Proposal for a Structure of the PROVO HTML document

## Introduction

* Talk about prov-dm
* Say that we are using OWL-RL

## PROVO at a Glance

The terms (classes and properties) in provo can be categorized into three categories: core terms, terms used to encode qualified relationships, and “common” (or what we can also call additional or convenient) terms. This section provides a rational on this categorization preferably in one paragraph, and then presents three indexes that list the terms in each category hyperlinked to their detailed description in Section 4 for quick reference. To see what we mean by an index, take a look at the following link <http://rdfs.org/sioc/spec/#sec-glance>

## The PROVO Ontology Description

This section provides the reader with an understanding about the structure of the section, so that they know which section they should go to read and where they should stop. That said, the message about the purpose of each section can still be repeated at each individual section.

### PROVO Core

This section presents a diagram illustrating the concepts and properties that compose the ontology, without talking about involvement. The following example diagram illustrates the level of details we want to present and the **exact** classes and properties we believe fall into this category. No need to show the “Thing” class in the diagram, as we are doing now. That just makes the diagram more complex. Neither SIOC nor OPMV do that. The classes and properties to show in this diagram are the following:

Entity, Agent, Activity, used, wasGeneratedBy, wasDerivedFrom, wasAssociatedWith (wasEndedBy? wasStartedBy??), actedOnBehalfOf, wasInformedBy, wasAttributedTo, “Temporal stuff”



Figure Core classes and properties in Prov-o

### Example

Turtle + Explanation. The example does not have to illustrate all the terms. Instead, it focuses on the main ones. (see eg16-journalism-simple-without-comments.ttl)

### Qualified Relationships in PROVO

This section shows how qualified relationships in prov-dm are encoded in provo. We need two diagrams: one to show the hierarchical relationship between involvement classes, and another to using Association as an exemplar to illustrate the general pattern to *qualify* a binary relationship.



Figure Express association between an activity and an agent using a binary relationship and an alternatie qualified relationship

We have in the current provo HTML document a diagram about qualified involvement in Section 8 “Overview of Qualified Involvement”, but we don’t think that such a diagram is great. We need to illustrate the hierarchy of involvements and some, not necessarily all, the associated object properties. The focus in the diagram will be on the elements of prov-dm that are illustrates in the previous overview sub-section.

The diagram can show the following kinds on involvements, and shows some (not necessarily all) the properties that link them together, e.g., entity, activity, agent, qualified.

Involvement, EntityInvolvement, ActicityInvolvement, AgentInvolvement, Derivation, Usage, Generation, Association (End, Start??), Responsibility, Attribution, Revision. Quatation, Trace, Source, Inform? StartByActivity?

### Example

Turtle + Explanation. We do not to cover all the kinds of involvement and properties in the example. We only show how Derivation, Usage, Generation and Association can be expressed using this reification pattern. If need be, examples of other types of involvement should be given in section 4, where they are defined. For the qualified relationships we provide the following example files:

\* eg16-journalism-derivation.ttl, to show how additional information about derivation can be expressed, such as when it all happened.

\* eg16-journalism-generation.ttl, to show how additional information about generation can be expressed, such as when it all happened.

\* eg16-journalism-usage.ttl, to show how additional information about usage can be expressed, such as when it all happened.

\* eg16-journalism-association.ttl, to show how we can express plan or recipe used in an activity by an agent

## “Common” Classes and Relationships in PROVO

This section presents the rest of classes and properties textually without using a diagram. We may use some examples in English.

The classes and properties covered in this section are:

Note, hasAnnotation, tracedTo, wasRevisionOf, hadOriginalSource, wasQuotatedFrom, hadQuotatedAgent, hadQuoterAgent, wasSummaryOf, alternateOf, terms related to Collection? Bundle? Role?? , hadRole, Location??, hadLocation?, wasStartedByActivity?? specializationOf?

### Extension of PROVO (This section is not mandatory and can be droped if deemed unnecessary)

This section contains a summary of what is currently in Section 9 “Specializing the PROV Ontology for Domain-specific Provenance Applications” of the HTML document of provo. It also references the best practices document.

## Cross-reference for PROVO classes and properties

### CORE Terms

### Terms used to encode qualified relationships

### Common, Additional, or Convenient Terms

This section, i.e., Section 4, presents the classes and properties that compose the ontology. No need to provide a diagram illustrating the domain and the range of the object properties. Also, no need to provide an example for each class and property. We should be able to present most of classes with an example stated using a sentence or two in English. In other words, we don’t need to provide the turtle for every small example if a sentence in English is enough. We should reserve the use of turtle only for the classes and properties that we think may be difficult for the user to grasp, e.g., involvement.

A class description contains:

* A textual description
* Its super-class (or sub-class). Sometimes not always. Only the direct ones, no need to go through all the hierarchy.
* Information about the properties that the class in question is used as a domain or range. Sometimes not always
* An example (one or two English sentences).

A property description contains:

* A textual description
* Domain
* Range
* Inverse property: if such a property exists.
* Super-properties: only the direct ones, no need to go through all the hierarchy.

Note that there is no need to reference terms listed in Section 4 from within the table of contents, this makes the table of contents difficult to browse. We think that the reader is likely to prefer to use the links provided in “PROVO in a glance”, as well as links used throughout the document.