DCPAPERS
Proc. Int'l Conf. on Dublin Core and Metadata Applications 2011

Satellites, the Elsevier Format for Ancillary Information to Scientific Journals and Books

David Kuilman
Elsevier Operations,
The Netherlands
d.kuilman@elsevier.com

Martin Ruck
Elsevier Operations,
United Kingdom
m.ruck@elsevier.com

Abstract

Elsevier presents the Satellite format -- a linked data compliant data format to capture, store and expose metadata objects using open standards based metadata frameworks e.g. SKOS, DCMI and SWAN. The satellite format allows for an array of configurable features to be defined on a per-project basis to specify the metadata object and its required business usage. A key use case presented in detail is the modeling of tagging information sourced by text mining and content enhancement suppliers to persist scientific document annotation expressed in RDF, linking text strings within the document to concept URIs in scientific vocabularies.

Keywords: Gold standard test, SKOS, RDF, RDF serialization, tagging, URIs

1. Introduction

Elsevier is a leading scientific global publisher, taking care of 2500+ journals and a few thousand books per year. The past years have shown a gaining demand for capturing content enhancement information to source online products with an improved navigation and search experience. As a companion to existing content flows for journal articles and book chapters, Elsevier has developed a satellite format to hold metadata resources, as persistent objects, with a URI-based coupling to its content host. It is an important conceptual step to handle metadata objects as persistent objects that can cross architectural boundaries, while preserving its meaning during its traversal. For these features the name "satellite" object was coined.

The metadata resources offer key ingredients to expose ancillary features to make products more versatile, accurate and intuitive to use. These features are commonly sourced by employing text mining techniques that gather and derive knowledge from data.

This paper presents an outline of this satellite format, detailing the key features based on web and metadata framework standards.

2. Requirement for a satellite format

Within Elsevier all content, metadata, relationship and management information resides in separate, but connected, warehouses. To introduce a dedicated container for content enhancement information, this will require the format to not only link enterprise data together, but also provide a linked open data interface for online products and integration with the semantic web. During the manufacture of the satellite data, the format must also provide expression to capture service information e.g. provenance, curation and administration based on different classes of metadata resources. This requirement has led to a format that utilizes linked open data principles (e.g. dereferenceble URIs) with closed, enterprise centric formats that provide API for enterprise quality assurance and control tooling.

In this chapter we will go into some detail explaining the design principles behind the satellite format and conclude with some samples.

https://doi.org/10.23106/dcmi.952135784

2.1 Standards based metadata encoding

To bridge the XML centric content flows within Elsevier with standards based metadata encoding initiatives, the choice was made to adopt the Resource Description Framework to:

- Implement properties provided by DCMI (Nilsson et al., 2008) and SWAN (W3C Interest Group, 2009) to encode property value classes;
- Implement SKOS (Miles et al., 2008) for encoding taxonomies and controlled vocabularies.

For both business use cases, RDF/XML serialization (W3C, 2004a) of the content aligns well with the existing XML-based validation frameworks. To allow for associating metadata at the fragment level, an overhaul of the existing scientific article base was made to insert URIs at the sub-document level. In this way, the full content store of Elsevier and its granular metadata has become compatible to linked data standards and scientific publishing platforms.

2.2 Quality control on metadata that is scalable for production

Quality control of content and metadata to content are enforced using XML-based validation frameworks. A series of business rules are applied through automation to secure the quality of content proof-readers and typesetters. Interestingly, from the perspective of a publisher, content enhancement suppliers (i.e. text mining service providers) operate a different engagement model with the customer by allowing manual curation as a learning feature to enhance the quality of the metadata. Because metadata resources can have distinct workflows, the quality control of the results will need to be integrated with existing content flows as the content and its metadata are interdependent. It is this requirement that drives a series of quality assurance (as complement to quality control) workflows, with subsequent tooling, that builds upon the satellite format. The satellite format becomes a hub document that holds in a single place:

- Metadata and identification on the resource that the satellite is related to;
- The metadata itself that can be modeled according to table or graph data structures;
- Provenance information of the encapsulated metadata
- Optional information e.g. scoring and confidence information, document fragment identification, supplier communication etc.

2.3 Classes of satellites

Depending on the business use case, different information resources must be associated, on demand, with core content assets being journal articles and (book) chapters. In this sense, satellites serve the role of being a business object that can be the subject of an independent workflow (e.g. taxonomy maintenance), or a complex, interconnected workflow that requires ad-hoc assemblies of satellites in a single application (e.g. quality assurance using Gold sets). The organization and selection of the information resources are therefore based on satellite manufacturing requirements and product feature requirements. The following classes of satellites have been devised:

- The Annotation class (or 'tagging' class) holds concept information at the document and sub-document level.
- The Vocabulary class holds taxonomies, controlled lists and thesauri.
- The Basic metadata property class holds a set of DCMI (Nilsson et al., 2008) and PRISM properties.
- A Document Structure class to capture the logical structure of a document.
- The Embedded metadata class is a general purpose container for holding an arbitrary XML-vocabulary, e.g. a Keyhole Markup Language document (KML).

The design of the satellite format is modular in the sense that a project-determined root-element will include subordinate schemas from a general namespace and a project namespace. Examples of namespaces are:

https://doi.org/10.23106/dcmi.952135784

TABLE 1: Namespaces of the satellite format

Label:	Formal identifier:	Description:
prj:	http://www.elsevier.com/xml/schema/rdf/Project-1/	Local: Project identifier
dct:	http://purl.org/dc/terms/	Common: Dublin core terms
pav:	http://purl.org/swan/pav/provenance/	Common: Provenance, authoring and versioning
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Common: basic RDF tags and document structure
rgn:	http://www.elsevier.com/xml/schema/rdf/LDR-Satellite/Regions-1/	Common: description for document regions
sat:	http://www.elsevier.com/xml/schema/rdf/LDR-Satellites/Base-1/	Common: the satellite structure itself.
skos:	http://www.w3.org/2004/02/skos/core#	Common: Simple Knowledge Organization System.
tag:	http://www.elsevier.com/xml/schema/rdf/LDR-Satellites/TagAnnot-1/	Common: description for tag assignments etc.
voc:	http://www.elsevier.com/xml/schema/rdf/SKOSsatellite-1/	Local: Wrapper for the vocabularies

3. The satellite format

This section details some key design aspects of the satellite format based on features viewing a satellite as a business object.

3.1 Stand-off and associated resources

The satellite object serves, conceptually, as an arbitrary add-on to core journal article and (book) chapters. However, a satellite can also be a stand-alone object that is not yet associated with another resource, or never will because it will be used as reference authority in its own right e.g. vocabularies and taxonomies.

It follows that a satellite resource must be a persistent object within the content architecture that is under version control. Every satellite is exposed to semantic web services using a URI. As a storage object, the satellite is a named graph that must be interrogated at the graph level.
A satellite class can be identified as an RDF-document holding a (satellite) head-element with a body-payload expressed as a series of `rdf:Description` elements (or non-RDF vocabulary for the class of embedded metadata).

It is an important feature that multiple satellites can refer to a single resource. Implementation detail will prescribe the way multiple satellites will get merged into a single, new graph with possible, inferred triples.

3.2 Annotations

Adding keywords, concepts, phrases and/or categories to documents or areas of documents, is collectively called tagging. With tagging it is possible to optimize search indexes or calculate similarities between documents that share the same selection of tags. The source of selectable concepts is commonly provided with a reference to a controlled vocabulary.

Within annotating text, it is important to express the actual concepts and its relationship towards fragments in the text. The following RDF-fragment presents the skeleton for a satellite instance.

https://doi.org/10.23106/dcmi.952135784

```
<x:satelliteWrapper ...>
        <rdf:RDF ... >
                <!-- Satellite header -->
                <sat:Satellite>
                ....
                </sat:Satellite>
                <!-- Used thesauri -->
                <skos:ConceptScheme/>
                <!-- Annotations at the whole-article level -->
                <rdf:Description rdf:about="whole-document">
                        <concept> + <annotation details>
                        <concept> + <annotation details>
                        <concept> + <annotation details>
                ....
                </rdf:Description>
                <!-- Document regions with their annotations (if any) -->
                <rgn:XMLDocumentRegion rdf:about="some-region"/>
                <rdf:Description rdf:about="the-same-region">
                        <concept> + <annotation details>
                        <concept> + <annotation details>
                        <concept> + <annotation details>
                        ....
                </rdf:Description>
```

FIG. 1: The Annotation satellite skeleton

This structure is followed by all annotation satellites. It has the following logical sections:

1. Outer wrapper with namespace declarations ("x:" being substituted for the appropriate namespace label)
2. The satellite as object: the identifier, which document it annotates, the satellite type, project, creation date, etc
3. A list of the thesauri that this satellite makes use of for its annotations
4. Annotations for the document as a whole. Each annotation consists of the annotating concept plus some supporting metadata (labeled as "annotation details")
5. Annotations for specific regions (down to individual words or characters) of the document. As above, each annotation consists of the annotating concept plus some supporting metadata.

3.3 Annotation detail using Dublin Core and Tag namespace

Additional to identifying the concept that qualifies the text as an annotation, is the need to express, for the user of the annotations, the confidence of the annotations measured against options, the foundation for the scoring both expressed in an applied algorithm and pattern recognition in the source document. For this, the tag-namespace is a general purpose structure to allow for grouping metadata properties using Dublin Core namespace elements or proprietary tagging elements to construct substructures.

Annotations are encoded as an internal assertion which can be a set of statements made about another statement in the same document. It is a specific example of the more general RDF principle of reification.

In annotation satellites, internal assertions are used to provide additional data about an annotation.

First, the annotation itself is declared. Then, an additional block of statements is added which refer to (that is, further describe) the annotation. The annotation and the further statements are linked via URLs. The URL of the annotation itself is declared in an attribute `rdf:ID` of the enclosing element (usually `dct:subject` or one of the named facets). Then, additional statements are added via a `tag:relatedAnnotation` element. This element contains a sub-element `tag:annotatesStatement` whose attribute `rdf:about` is equal to the previously declared `rdf:ID`.

https://doi.org/10.23106/dcmi.952135784

```
<rdf:Description rdf:about="http://api.elsevier.com/content/article/DOI:10.1016/j.am#id110085">
   <dct:subject rdf:ID="stmt-83" rdf:resource="http://data.elsevier.com/vocabulary/EMMeT/Concept-222" />
      <!-- Risk Factors -->
      <tag:relatedAnnotation>
        <tag:TaggingAnnotation rdf:about="#stmt-83">
           <tag:score>0.5773502691896258</tag:score>
           <tag:relevance rdf:resource="http://data.elsevier.com/ns/ Satellite/RelevanceCodes-1/Major"/>
           <tag:targetText>risk factors</tag:targetText>
           <tag:status rdf:resource=http://data.elsevier.com/ns/ Satellite/TagAnnot-1/Unreviewed"/>
           <pav:createdBy rdf:resource=
                        "http://data.elsevier.com/enh-service/Project-1/v1"/>
           <pav:createdOn>2010-12-02T19:53:19Z</pav:createdOn>
         </tag:TaggingAnnotation>
     </tag:relatedAnnotation>
```

FIG. 2: Detail markup for adding annotation detail

The URI reference to the RDF predicate `dct:subject` can also be replaced by another predicate in a customer namespace, e.g. `med:procedure`. This predicate is used to qualify the enclosing SKOS-concept as being a scope field for the 'procedure' facet.

```
...
<med:procedure>
          <skos:Concept rdf:about="...code...">
                  <skos:prefLabel>Diabetes</skos:prefLabel>
         </skos:Concept>
</med:procedure>

...
```

FIG. 3: Optional use of customer namespace for setting facets

Other DC properties used for metadata encoding are, `dct:coverage` to denote geo-spacial information that is used to plot locations on a map, `dct:isPartof` to denote the logical inclusion to the identified fragment and `dct:title` for all labeling purposes.

3.4 Fragment handling

Having supplied a URI for the source document as a whole, it is often necessary to identify a specific region within a document. Document region identifiers are required wherever annotations are relevant to a particular part of the document, rather than the document as a whole. Identifying a region is done based on the Document Object Model in conjunction with the substring function to calculate character offsets and ranges.

A document region is defined by using a `rgn:XMLDocumentRegion` element with an attribute `rdf:about` equal to some Xpath-e expression [XPath-e] that defines the document region. Xpath-e is a variant of the Xpath standard which allows a few more functions and thus more flexibility.

Xpath-e allows the definition of substrings taken either from the root of the document or from an identified section of the document. In general, content to be annotated will be supplied with IDs in place on significant document structures, such as sections, paragraphs, lists, figures, and so on.

These IDs are unique and can form part of the Xpath-e expression to define a document region. If IDs are not supplied, document regions can be expressed using absolute Xpaths.

Depending on the available markup and identifiers, in conjunction with the required precision of identifying a region, a cascaded approach can be followed that offers the correct, available detail of expressing a region in a structured document.

1. Where the document region is completely described by an existing ID, use that ID to define the region.
   Example: `http://data.elsevier.com/content/article/DOI:10.1016/S0030-3992(02)00069-5#p0100` specifies a document region as the element with ID "p0100".
2. Where the document region can be completely described by an element within an ID'd element, navigate outwards to an ID that encloses the region, and use a relative Xpath.

https://doi.org/10.23106/dcmi.952135784

Example:
`#xpath-e(id('s0050')/ce:para[4])` specifies a document region as the fourth ce:para element within an element with ID "s0050".

3. Where the document region cannot be completely described by an element within the content, use the above locators combined with substrings.
   Examples: `#xpath-e(substring(id('p0100'),10,20))` specifies a document region as being characters 10–20 in the element with ID "p0100". And `#xpath-e(substring(id('s0050')/ce:para[4],27,42))` specifies a document region as being characters 27–42 of the fourth `ce:para` in the element with ID "s0050".
4. Where the source content does not contain IDs, use absolute Xpaths to navigate to the appropriate element, and use substrings as required.
   Example: `#xpath-e(article/body/ce:sections/ce:section[4]/ce:para[4])` points to a particular ce:para as defined by the given Xpath.

3.5 Quality Analysis

Annotation data can be the result of applying text mining algorithms, statistics and pattern matching algorithms that inherently have a margin of uncertainty. The process of manual curation is an essential element of QA that must be modeled as an integral process within the end-to-end workflow, without blocking or delaying production turn-around-times.

To manage uncertainty in quality, a number of methods and tools are applied that have in common that they need to be sourced with data; data that must be related to the annotation under consideration. Within Elsevier, the quality is assessed by:

- Visual inspection by a subject matter expert on keywords-in-context, and/or,
- An analysis sheet of key metrics describing the frequencies, distributions and concept-specificity of annotations compared to a Gold standard test and the reference vocabulary, and,
- Validation of concept URIs that must align with valid concept URIs in the reference vocabulary. The reference vocabularies are listed as `skos:conceptscheme` within the core satellite structure.

```
...
<tag:relatedAnnotation>
   <tag:TaggingAnnotation rdf:about="#stmt-83">
      <tag:score>0.5773502691896258</tag:score>
      <tag:relevance rdf:resource="http://data.elsevier.com/ns/ Satellite/RelevanceCodes-1/Major"/>
      <tag:targetText>risk factors</tag:targetText>
      <tag:status rdf:resource="http://data.elsevier.com/ns/ Satellite/TagAnnot-1/Unreviewed"/>
      <pav:createdBy rdf:resource="http://data.elsevier.com/enh-service/Project-1/v1"/>
      <pav:createdOn>2010-12-02T19:53:19Z</pav:createdOn>
   </tag:TaggingAnnotation>
 </tag:relatedAnnotation>...
```

FIG. 3: keeping an audit trail

To keep track of versions and construct an audit trail, date information and provenance are essential metadata properties. However, this information must be handled diligently to avoid ambiguity on the scope of what has been changed: the satellite instance *itself* that holds the annotations or the discrete annotations in their own right. For this the `dct:date` and `dct:creator` predicate is used at the outer, satellite scope, while the `pav:date` and `pav:createdBy` predicates, taken from the ontology (Ciccarese et. al., 2008) are used at the atomic annotation scope level.
During the setup and tuning of the text mining services, it has been recognized that the satellites create value by maintaining consistency between related resources (articles, taxonomies, annotations and customer facing product features) that would otherwise be nearly impossible to keep in alignment.

https://doi.org/10.23106/dcmi.952135784

On many occasions, modifications to the taxonomy, deprecation of search terms in a product and rejection of false positives during QA of tagging, are handled in parallel, making the satellite file an authority file in its own right that can expose inconsistencies that could not be detected by managing the resource in isolation. By tracking the lifecycle of a satellite, a record is kept to measure, in a quantitative way, the cost effects of modifying the vocabularies, rerunning tagging jobs, manual curation and viewing the precision and recall of search result in the product.

3.6 Vocabulary

Taxonomies and controlled vocabularies serve as dedicated resources for content enhancement suppliers to be used for categorizing and annotating content. Because SKOS is an application of RDF, the vocabulary can be embedded as a data payload within the satellite.

```
<voc: satelliteWrapper ...>
        <rdf:RDF ... >
                <!-- Satellite -->
                < sat:Satellite >
                <!-- Which vocabulary the satellite is about -->
                <skos:ConceptScheme rdf:about="the URI for the whole scheme">
                <!-- Individual concepts within the thesaurus -->
                <skos:Concept rdf:about ="the URI for this concept">
                <skos:inScheme rdf:resource="the URI for the above scheme">
                <labels, cross-references and other data for this concept>
                </skos:Concept>
                ....
        </rdf:RDF>
</voc: satelliteWrapper >
```

FIG. 4: The Vocabulary satellite skeleton

The above structure is followed by all vocabulary satellites. It has the following logical sections:

1. Outer wrapper with namespace declarations
2. Header information: the satellite identifier, which vocabulary it describes, the satellite type, project, creation date, etc
3. A SKOS Concept scheme to identify a set of concepts.
4. The individual concepts within the vocabulary with any of the available SKOS Core classes (modeled in RDFS) and properties.

4. Conclusion and future work

The satellite format has delivered a technical format for expressing the creation, transport and manual curation of metadata objects throughout its publishing lifecycle. Important aspects of the implementation of the satellite format is the embedding of quality assurance workflows in the editorial back-office, the technical engagement with content enhancement suppliers and the enablement of features on customer facing platforms, while avoiding disruption on production. A careful process of release management on impacted content processing systems, intensive knowledge exchange with suppliers, and re-tooling the business process of QA on annotation runs, have made the satellite format a key metadata resource within Elsevier production. Field testing with external content enhancement suppliers have both validated the design of the satellite and offered a number of extensions to the current framework:

- Provenance of annotations can be extended by recording applied features from the text mining domain to support the evidence for tagging, e.g. Natural Language Processing analysis, applied dictionaries, entity extractions and white/black listings.
- In its current application, the satellite instance holds data both for transport of primary metadata and secondary data used for QA. The potential redundancy of data can be managed by decomposing the satellite container further. A mechanism for shredding or filtering satellites is the proposed implementation path.
- Change information for concepts in a vocabulary satellite is currently not supported. As vocabularies are subject to change, this will impact satellites that make use of concepts

https://doi.org/10.23106/dcmi.952135784

that may no longer be current. Change information within a vocabulary satellite can facilitate impact analysis on existing satellite resources.

- A formal metric will be implemented showing key performance indicators on manual curation turnaround times, number of reruns for tagging and precision/recall ratios against Gold sets. These metrics will be used to guide taxonomy maintenance and tagging jobs in a cost effective way.

Currently, XML Schema and RDF-based technologies are sufficient to engineer the satellite workflows. It is anticipated that the implementation will move towards RDFS (W3C, 2004b) to allow for model-driven validation of the satellite vocabularies. The implementation of OWL, as proposed by the work of Ciccarese et al. (2008), is not yet considered as the impact on existing technology stacks will prove to be counterproductive in this phase of the lifecycle of the satellite model.

References

Ciccarese, P., E. Wu, G. Wong, M. Ocana, J. Kinoshita, A. Ruttenberg, and T. Clark. (2008). The SWAN biomedical discourse ontology. J Biomed Inform 41, 739-51.

Miles, A. and S. Bechhofer. (2009) SKOS Simple Knowledge Organization System Reference. W3C Recommendation. Available: http://www.w3.org/TR/skos-reference/

Nilsson, M., A. Powell, P. Johnston and A. Naeve, A. (2008). Expressing Dublin Core metadata using the Resource Description Framework, DCMI Recommendation. Available: http://dublincore.org/documents/dc-rdf/

W3C Interest Group. (2009) Semantic Web Applications in Neuromedicine (SWAN). Ontology. Note 20, October 2009. Available: http://www.w3.org/TR/hcls-swan/

World Wide Web Consortium (W3C). (2004a). D. Beckett ed. RDF/XML Syntax Specification (Revised), W3C Recommendation. D. Beckett ed. Available: http://www.w3.org/TR/REC-rdf-syntax/

World Wide Web Consortium (W3C). (2004b) RDF Vocabulary Description Language 1.0: RDF Schema. Available: http://www.w3.org/TR/rdf-schema/

https://doi.org/10.23106/dcmi.952135784