



VA
HEALTH
CARE

Defining
EXCELLENCE
in the 21st Century



Linked Vitals

*A Linked Data Approach to
Semantic Interoperability*

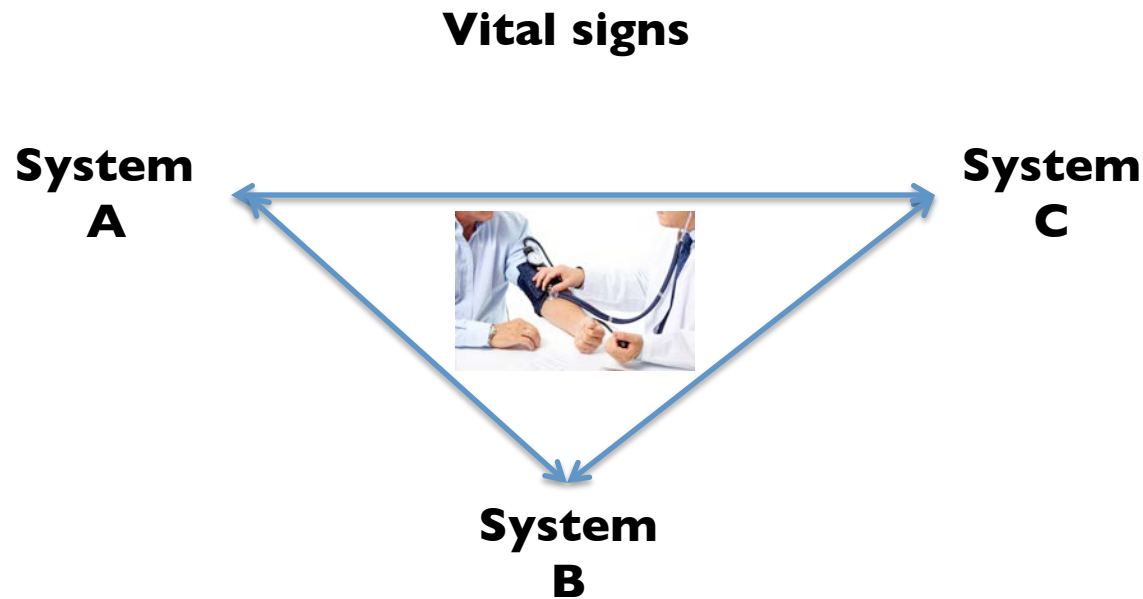
*SemTech Conference
San Jose, CA
August 20, 2014*

Rafael M Richards MD MS
*Physician Informaticist
Office of Informatics and Analytics
Veterans Health Administration
U.S. Department of Veterans Affairs*



Problem Statement: General

How do we integrate data such as vital signs *meaningfully* from different information systems?

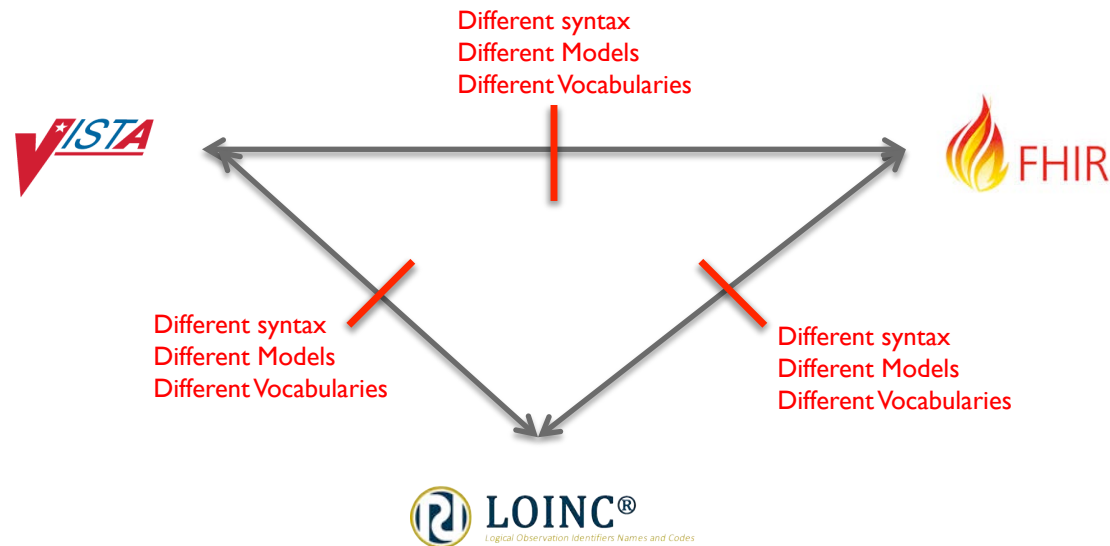




Problem Statement: Specific



How do we integrate vital signs between VistA and FHIR each with different syntax, different models, and different vocabularies ?



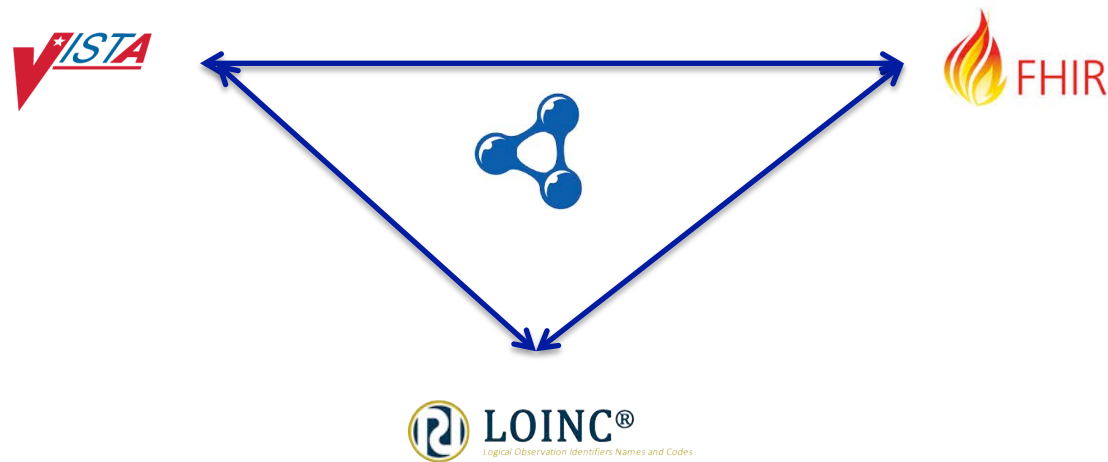
Language barriers to exchange



Solution Statement



How do we integrate vital signs between VistA and FHIR each with different syntax, different models, and different vocabularies ?



A common exchange language



Overview of Translation Process



Source
Data

Syntatic
alignment



Semantic
alignment

Integrated
Data

Syntax A
Model A
Vocabulary A



FM
Schema



Vocabulary B



Syntax C
Model C
Vocabulary C



**Shared Syntax
Shared Model
Shared Vocabulary
Shared Meaning**

Different Syntax
Different Models
Different Vocabularies

Common syntax within
model-flexible medium
(Linked Data)

Rule-based mapping
Model alignment
Vocabulary alignment



Linked Data (RDF): What is it?



The World Wide Web Consortium (W3C) standard for semantic information integration for the Internet of Data.



HTML (hypertext markup language)
For **humans** to exchange information

enables
→

Linked Documents
(Document Web)



RDF (resource description framework)
For **computers** to exchange information

enables
→

Linked Data
(Semantic Web)



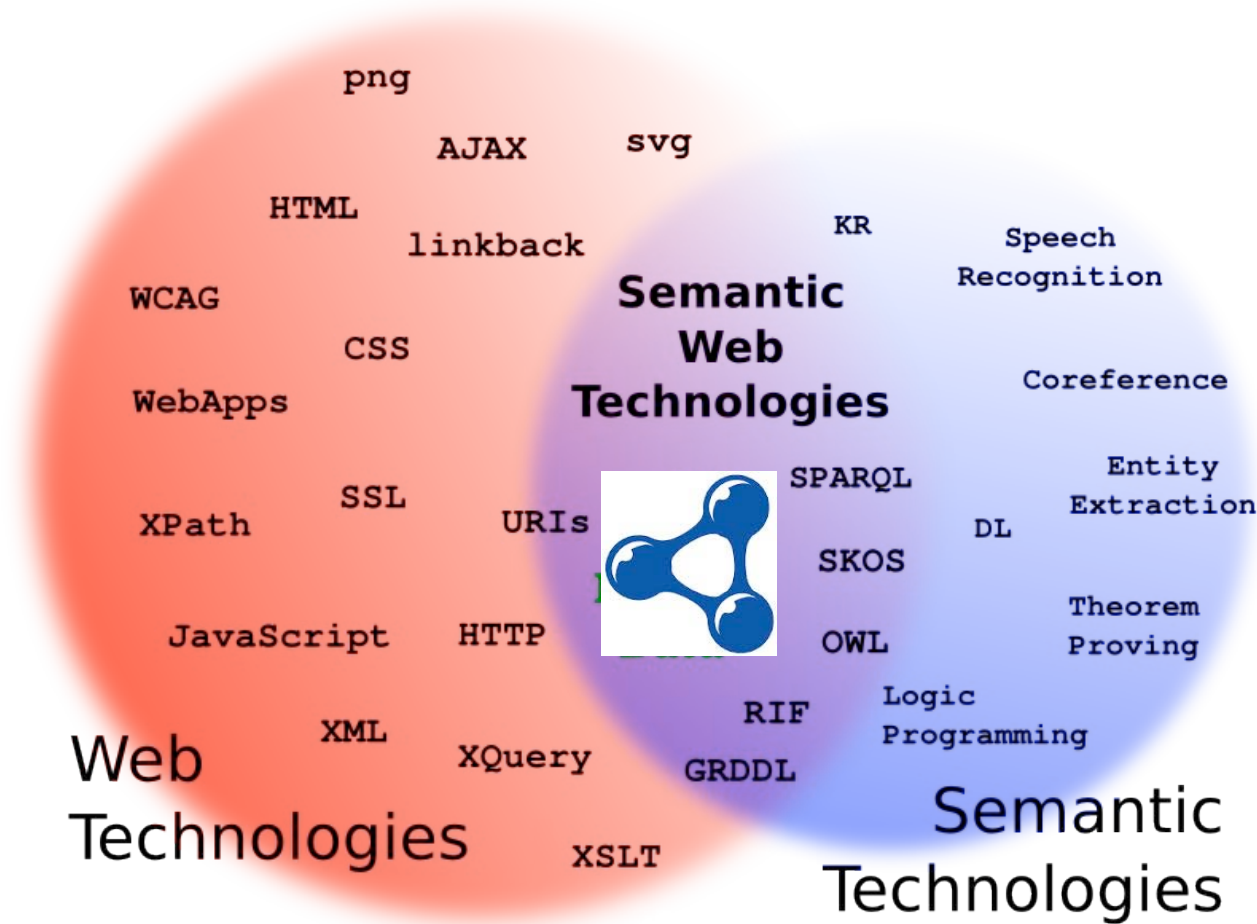
“The Semantic Web [Linked Data] provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.”

Tim Berners-Lee, MIT Professor and Inventor of the World Wide Web (HTML and RDF protocols)



Linked Data:

Intersection of Semantic Technologies and Web Technologies

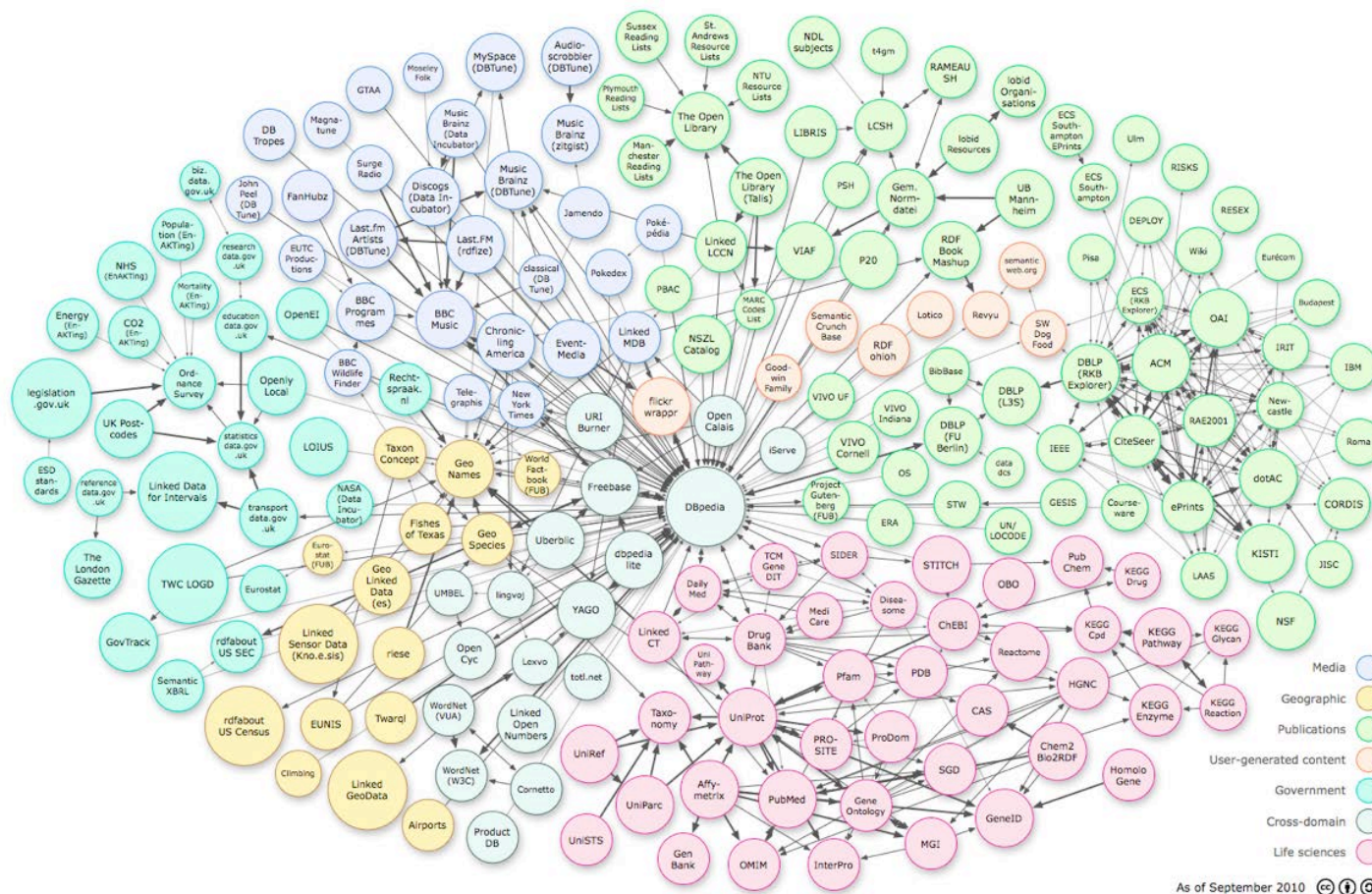




Linked Data and the Internet of Data



Linking media, geographic, publications, government, and life sciences ...



As of September 2010



Linked Data

This represents over 300 linked data sources and databases, comprising billions of data elements and millions of semantic links.

Each one of these circles represents a data source, which is semantically linked to other data sources, creating one virtual federated queryable web of data.

Wikipedia is one of the resources converted to Linked Data, and is called DBpedia (center circle).

Why not linked healthcare?



What is VISTA?



- Veterans Information Systems and Technology Architecture
- Information system of all VA care sites
- Foundation of other public health information systems
 - VA (VISTA): 1200+ care sites
 - DoD(CHCS): 900+ care sites
 - IHS(RPMS): 500+ care sites
 - NY State: 24 hospitals
- Most physicians in U.S. have used VISTA
- Open source
 - Deployed in many other settings in U.S. and internationally
 - Many developments by open source community



VISTA in the U.S.





VistA Architecture: Overview



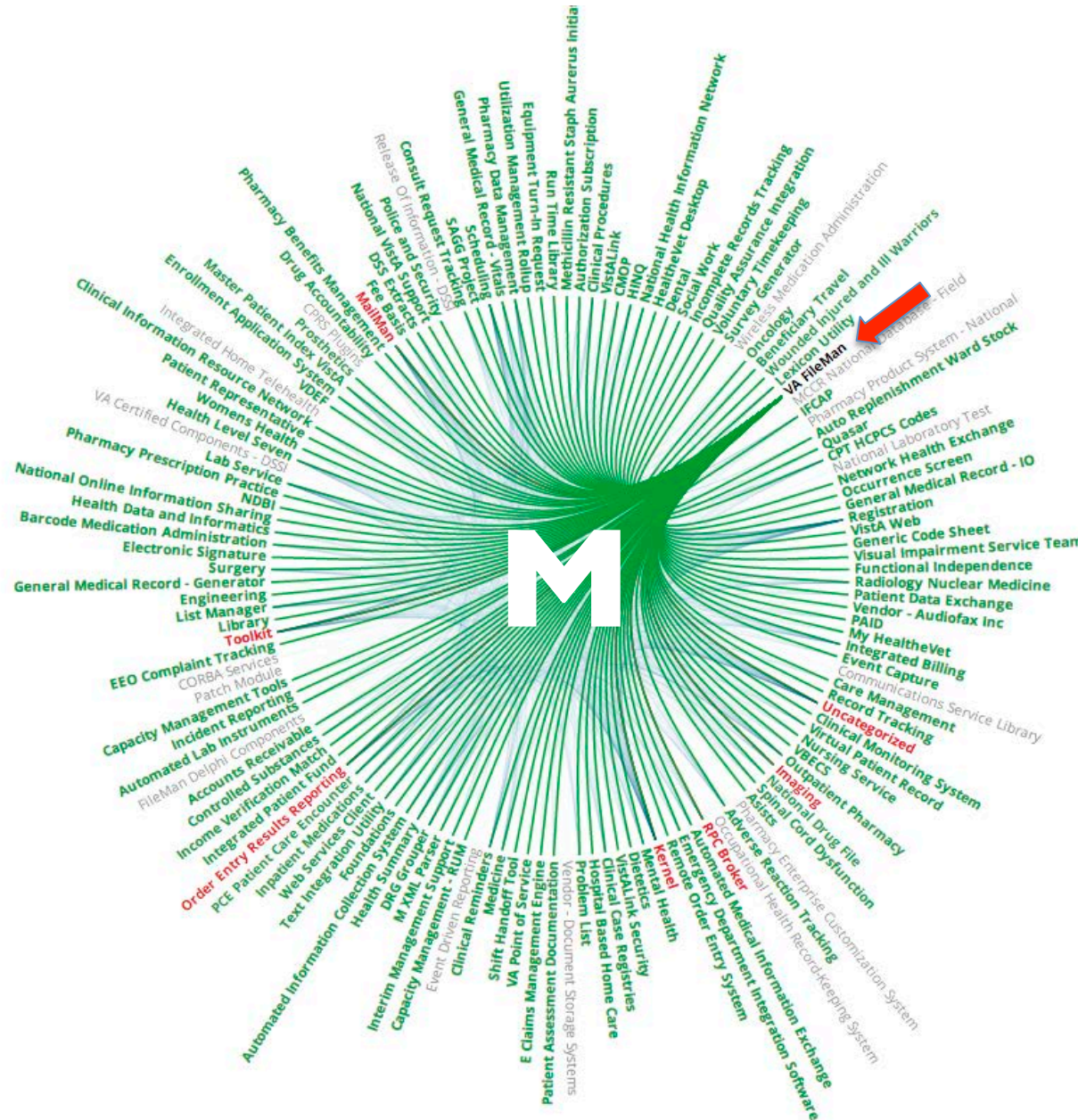
All 150+ VistA applications are integrated within one shared multidimensional data engine.



FMQL is the Fileman Query Language

Fileman is VistA's database management system

M is VistA's application-data integration engine.



VistA's integrated data engine.
All 150+ applications are integrated to their data inside the multidimensional data engine, represented by the inner circle.



Exposing VistA's native data model



Objective: Make machine-processable schemas, vocabularies, and datasets from VISTA

<http://vista.caregraf.info/analytics>

Challenge: Poor RDF/OWL model

Solution: Refine RDF/OWL model (example here)



Exposing VistA's native data model



FileMan Schema Browser

Populated Files

In this system 1245 out of 2356 have entries.

#	Name	Global	Count
.11	INDEX	^DD("IX",	644
.31	KEY	^DD("KEY",	64
.4	PRINT TEMPLATE	^DIPT(1234
.401	SORT TEMPLATE	^DIBT(773
.402	INPUT TEMPLATE	^DIE(1519
.403	FORM	^DIST(.403,	144
.404	BLOCK	^DIST(.404,	519
.44	FOREIGN FORMAT	^DIST(.44,	11
.5	FUNCTION	^DD("FUNC",	169
.7	MUMPS OPERATING SYSTEM	^DD("OS",	7
.81	DATA TYPE	^DI(.81,	11
.84	DIALOG	^DI(.84,	2574
.85	LANGUAGE	^DI(.85,	11
1	FILE	^DIC(2356
1.2	ALTERNATE EDITOR	^DIST(1.2,	4
2	PATIENT	^DPT(40
3.07	PROGRAMMER MODE LOG	^%ZUA(3.07,	930

VistA's native data model is comprised of hierarchical files and subfiles, each which addresses a specific M Global storage.



Fileman Query Language (FMQL)



Query: DESCRIBE 120_5 FILTER(.02=2-9) LIMIT 1

Format: HTML JSON RDF JSON-LD

Send Query

FMQL is the Fileman Query Language. This provides real-time web-based query access to the entirety of VistA's data.

It exposes the native hierarchical data model of Fileman in web standard forms including HTML, JSON, and RDF.

HTML: hypertext markup language (visual document markup)
JSON: javascript object notation (data serialization / packaging)
RDF: resource description framework (linked data / semantics)
JSON-LD: JSON with linked data capability





VistA Vitals: HTML output



FMQL query of VistA for vital signs with output in **HTML**.

DESCRIBE 120_5 FILTER(.02=2-9) LIMIT 2

FMQL Query Maker

Query a live VistA!

Help? [Caregraf Support](#)

Query:

Format:

1. GMRV VITAL MEASUREMENT > 2005-09-01T13:00:00Z (1)

date time vitals taken
2005-09-01T13:00:00Z
patient
PATIENT/JONES,CHRISTOPHER
vital type
GMRV VITAL TYPE/BLOOD PRESSURE (VA:4500634)
date time vitals entered
2005-12-28T13:48:44Z
hospital location
HOSPITAL LOCATION/4 SOUTH - MED
entered by
NEW PERSON/NOTHER,NADA (LOCAL)
rate
150/10
entered in error
true
error entered by
NEW PERSON/MANAGER,SYSTEM (LOCAL)
reason entered in error



**HTML output:
Human-readable**



VistA Vitals: RDF output



FMQL query of VistA for vital signs with output in **RDF**.

DESCRIBE 120_5 FILTER(.02=2-9) LIMIT 1

FMQL Query Maker

Query a live VistA!

Help? [Caregraf Support](#)

Query:

Format:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:vs="http://datasets.caregraf.org/vs/"
  xmlns:fms="http://datasets.caregraf.org/fms/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
>

  <rdf:Description rdf:about="http://livevista.caregraf.info/120_5-1">
    <rdf:type rdf:resource="http://datasets.caregraf.org/vs/120_5"/>
    <rdfs:label>2005-09-01T13:00:00Z</rdfs:label>
    <vs:date_time_vitals_taken-120_5 rdf:datatype="xsd:dateTime">2005-
    <vs:patient-120_5 rdf:resource="http://livevista.caregraf.info/2-
    <vs:vital_type-120_5 rdf:resource="http://livevista.caregraf.info
    <vs:date_time_vitals_entered-120_5 rdf:datatype="xsd:dateTime">20
    <vs:hospital_location-120_5 rdf:resource="http://livevista.caregr
    <vs:entered_by-120_5 rdf:resource="http://livevista.caregraf.info
    <vs:rate-120_5>150/10</vs:rate-120_5>
    <vs:entered in error-120 5 rdf:datatype="xsd:boolean">true</vs:en
```



**RDF output:
Machine readable**



VistA Vitals in RDF



The screenshot displays a web application interface for managing RDF classes. On the left, a 'Navigator' pane shows a hierarchical tree of classes. The tree starts with 'rdfs:Resource', followed by 'dtype:numericUnion', 'owl:Thing', 'cg:Thing', and 'cg:PatientResource'. Under 'cg:PatientResource', a list of URIs is shown, each with a count in parentheses. The URI '<http://datasets.caregraf.org/vs/120_5>' (239) is highlighted with a blue background. On the right, the 'Class Form' pane shows details for the selected class. It includes a 'Name' field with the URI '<http://datasets.caregraf.org/vs/120_5>'. Below this, the 'Annotations' section shows 'rdfs:label' with the value 'GMRV VITAL MEASUREMENT'. The 'Class Axioms' section shows 'rdfs:subClassOf' with the value 'cg:PatientResource', 'owl:equivalentClass', 'owl:disjointWith', and 'owl:hasKey'. An 'Other Properties' section is also visible at the bottom.

Navigator

- Classes
- rdfs:Resource
 - dtype:numericUnion
 - owl:Thing
 - cg:Thing
 - cg:EnumConcept
 - cg:InternalConcept
 - cg:OutsideConcept
 - cg:PatientResource
 - <http://datasets.caregraf.org/vs/100_008> (259)
 - <http://datasets.caregraf.org/vs/100_045> (1606)
 - <http://datasets.caregraf.org/vs/100_09> (6)
 - <http://datasets.caregraf.org/vs/100> (240)
 - <http://datasets.caregraf.org/vs/102_3> (1)
 - <http://datasets.caregraf.org/vs/102_4> (178)
 - <http://datasets.caregraf.org/vs/120_5> (239)**
 - <http://datasets.caregraf.org/vs/120_8> (12)
 - <http://datasets.caregraf.org/vs/120_81> (15)
 - <http://datasets.caregraf.org/vs/120_813> (2)
 - <http://datasets.caregraf.org/vs/120_826> (2)

Class Form

Name: <http://datasets.caregraf.org/vs/120_5>

Annotations

rdfs:label
GMRV VITAL MEASUREMENT

Class Axioms

rdfs:subClassOf
cg:PatientResource

owl:equivalentClass

owl:disjointWith

owl:hasKey

Other Properties

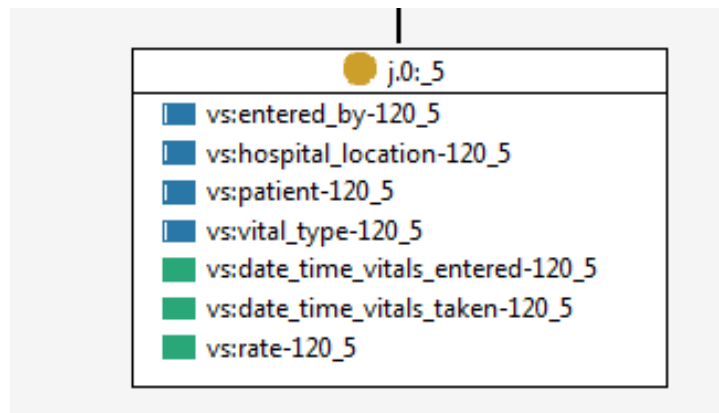
239 instances in the sample dataset



VistA Vitals in RDF



Exposing VistA's intrinsic data model for vitals



bp-vista.rdf

Object Property Form

Name: vs:vital_type-120_5

▼ Annotations

▼ Property Axioms

rdfs:domain ▼
j.0:_5

rdfs:range ▼

rdfs:subPropertyOf ▼

owl:equivalentProperty ▼

owl:inverseOf ▼

owl:propertyDisjointWith ▼

owl:propertyChainAxiom ▼

▼ Other Properties

rdf:type ▼
owl:ObjectProperty



VistA Vitals in RDF

- Native VistA in RDF did not contain “common concepts” (so we created some).
- URIs are “messed up” (so we fixed them)

vs:vital_type-120_5 ▾

✚ <http://livevista.caregraf.info/120_51-1> ▾

rdfs:label ▾
BLOOD PRESSURE ▾

rdf:type ▾
j.0:_51 ▾

owl:sameAs ▾
✚ <http://datasets.caregraf.org/va/4500634> ▾

rdfs:label ▾
BLOOD PRESSURE ▾

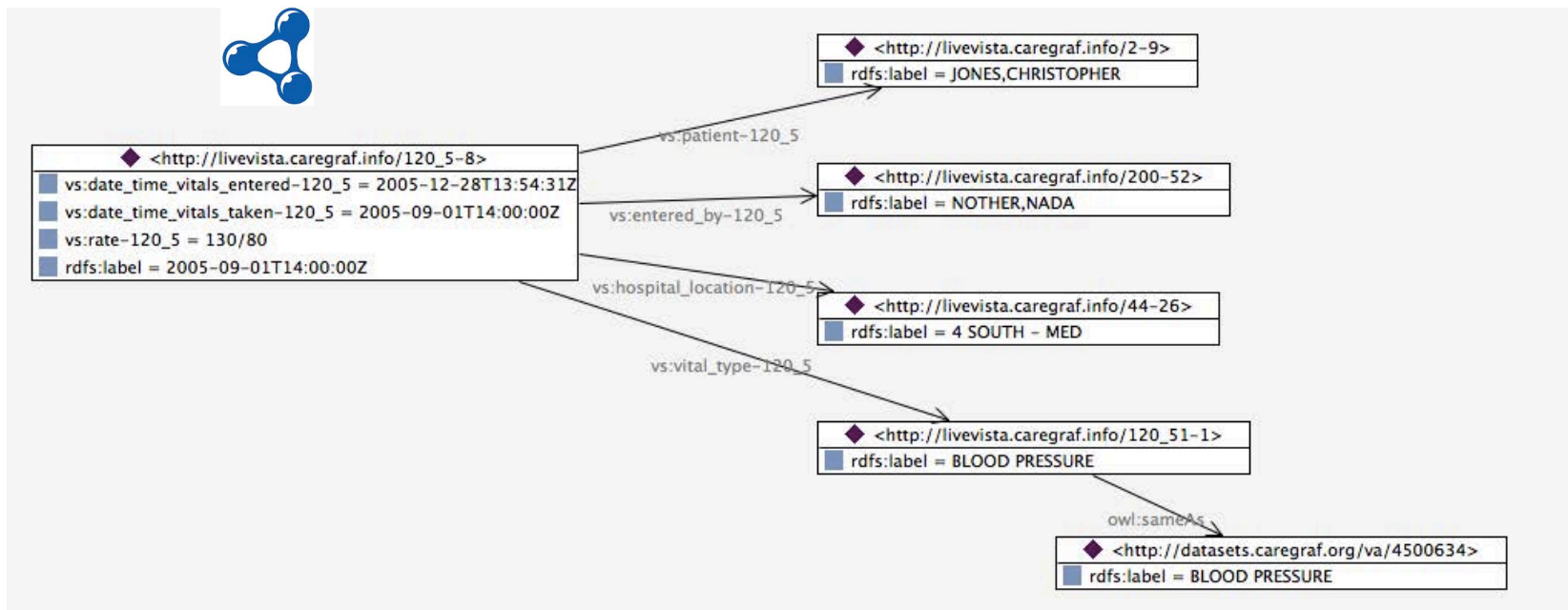
rdf:type ▾
fms:CommonConcept ▾

vs:units ▾
S mm[Hg] ▾



VistA Vitals in RDF

Exposing VistA's native data model for vitals



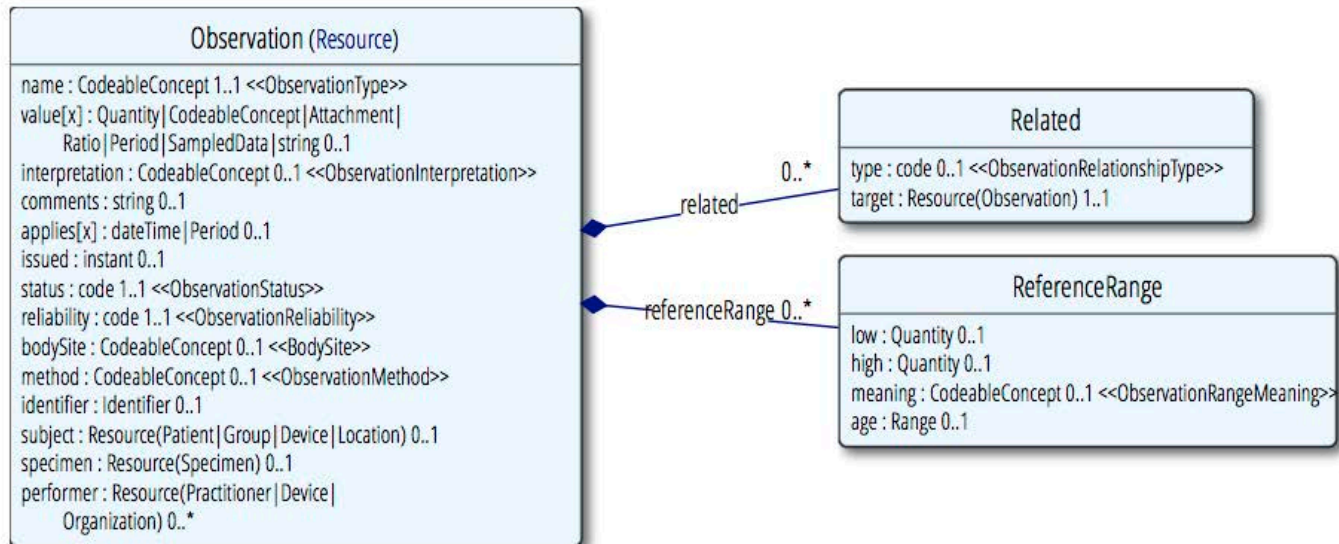


FHIR: Native model



- FHIR - Observation
- XML model in XML Schema

4.15.3 Resource Content

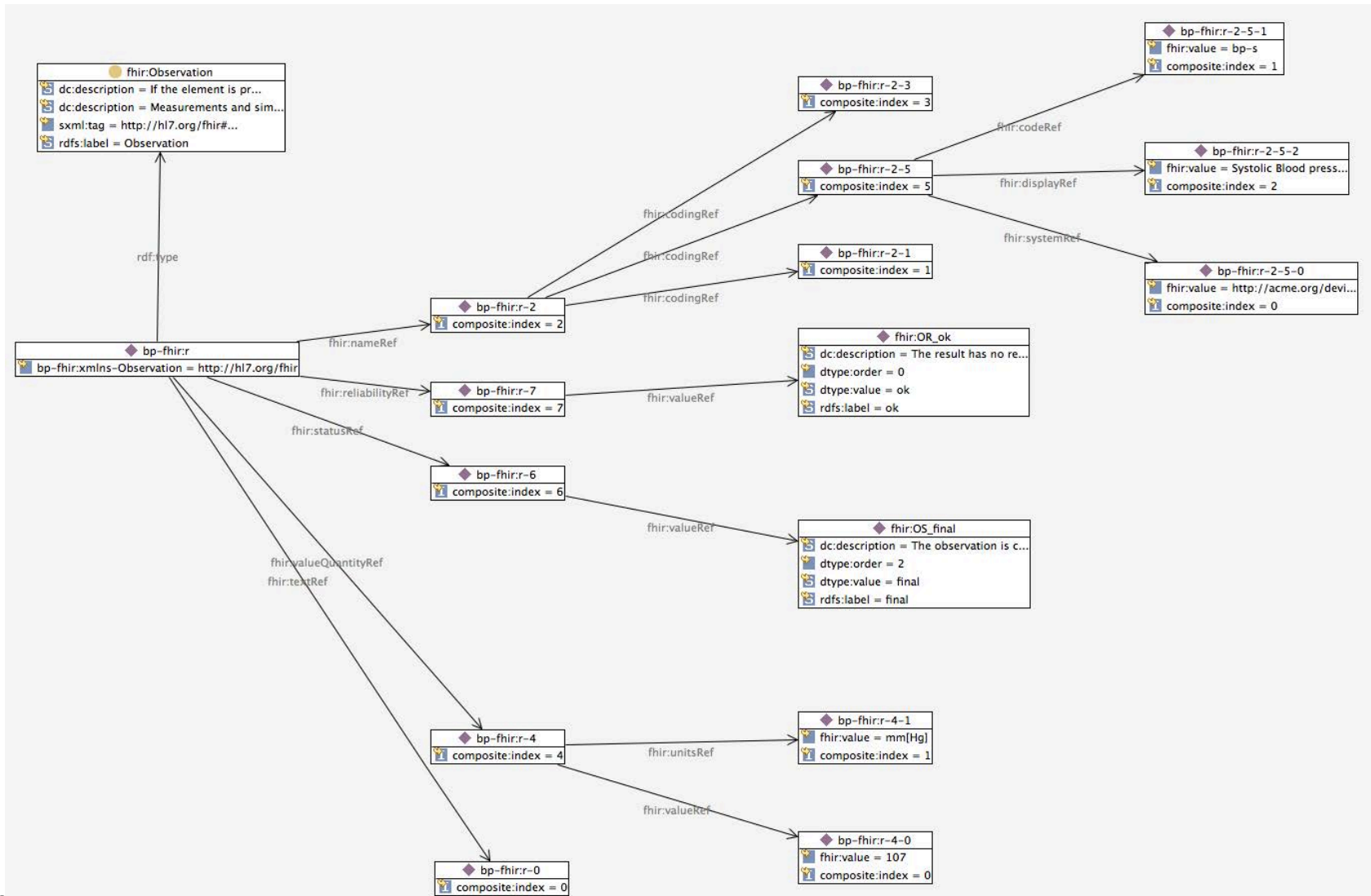




FHIR Vitals in RDF Enhanced

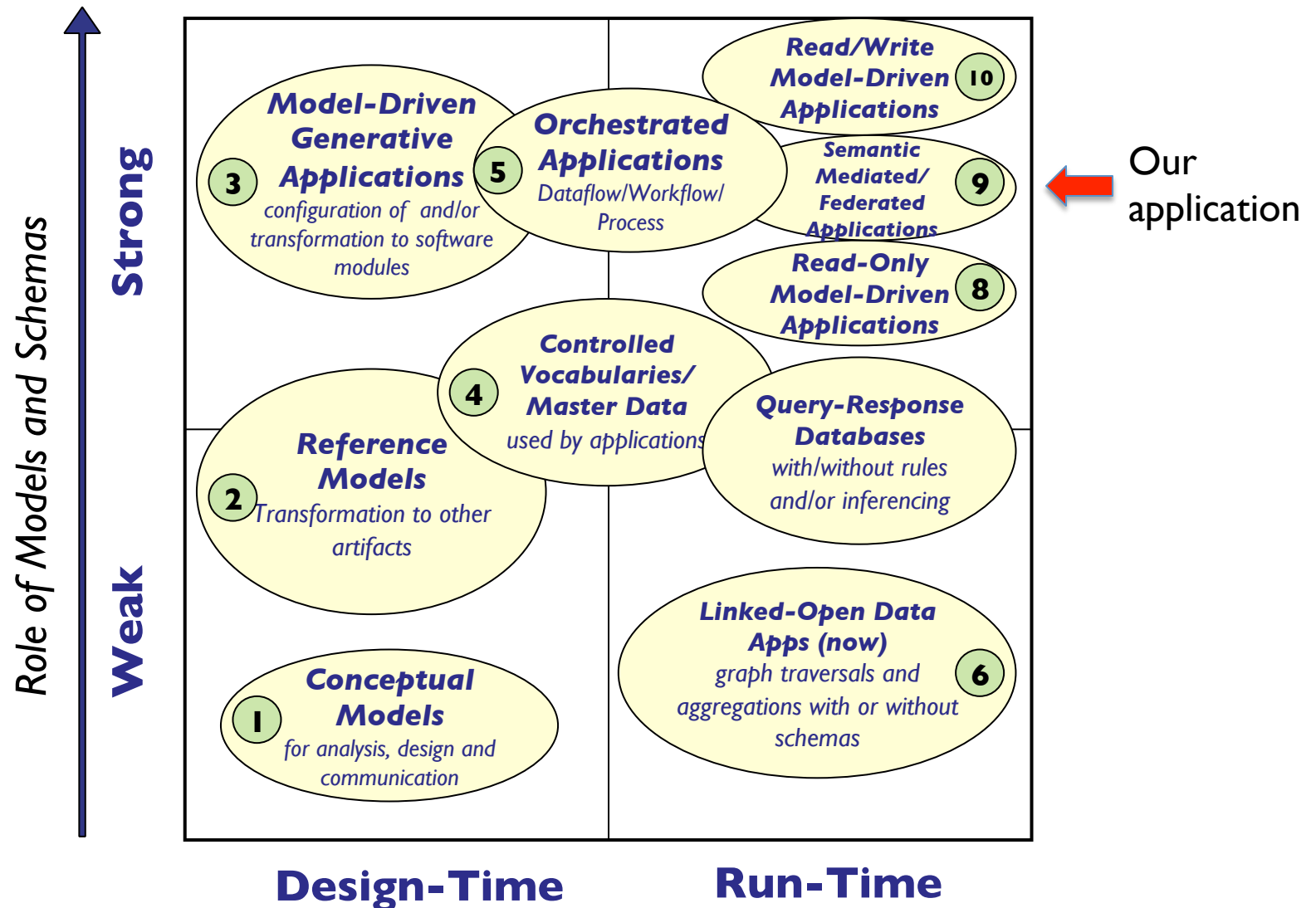


Enhanced bridge model for FHIR in RDF





Many ways of using “Ontologies”

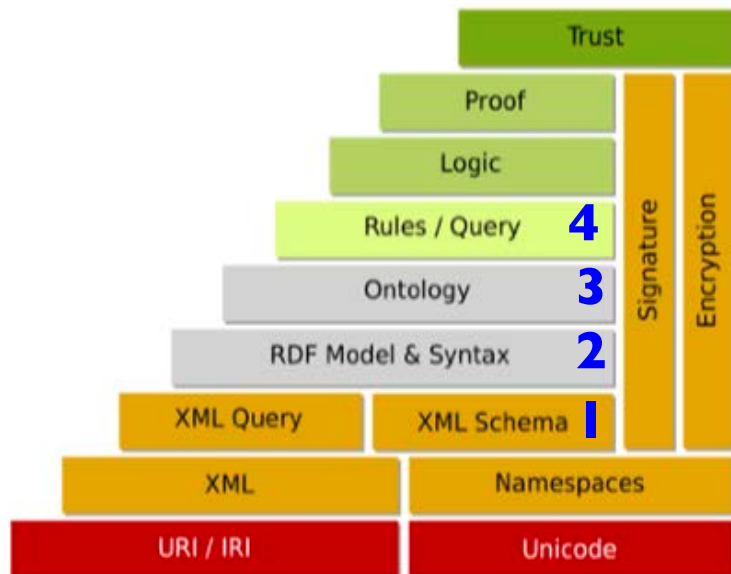




Context of Translation: RDF Technology Stack



- FHIR: XML Schema -> RDF **(1->2)**
- VISTA: Fileman model -> RDF **(2)**
- LOINC: CSV -> RDF **(2)**
- Create bridge model **(3)**
- Merge models and terminology: SPIN Map **(4)**



1. FHIR native model
(XML Schema)
2. Translation
(RDF Schema)
3. Bridge schema
(OWL)
4. SPARQL rules
(SPINMap)



SPIN: SPARQL Inferencing Notation



- A W3C candidate standard
- A SPARQL Rules Language
- Builds on top of the SPARQL query language
- Enables exchangeable rules and transformations

SPIN is a SPARQL Rules Language

The property `spin:rule` can be used to link a class with SPARQL CONSTRUCT queries that define inference rules for the members of the class

Class Form

Name: `ss:Rectangle`

Annotations

`rdfs:label` `Rectangle`

Class Axioms

`rdfs:subClassOf` `rdfs:Resource`

Other Properties

`spin:rule`

```
# Computes area := width * height
CONSTRUCT {
  ?this ss:area ?area .
}
WHERE {
  ?this ss:width ?width .
  ?this ss:height ?height .
  LET (?area := (?width * ?height)) .
}
```

Benefits

- Natural object-oriented way of modeling
- SPARQL is very expressive
- Rules can be natively executed by SPARQL engines of the database
- Easy to combine with other SPARQL rule bases like OWL RL

SPIN Standard Modules Library

<http://spinrdf.org/spl>

Reusable modeling constructs like cardinality, instanceOf

SPIN Modeling Vocabulary

<http://spinrdf.org/spin>

Rules and Constraints

Define the semantics of classes and their relationships

Functions and Templates

Encapsulate reusable queries with template arguments

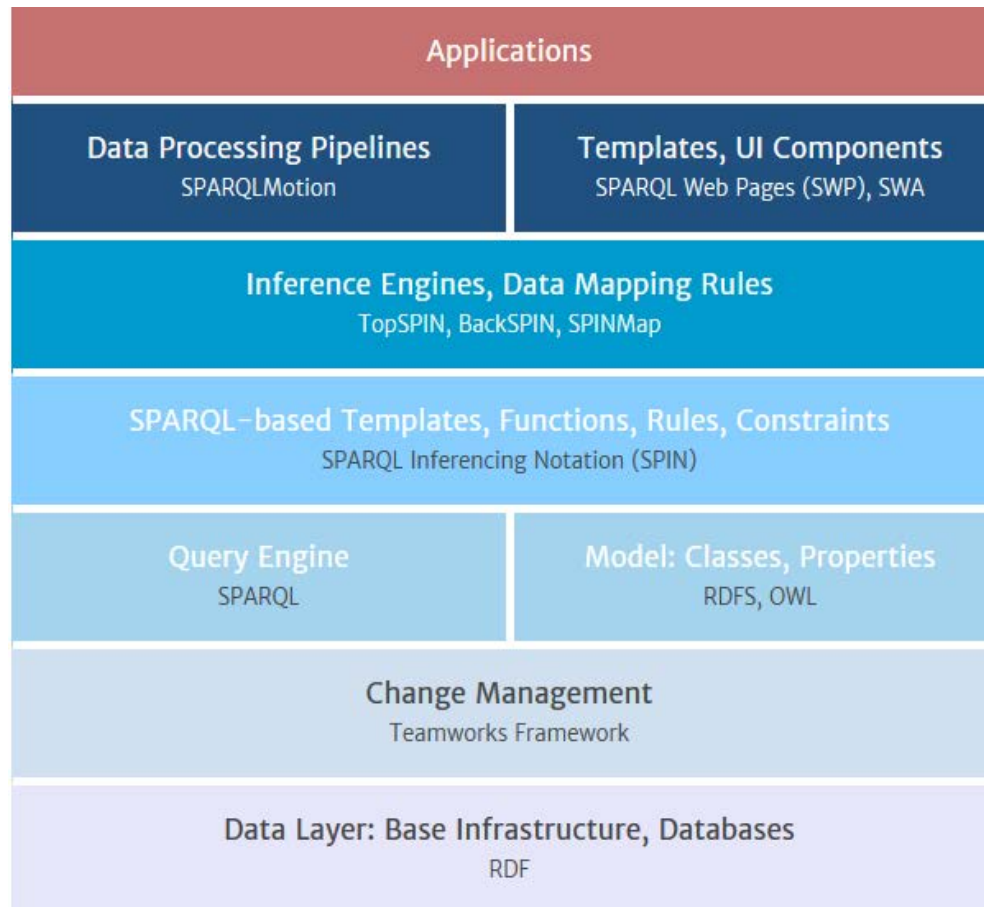
SPIN SPARQL Syntax

<http://spinrdf.org/sp>

An RDF vocabulary for representing queries, variables, filter clauses etc.



SPINMap: Data transformation stack

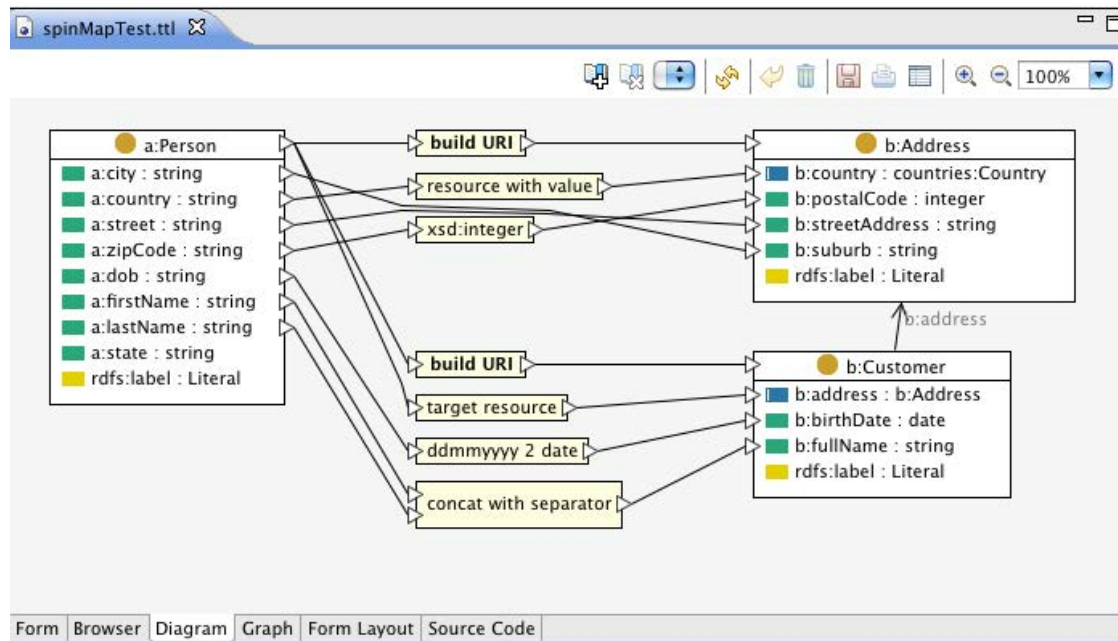




SPINMap: Data mapping rules engine



Motivation: Simplifies mappings between different models
Key Features: Creates executable transformations





SPINMap: Field mapping with rules



Easier to create deep nested structures in the target

Class Form

Name:

▼ Annotations

xml:element ▼

rdfs:label ▼

GMRV VITAL MEASUREMENT ▼

▼ Class Axioms

rdfs:subClassOf ▼

owl:Thing ▼

spin:constraint ▼

spin:constructor ▼

spin:rule ▼

★ # RULE 101: handle match second value mapping

```
CONSTRUCT {  
  ?target a fhir:Observation .  
  ?target fhir:valueQuantityRef ?b0 .  
  ?b0 a fhir:Quantity .  
  ?b0 composite:index 4 .  
  ?b0 fhir:unitsRef ?b1 .  
  ?b1 fhir:value ?unitValue .  
  ?b1 composite:index 1 .  
  ?b1 a fhir:String .  
  ?b0 fhir:valueRef ?b2 .  
  ?b2 fhir:value ?measurementValue .  
  ?b2 composite:index 0 .  
  ?b2 a fhir:Decimal .  
  ?target fhir:nameRef ?b3 .  
}
```



The screenshot shows the TopBraid Vistas interface. The top bar displays the project name "TopBraid - hl7.fhir.linkedmodel.org" and the active file "vista-fhir-map.spin.ttl". The left sidebar contains a "Navigator" pane listing various classes and properties, including "fhir:SupplyStatus", "fhir:UnitsOfTime", "fhir:Uri (6)", "fhir:Uuid", "fhir.ValueSetStatus", "fhir.Resource.Inline", "fms:CommonConcept (2)", "j.o._5 (1)", "loinc:LOINCCode (73115)", "skos:Collection", "skos:Concept", "skos:ConceptScheme", "xml:Node (9)", "vaem:Dimension", "vaem:GraphMetaData (2)", "vaem:GraphRole (16)", "vaem:Party (1)", "xhtml:Base", "xhtml:Blocktext (1)", "xhtml:Cellhalign", "xhtml:Cellvalign", "xhtml:Coreattrs (1)", "xhtml:Events (1)", "xhtml:Focus", "xhtml:Fontstyle", "xhtml:Head.misc", "xhtml:Heading (1)", "xhtml:i18n (1)", "xhtml:inline.forms", "xhtml:Lists (1)", "xhtml:Misc.inline (1)", "xhtml:Param", "xhtml:Phrase", "xhtml:Script", "xhtml:Special.pre", "rdf:Property (1224)", "rdf:Statement", "rdfs:Class (4479)", "sp:SystemClass (265)", "spin:Column", and "spin:ConstraintViolation".

The main workspace is divided into two panes. The top pane shows the "CONTROLLER_vista-fhir.ui.ui.ttl" file, which contains a SPARQL query rule named "RULE 103: handle sameAs mapping". The query is as follows:

```
# RULE 103: handle sameAs mapping
CONSTRUCT {
    ?target a fhir:Observation .
    ?target fhir:valueQuantityRef _b0 .
    _b0 a fhir:Quantity .
    _b0 composite:index 4 .
    _b0 fhir:unitsRef _b1 .
    _b1 fhir:value ?unitValue .
    _b1 composite:index 1 .
    _b1 a fhir:String .
    _b0 fhir:valueRef _b2 .
    _b2 fhir:value ?measurementValue .
    _b2 composite:index 0 .
    _b2 a fhir:Decimal .
    ?target fhir:nameRef _b3 .
    _b3 a fhir:CodeableConcept .
    _b3 composite:index 2 .
    _b3 fhir:codingRef _b4 .
    _b4 composite:index 1 .
    _b4 a fhir:Coding .
    _b4 fhir:codeRef _b5 .
    _b5 fhir:value ?codeValue .
    _b5 composite:index 1 .
    _b5 a fhir:Code .
    _b4 fhir:systemRef _b6 .
    _b6 fhir:value "http://loinc.org" .
    _b6 composite:index 0 .
    _b6 a fhir:Uri .
    _b4 fhir:displayRef _b7 .
    _b7 fhir:value ?displayValue .
    _b7 composite:index 2 .
    _b7 a fhir:String .
}
WHERE {
```

The bottom pane shows the "Form" tab, which displays a table with three columns: "[Subject]", "Predicate", and "Object". The table is currently empty.



SPINMap Output: Linked Vitals



VistA Patient Records of Interest



VistA Patient Record	Blood Pressure
http://livevista.caregraf.info/120_5-8	130/80



Same As

FHIR Patient Records



FHIR Patient Record	Blood Pressure	Value
http://hl7.org/fhir#OBS_cf8c9913-2aa1-11b2-80b1-f6177b67abba	Diastolic blood pressure	130/80
http://hl7.org/fhir#OBS_cf8c9912-2aa1-11b2-80b1-f6177b67abba	Systolic blood pressure	130
http://hl7.org/fhir#OBS_cf8c9909-2aa1-11b2-80b1-f6177b67abba	Diastolic blood pressure	80



Appendix

*Review of Linked Data and its features
as a semantic interoperability language*



Linked Data:

Accommodates both Standardization and Innovation

Information Models:

An apparent conflict between standardization and innovation?

Standardization: need to remain static in order not to be disruptive for adopters.

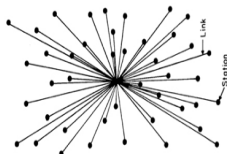
- Static, Brittle
- Centralized
- General (Lowest Common Denominator)
- Committee-driven
- Large, “all-or-nothing”, disruptive updates

VS.

Innovation: requires continuous evolution of thousands of new information models.

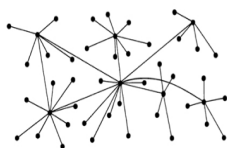
- Adaptive / Evolutionary
- Decentralized
- Highly Specialized, “Best of breed”
- End-user / specialist – driven
- Small, continuous, low-impact updates

What are the options?



Centralized Model-rigid approach: For exchange of information to occur all models must remain fixed, and data must go through only one central ‘broker’ model. *Technologies that support this method are HL7 and XML.*

The current approach to healthcare data

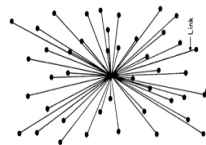


Decentralized Model-flexible approach: Multiple models peacefully co-exist and evolve, mediated by their ability to freely link to any model at all times. In this approach, all models are free to evolve AND are capable of resolving to a common standard model at all times. *The only technology that currently supports this approach is RDF (Linked Data).*

Linked Data supports both standardization and innovation



Data Integration: Legacy vs. Linked Data Approach



Architectural Issues

Legacy Data (HL7, XML)

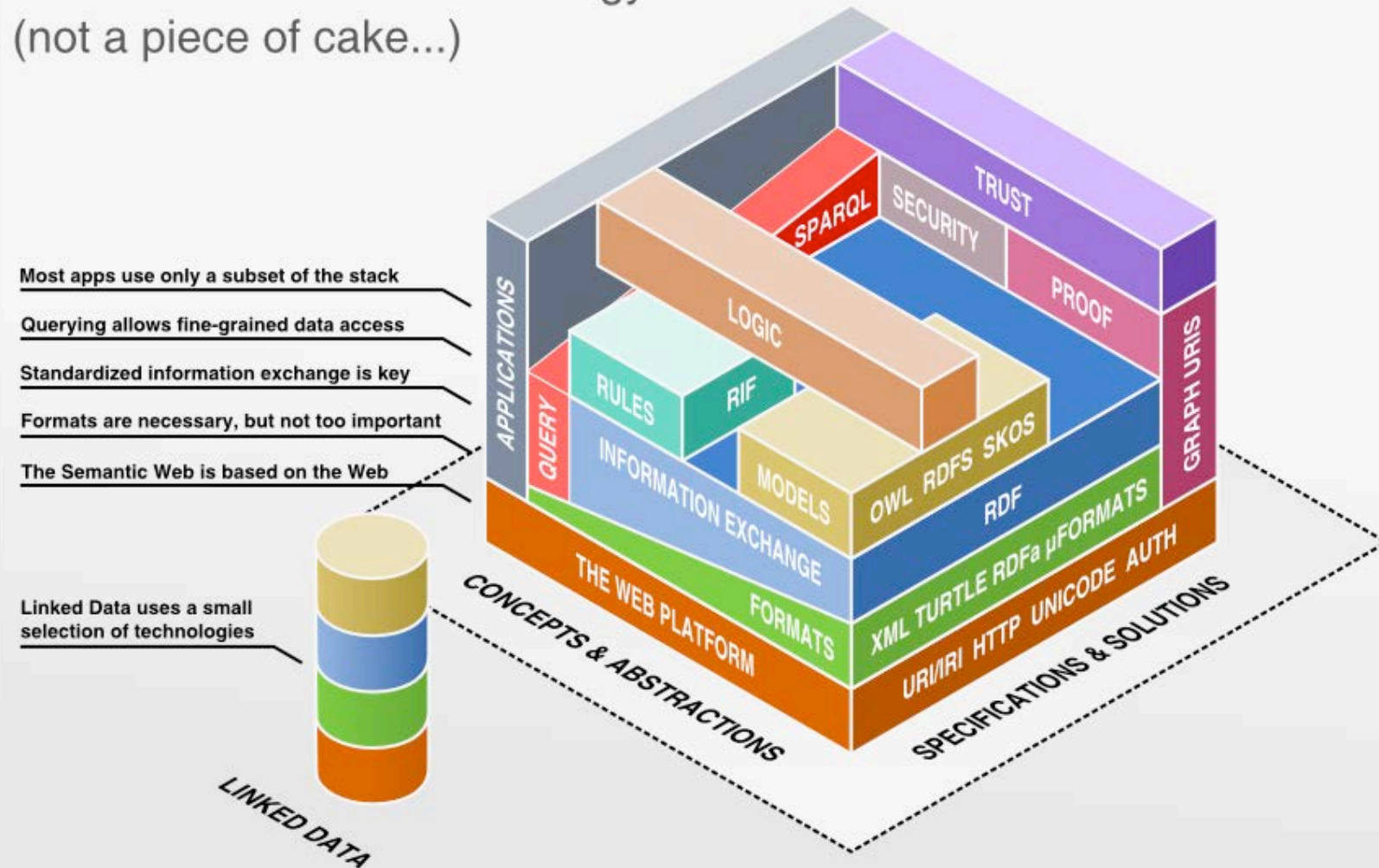
Linked Data (RDF)

Function	Serialization format	Data model
Granularity	Message-centric (documents)	Data-centric (data elements)
Semantics	Weak semantics. Extrinsic to the data. Depends on an external data model.	Strong semantics. Intrinsic to the data.
Data model characteristics	Model-rigid architecture. Only the least common denominator of model unifies information; must remain unchanged to orchestrate. Restrictive expression.	Model-flexible architecture. All data models may independently evolve. Maximizes expressivity.
Data model compatibility	No model diversity permitted Requires one-size-fits-all mega-model	Multiple models peacefully coexist Data model agnostic
Data model evolution	Costly and difficult to evolve models. Due to model-rigid architecture.	Cheaper and easier to evolve models. Due to model-flexible architecture.
Data access method	Downloading + Aggregating	Linking + Federating
Scalability: incremental effort required to add new data sources	Common model must be updated	Individual models can be independently and incrementally semantically linked.



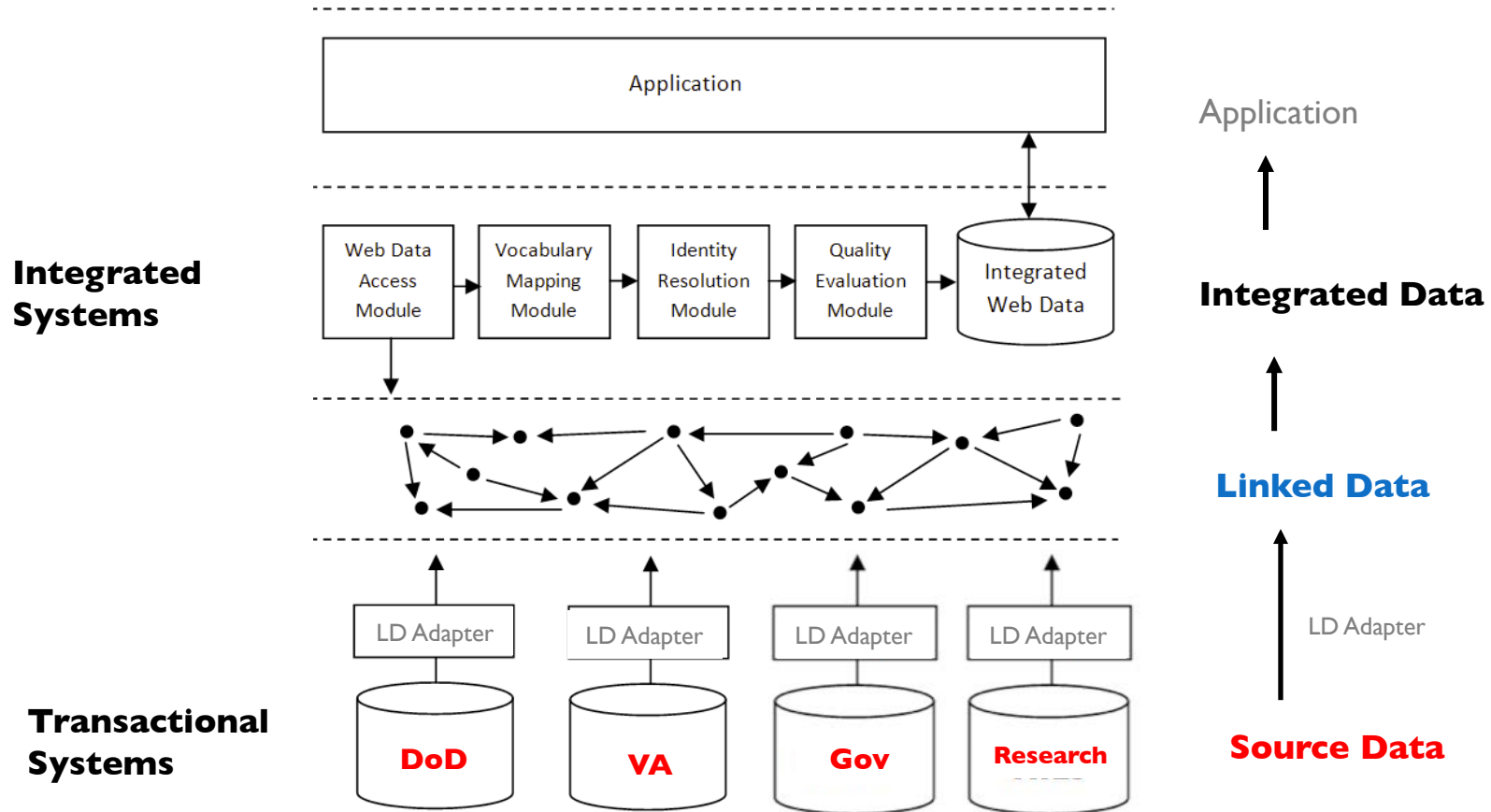
Linked Data: Technology Stack

The Semantic Web Technology Stack (not a piece of cake...)





Linked Data: General Approach to Semantic Integration





End