Scope

RDF Backed multiple data sources (RDF import) knowledge database kind. Distributed 'peers' support for instances sharing of knowledge.

Type inference. Align & merge of different (imported) ontologies. Relationships & link discovery. Augmentation of source knowledge.

Data coming from different sources in the form of raw triples without a schema will be classified, linked and merged if they refer to the same subjects.

Set oriented metamodel layers abstraction for resource aggregation. Temporal contexts for statements. Eventual & logical order.

Functional programming interface for models. 'Ports' for interfacing via known protocols with the knowledge base.

Raw categorized data to be aggregated into facts, information (semiotically) and knowledge (behavior) metamodel levels. Rules (productions), Events and Flows (state) inference for declarative building of application logic.

BI and Linked Data enabled platform for semantic knowledge backend: The above mentioned features could make the concepts of the framework a solid backend for this kind of applications, suitable for flexible, platform independent, middle and presentation tiers implementations. This solution and the ability to merge diverse datasources / formats and to provide uniform interfaces are advantageous at the time to decide for a specific backend or narrow to a RDF only solution.

1. Introduction

The framework proposed here is not a 'pure' semantic web application framework in its strict sense but is more a workaround for implementing a functional programming frontend backed with set theory reasoning for type inference and ontology alignment and merge of diverse knowledge bases.

Various datasources and distributed capabilities are also considered as there are also many 'ports' supported for knowledge base interaction (REST, OData, Solid, SOAP).

The rest of this (very early draft) document attempts to summarize features and capabilities of the desired platform.

1.1. Peers. Datasources.

A Peer is a repository of semantic data, information and knowledge. It performs loading plain RDF triples from enabled datasources and interacting with other Peers sharing knowledge. Datasources have 'drivers' which enables them via an events mechanism to keep in sync with Peer's CRUD operations. Lastly, Peers offer a set of APIs & ports to many protocols exposing internal (augmented) knowledge.

Population of Peer's internal models is done through decomposition and aggregation of input triples. Apache Jena models are kept for input sources (provenance) and for aligned models (ports).

1.2. Type inference

Given the fact the only input a Peer expects is 'plain' RDF (no RDFS, no OWL) it's necessary for a given set of input triples to assert some kind of type information for each Subject, Predicate and Object part of a statement.

The approach taken (better explained in sets section below) is to regard, for example for a given Subject occurring in a set of statements, the Predicates and Objects of those statements as describing a 'class' and a 'metaclass' for this kind of Subject, maybe shared with other Subjects of the same kind (same Predicates and Objects).

In fact, for a Subject sharing a set of Predicate properties in its Triple's occurrences we say it is of some 'class' and given the value of those properties (Objects) we say it is of some 'metaclass' being both class and metaclass the Kind of the Subject. Same applies for Predicates and Objects having corresponding classes and metaclasses aggregated through their corresponding attributes and values (see diagrams below).

Kinds aggregates classes and metaclasses for occurrences of attributes and values relative to each SPO taken into account, for example, given a Subject the attributes / values to be considered are on the intersection of the P and O sets and being the Kind corresponding for a given Subject according its Triple occurrences.

Kinds are the intersection of attributes and a values aggregated according a given Resource triple occurrence. For the triple (a, b, c) if a is the Resource (Subject), b and c are the attribute and value and all attributes and values of the same Subject are aggregated with common attributes and values from other resources to infer class and meta class relationships.

Resources may have multiple occurrences, as subjects, predicates and objects. Regarding Kinds, for example for a given Subject, it SubjectKinds will be the set of all Predicate attributes and Object values according their occurrences in triples where there is that Subject (the set with kinds attrs/values intersection is populated from source triples correspondingly). Then aggregation is done for class / metaclass inference.

The aggregated form of a SubjectKind Resource is the quad:
(metaclassID) (classID) (attribute) (value);
Where both IDs are aggregated for same attributes / values.

A Subject Resource bearing Kind information (for a given Context) is of the form:
(Context) (SubjectURI) (classID) (metaclassID);

In different metamodels (Sets), SPO are named differently but the same logic for calculating Kinds applies: the three outer sets are the 'SPO' sets, the intersection between two of them are the 'Kinds' and the intersection of all them three are the 'Triples' (Contexts).

Kind hierarchies are given being a subclass having a superset of the attributes of its superclass and given an identification mechanism for Kinds which allows to infer the relationship (names, paths, ResourceIDs).
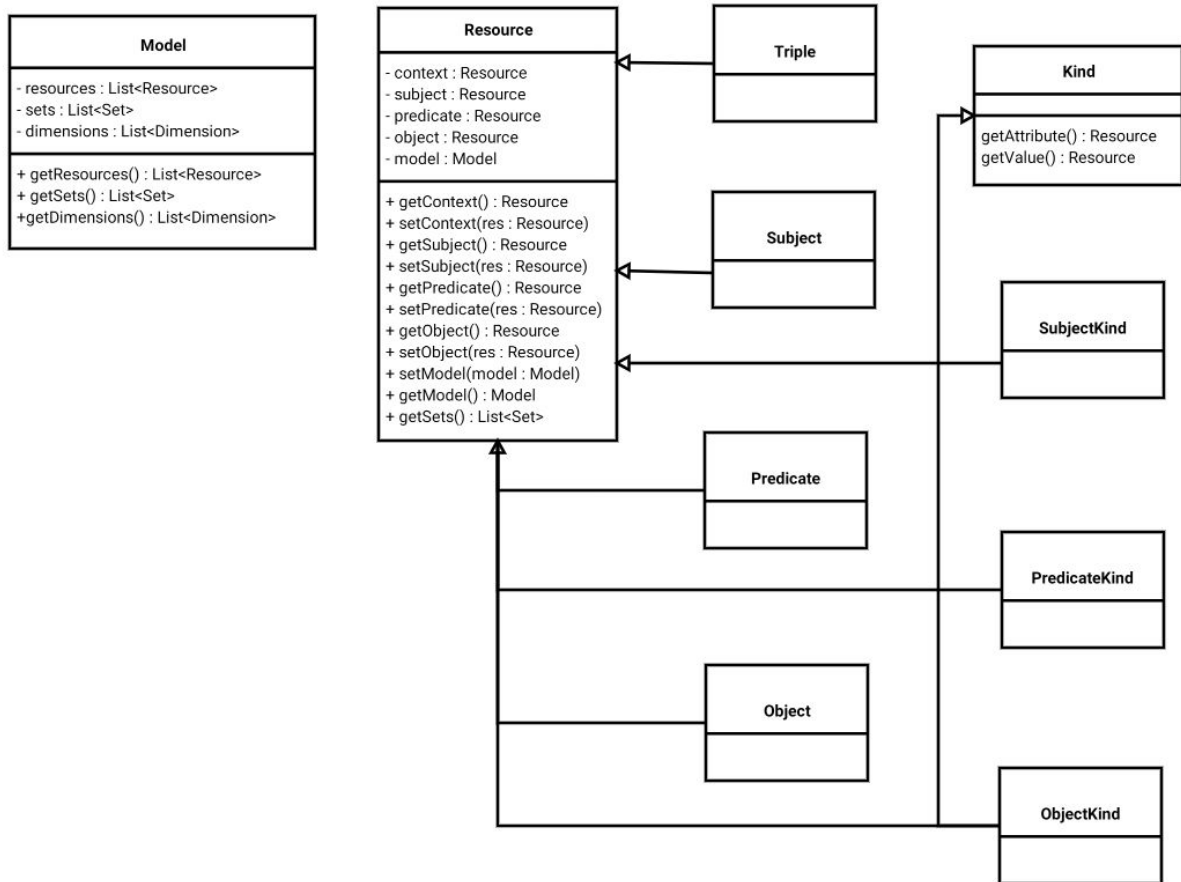
One example of a Subject Kind would be, given the Triple:
(2001) (John Doe) (worksAt) (XYZ Corp);
Kind class: Employee.
Kind metaclass: XYZ Corp Employee.

One issue would be how to gather the names (strings, URIs) in a human readable format (like 'Employee' in the example) given there is no knowledge available for such inference. TBD.

1.3. Resources, Sets, Services

1.3.1. Resources

Resources extracted from input triples are Subjects, Predicates, Objects, SubjectKinds, PredicateKinds, ObjectKinds and Triples. All resources are reified so, for example a Subject in the source graph is represented as a resource which have this Subject as its subject part and which identifies that Subject across occurrences and contexts with a unique name (for example for align & merge purposes)

Model

- resources : List<Resource>
- sets : List<Set>
- dimensions : List<Dimension>

+ getResources() : List<Resource>
+ getSets() : List<Set>
+getDimensions() : List<Dimension>

Resource

- context : Resource
- subject : Resource
- predicate : Resource
- object : Resource
- model : Model

+ getContext() : Resource
+ setContext(res : Resource)
+ getSubject() : Resource
+ setSubject(res : Resource)
+ getPredicate() : Resource
+ setPredicate(res : Resource)
+ getObject() : Resource
+ setObject(res : Resource)
+ setModel(model : Model)
+ getModel() : Model
+ getSets() : List<Set>

Triple

Subject

Predicate

Object

Kind

getAttribute() : Resource
getValue() : Resource

SubjectKind

PredicateKind

ObjectKind

Resource 'definitions': Relative names (this, that, brother) evaluate functionally via semiotic sets metamodel pointers. Resolution of grammars till primitives. ResourceIDs. TBD.

Metamodel Dimensions:
Dashboard, cube axes browsing & functional services: SPOs, Kinds, Triples aggregated through Semiotic metamodel Kinds (Extraction, functional, ResourceIDs). Browse aggregated SPO Kinds, Concept Kinds, Object Kinds (dimensions). Selected Topic expands into Behavior metamodel faceted view. ResourceIDs functional extraction / mask composites.

All sets resources are 'reified' statements about the entity they represent (ie.: quads for SPOs, Kinds and Triples).

Kinds are a special type of resource (statement) which have, for a subject kind example, a predicate and a value extracted from subject's triple occurrences. So they can play, for example, predicate and object roles in sets.

Kinds are extracted from source triples and instantiated into new resources (statements) with class and metaclass metadata in their statement context (metaclass URI) and in their statement subject (class URI, subject kind example). Then, kinds may be reified into their corresponding SPOs and evaluated as classes/metaclasses definition by intention.

For the last question I think I'll be able to encode much more metadata this way without resorting to constructs external to the ontology and do this augmenting existing resources by reification. The main goal would be to develop an 'algebraic' form of inference, reasoning, extraction and transformation of entities in the knowledge base.

1.3.2. Sets

Sets representations of Resource (meta) models provide the means for functional query, extraction and traversal of entities. Sets population, due Set definition predicates, provides for Resource class hierarchy instantiation and Services layer population. Sync and callbacks between models.

Models are instances of other (meta) models, initially SPO triples. Given a metamodel, all its SPOs are taken as the Subjects of the next model, the Kinds are taken as the Predicates and the Triples as the Objects conforming new (Set) triples from which to infer new Kinds.

In the Sets models diagrams attributes / values refer to the corresponding ones for each SPO Kinds aggregation sources.

SPO (Initial) Metamodel:
This model holds initial input data comming from (RDF) Datasources. It basically represents statements and the type relationships between their components.

SPO Model (Facts)
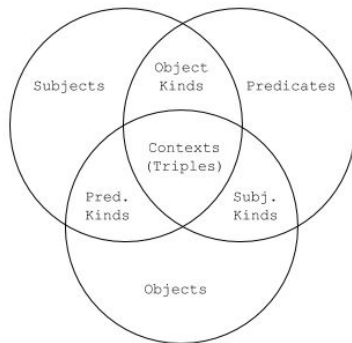
Occurrence	Attribute	Value
Subject	Predicate	Object
Predicate	Subject	Object
Object	Predicate	Subject

Triples:

```
Occurrences (Subject ex.):
[context / time] [SubjectURI] [classID] [metaClassID]

Kinds:
[metaClassID] [classID] [attribute] [value]

Contexts:
[context / time] [Subject] [Predicate] [Object]
```

Example Context (Triple):
(2001) (aPerson) (worksAt) (aCompany);

Kinds: Employee, Employment, Employeer.

Functional Extraction (properties):
Triple(Employee) : aPerson.
aPerson(Employeer) : aCompany.
aCompany(Employee) : [aPerson].

Properties (functional) are of the form:
A of B is C in Context. Inferred from each model components relationships, given a Context, traversal triples matching kinds.

SCO (Sign, Concept, Object) Metamodel:
This model aggregates semiotically the SPO Metamodel. It regards previous (SPO) metamodel as its data source being all elements from SPO's sets its 'Signs' set, all elements from SPO Kinds its 'Concepts' set and all elements from SPO Triples its 'Objects' set. This level Kinds are aggregated the same way than in SPO bu the sets have different names.

Population of sets starts with this level triples ('Topics') which comes from meaningful aggregation of the SPO elements.

This model is 'semiotic' because it aggregates and adds 'meaning' to the three types of 'signs' of the previous model via its own Kinds.

Semiotic Model (SCO, Contexts)
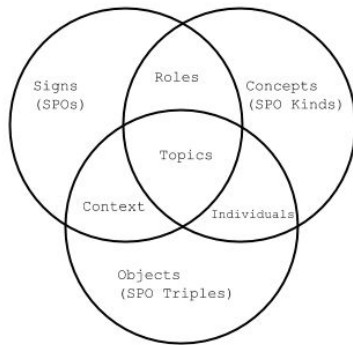
Occurrence	Attribute	Value
Sign	Concept	Object
Concept	Object	Sign
Object	Concept	Sign

Triples:

```
Occurrences (Object ex.):
[context / Topic] [ObjectURI] [classID] [metaClassID]

Kinds:
[metaClassID] [classID] [attribute] [value]

Contexts:
[Topic] [Object] [Concept] [Sign]
```

Example Topic (Topic, Triple, Kind, SPO):
(Work) (aWorkPosition) (Skill) (developer);

Kinds: WorkPosition, PositionRequirement, PositionSkill.

Functional Extraction (properties):
aWorkPosition(PositionSkill) : developer.

Behavior Model:
This model attempts to aggregate previous model concepts in a DCI design pattern with behavior Data, Context and Interactions form. Inputs comming from 'semiotic' models should be triples arranged in such a way 'Purposes' (this level triples) renders this level Topics (Interactions) of Players (Data) into Scenarios (Contexts).

The principal feature of this model is to aggregate semiotic knowledge into the notion of events, flows and rules (discussed below).

Behavior Model (TSP)
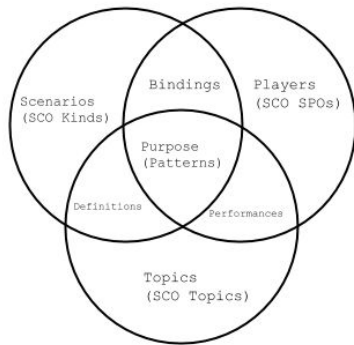
Occurrence	Attribute	Value
Scenario	Topic	Player
Player	Scenario	Topic
Topic	Scenario	Player

Triples:

```
Occurrences (Topic ex.)
[context / Purpose] [TopicURI] [classID] [metaClassID]

Kinds:
[metaClassID] [classID] [attribute] [value]

Contexts:
[Purpose] [Topic] [Scenario] [Player]
```

Example Purpose (Triple):
(orderRelTag) (Work) (SomeCompany) (aDeveloper)
TBD.

Events, Flows and Rules:
According the type of predicate (Scenario) of a triple at this knowledge level there are three kind of possible Bindings:

For a SCO Role: Event Binding.
For a SCO Context: Rule Binding.
For a SCO Individual: Flow Binding.

Definition holds for possible Bindings state transition kinds. Performance holds for instances of given state transition executions.

Rule: A & B : C. C & D : cause(D, C).

Flow: A : B, C... according condition.

Event: Employee : Promotion (good emp.). Identify / reify event and kind of event.

TSP Purpose contexts aggregates / sorts Rule's, Event's and Flow's instance triples.

1.3.3. Services

Services layer provides a functional abstraction over Resources and Sets layers. Basically provides mappings between the main three kinds of entities the framework provides: URIs: SPOs (Names), Kinds ('Content' Types) and Triples (Representations). These entities are 'uniform' resources with dereferenceable URIs. It also provides 'grammars' for those mappings being a grammar the representation for a Content Type (Kind). Mappings and grammars are encoded as Triples / Quads. A 'runtime' or environment may be enforced by the use of OWL and an upper ontology in which to align resources given their abstract (grammar) structure.

Besides Services, functional properties extraction (discussed above) and functional query of entities are means to provide access to query and traversal of models. Kinds can be functionally built using a Template constructor with a Parent and two Kind arguments, maybe as a predicate for filtering or for 'assisted' knowledge modeling (ResourceIDs).

An application specific OWL ontology is to be provided as a type of declarative runtime modelling given an upper ontology to which adheres 'inferred' OWL application description. This will allow for an ontology 'model by example' application design and tailor, for example, Peer's events, reasoning, ontology alignment and goals.

Services: internal storage & representations (ResourceIDs, set predicates). Addressing, Encoding. URI IDs map. Functional features. Merge.

Similarity search (Resources equivalent grammars). TBD.

Services alignement to Behavior metamodel sets:
D: Players - Naming
C: Scenarios - Index
I: Topics - Registry

1.3.3.1. Index

DCI Pattern Interactions.
Holds grammars and definitions for 'types' (here Kinds, Content-Type).

Index mapping func.: idx(name: ctx, representation) : type (ctx optional).
Index quads:
(Ctx) (Name) (Representation) (Kind)
Index grammar:
(Kind : parent) (Kind : name) (Kind : representation) (Kind : type)

1.3.3.2. Naming

DCI Pattern Data.
Holds grammars and definitions for 'names' (here URIs, SPOs).

Naming mapping: nam(representation : ctx, type) : name (ctx optional).
Naming quads:
(Ctx) (Representation) (Kind) (URI)
Naming grammar:
(Kind : parent) (Kind : rep) (Kind : type) (Kind : name)

1.3.3.3. Registry

DCI Pattern Contexts.
Holds grammars and definitions for 'representations' (here Triples).

Registry mapping: reg(type : ctx, name) : representation (ctx optional).
Registry quads:
(Ctx) (Kind) (Name) (Representation)
Registry grammar:
(Kind : parent) (Kind : type) (Kind : name) (Kind : representation)

2. Features

Protocol: Component. Listeners (in / out, Behaviors). Stack. Input / Output: RDF w./ reified Kinds. Submission of triples returns triples. The main purpose of this framework is to provide a low level lightweight knowledge database component.

Protocol modes:

Definition: SPO Triples input returns related knowledge triples.

Manipulation: SCO triples browsing through aggregation and services.

Modification: TSP layer. Submission of (foreignProducts, increaseTax, 5) triple as definition input produces and returns the TSP generated triples ('fires' rules, events, flows) for the definition and manipulation layers that, when submitted, materializes / updates definitions and manipulation sets and services models. Also a 'priceIncrease' event may be fired if two correlated (by purpose context) occurs and there exist a performance and roles kinds for the scenario and player parts of the TSP triple.

Definitions: Materialize SCO, TSP .(Reified Kinds) into SPO. Query for 'priceIncrease' events, rules and flows matching given criteria. TBD.

2.1. Links type / instance inference

Relationship (links) inference example: X coworker Y (same employer). Develop discover algorithms. Infer link types (grammars). Use Kinds, classes, metaclasses (Kinds) relations.

Infer attributes / rels from class (emp, sal, dept, manager) from links. Mgr. is emp's dept. leader.

Infer type by contents: Occurrence having other Kinds in other contexts. Grammar (abstract) occurrences of subject, context merge. Sort Kinds: Grammar hiers (parent). Adult - CanDrive. Employee must be Person & Student. ResourceIDs. Phone number has area code. Infer keys.

2.2. Type grammar inference

Discover primitives (metamodels). Aggregate Kinds and Kinds of class / metaclass. ResourceIDs. Keys. TBD.

2.3. Align & merge of ontologies

Similarity from grammar equivalence (equivalent grammar graphs). Infer 'keys' for types (PK like). Resource 'definitions' & semiotic primitives.

Similarity statements (equiv Kinds). Similarity Kinds (cls, meta Kinds). Similarity resources (equiv occurs / grammars).

ResourceIDs and graph similarity. Infer keys. Infer type by value (same 'grammar' string in different occurrences).

TBD.

2.4. Ordering of triples & events

Triple context (Quad context) holds temporal relationships in metamodels. Query for specific time range, specified interval (bounds) may fire 'events'. Events may be materialized into model triples. Mappings grammar could specify 'listeners' and templates for goals / purposes.

At first metamodel level (SPO) the time context should be 'absolute' (timestamp). Along the metamodel chain the time context could be more general / abstract allowing for time relationship calculations not only by time ordering but also for logical ordering of events / triples.

TBD.

3. APIs

3.1. Services REST API

Service layer provides functionality for seamless implemention of a REST HATEOAS API directly over the model. Roughly, protocol would be like clients requesting, previously potentially sending a state set of triples, and retrieving an 'index' triple(s) Services representation. Then client chooses a triple and a name in that triple to submit. Then it obtains a type (Kind) for that name in that context and given this Kind it can query for further properties and retrieve them again in the form of new triples.

Client(triples) - Server(Triples)
Client(name) - Server(type rep.)
Client(type name) - Server(triples)

Protocol CRUD: Performs Align & merge. Store all requests (only POST, sends, receive only triples): possible individuals. worksAt example. TBD.

3.2. Functional (Dimensional) API

Functional (jQuery like) interface for uniform Resource operations. Query, filter, traversal, predicates, assertions. TBD.

Metamodel Dimensions:
Dashboard, cube axes browsing & functional services: SPOs, Kinds, Triples aggregated through Semiotic metamodel Kinds (Extraction, functional, ResourceIDs). Browse aggregated SPO Kinds, Concept Kinds, Object Kinds (dimensions). Selected Topic expands into Behavior metamodel faceted view. ResourceIDs functional extraction / mask composites.

3.3. Ports

3.3.1. RDF(S) / OWL

3.3.2. SPARQL

3.3.3. OData

3.3.4. SOAP

3.3.5. Solid

3.4.6.Activation Bundles (DOM Model)

ORM + Services like bindings for specific platforms. Export bundles (JAR files? JS?) with concrete APIs. DOM model. TBD.

4. Lab

4.1. Encoding & addressing

Peers hub dialog align: entity/kinds resolution. Equivalent occurrences in local triples. Self node resolution.

Beside URIs Resources have a ResourceID identifying diverse occurrences of the same Resource regardless of different URIs representating the same concept for example when aligning different ontologies.

ResourceID is a set of the form:
ResourceID : ([TripleID, ResourceID pattern | TripleID mask])*;

Where TripleID is a sequence of uploaded / inferred / derived object triples sequences of the form:

[CtxID] [SubjectID] [PredicateID] [ObjectID]

Where each ID is a URI sequence and ResourceIDs are a mask over TripleIDs:

The left part of the expression is a (variable) input and the TripleID is the context in which to evaluate the output of the expression.

For Triples: Mask filtering all triple contents.

For Kinds: Mask filtering Kind's attributes and values for the occurrence in the corresponding Triple.

For SPOs: Filters corresponding SPOs ocurrences in the given Triple.

Filters may be chained / nested to form more complex queries.

ResourceID expressions: A resource can match to one or more resource expressions of the form:

(ResultAggregate, SourceAggregate) (LHSJoinMask, RHSJoinMask) (Mask, InputMask)

where each of the expression parts are ResourceIDs having their input on its right and their outputs on their left sides.

Services mappings & grammars: ResourceIDs. Composite functional expressions. URI encoding of ResourceIDs (operable).

Subject, Predicate, Object ResourceIDs: Patterns attributes/values of their occurrences.

SubjectKind, PredicateKind, ObjectKind ResourceIDs: Patterns attributes/values of their occurrences (Join SPOs). Kind qualified SPOs. TBD.

Triple ResourceIDs: Patterns of their SPOs.

ResourceID Pattern (for a Kind Triple):
(ctx sel(subj sel(attr mask(val mask, result mask))))

Compose & operate on identifiers. Expressions. Criteria. Relationship extraction. Hash lookup: IDs and patterns of IDs represents the TripleIDs for which a ResourceID (set) holds. TripleIDs and patterns/mask are composed together using the binary OR of its parts and the resulting mask applied to an (ordered) set of IDs resulting in the selection of matching instances.

4.2. Octal order rel. encoding

4.3. Node, containers

5. Application

5.1. Dashboard example

Services Dashboard matrix
X Axis: Names (SPO URIs) : DCI Data
Y Axis: Representations (Triples) : DCI Context
Points: Content Types (Kinds) : DCI Interaction

Metamodel Dimensions:
Dashboard, cube axes browsing & functional services: SPOs, Kinds, Triples aggregated through
Semiotic metamodel Kinds (Extraction, functional, ResourceIDs). Browse aggregated SPO
Kinds, Concept Kinds, Object Kinds (dimensions). Selected Topic expands into Behavior
metamodel faceted view. ResourceIDs functional extraction / mask composites.

Tool for analysis, discovery & mining. Develop views through the use of facets for Activation
Bundles. TBD.

6. GNU Free Documentation License

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way

to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
world-wide, royalty-free license, unlimited in duration, to use that
work under the conditions stated herein. The "Document", below,
refers to any such manual or work. Any member of the public is a
licensee, and is addressed as "you". You accept the license if you
copy, modify or distribute the work in a way requiring permission
under copyright law.

A "Modified Version" of the Document means any work containing the
Document or a portion of it, either copied verbatim, or with
modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of
the Document that deals exclusively with the relationship of the
publishers or authors of the Document to the Document's overall
subject (or to related matters) and contains nothing that could fall
directly within that overall subject. (Thus, if the Document is in
part a textbook of mathematics, a Secondary Section may not explain
any mathematics.) The relationship could be a matter of historical
connection with the subject or with related matters, or of legal,
commercial, philosophical, ethical or political position regarding
them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title,

preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on

the back cover. Both covers must also clearly and legibly identify
you as the publisher of these copies. The front cover must present
the full title with all words of the title equally prominent and
visible. You may add other material on the covers in addition.
Copying with changes limited to the covers, as long as they preserve
the title of the Document and satisfy these conditions, can be treated
as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit
legibly, you should put the first ones listed (as many as fit
reasonably) on the actual cover, and continue the rest onto adjacent
pages.

If you publish or distribute Opaque copies of the Document numbering
more than 100, you must either include a machine-readable Transparent
copy along with each Opaque copy, or state in or with each Opaque copy
a computer-network location from which the general network-using
public has access to download using public-standard network protocols
a complete Transparent copy of the Document, free of added material.
If you use the latter option, you must take reasonably prudent steps,
when you begin distribution of Opaque copies in quantity, to ensure
that this Transparent copy will remain thus accessible at the stated
location until at least one year after the last time you distribute an
Opaque copy (directly or through your agents or retailers) of that
edition to the public.

It is requested, but not required, that you contact the authors of the
Document well before redistributing any large number of copies, to
give them a chance to provide you with an updated version of the
Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under
the conditions of sections 2 and 3 above, provided that you release
the Modified Version under precisely this License, with the Modified
Version filling the role of the Document, thus licensing distribution
and modification of the Modified Version to whoever possesses a copy
of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct
   from that of the Document, and from those of previous versions

(which should, if there were any, be listed in the History section
of the Document). You may use the same title as a previous version
if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities
responsible for authorship of the modifications in the Modified
Version, together with at least five of the principal authors of the
Document (all of its principal authors, if it has fewer than five),
unless they release you from this requirement.

C. State on the Title page the name of the publisher of the
Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications
adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice
giving the public permission to use the Modified Version under the
terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections
and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add
to it an item stating at least the title, year, new authors, and
publisher of the Modified Version as given on the Title Page. If
there is no section Entitled "History" in the Document, create one
stating the title, year, authors, and publisher of the Document as
given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for
public access to a Transparent copy of the Document, and likewise
the network locations given in the Document for previous versions
it was based on. These may be placed in the "History" section.
You may omit a network location for a work that was published at
least four years before the Document itself, or if the original
publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications",
Preserve the Title of the section, and preserve in the section all
the substance and tone of each of the contributor acknowledgements
and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document,
unaltered in their text and in their titles. Section numbers
or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section
may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements"

or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document, you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains
nothing but endorsements of your Modified Version by various
parties--for example, statements of peer review or that the text has
been approved by an organization as the authoritative definition of a
standard.

You may add a passage of up to five words as a Front-Cover Text, and a
passage of up to 25 words as a Back-Cover Text, to the end of the list
of Cover Texts in the Modified Version. Only one passage of
Front-Cover Text and one of Back-Cover Text may be added by (or
through arrangements made by) any one entity. If the Document already
includes a cover text for the same cover, previously added by you or
by arrangement made by the same entity you are acting on behalf of,
you may not add another; but you may replace the old one, on explicit
permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License
give permission to use their names for publicity for or to assert or
imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this
License, under the terms defined in section 4 above for modified
versions, provided that you include in the combination all of the
Invariant Sections of all of the original documents, unmodified, and
list them all as Invariant Sections of your combined work in its
license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and
multiple identical Invariant Sections may be replaced with a single
copy. If there are multiple Invariant Sections with the same name but

different contents, make the title of each such section unique by
adding at the end of it, in parentheses, the name of the original
author or publisher of that section if known, or else a unique number.
Make the same adjustment to the section titles in the list of
Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History"
in the various original documents, forming one section Entitled
"History"; likewise combine any sections Entitled "Acknowledgements",
and any sections Entitled "Dedications". You must delete all sections
Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other
documents released under this License, and replace the individual
copies of this License in the various documents with a single copy
that is included in the collection, provided that you follow the rules
of this License for verbatim copying of each of the documents in all
other respects.

You may extract a single document from such a collection, and
distribute it individually under this License, provided you insert a
copy of this License into the extracted document, and follow this
License in all other respects regarding verbatim copying of that
document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate
and independent documents or works, in or on a volume of a storage or
distribution medium, is called an "aggregate" if the copyright
resulting from the compilation is not used to limit the legal rights
of the compilation's users beyond what the individual works permit.
When the Document is included in an aggregate, this License does not
apply to the other works in the aggregate which are not themselves
derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these
copies of the Document, then if the Document is less than one half of
the entire aggregate, the Document's Cover Texts may be placed on

covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the

violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License. If your rights have been terminated and not permanently
reinstated, receipt of a copy of some or all of the same material does
not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the
GNU Free Documentation License from time to time. Such new versions
will be similar in spirit to the present version, but may differ in
detail to address new problems or concerns. See
http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number.
If the Document specifies that a particular numbered version of this
License "or any later version" applies to it, you have the option of
following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation. If the Document
specifies that a proxy can decide which future versions of this
License can be used, that proxy's public statement of acceptance of a
version permanently authorizes you to choose that version for the
Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any
World Wide Web server that publishes copyrightable works and also
provides prominent facilities for anybody to edit those works. A
public wiki that anybody can edit is an example of such a server. A
"Massive Multiauthor Collaboration" (or "MMC") contained in the site
means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0

license published by Creative Commons Corporation, a not-for-profit
corporation with a principal place of business in San Francisco,
California, as well as future copyleft versions of that license
published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in
part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this
License, and if all works that were first published under this License
somewhere other than this MMC, and subsequently incorporated in whole or
in part into the MMC, (1) had no cover texts or invariant sections, and
(2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site
under CC-BY-SA on the same site at any time before August 1, 2009,
provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

    Copyright (c) YEAR YOUR NAME.
    Permission is granted to copy, distribute and/or modify this document
    under the terms of the GNU Free Documentation License, Version 1.3
    or any later version published by the Free Software Foundation;
    with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
    A copy of the license is included in the section entitled "GNU
    Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the "with...Texts." line with this:

    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.