Source Definiton
CDR 1
CDR 2
CDR 3
DDO 1
DDO 1
DDO 2
NEWT ATC France
NEWT ATC Int'l
Local Bacteria ATC Int'l
hl7 CDR
legacy CDR

From data values to formal resources



DB: database
CIS: Clinical Information System
DDO: Data Definition Ontology
CDR: Clinical Data Repository
DO: Domain Ontology
CT: clinical terminology
OO: Operational Ontology
vCDR: virtual Clinical Data Repository - e.g. EPR: Electronic Patient Record
CASQ: Clinical Analysis SPARQL Query
DSSQ: Data Set SPARQL Query
GUI: Graphical User Interface / SQL: Structured Query Language

At development (deployment) **time**: creating "formal library"

At runtime: formalizing (via querying) and deducting (via reasoning with rules)

Using formalisms declared in ontologies

Local to Global

- Domain ontology formalise entities, activities, procedures… Data ontology has only informational entities
- Open world logic of our system (scalability) has driven us to go for **local to global «** mapping » and so query rewriting since **global to local** is *semi-decidable* in a open world assumption

- >> MAPPING RULES ARE MONOTONIC (so is the system) !

Step 1: CDR

DDO

- **Creation of Data Definition Ontologies for 6 DBs:**
  - Local formal semantics
  - Express the database structure and values in RDF/S and OWL
  - Table name represented by class (rdfs:Class)
  - Column name represented by property (rdf:Property)
  - Field value represented by individual or (datatyped) literal (e.g. code)
  - Row (grey): graph

DB

Table name:

Column names:

Culture results	**Drug**	**Identified bacteria**	**Antibiotic tested result**
culture results 1	J01XA01 (vancomycin)	1280 (S. aureus)	sensitive

DDO	**RDF/S OWL**
CultureResults	Class
hasDrug	Property
hasIdentifiedBacteria	Property
hasAntibioticTestedResult	Property
cultureResults1	individual of Class
"J01XA01" ATC	Literal (typed)
"1280" UniProt	Literal (typed)
Sensitive	individual

debugIT

DDO

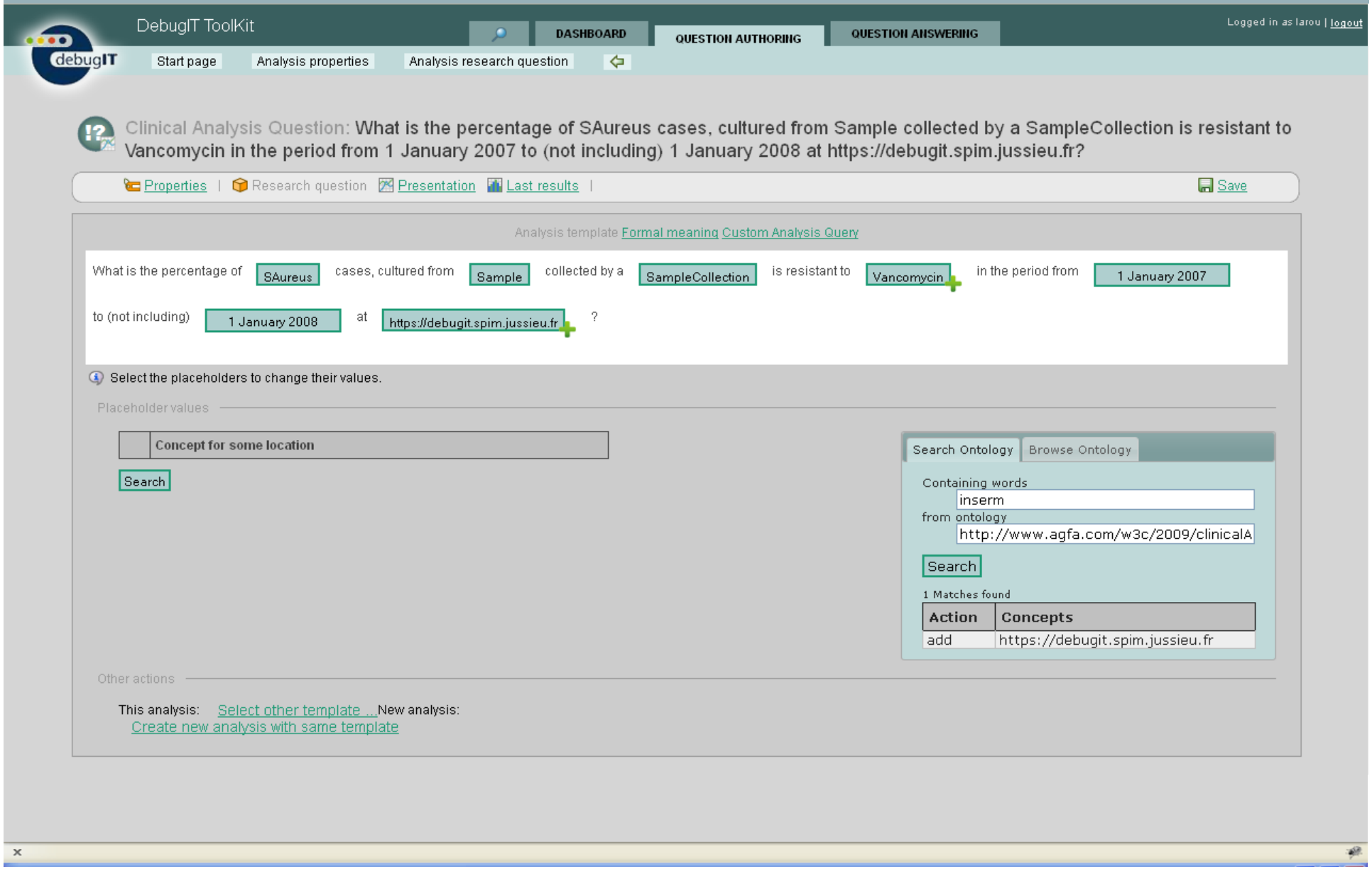
Step 1: CDR

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix ddo: <https://debugit.spim.jussieu.fr/ddo#>.
@prefix clisko: <http://www.agfa.com/w3c/2009/clinicalSKOSSchemes#>.
@prefix biosko: <http://eulersharp.sourceforge.net/2003/03swap/bioSKOSSchemes#>.

ddo:CultureResults rdfs:subClassOf [
    a owl:Restriction; owl:onProperty ddo:hasDrug;
      owl:someValuesFrom ddo:Drug], [
    a owl:Restriction; owl:onProperty ddo:hasIdentified_Bacteria;
      owl:someValuesFrom ddo:Bacteria], [
    a owl:Restriction; owl:onProperty ddo:hasAntibioticTestedResult;
      owl:someValuesFrom ddo:AntibioticTestedResult].

ddo:Drug
  rdfs:subClassOf [a owl:Restriction; owl:onProperty skos:notation;
    owl:someValuesFrom clisko:atc20090101DT].

ddo:Bacteria
  rdfs:subClassOf [a owl:Restriction; owl:onProperty skos:notation;
    owl:someValuesFrom biosko:uniProtTaxonomyDT].
```


debugIT

00

Query builder



CASQ template

CASQ

Looking for percentage
of **bacterium** cases (representing set of total)
    cultured from **sample**
        resulting from a **collection**
            during period
                starting date time yyyy-mm-dd/hh:mm:ss,
                ending date time yyyy-mm-dd/hh:mm:ss
            in **hospital**
    counting members;
percentage that is subset of the former (same properties)
    and resistant to **antibiotic**
    counting members;
having measurement with
    factor 'percent' (/100)
    value

CASQ instance

CASQ

Looking for percentage
of **Staphylococcus aureus** cases (representing set of total)
    cultured from **all samples**
        resulting from a **collection**
            during period
                starting date time **2007-01-01/00:00:00**,
                ending date time **2008-01-01/00:00:00**
            in **INSERM OR HUG** hospital
    counting members;
percentage that is subset of the former (same properties)
    and resistant to **vancomycin**
    counting members;
having measurement with
    factor 'percent' (/100)
    value

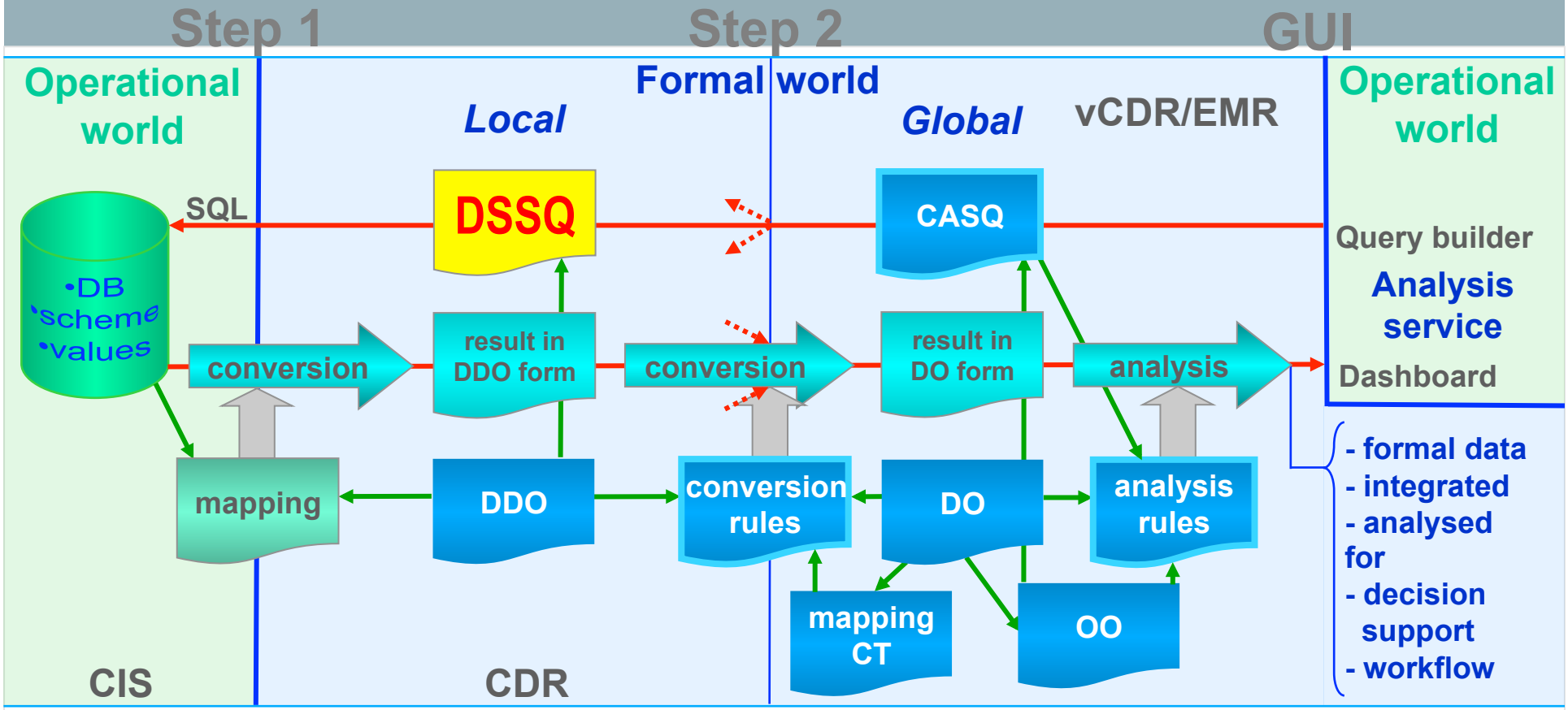
CASQ instance

CASQ

```
WHERE {
 _:percentage
  quant:percentageOf _:total; # set
  quant:percentageThat _:part;
  quant:hasMeasurement [quant:hasValue ?percentageValue; quant:hasFactor
    quant:percent].
 _:total
  quant:counts ?total; # number
  rdfs:subClassOf cao:SAureus, [
   a owl:Restriction; owl:onProperty cao:culturedFrom; owl:someValuesFrom [
    rdfs:subClassOf dco:Sample, [
     a owl:Restriction; owl:onProperty biotop:outcomeOf; owl:someValuesFrom [
      rdfs:subClassOf dco:SampleCollection, [
       a owl:Restriction; owl:onProperty event:during; owl:someValuesFrom [
        dco:hasStartDateTime "2007-01-01T00:00:00"^^xsd:dataTime;
        dco:hasEndDateTime "2008-01-01T00:00:00"^^xsd:dataTime]], [
         a owl:Restriction; owl:onProperty biotop:hasLocus; owl:someValuesFrom
          <https://debugit.spim.jussieu.fr>]]]]].
 _:part
  quant:counts ?part;
  rdfs:subClassOf _:total, [
   a owl:Restriction; owl:onProperty cao:resistantTo; owl:someValuesFrom
    dco:Vancomycin]}
```


debugIT

From data values to formal resources



DB: database
CIS: Clinical Information System
DDO: Data Definition Ontology
CDR: Clinical Data Repository
DO: Domain Ontology
CT: clinical terminology
OO: Operational Ontology
vCDR: virtual Clinical Data Repository - e.g. EPR: Electronic Patient Record
CASQ: Clinical Analysis SPARQL Query
DSSQ: Data Set SPARQL Query
GUI: Graphical User Interface / SQL: Structured Query Language

→ **At development** (deployment) **time**: creating "formal library"
→ **At runtime**: formalizing (via querying) and deducting (via reasoning with rules)
Using formalisms declared in ontologies

debugIT

Data Set SPARQL Query

DSSQ

- **Created at development time:**
  - CONSTRUCT query
  - Expressed in DB specific DDO terms
  - Part of CASQ semantics retrievable from DB
  - Difference between DDOs
- **Automatically invoked from CASQ at runtime:**
  - Retrieving data set from 1 DB as RDF
  - Binding universally quantified variables in WHERE and CONSTRUCT clause

DSSQ: INSERM

DSSQ

Looking for result:
- of a **culture**
- with identified **bacterium**
- tested **antibiotic**
- **susceptibility state**

Culture:
- having **sample type code x** (e.g. wound; SNOMED CT: 258531008)
- having **result date time y** (yyyy-mm-dd/hh:mm:ss)

Bacterium having **code z**

Antibiotic having **code q**

Filtering on:
- x: SNOMED CT codes (list of available)
- y: ≥ 2007-01-01, < 2008-01-01
- z: UniProt 1280 (S. aureus)
- q: ATC J01XA01 (vancomycin)

DSSQ: INSERM

DSSQ

```
...
FILTER (?uniProtCode = "1280"^^biosko:uniProtTaxonomyDT)
  # Staphylococcus aureus
FILTER (?atcCode = "J01XA01"^^clisko:atc20090101DT)
  # vancomycin
FILTER (?sampleTypeCode = "119295008"^^clisko:sct20080731DT
  # Specimen obtained by aspiration
  || ...
  || ?sampleTypeCode = "258531008"^^clisko:sct20080731DT) # wound swab
FILTER ("2007-01-01T00:00:00"^^xsd:dateTime <= ?resultDate
        && ?resultDate < "2008-01-01T00:00:00"^^xsd:dateTime)}

CONSTRUCT {...} # same as WHERE, without filters
```


debugIT

DSSQ: HUG

DSSQ

Looking for an **antibiogram**:

of a culture

with identified bacterium

tested antibiotic

outcome (susceptibility state)

Culture:

having sample type code x

having result date time y (yyyy-mm-dd/hh:mm:ss)

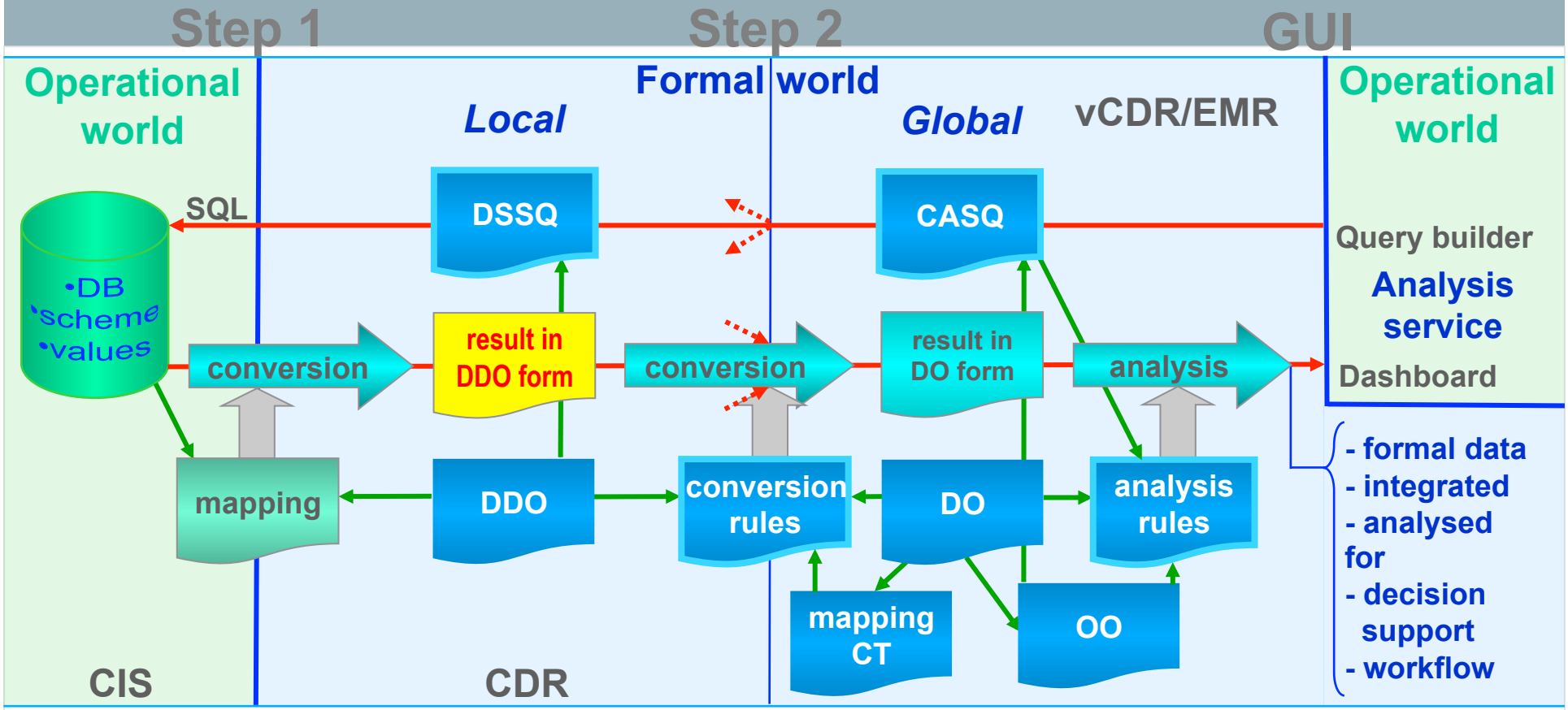
Bacterium having code z

Antibiotic having code q

Filtering on:

- x: SNOMED CT (list of available)
- y: ≥ 2007-01-01, < 2008-01-01
- z: UniProt 1280 (S. aureus)
- q: ATC J01XA01 (vancomycin)

From data values to formal resources



DB: database
CIS: Clinical Information System
DDO: Data Definition Ontology
CDR: Clinical Data Repository
DO: Domain Ontology
CT: clinical terminology
OO: Operational Ontology
vCDR: virtual Clinical Data Repository - e.g. EPR: Electronic Patient Record
CASQ: Clinical Analysis SPARQL Query
DSSQ: Data Set SPARQL Query
GUI: Graphical User Interface / SQL: Structured Query Language

→ **At development** (deployment) **time**: creating "formal library"
→ **At runtime**: formalizing (via querying) and deducting (via reasoning with rules)
Using formalisms declared in ontologies

debugIT

DSSQ result in DDO form: INSERM

result in DDO form

1 example of result set:

Result <https://debugit.spim.jussieu.fr/resource/culture_result/1970194>
  of a culture <https://debugit.spim.jussieu.fr/resource/culture_normalized/56313>
  with identified bacterium <https://debugit.spim.jussieu.fr/resource/bacteria/129>
  tested antibiotic <https://debugit.spim.jussieu.fr/resource/drug/2491>
  susceptibility state <http://debugit.spim.jussieu.fr/ddo#Sensitive>
Culture <https://debugit.spim.jussieu.fr/resource/culture_normalized/56313>
  having sample type code **"119311002"** *(SNOMED CT typed)*
  having result date time **"2007-01-14T23:00:00"** *(XSD typed)*
Bacterium having code **"1280"** *(UniProt typed)*
Antibiotic having code **"J01XA01"** *(ATC typed)*

debugIT

DSSQ result in DDO form: HUG

result in DDO form

1 example of result set:

Antibiogram <https://babar.unige.ch:8443/cdr/resource/Antibiogram/771531>
- **of a culture** <https://babar.unige.ch:8443/cdr/resource/Culture/729161>
- **with identified bacterium** <https://babar.unige.ch:8443/cdr/resource/Bacterium/260>
- **tested antibiotic** <https://babar.unige.ch:8443/cdr/resource/Drug/182>
- **outcome "131196009"** *(SNOMED CT typed)*

Culture <https://debugit.spim.jussieu.fr/resource/culture_normalized/56313>
- **having sample type code "119311002"** *(SNOMED CT typed)*
- **having result date time "2007-01-14T23:00:00"** *(XSD typed)*

Bacterium having code "1280" *(UniProt typed)*

Antibiotic having code "J01XA01" *(ATC typed)*

debugIT

Reasoner

- **EYE: Euler Yap Engine**
  - Backward-chaining reasoner enhanced with Euler path detection
  - Adhering coherent logic (close to first order logic)
  - Inference engine supporting logic based proofs
  - Implementation in Java/Prolog (logic programming language)
  - Open source: http://eulersharp.sourceforge.net/
  - **60 million lips** (logical inferences / second) on an Intel Core Duo 2.2 GHz

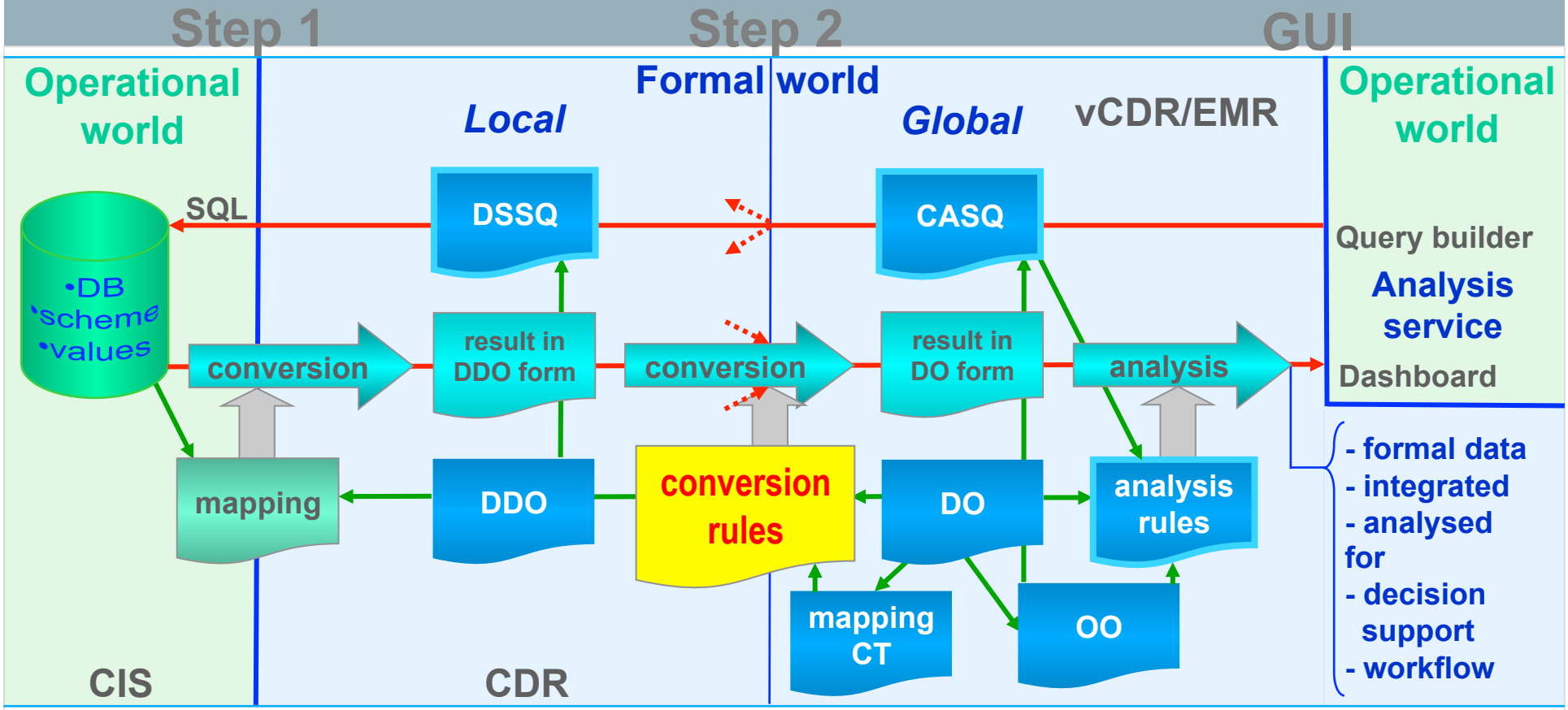
Reasoner: deep taxonomy benchmark

- http://ruleml.org/WellnessRules/files/WellnessRulesN3-2009-11-10.pdf

Depth	colog	cwm	eye	eulerj	jDREW	jena	pellet
10	0.007	0.071	**0.000**	0.006	0.004	0.121	0.075
100	0.511	1.449	**0.004**	0.179	0.172	0.783	0.442
1000	**500.600**	**115.820**	**0.040**	**3.907**	**98.467**	**29.330**	**38.836**
10000	498137.000	16016.625	**0.436**	155.710	91614.000	(outOfMem)	(outOfMem)
100000	*16 year*		**4 sec**		*4 year*		

Unit: seconds if not indicated otherwise

From data values to formal resources



DB: database
CIS: Clinical Information System
DDO: Data Definition Ontology
CDR: Clinical Data Repository
DO: Domain Ontology
CT: clinical terminology
OO: Operational Ontology
vCDR: virtual Clinical Data Repository - e.g. EPR: Electronic Patient Record
CASQ: Clinical Analysis SPARQL Query
DSSQ: Data Set SPARQL Query
GUI: Graphical User Interface / SQL: Structured Query Language

→ **At development** (deployment) **time**: creating "formal library"
→ **At runtime**: formalizing (via querying) and deducting (via reasoning with rules)
Using formalisms declared in ontologies

debugIT

Conversion rules

- **Bridging the semantic gap:**
  - Local → global semantics: DDO → DO
  - Set for every DDO created at development time
- **Executed at runtime using a reasoner (EYE) within the query rewriting mechanism -> permits in some extent query entailement at runtime and result inference**

debugIT

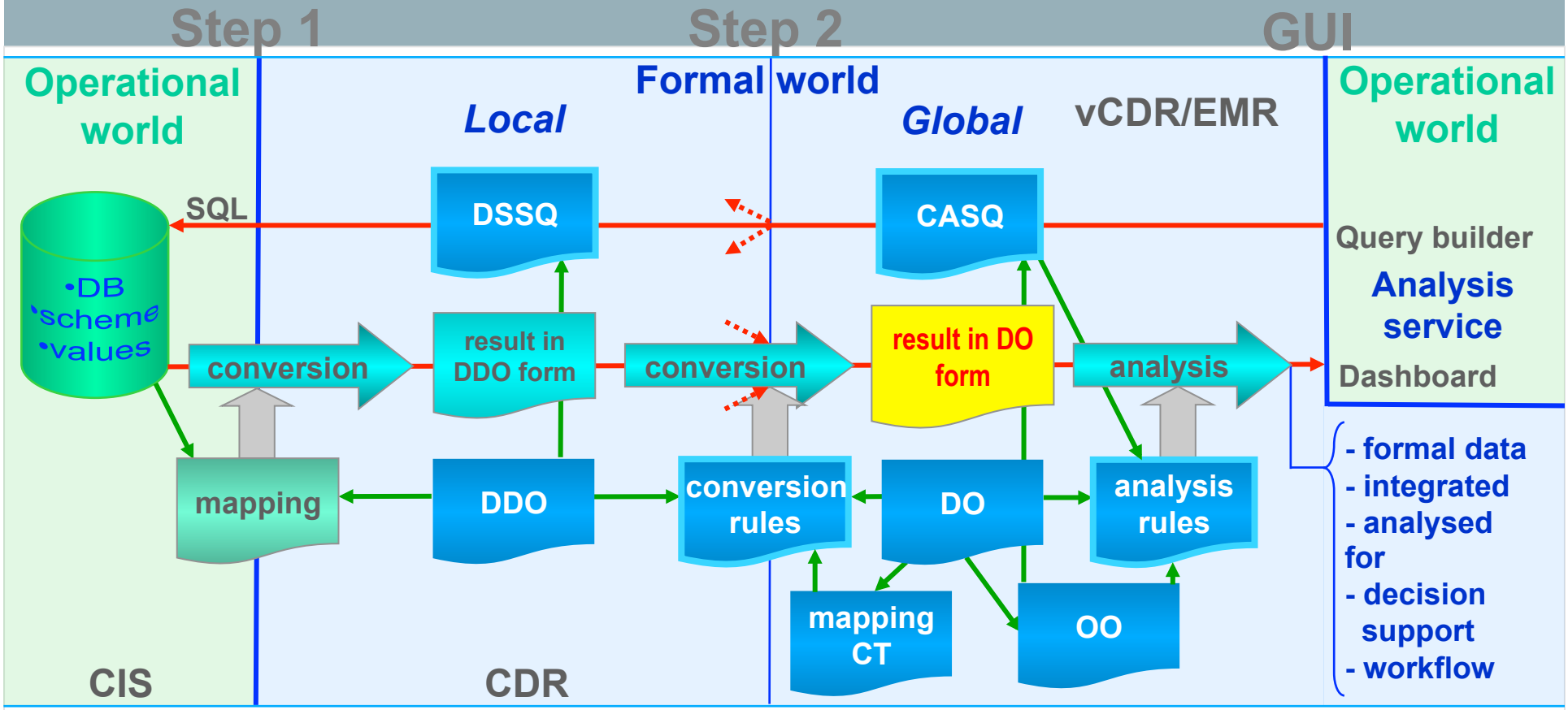
Conversion rules

conversion rules

E.g. 1
DDO expression converted to DO
not having identical but related semantics and mappable,
resp. DDO: culture results and
DO: susceptibility test result = antibiogram,
with susceptibility state as 1 of the 3 result items,
besides bacterium and antibiotic

E.g. 2
Needed elements for desired expressions absent in DDO:
e.g. evaluation of antibiogram,
combining 3 result items + time indicator
→ deducting semantics from other conversions

From data values to formal resources



DB: database
CIS: Clinical Information System
DDO: Data Definition Ontology
CDR: Clinical Data Repository
DO: Domain Ontology
CT: clinical terminology
OO: Operational Ontology
vCDR: virtual Clinical Data Repository - e.g. EPR: Electronic Patient Record
CASQ: Clinical Analysis SPARQL Query
DSSQ: Data Set SPARQL Query
GUI: Graphical User Interface / SQL: Structured Query Language

→ **At development** (deployment) **time**: creating "formal library"
→ **At runtime**: formalizing (via querying) and deducting (via reasoning with rules)
Using formalisms declared in ontologies

debugIT

Result in DO form

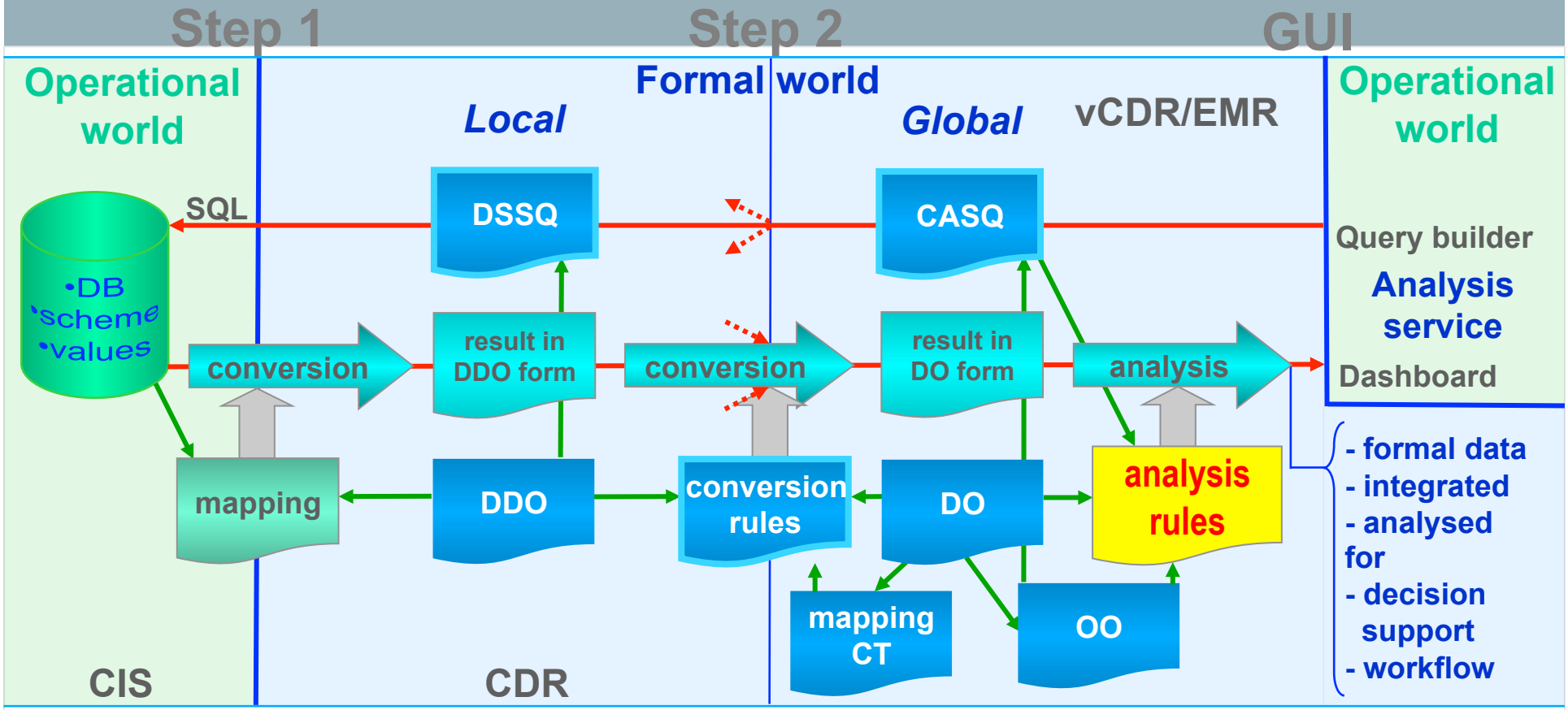
result in DDO form

1 example of result set:

A bacterial antibiogram *analysis (evaluation)*
having participant <https://debugit.spim.jussieu.fr/resource/culture_result/1970194>
having an outcome
encoding a microbiological susceptibility
having quality located <http://purl.org/imbi/dco/dco#Sensitive>

<https://debugit.spim.jussieu.fr/resource/culture_result/1970194> ***(antibiogram)***
spatially related to a taxon quality
having quality located <...dco#SpeciesStaphylococcusAureusValueRegion>
encoding <...dco#Vancomycin>

From data values to formal resources



DB: database
CIS: Clinical Information System
DDO: Data Definition Ontology
CDR: Clinical Data Repository
DO: Domain Ontology
CT: clinical terminology
OO: Operational Ontology
vCDR: virtual Clinical Data Repository - e.g. EPR: Electronic Patient Record
CASQ: Clinical Analysis SPARQL Query
DSSQ: Data Set SPARQL Query
GUI: Graphical User Interface / SQL: Structured Query Language

→ **At development** (deployment) **time**: creating "formal library"
→ **At runtime**: formalizing (via querying) and deducting (via reasoning with rules)
Using formalisms declared in ontologies

debugIT

Analysis rules

- Integrated data from different DBs
- Processing globally formalized data
- E.g. calculating percentage

"What is the percentage of S. Aureus cases - cultured from all samples collected in 2007 at INSERM and HUG hospital - that is resistant to vancomycin?"

Analysis rules

analysis rules

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix e: <http://eulersharp.sourceforge.net/2003/03swap/log-rules#>.
@prefix event: <http://eulersharp.sourceforge.net/2003/03swap/event#>.
@prefix quant: <http://eulersharp.sourceforge.net/2003/03swap/quantities#>.
@prefix biotop: <http://purl.org/biotop/biotop.owl#>.
@prefix dco: <http://purl.org/imbi/dco/dco#>.
@prefix cao: <http://www.agfa.com/w3c/2009/clinicalAnalysisOntology#>.
```

Analysis rules: antecedent

analysis rules

{ Search for all susceptibility states (R, S, I)
of all cases of S.aureus
for vancomycin,
and put them in a list.

Search for all states 'resistant'
of all cases of S.aureus
for vancomycin,
and put them in a list.

Calculate from lists percentage of S.aureus cases
that have 'resistant' state
for vancomycin,
rounded to 1 decimal}

Analysis rules: conclusion

analysis rules

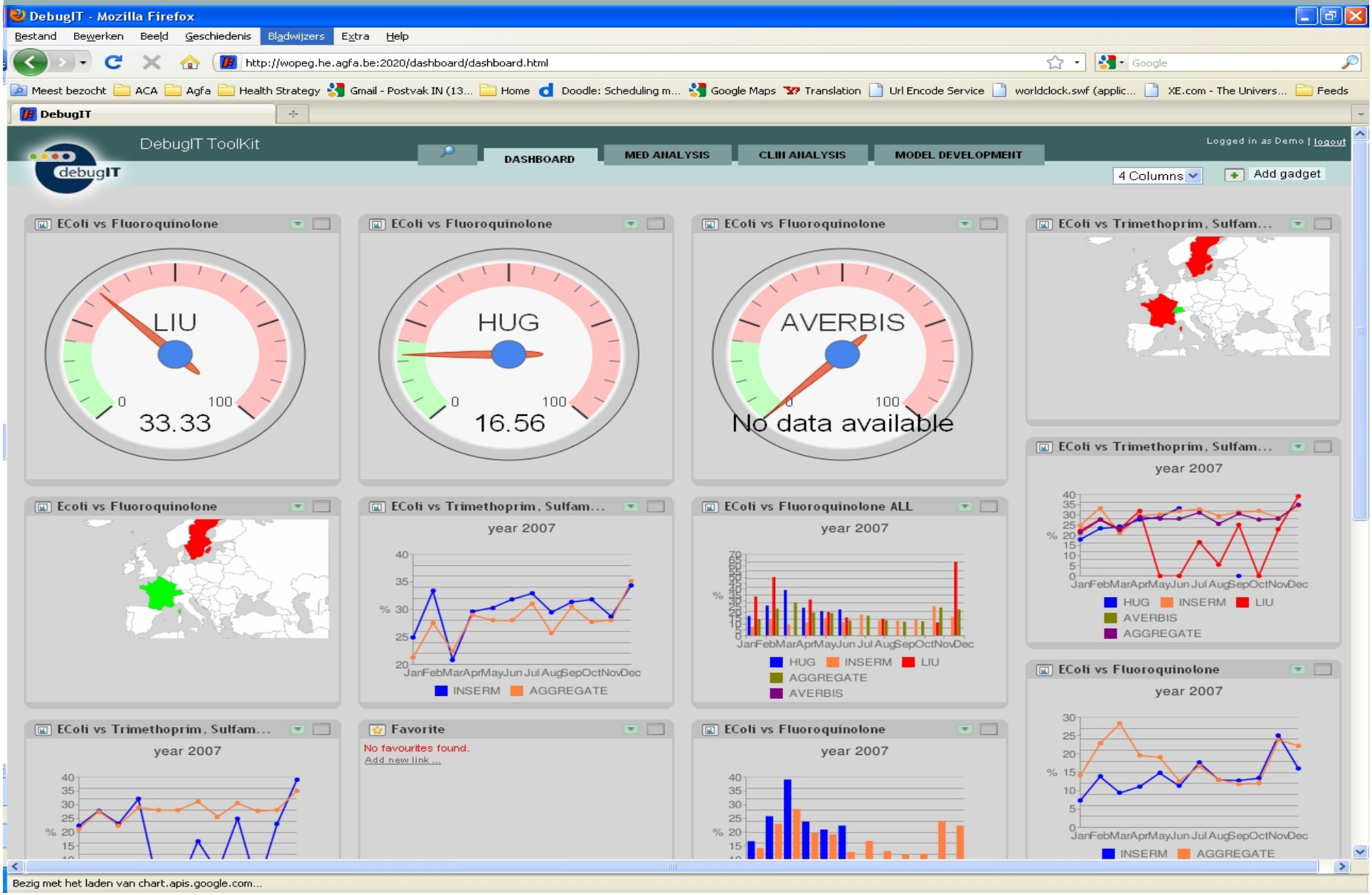
```
=> # same form as CASQ CONSTRUCT
{_:percentage
  quant:percentageOf _:total;
  quant:percentageThat _:part;
  quant:hasMeasurement [quant:hasValue ?percentageValue; quant:hasFactor
    quant:percent].
 _:total
  quant:counts ?total;
  rdfs:subClassOf cao:SAureus, [
   a owl:Restriction; owl:onProperty cao:culturedFrom; owl:someValuesFrom [
    rdfs:subClassOf dco:Sample, [
     a owl:Restriction; owl:onProperty biotop:outcomeOf; owl:someValuesFrom [
      rdfs:subClassOf dco:SampleCollection, [
       a owl:Restriction; owl:onProperty event:during; owl:someValuesFrom [
        dco:hasStartDateTime "2007-01-01T00:00:00"^^xsd:dataTime;
        dco:hasEndDateTime "2008-01-01T00:00:00"^^xsd:dataTime]], [
         a owl:Restriction; owl:onProperty biotop:hasLocus; owl:someValuesFrom
           <https://debugit.spim.jussieu.fr>]]]]].
 _:part
  quant:counts ?part;
  rdfs:subClassOf _:total, [
   a owl:Restriction; owl:onProperty cao:resistantTo; owl:someValuesFrom
    dco:Vancomycin]}.
```

Analysis rules: result

analysis rules

```
Percentage
    of Staphylococcus aureus cases
        cultured from all samples
            resulting from a collection
                during period
                    starting 2007-01-01/00:00:00,
                    ending 2008-01-01/00:00:00
                in Inserm hospital
    counting 39 members;
percentage that is resistant to vancomycin
    counting 0 members;
having measurement with
    value 0.0
    factor 'percent'
```

Resistance Comparison & Monitoring



What did we do?

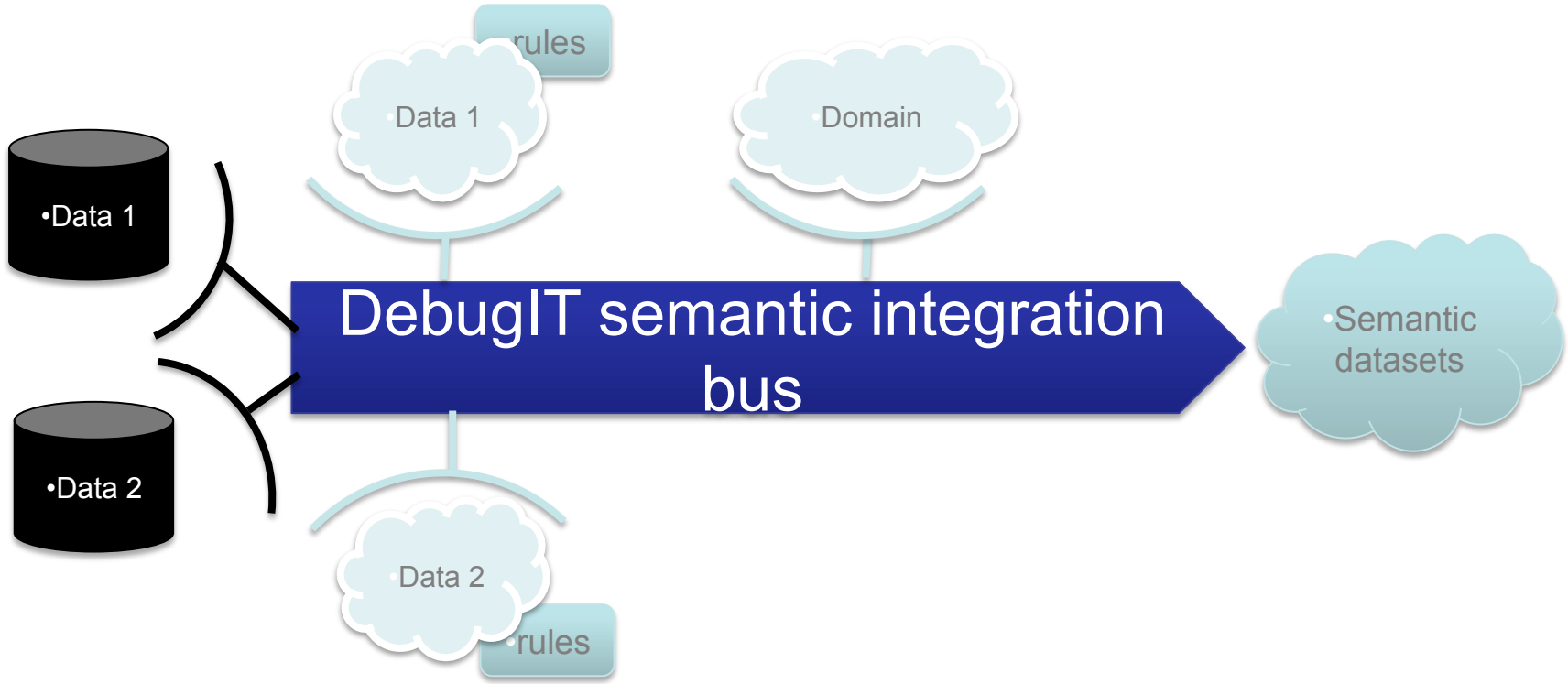
- We tried to follow a **pragmatic** approach by leaving things where they belong (data / concepts) but **formalizing** them so the **machine** can use them to **exchange data** and knowledge and optionally, to infer new knowledge

- The formal semantics does not resolve everything (= the machine does not interpret implicit knowledge yet) but it is a step towards it

- We did not follow the strict standardization route although we use standards to limit the wideness of the problem

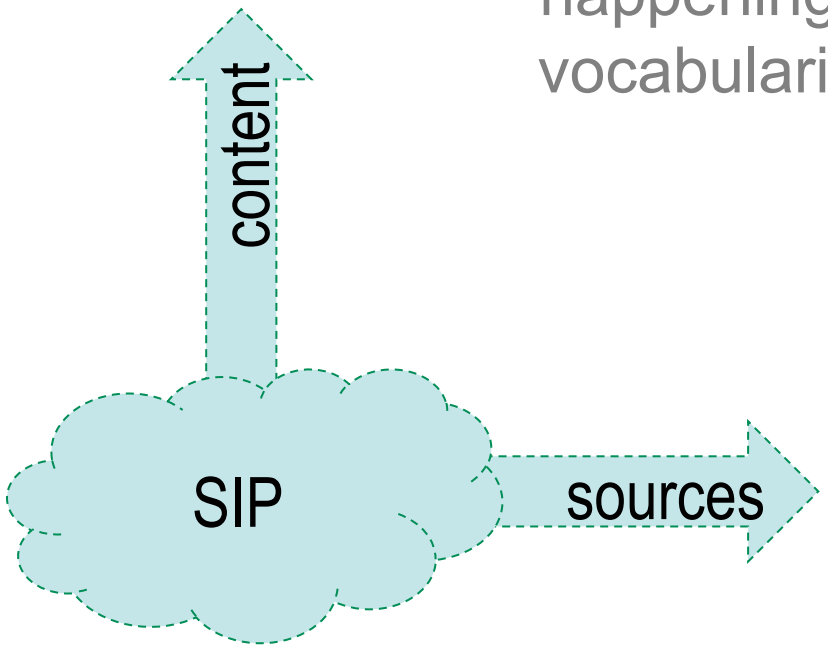
Independent platform

Declarative world



Scalability and interoperability

- The more liberty you give to express things, the less your system is performing and scalable
- We have set a framework where scalability is happening at both vertical (questions, data, vocabularies) and horizontal (data sources) levels



debugIT