From Markup to Linked Data: Mapping NISO JATS v1.0 to RDF using the SPAR (Semantic Publishing and Referencing) Ontologies

S. Peroni,¹ D.A. Lapeyre,² and D. Shotton³.

¹ Department of Computer Science, University of Bologna
essepuntato@cs.unibo.it
² Mulberry Technologies, Inc.
dalapeyre@mulberrytech.com
³ Research Data Management and Semantic Publishing Research Group, Department of Zoology, University of Oxford
david.shotton@zoo.ox.ac.uk

The Journal Article Tag Suite (JATS), published on 22 August 2012 as ANSI/NISO Z39.96-2012, JATS: Journal Article Tag Suite (version 1.0), is the successor to the National Library of Medicine (NLM) DTD. It is de facto standard for the XML markup of scholarly journal articles that is widely used by many academic publishers within their routine publication workflows, and also used as the ingest format for PubMed Central.

The Resource Description Framework (RDF) is the key enabling technology for the Semantic Web, also know as the web of linked data. By defining statements about entities and their relationships in RDF syntax using publicly available ontologies, such statements can be combined into interconnected information networks (RDF graphs) in which the truth content of each original statement is maintained, thereby creating a web of linked data, the Semantic Web.

The SPAR (Semantic Publishing and Referencing) Ontologies are a suite of complementary and orthogonal OWL 2 DL ontologies. They were created to permit RDF descriptions of bibliographic entities, citations, reference collections and library catalogues, the structural and rhetorical component parts of documents, and roles, statuses and workflows in publishing.

This paper describes JATS2RDF v1.0 (http://purl.org/spar/jats2rdf/), a mapping of the principle metadata components of the ANSI/NISO JATS Journal Publishing Tag Library Version 1.0 from XML to RDF. Our mapping uses the SPAR ontologies, together with elements from other well-known vocabularies such as the Dublin Core Metadata Initiative (DCMI) Metadata Terms and the Friend of a Friend (FOAF) Vocabulary. By means of an Extensible Stylesheet Language Transformation (XSLT) transform that we have also created (http://purl.org/spar/jats2rdf/xslt), this JATS2RDF mapping now permit the JATS metadata elements and their attributes, from documents marked up in XML using the NISO-JATS Journal Publishing Tag Library v1.0, to be converted automatically to RDF, enabling this information to be published to the Semantic Web as linked open data in a manner that is unambiguous and universally understood.

We hope that this ability to express in RDF the JATS Journal Publishing Tag Library metadata descriptions will promote the use of JATS to a wider community.

Introduction

The changing world of scholarly communication

On 6th March 1665, the first issue of the Philosophical Transactions of the Royal Society appeared, edited and published by the Royal Society’s first secretary, Henry Oldenburg (4). In the intervening 347 years since this that event, widely regarded as the start of modern scholarly communication by means of journal articles, very little has changed in terms of the characteristics of such articles. Most research articles today still have a linear narrative, with a standardized sequence of sections having different semantic purposes (e.g. Introduction, Methods, Results, Discussion, Conclusions), and conclude with a reference list acknowledging the previously published scholarly papers that have influenced the author’s own work.

While now, at the start of the 21st Century, the majority of scholarly papers are available on-line, the prevailing norm is to publish such on-line journal articles as static PDF files that are facsimiles of the printed page. However, this is totally antithetical to the spirit of the Web, and ignores its great potential.

In the previous world of printed scholarship:

- articles were finite, and were published as finished documents - ‘versions of record’ - that had been privately peer reviewed before publication;
- publication costs were high, leading to size limitations being placed on papers;
- paper publications did not scale or link, and references did not ‘work’, in the sense that one could not click on a reference and be taken to the cited paper;
- large quantities of related data were hard to include; and
- print collections were created in physical libraries near to communities of users.

Indeed, the world of printed scholarship was a closed world: if an article had been published, the knowledge ‘existed’, and it was up to the scholar to find it in the library. Conversely, if the knowledge had not been published in the scholarly literature, it does not officially ‘exist’.

The current world of the Web offers a distinct contrast:

- information is extensive, scattered, incomplete and of variable quality - journal articles are just a small part of the overall picture;
- discussion and peer review of an article can occur both before and after publication, as an on-going process;
- publication costs are very low, and there are no practical limits to size;
- the Web scales, links are everything, and resolvable Digital Object Identifiers (DOIs) accompanying references take you to cited papers;
- datasets are easy to include or to link to; and
- for digital collections of scholarly information, the geographical locations are largely irrelevant, and the need for physical library buildings is reduced.

In future, if a journal article does not fully embrace Web technologies, so that it can be easily found, and so that the data within it are made available in actionable form, there is a danger that the knowledge contained within the article will be increasingly ignored in an open world with alternative media for scholarly communications, including blogs and research data repositories.

One of the significant changes that is required if scholarly publishing is to adopt Web technologies is that of ensuring that the metadata used by publishers to describe journal articles conforms to the best available Web technologies, and that these are encoded in such a form that they can easily be integrated with similar metadata describing resources from other publishers. This makes JATS highly relevant.

**JATS**

The Journal Publishing Tag Suite (JATS) defines a vocabulary of XML elements and attributes that describe the content and metadata of journal articles, where “journal article” has been defined broadly to include as both research and non-research articles. Thus original research, review articles, letters to the editor, editorials, instructions to authors, and book and product reviews are all defined as journal articles. This Tag Suite contains elements that describe the full narrative content of such an article, its graphical and media components, and the article header metadata.

JATS was based on previous work at the National Library of Medicine and on a thorough study of available proprietary and standard journal article DTDs and schemas. In 2003, the National Centre for Biotechnology Information (NCBI), a division of the National Institutes of Health (NIH), released the National Library of Medicine (NLM) Archiving and Interchange Tag Suite, and two article Document Type Definitions (DTDs) for tagging journal articles: the Archiving and Interchange DTD and the Journal Publishing DTD. In 2005, for version 2.1, the third article model, the Article Authoring DTD, was added. Version 3.0, which was a major revision, was released in 2008.

This NLM DTD became a de facto standard for the XML markup of scholarly journal articles, widely used by many academic publishers within their routine publication workflows. The NLM DTD was also used as the standard ingest format for PubMed Central, a large public repository of full-text journal articles in the field of biomedical sciences.

Further development of the NLM DTD was then switched to a working group of the National Information Standards Organization (NISO), and earlier this year NISO released a draft version (version 0.4) of the standard, renamed the Journal Article Tag Suite (JATS), as a minor update to version 3.0 of the NLM Tag Suite that was fully backward compatible with NLM version 3.0.


As does the NLM DTD, JATS contains three tag sets, the Journal Archiving and Interchange Tag Set, the Journal Publishing Tag Set, and the Article Authoring Tag Set, intended for slightly different purposes. Both for reasons of tag set modelling tightness and for maximum user uptake, we chose the JATS Journal Publishing Tag Set Version 1.0, to which we were given pre-publication access, for the JATS2RDF mapping that is the subject of this paper.

Journal Publishing Tag Set is a moderately prescriptive Tag Set, optimized to regularize and control the
sequence of the XML content, not to accept whatever arrangement is delivered by any particular publisher. It can be used for the initial XML tagging of journal material, usually as converted from an authoring format such as Microsoft Word. The philosophy of the Publishing Tag Set is to prefer a single structural form whenever possible. Elements and tagging choices have been limited to produce more consistent data structures and to provide a single location of information for searching.

RDF
The Resource Description Framework (RDF) is the key enabling technology for the Semantic Web, also known as the web of linked data. Uptake by a number of influential parties such as the British Broadcasting Corporation has recently brought Semantic Web technologies into widespread acceptance. The principles are quite simple. If entities and their relationships are identified and defined in machine-readable form by the use of unique URIs referencing publicly available and commonly accepted structured vocabularies (ontologies) in which the meaning of terms are defined, and if each of these relationships is expressed, using the RDF syntax, as a simple subject - predicate - object ‘triple’, such as the statement

\[
\text{<Paper a> cites <Paper b> .}
\]

then such statements can be combined into interconnected information networks (RDF graphs) in which the truth content of each original statement is maintained, thereby creating a web of linked data, the Semantic Web.

The SPAR (Semantic Publishing and Referencing) Ontologies
The use of common ontological descriptions of entities and their relationships enables data from independent sources to be integrated without ambiguity or loss of precision of meaning, a situation that would be impossible if the entities were to be described using a variety of XML vocabularies, since the lack of universally agreed meanings for markup terms could easily lead to confusion with respect to homonyms or synonyms (for example, whether “creator” in one schema is equivalent to “author”, “contributor”, “composer” or “choreographer” in another).

The SPAR (Semantic Publishing and Referencing) ontologies (http://purl.org/spar) are a suite of complementary and orthogonal OWL 2 DL ontologies that we have published to permit the creation of RDF descriptions covering all aspects of the scholarly publishing world, including bibliographic entities, citations, reference collections and library catalogues, the structural and rhetorical component parts of documents, and publishing roles, statuses and workflows (9).

Together with other well-known and commonly used vocabularies such as the Dublin Core Metadata Initiative (DCMI) Metadata Terms and the Friend of a Friend (FOAF) Vocabulary, the SPAR ontologies provide the ontological tools required to encode publication metadata in RDF.

The desirability of expressing document metadata in RDF
Scholarly authoring and publishing are in the throes of a revolution, as the full potential of on-line publishing is explored. Yet, to date, publishers have not adopted Web standards for their work, but rather continue to employ a variety of open or proprietary XML-based informational models and document type definitions (DTDs). JATS is one of these.

While such heterogeneity and independence was reasonable in the pre-web world of paper publishing, it now appears anachronistic, since publications and their metadata from different sources are incompatible, requiring hand-crafted mappings to convert from one to another. For a large community such as publishers, this lack of standard definitions that could be adopted and reused across the entire industry represents losses in terms of money, time and effort.

In contrast, modern web information management techniques employ standards such as RDF (1) and OWL 2 (12) to encode information in ways that permit computers to query metadata and integrate web-based information from multiple resources in an automated manner. This opens the possibility for semantic publishing.

Semantic publishing (10) is the use of Web and Semantic Web technologies to enhance the meaning of a published journal article, to facilitate its automated discovery, to enable its linking to semantically related articles, to provide access to data within the article in actionable form, and to facilitate integration of data between articles, opening up the possibility of major advances in the digital publishing world.

Since the processes of scholarly communication are central to the practice of science, it is essential that publishers now adopt such standards and technologies to permit inference over the entire corpus of scholarly communication represented in journals, books and conference proceedings. The purpose of this paper is to describe one step towards that goal, namely the mapping of metadata elements from the Journal Publishing Tag Set of JATS to RDF to enable publishers’ XML metadata encoded using JATS to become part of the web of linked data.

Differing philosophical viewpoints for XML and RDF
Open versus Closed
In Section 1.1, we have already contrasted the open world view of the Web information publishing with the
closed world of print publishing. This 'open world' philosophical viewpoint underpins Semantic Web technologies, and is commonly contrasted with the 'closed world' of database technologies, whereby, if an item of information is not present in a database, its converse is assumed to be true. For example, if a journal article is not recorded as being open access, it is assumed not to be. In the contrasting RDF world view, if an article is not describes as being open access, one has to keep an open mind - it might be, or it might not.

Precise Semantics for Markup Terms

A second interesting contrast arises when comparing the open world of RDF descriptions with that of XML markup. Here, the contrast is not between the different assumed meanings of unstated assertions, but between the semantic meanings of markup terms. A cornerstone of the Semantic Web is the use of open published ontologies to give precise and universally available definitions to terms, so that RDF statements, whatever else they are, are unambiguous in their meaning. This is not the case in the world of XML, where markup terms can take on different meanings (5), depending upon who is using them, reminiscent of Humpty Dumpty's statement in Alice's Adventures in Wonderland (2):

'When I use a word,' Humpty Dumpty said in rather a scornful tone, 'it means just what I choose it to mean - neither more nor less.'

For JATS, this is by design. JATS is a descriptive, not a prescriptive model, that endeavours to capture and document the actual practice of current publishing. It does not tell publishers what they should call their content; rather, if a term usage is widely practiced it is likely to appear in the JATS, which aims to provide a vocabulary that will be used more or less consistently across publishers. Furthermore, suggested values for JATS elements and attributes lists are just that - suggested, since JATS provides structures for recording different types of information, but does not attempt to regularize their usage.

For example, the JATS documentation describes the central element <article> as follows:

Usage: This element can be used to describe not only typical journal articles (research articles) but also much of the non-article content within a journal, such as book and product reviews, editorials, commentaries, and news summaries.

Thus the JATS element <article> may be used to describe an XML representation of a research article, or an XML representation of many other kinds of journal content, such as an editorial, obituary, list of events, book review, puzzle, game, quiz, interview or photo-essay, depending upon the meaning an individual publisher chooses for this tag element. This goes beyond what the average publisher or average person means by "journal article."

In other words the JATS standard is deliberately vague and non-committal about the meaning of many terms, both because there is no intention to tell any publisher what or how much metadata to publish, and also because it is intended that conversion from other markup systems into JATS should be achievable automatically and relatively painlessly.

As a consequence of this 'loose' design, the first barrier we came up against when mapping JATS to RDF was that a JATS element might mean what its name implies, but equally might be used by some publishers to mean something entirely different. What JATS means by <article> is most frequently what is defined in FaBio, the FRBR-aligned Bibliographic Ontology, as a fabio:JournalArticle, but it can also mean fabio:JournalEditorial, fabio:JournalNewsItem, fabio:BookReview, fabio:ProductReview, etc., all of which are journal content items that can be mapped in RDF as follows:

:periodical-entity a fabio:PeriodicalItem ;
  frbr:partOf [ a fabio:JournalIssue ] .

However, it is also permissible to use JATS <article> to describe textual entities before they appear in a journal issue, for example to describe a preprint or a revised manuscript submitted to a publisher. Clearly, this brings problems for unambiguous XML mapping to specific RDF terms. Our solution to this dilemma has been, where necessary, to map the entity described by <article> to :textual-entity, a resource name that is broad enough to include all relevant possibilities, thereby achieving semantic accuracy, if not detailed specificity.

Hierarchies versus Triples

Finally, the last clear difference between XML and RDF we would like to discuss concerns the structural organisation of items (i.e. the elements in XML and the resources in RDF). XML is able to structure elements according to a particular containment order, thus creating hierarchies of nested elements. Of course, a containment relation between two XML elements always carries a particular semantics, although it is not formalised and implicitly lives outside the XML schema of the language. Let us consider the following two excerpts of JATS markup:

<article-meta>
  <title-group>
    <article-title>
Dealing With Markup Semantics
</article-title>
</title-group>
<article-meta>

<element-citation>
  <article-title>
    Dealing With Markup Semantics
  </article-title>
</element-citation>

Above, the element article-title is used in two different contexts, thus having two alternative interpretations. In the former excerpt (i.e. when it is descendant of the element article-meta) it is the title of the article under consideration, while, in the latter excerpt, it represents the title of another bibliographic work that the article under consideration cites.

The semantics hidden behind XML containment relationships is the main issue one has to address to correctly map XML schemas to RDF vocabularies, since:

- RDF is not able to represent the hierarchical relation of XML elements using native constructs, since everything is described as a 'flat' graph of resources; and
- the semantics of the containment of XML elements, such as the aforementioned article-meta/article-title and element-citation/article-title, is neither explicitly nor formally defined - it can live either in a natural language definition of the element or in the mind of the developer of the schema or, sometimes, in the mind of the author of the XML document.

Of course, RDF can express hierarchical relationships, that are clearly defined by the DL logic of the ontologies from which terms are used. Thus fabio:hasShortTitle and fabio:hasTranslatedTitle are both sub-properties of dcterms:title. However, such hierarchical definitions do not address the contextual semantics of XML determined by containment relationships.

Thus one of the main goals of our mapping has been to try to disambiguate (as far as possible) the semantic relations between the entities referred to by the JATS XML elements.

Mapping to the FRBR data model

The FRBR (Functional Requirements for Bibliographic Records) classification model (11), developed by the United States Library of Congress for characterizing different aspects of a publication, distinguishes Works, Expressions, Manifestations and Items:

- A Work is a distinct intellectual or artistic creation, an abstract concept recognised through its various expressions.
- An Expression is the specific form that a Work takes each time it is 'realized' in physical or electronic form.
- A Manifestation is a particular physical or electronic embodiment of an Expression. Typically, the print version of an article, the on-line HTML version of that article, and the downloadable PDF file are three separate Manifestations of the same Expression, all bearing the same DOI. They can be viewed as alternative 'containers' or 'channels' for the same information.
- In FRBR, an Item is one single exemplar copy of a Manifestation, i.e. a physical or electronic copy of a document that can be owned by a person.

In creating FaBIO (6), we aligned our ontology to this FRBR model, deciding that some things (e.g. an Opinion, a Research Paper, a Novel) were Works that could have various forms of Expression, while others (e.g. an Editorial, a Journal Article, a Book) were Expressions, and yet others (e.g. a Reprint) were clearly Manifestations.

We adopted this same methodology in creating the mapping from JATS to RDF, using the following general resource names as appropriate:

- :conceptual-work (the Work from which the JATS article derives),
- :textual-entity (the Expression of the article, bearing the JATS XML markup),
- :digital-embodiment (the Manifestation a digital Manifestation of the article),
- :digital-item (the Item a single copy of the JATS article).

Because JATS markup is applied to particular documents in publishers' production workflows (i.e. to particular FRBR Expressions), we chose that JATS entities should be mapped to Expression-level classes within FaBIO.

The mapping statements in our JATS2RDF mapping document presume the following RDF relationships to exist between these entities:
These relationships are specified explicitly early in the mapping document (as the first item in Table 2, see JATS2RDF mapping document), and are implicit in the subsequent mapping statements.

Similarly, other statements, such as


are presumed in subsequent mappings, after having been first stated.

In some mappings, we specify more than one FRBR element for greater clarity, for example encoding the element research-article as an article (Expression) as a realisation of a research paper (Work), using the two statements:

: textual-entity a fabio:Article .

: conceptual-work a fabio:ResearchPaper .

Choosing what JATS entities to map to RDF

The JATS Journal Publishing Tag Library Version 1.0 specification is large, containing 246 elements and 134 attributes. We chose to map the JATS metadata entities that describe an article (e.g. <journal-meta> for metadata about the journal in which the article was published, such as the name of the journal), and to leave aside (possibly for a later mapping exercise using DoCO, the Document Components Ontology) those entities describing the textual and graphical structure and content of the article (e.g. <title>, <body>, <fig>, <table>).

The principle metadata elements that we chose to map are <article>, <article-meta>, <journal-meta>, <contrib>, and <ref-list>, plus their principle contained elements and attributes. The mappings for each of these five principle metadata elements are detailed in five separate tables in the JATS2RDF mapping document (Tables 2-6). In all, 242 separate XML to RDF mapping statements have been made.

Choosing the ontologies and RDF vocabularies to use for the mappings

For the mapping, we have chosen to use terms from the following SPAR (Semantic Publishing and Referencing) Ontologies:

- BIBO, the Bibliographic Reference Ontology,
- CiTO, the Citation Typing Ontology,
- the DataCite Ontology,
- FaBiO, the FRBR-aligned Bibliographic Ontology,
- DEO, the Discourse Elements Ontology,
- PRO, Publishing Roles Ontology,
- PSO, Publishing Status Ontology, and
- PWO, Publishing Workflow Ontology;

and from the other specialized ontologies listed in Table 1 of the mapping document.

These have been supplemented, where necessary, by terms from:

- FRBR (Functional Requirements for Bibliographic Records),
- FRAPO, the Funding, Research Administration and Projects Ontology,
- SCORO, the Scholarly Contributions and Roles Ontology;

In addition, wherever possible, we have used terms from the following commonly used vocabularies such as:

- the Dublin Core Metadata Initiative (DCMI) Metadata Terms,
- FOAF (the Friend of a Friend Vocabulary),
- PRISM (Publishing Requirements for Industry Standard Metadata) and
- vCard, used to store contact information for electronic business cards.

Adding new classes and properties to FaBiO and FRAPO

Our mapping activity has revealed a number of document types and relationships which had no previous representation within the relevant ontologies.
To enable accurate mapping, we have thus added the following classes to FaBiO and FRAPo:

- `fabio:InBrief`
- `fabio:Correction`, making the following existing ontology classes to be sub-classes of this:
  - `fabio:Corrigendum`
  - `fabio:Erratum`
- `fabio:ExecutiveSummary`
- `frapo:PostalAddress`
- `fabio:Supplement`

We further added the following object property to SCoRO:

- `scoro:isEquivalentToContributionSituation`

and the following additional data properties to CITO, FaBiO and FRAPo:

- `cito:citesAsRecommendedReading`
- `fabio:hasCODEN`
- `fabio:hasPII`
- `fabio:hasStandardNumber`
- `fabio:hasTranslatedSubtitle`

- `frapo:hasInitial`, with two sub-properties:
  - `frapo:hasFamilyNameInitial`
  - `frapo:hasGivenNameInitial`
- `frapo:hasNameSuffix`, with three sub-properties:
  - `frapo:hasDegreeSuffix`
  - `frapo:hasFamilialSuffix` and
  - `frapo:hasHonorificSuffix`
- `frapo:hasPostalAddressLine`

In addition, we modified the super-class restrictions on and/or the textual descriptions of the following classes and properties within FaBiO to fit more precisely to the intended JATS meaning:

- `fabio:Abstract`
- `fabio:BriefReport`
- `fabio:RapidCommunication`
- `fabio:Reply`
- `fabio:hasShortTitle`
- `fabio:hasRequestDate`
- `prism:section`

Finally, we added the following new members to the class `pro:Role`:

- `pro:guest-editor`
- `pro:photographer`

**JATS2RDF mapping decisions**

In this section of the paper, we describe the mapping decisions we made, in those cases where these were not obvious. Many of these decisions are also noted as footnotes in the JATS2RDF v1.0 mapping document (http://purl.org/spar/jats2rdf/).

The following 18 paragraphs are grouped under the following general headings:

- Publications
- Identifiers
- References
- Publication types and formats
• Dates and places
• People and roles

Publications

<article>
As explained in above, the JATS element <article> may be used to describe an XML representation of an article, or an XML representation of another kind of journal content, such as a book review, or even a document before publication. As a consequence of this 'loose' design, we chose to map <article> not to :this-article, but to a very general resource name, :textual-entity, that is broad enough to include all relevant possibilities, thereby achieving semantic accuracy if not detailed specificity.

We thus map the XML:

<article xml:lang="en" ... </article>

to :textual-entity, and specify the language of :textual-entity, and the relationships to its related FRBR entities as follows:

:textual-entity
  a fabio:Expression ;
  frbr:realizationOf :conceptual-work ;
  frbr:embodiment :digital-embodiment ;
  fabio:hasRepresentation :digital-item ;
  dcterms:language [ a dcterms:LinguisticSystem ;
    dcterms:description "en" "~dcterms:RFC5646" ].

The @article-type attribute was used to determine the appropriate FaBiO class to describe the JATS <article> to be mapped. Consistency of access was deemed more important than specificity, so we have consistently only used the general categories :conceptual-work, :textual-entity, :digital-embodiment and :digital-item, depending on the FRBR layer in consideration, rather than the more specific resource names such as :this-article or :this-calendar.

Identifiers

@contrib-id-type

The JATS documentation says that the attribute @contrib-id-type "names the type of contributor identifier or the authority that is responsible for the creation of the contributor identifier." In other words, the attribute can be used either to identify an identifier scheme, or to identify the organization or authority responsible for that identifier scheme. Thus, in the generic case, in which the XML is:

<contrib-id
  contrib-id-type="YYY">
  XXX</contrib-id>

we provide the alternative mappings:

:this-agent datacite:hasIdentifier [ a datacite:Identifier ;
  literal:hasLiteralValue "XXX" ;
  datacite:usesIdentifierScheme [ a datacite:IdentifierScheme ;

or

:this-agent datacite:hasIdentifier [ a datacite:Identifier ;
  literal:hasLiteralValue "XXX" ;
  prov:wasAttributedTo [ a prov:Agent , foaf:Organization ;
    dcterms:description "Identifier scheme authority" ;

@journal-id-type

In exactly the same manner, JATS describes @journal-id-type as "Type of journal identifier or the authority that created a particular journal identifier." For this reason, for the generic case:

<journal-id
  journal-id-type="YYY">
we provide the following alternative mappings:

:journal datacite:hasIdentifier [  
a datacite:Identifier ;  
literal:hasLiteralValue "XXX" ;  
datacite:usesIdentifierScheme [  
a datacite:IdentifierScheme ;  
rdfs:label "YYY" ] ] ]

.or

:journal datacite:hasIdentifier [  
a datacite:Identifier ;  
literal:hasLiteralValue "XXX" ;  
prov:wasAttributedTo [  
a prov:Agent , foaf:Organization ;  
dcterms:description "Identifier scheme authority" ;  

<isbn>

The JATS specification states that the element <isbn> should be used "for identifying a particular product form or edition of a publication, typically a monographic publication." Since different International Standard Book Numbers are used for hardback and paperback editions of the same book, this number clearly applies to the FRBR Manifestations of the book. However, the JATS definition goes on to state: "In the rare case that a serial publication has an ISBN, <isbn> is used to record that value." In other words, while an International Standard Book Number normally identifies a book, it can also be used to identify an issue or a volume of a periodical. Because of this, the following alternatives are valid, since the mapping depends on the subject entity to which the ISBN relates.

The generic mapping of <isbn> is:

:textual-entity frbr:embodiment [  
a fabio:Manifestation ; prism:isbn "XXX" ] .

.or

:textual-entity frbr:partOf [  
a fabio:Expression ; frbr:embodiment [  
a fabio:Manifestation ; prism:isbn "XXX" ] ] .

When identifying a periodical issue or volume, as part of <journal-meta>:

<journal-meta>
...
<isbn>XXX</isbn>
...
</journal-meta>

can be mapped:

:periodical-issue a fabio:PeriodicalIssue ;  
frbr:part :textual-entity ; frbr:partOf :journal ;  
frbr:embodiment [ a fabio:Manifestation ; prism:isbn "XXX" ] .

.or

:periodical-volume a fabio:PeriodicalVolume ;  
frbr:part :textual-entity ; frbr:partOf :journal ;  
frbr:embodiment [ a fabio:Manifestation ; prism:isbn "XXX" ] .

And when it is part of <article-meta>:

<article-meta>
...
<isbn>XXX</isbn>
...
</article-meta>

can be mapped:

:periodical-issue frbr:embodiment [  
a fabio:Manifestation ; prism:isbn "XXX" ] .
or
:periodical-volume frbr:embodiment [a fabio:Manifestation ; prism:isbn "XXX"].

In each case, the mapping depends on the subject entity to which the ISBN relates.

References

<ref>
In JATS, the element <ref> defines one item in a bibliographic list. We use the Collections Ontology to specify the content of ordered lists in RDF, each numbered item in the list (:ref-XXX, :ref-YYY, etc.) being regarded as a content container, within which the actual reference (:ref-XXX, :ref-YYY, etc.) is the content. Each reference, e.g. :ref-XXX, is defined as a bio:BibliographicReference.

Thus the JATS XML markup:

<ref-list>
  <ref id="XXX"> ... </ref>
  <ref id="YYY"> ... </ref>
  ...
</ref-list>

is mapped:
:ref-list co:item :iref-XXX .
:iref-XXX a co:ListItem ;
  co:itemContent :ref-XXX ;
  co:nextItem :iref-YYY ;
  co:index "1" .
:iref-YYY a co:ListItem ;
  co:itemContent :ref-YYY ;
  co:nextItem :iref-ZZZ ;
  co:index "2" .
:ref-XXX a bio:BibliographicReference .
:ref-YYY a bio:BibliographicReference .
# (etc. until list is complete)
:ref-XXX (the actual bibliographic reference) is then used as the subject in the mapping of <element-citation> and of <mixed-citation>. For example:

<ref id="XXX">
  <element-citation>
    ...
  </element-citation>
</ref>

is mapped as:
:ref-XXX a bio:BibliographicReference ;
  bio:references :textual-entity-XXX .
:textual-entity cites :textual-entity-XXX .
:textual-entity-XXX a fabio:Expression ;
  frbr:realizationOf :conceptual-work-XXX .

In our mapping, we have assumed the normal scientific literature usage that each reference list element contains only one bibliographic reference (a bio:BibliographicReference). However, the following complication arises: in JATS, a <ref> may alternatively contain just a <note> (i.e. a comment, such as an end note), or several bibliographic references, or a mixture of comments and references - the way reference lists are used varies widely between publishers, and JATS <ref> accommodates all such usages.

While we could manually encode in RDF the fact that one reference list item, e.g. :iref-XXX, contains either a single bibliographic reference, e.g. :ref-XXX (a bio:BibliographicReference), or an ordered list of such bibliographic references (each a bio:BibliographicReference), or a note or comment (a fabio:Comment), or a mixture of references and comments, the fact that <ref> permits all of these situations makes it impossible to
devise a mapping to RDF that can be undertaken automatically, without resorting to text mining to parse the reference list item in order to determine the entities it contains.

For this reason, our mapping is, at present, restricted to the simplest and most common case in which a reference list element contains only a single bibliographic reference.

**Publication types and formats**

<article-categories>

This JATS element is used to group together clusters of articles by subject, by article types, by length, by date, by author affiliation, or by any other criterion the publisher wishes to choose. These groupings are often used to create Tables of Contents, which may have sub-categories. Because of this heterogeneity, it is not possible automatically to map this element either to article types or to subject categories.

For this reason, our mapping is deliberately vague and generic. For classification by subject group, we would map:

<article-meta>

...  
<article-categories>  
  <subj-group>  
    <subject>XXX</subject>  
  </subj-group>  
</article-categories>  
...
</article-meta>

to:

textual-entity

  fabio:hasSubjectTerm [  
    a fabio:SubjectTerm ; rdfs:label "XXX" ] .

One could of course be more specific. JATS XML:

<article-meta>

...  
<article-categories>  
  <subj-group>  
    <subject>Biological Sciences</subject>  
  </subj-group>  
  <subj-group>  
    <subject>Entomology</subject>  
  </subj-group>  
</article-categories>  
...
</article-meta>

could be mapped as

textual-entity fabio:hasSubjectTerm :bio-sciences , :ent .

:bio-sciences a fabio:SubjectTerm ;  
  rdfs:label "Biological Sciences" ; skos:narrower :ent .

:ent a fabio:SubjectTerm ; rdfs:label "Entomology" .

Other types of <article-categories> grouping, e.g. by article type or by date, would of course require other RDF descriptions, which are not presently represented in the JATS2RDF mapping document.

@article-type "research article"

Here, by specifying that the article type is "research article", the purpose is to emphasise that a journal article or a conference presentation reports research, rather than being, for example, an editorial or an obituary. In FaBiO terms, it is thus clearly an expression of a research paper, giving the straightforward mapping from XML:

<article

  article-type="research-article">  
...
</article>

to RDF:

:textual-entity a fabio:Article .
conceptual-work a fabio:ResearchPaper.

@publication-type

The attribute @publication-type defines the category of referenced publication being cited within <element-citation> or <mixed-citation>. We specify mappings for the examples of @publication-type given in the JATS documentation (namely "book", "letter", "review", "journal", "patent", "report", "standard", "working-paper"), and also for "ZZZ". One example will suffice to show usage:

<element-citation
    publication-type="book">
    ...
</element-citation>

is simply mapped as:

:textual-entity-XXX a fabio:Book.

Other specific publication types can easily be added to expand the list as required - FaBiO was designed precisely to define such different types of bibliographic entities.

@publication-format

The JATS documentation for this optional attribute that defines the format of a publication suggests the following values: "print", "electronic", "video", "audio", "ebook", "online-only". "Online" and "web" are additional possibilities.

The first point to make is that in the open world of RDF, we would not wish to state that a publication is "online-only", since we cannot read the future - someone might come out with a print edition later on, and we wish our RDF encoding to be as true in the future as it is in the present.

The second point is that the classification of resources here potentially involves three different types of metadata, describing:

1. the nature of the information, e.g. text, image, sound;
2. the nature of the storage medium, e.g. paper, analogue tape, digital tape, CD/DVD, Web; and
3. the analogue or digital file format, e.g. PDF, XML, VHS and JPEG.

In FaBiO, we have the following classes, separate from the FRBR categories relating to bibliographic entities:

    fabio:StorageMedium (something independent of the content), with sub-classes

    fabio:AnalogStorageMedium

    (with members fabio:analog-magnetic-tape, fabio:film, fabio:paper and fabio:vinyl-disk),
    and

    fabio:DigitalStorageMedium


Other individuals can easily be added to these classes as need arises, without changing the structure of the ontology.

The types of FRBR Manifestation that can be stored on these various storage media in different formats are simply classified as follows:

    fabio:AnalogManifestation, with sub-class

    fabio:PrintObject, which has its own sub-classes

    fabio:Hardback and fabio:Paperback; or

    fabio:DigitalManifestation - "any representation of data in binary form", and its sub-class

    fabio:WebManifestation - "A digital manifestation on the Web, such as a wiki, a web site, a web page or a blog", with these as its sub-classes:


We also have the following types of FRBR Items:

    fabio:AnalogItem, and
To specify (an analogue and digital) file formats such as PDF, XML, VHS and JPEG, we use the Dublin Core property dcterms:format and the Dublin Core class dcterms:MediaTypeOrExtent, as shown in the mapping document and illustrated below.

This distinction between what is stored, the storage medium, and the file format gives us clarity and flexibility in describing things. However, the JATS proposal that "print", "electronic", "video", "audio", "ebook", "online" and "web" are all permissible values for the attribute @publication-format mix the three categories described above, referring in one group of suggested values either to the nature of the information ("video", "audio"), or to the digital nature of the manifestation ("electronic", "ebook"), or to the nature of the storage medium ("print", "online", "web").

Furthermore, the classification "electronic" is itself problematical. "Video" and "audio" information can be stored either on analogue or digital media. While analogue storage media like vinyl records and magnetic tapes are not themselves "electronic", the rendition of the information contained on them is almost always electronic, but can also be analogue - think of Scott Joplin-type punched paper player piano rolls, and vinyl discs played on ancient non-electronic record players equipped with an acoustic horn attached to the needle. Similarly, a DVD, containing digital information, is not itself electronic, but the rendition of the information contained on it is. "Electronic" thus seems a difficult term to handle, unless, as we have done, one makes the assumption that "electronic" really means "digital". Our mapping of the value "electronic" for the JATS attribute @publication-format is to something that has a digital manifestation, i.e.:


Similarly, mapping the value "print" for the JATS attribute @publication-format is simple:


One could additionally say, if one wished,

`:textual-entity frbr:embodiment [ a fabio:PrintObject ;

to distinguish it from something printed on plastic, metal, velum or some other two-dimensional analogue material.

In mapping "video" and "audio", we specify the nature of the information being stored, but, without further details being provided, we are unable to specify whether their manifestations are encoded in analogue or digital format. For example:

```
<pub-date
  publication-format="video">
  <day>DD</day>
  <month>MM</month>
  <year>YYYY</year>
</pub-date>
```

is mapped as:

```
:conceptual-work a fabio:MovingImage ; fabio:hasManifestation [ a fabio:Manifestation ; dcterms:date "YYYY-MM-DD"^^xsd:date ]
```

In mapping "online" and "web", it is the nature of the publication platform that is the important thing to specify. Thus, for example:

```
<pub-date
  publication-format="online">
  <day>DD</day>
  <month>MM</month>
  <year>YYYY</year>
</pub-date>
```

is mapped as follows:

```
:conceptual-work a fabio:DigitalObject ;
  frbr:embodiment [ a fabio:ComputerFile ;
    frbr:isExemplar [ a fabio:ComputerFile ;
      fabio:isExemplarOf fabio:computerfile ] ;
    dcterms:date "YYYY-MM-DD"^^xsd:date ] .
```

Ebooks (i.e. books that have a digital manifestation) are published in a large variety of formats, so we use dcterms:format and dcterms:MediaTypeOrExtent to specify the specific format used (if know), after declaring:
For example:

```
:textual-entity a fabio:Book ; frbr:embodiment [ 
  a fabio:DigitalManifestation ] .
```

Dates and places

`<pub-date>` and `<date>`

`<pub-date>` can, despite its name, be defined by its `@date-type` to be a date other than the publication date of an article. Thus `<pub-date>` can be used to specify the correction date of the Work, or the retraction date of the Expression. For this reason, we cannot use the more specific data property `prism:publicationDate` to map `<pub-date>` to RDF, but rather have to use the generic property `dcterms:date`. The choice of mapping will depend on the context. Hence, for XML:

```
<article-meta>
  ...
  <title-group>
  ...
  </title-group>
  <pub-date>
    <day>DD</day>
    <month>MM</month>
    <year>YYYY</year>
  </pub-date>
  ...
</article-meta>
```

we have provided the following alternative mappings:

```
:conceptual-work dcterms:date "YYYY-MM-DD"^^xsd:date .
```

or

```
:textual-entity dcterms:date "YYYY-MM-DD"^^xsd:date .
```

or

```
```

The choice between these will depend upon context, and upon what is specified by the `@date-type` attribute.

For `<date>`, as for `<pub-date>`, the same alternatives are valid, since `<date>` can be applied to so many different things in different contexts.

`@date-type ("corrected", "ecorrected", "pcorrected")`

If one thinks of the FRBR classification of Works, Expressions, Manifestations and Items, it is clear that the FRBR Work is the only one that may change during time, from the first draft to the final published version or subsequently corrected version, since the individual Expression at each stage is a static document that itself does not change, with every revision of the Work resulting in a new Expression. For this reason, the statement defining the date on which a correction is made to the content of the document applies specifically to the Work that underlies the particular document bearing the JATS markup (in our mapping, the entity :conceptual-work).

The XML:

```
<pub-date date-type="corrected">
  <day>DD</day>
  <month>MM</month>
  <year>YYYY</year>
</pub-date>
```

thus maps to

```
:conceptual-work 
  dcterms:hasCorrectionDate 
  "YYYY-MM-DD"^^xsd:date ;
  frbr:realization [ a fabio:Expression ;
  frbr:revision [ a fabio:Expression ;
```

This also applies to the mapping of `@date-type` with value "ecorrected" and "pcorrected".
@date-type ("retracted")

In contrast, retractions apply to published Expressions (here :textual-entity), or to a Manifestation of such an Expression, since one cannot retract a Work. Hence:

```xml
<pub-date date-type="retracted">
  <day>DD</day>
  <month>MM</month>
  <year>YYYY</year>
</pub-date>
```

is simply mapped as:

```xml
:textual-entity
  fabio:hasRetractionDate "YYYY-MM-DD"^^xsd:date .
```

<publisher-loc> and <address>

The <i>publisher-loc</i> element (<i>publisher-loc</i>) is used to define the place of publication of the entity being described ("Place of publication, usually a city, such as New York or London."). and forms part of <i>journal-meta</i> or of a bibliographic reference defined by <i>element-citation</i> or of <i>mixed-citation</i>.

A primary reason for encoding metadata in RDF is to permit integration of such information from different sources without loss of meaning. While one could simply define the location using the data property <i>prism:location</i>, this could lead to loss of contextual information when combining such RDF statements relating to different publications by the same international publisher. Thus, if the JATS RDF mapping from one journal simply said:

```xml
:this-publisher prism:location "London" .
```

while the JATS RDF mapping from a second article said

```xml
:this-publisher prism:location "Boston" .
```

and these statements were then combined into a single RDF graph

```xml
:this-publisher prism:location "London" , "Boston" .
```

the location information would be correct (i.e. that the publisher has at least two locations, London and Boston), but the context relating each particular location to a specific journal would be lost.

To avoid this ambiguity, we instead employ a standard ontology design pattern called the Time-indexed Value in Context Pattern (TVC; http://www.essepuntato.it/2012/04/tvc) (7). This design pattern, which involves one level of indirection between the object and the location, via the class <i>tvc:ValueInTime</i>, has the advantages both of enabling contextual information to be easily recorded (in this case, to which journal each publication location relates) and also of permitting any temporal constraints accompanying such relationships to be specified (a facility that we have not needed to use in the JATS2RDF mappings), as shown in Figure 1.

Our use of TVC permits us to state that a publisher claims the publication location "London" for one journal and the publication location "Boston" for another. Thus for the JATS markup:

```xml
<publisher>
  ...
  <publisher-loc>XXX</publisher-loc>
</publisher>
```

we create the RDF mapping:

```xml
:this-publisher tvc:hasValueInTime [  
  a tvc:ValueInTime ; tvc:withValue [  
    a vcard:VCard ;  
    vcard:addr [ a vcard:Address ; 
      vcard:locality "XXX" ] ] ;  
```

For the same reasons, we use TVC when mapping the address of a contributor, who, because of moving between institutions, may have one address recorded for one publication and another address for another:

```xml
<contrib>
  ...
  <address> ... </address>
  ...
</contrib>
```

being mapped to:

```xml
:this-agent tvc:hasValueInTime [  
  a tvc:ValueInTime ;
```
tvc:withValue :this-agent-contact-info ;

:this-agent-contact-info a vcard:VCard .

Our use of VCard permits great flexibility in the specification of location, addresses, emails, phone numbers, etc., and is therefore a mapping choice that we have used not only for <publisher-loc> and <address>, but also for other entities requiring items of contact information, for example <fax> and <phone>, thus:

<address>
  <fax>XXX</fax>
</address>

is mapped to:

:this-agent-contact-info vcard:tel [ 
  a vcard:Fax ;
  literal:hasLiteralValue "XXX" ] .

People and roles

<contrib>

In JATS <contrib> is specified as a "container element for information about a single author, editor, or other contributor." However, <collab>, referring to a group of people, is permitted within <contrib>, as well as within <contrib-group>, enabling, for example, an organization or a consortium of people to be named as a single author in an article's author list. So, rather than use foaf:Person as the object of the mapping here, we make a more generic mapping to foaf:Agent, which has sub-classes foaf:Person, foaf:Group and foaf:Organization. Thus the JATS XML:

<article>
  ...
  <contrib>...</contrib>
  ...
</article>

is mapped as:

:conceptual-work dcterms:contributor :this-agent .

:this-agent a foaf:Agent .

<person-group>

In contrast to <contrib>, which specifies information about a person or group of people directly, JATS defines the element <person-group>, as it is used within <element-citation> and <mixed-citation>, to be a "container element for [the names of] one or more contributors in a bibliographic reference", i.e. the list of authors within the bibliographic reference. The most direct way to map this element is to define, using PRO (7), a group of contributors (i.e. a foaf:Group) having the role of contributor, specifying the string of their individual names as the group name, and asserting that their contribution role relates to the particular bibliographic entity (i.e. :textual-entity-XXX) that is the object of the reference in question (i.e. :ref-XXX2) :

:ref-XXX biro:references :textual-entity-XXX .

:person-group a foaf:Group ;
  foaf:name "YYY" ;
  pro:holdsRoleInTime [ a pro:RoleInTime ;
    pro:withRole pro:contributor ;

Publishing roles, as in <role>, @contrib-type and @collab-type

It is quite common in the publishing world for people to have different roles in the context of different publications. For example, someone may be editor of a scholarly journal, and hence of the articles within it, while at the same time being author of an article published in another journal. To permit accurate description of such situations, we again employ the TVC design pattern within PRO, the Publishing Roles Ontology (which imports TVC), to permit the role of a person to be defined within the context of a particular publication, and when necessary within a particular time period (a facility that we have not needed to use in the JATS2RDF mappings). This is made possible by invoking one level of indirection between the person and that person's role via the class pro:holdsRoleInTime (7).

We use PRO at several points in the JATS2RDF mapping (as also shown in the previous section) to permit the role of a person or a group to be specified, for example when specifying the elements <role>, <copyright-
holder>, <principal-investigator>, <on-behalf-of>, <author-comment> and <aff>, and when defining roles relating to the attributes @contrib-type, @collab-type, @corresp and @person-group-type.

Since, in the FRBR context, authorship and similar roles relate to Works, while editorship, translators and similar roles relate to Expressions (i.e. submitted manuscripts), we have mapped roles to their appropriate contexts, using :conceptual-work or :textual-entity as appropriate. Two examples will serve to illustrate these common usages:

```xml
<contrib>
  ...
  <role>editor-in-chief</role>
  ...
</contrib>
```

is mapped to:

```xml
:this-agent pro:holdsRoleInTime [ 
  pro:withRole pro:editor-in-chief ; 
```

while:

```xml
<contrib contrib-type="author">
  ...
</contrib>
```

is mapped as:

```xml
:conceptual-work dcterms:creator :this-agent .
```

```xml
:this-agent pro:holdsRoleInTime [ 
  pro:withRole pro:author ; 
```

One advantage of this design pattern is that new roles can be easily added as individual members of the class pro:PublishingRole, without having to change the structure of the ontology.

**Other personal roles, as in @collab-type and <principal-award-recipient>**

SCoRO, the Scholarly Roles Ontology, imports PRO, and thus indirectly imports TVC. This permits us to use the same TVC ontology design pattern to specify other roles relating to scholarly research activity that are not specifically publishing roles. SCoRO contains seven other role classes in addition to the imported pro:PublishingRole, for example, scoro:InvestigationRole that contains 15 members including scoro:inventor.

This enables us to map roles that are not strictly publishing roles:

```xml
<collab collab-type="inventors">
  ...
</collab>
```

being mapped as shown:

```xml
:this-agent pro:holdsRoleInTime [ 
  pro:withRole scoro:inventor ; 
```

Similarly, for <principal-award-recipient>, we map:

```xml
<award-group>
  <principal-award-recipient>
    XXX
  </principal-award-recipient>
</award-group>
```

to:

```xml
:funding-recipient-agent a foaf:Agent ; 
foaf:name "XXX" ; 
pro:holdsRoleInTime [ 
  pro:withRole scoro:funding-recipient ; 
  pro:relatesToEntity :project ] .
```

@equal-contrib ("yes" or "no")

SCoRO also permits one to record that a person, a group or an organization has made a contribution to a particular endeavour such as a research article, again using the TVC pattern and the indirection class scoro:ContributionSituation. It is thus possible, using the new object property scoro:isEqualToContributionSituation added to the ontology for this purpose, to assert that the contribution
situation involving one author is equal to the contribution situation involving another author, by asserting for each of them:

<contrib equal-contrib="yes"> <!-- For author 1 -->
  ...
</contrib>

<contrib equal-contrib="yes"> <!-- For author 2 -->
  ...
</contrib>

which is mapped as:

:this-agent-1 scoro:makesContribution [  
a scoro:ContributionSituation ;
  scoro:hasContributionContext :conceptual-work ;
  scoro:isEqualToContributionSituation _:contribution-2 ] .

:_contribution-2 a scoro:ContributionSituation ;
  scoro:hasContributionContext :conceptual-work .

:this-agent-2 scoro:makesContribution _:contribution2 .

Here equality of contribution is expressed between all persons involved in contribution situations for whom the property scoro:isEqualToContributionSituation is asserted.

The alternative possibility, that the contribution is not equal, is mapped simply by asserting that a contribution is made, without asserting its equality to that of another's. Thus:

<contrib equal-contrib="no">
  ...
</contrib>

is mapped as:

:this-agent scoro:makesContribution [  
a scoro:ContributionSituation ;
  scoro:hasContributionContext :conceptual-work ] .

Using XSLT to automate the conversion from JATS XML to RDF

We developed an XSLT 2.0 transform to produce RDF descriptions from JATS documents according to the mappings specified in the JATS2RDF mapping document. The transform specifies the conversion strategy from JATS XML elements and attributes into RDF statements.

The entire process is based on a (recursive) named template called "assert" that takes a sequence of strings (s,p,o, t, l, p, o, z, p, o, z, p, o, z, ...) as input (where s = subject, p = predicate, o = object, and where the object can have a datatype d or a language l when appropriate), each representing either a resource IRI or a literal, and returns one or more RDF statements. This allows us to make RDF statements simply by parsing the list of strings. Table 1 shows some calls to this template on the left column and the resulting RDF translation in Turtle format in the right column.

Table 1 Pseudo-codes describing the calling to the template "assert" (left column) and the resulting RDF statements created (right column).

<table>
<thead>
<tr>
<th>Call to template &quot;assert&quot;</th>
<th>Resulting RDF statements (in Turtle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>assert(s,p,o)</td>
<td>s p o .</td>
</tr>
<tr>
<td>E.g.: assert(</td>
<td>E.g.: :textual-entity a fabio:Expression .</td>
</tr>
<tr>
<td>:textual-entity,a,fabio:Expression)</td>
<td></td>
</tr>
<tr>
<td>assert(s,p,&quot;o&quot;^^d)</td>
<td>s p &quot;o&quot;^^d .</td>
</tr>
<tr>
<td>E.g.: assert(</td>
<td>E.g.: :textual-entity dcterms:date</td>
</tr>
<tr>
<td>:textual-entity,dcterms:date,</td>
<td>&quot;2012-05-12&quot;^^xsd:date .</td>
</tr>
<tr>
<td>&quot;2012-05-12&quot;^^xsd:date)</td>
<td></td>
</tr>
<tr>
<td>assert(s,p,&quot;o&quot;@l)</td>
<td>s p &quot;o&quot;@l .</td>
</tr>
<tr>
<td>E.g.:</td>
<td>E.g.:</td>
</tr>
<tr>
<td>E.g.:</td>
<td>assert(s,p1,o1,p2,o2)</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
</tr>
<tr>
<td>assert()</td>
<td>:textual-entity,a,fabio:Expression, frbr:realizationOf,:conceptual-work)</td>
</tr>
<tr>
<td>E.g.:</td>
<td>assert(s,p1,,p2,o)</td>
</tr>
<tr>
<td>assert()</td>
<td>:textual-entity,cito:discusses,, a,fabio:Article)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E.g.:</th>
<th>assert(s,p1, p2,o2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>assert()</td>
<td>:textual-entity a fabio:Expression; frbr:realizationOf ;conceptual-work .</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E.g.:</th>
<th>assert(s,p1 [ p2 o ])</th>
</tr>
</thead>
</table>

Each JATS element/attribute in the mapping is caught by a specific XSLT template, which in turn calls appropriately the template “assert” so as to produce the related RDF statements as specified in the mapping document.

We decided to develop the XSLT transform in this way so as to keep it easily extensible in the future, when more JATS markup will be handled in the mapping (by adding new templates) or when the JATS tag suite will be modified (by modifying existing templates).

The main difficulties addressed when developing the XSLT transform relate to the ambiguities of tags, with particular consideration to the context in which they are found and the entities to which they refer. The following list introduce the main problems addressed during the implementation phase:

- **Contextual dependency.** A particular tag may describe metadata of certain entities rather than others, according to the context in which it appears in the XML (i.e. the other tags that contain it). For instance, the tag `<date>` can mean either publication date when it is contained by the tag `<element-citation>` or revision date when it is contained by the tag `<history>`. This issue is fully and correctly handled in the XSLT.

- **Incorrect context.** This is the case for all those tags that the JATS spec permits to be contained "inappropriately" by another tag. For instance, the tag `<isbn>` may be contained within the tag `<journal-meta>`, along with other tags related to the journal in which the article under consideration was published. However, journals themselves do not possess International Standard Book Numbers. ISBNs can, rarely, be assigned either to a journal `<issue>` or to a journal `<volume>` (both FRBR Manifestations). However it is impossible to understand from the context (being enclosed within the `<journal-meta>` tag) to which kinds of bibliographic entity the ISBN relates.

The solution we implemented in the XSLT transform for the `<isbn>` tag is enclosed within the `<journal-meta>` tag is as follows:

- The tag `<issue>` is preferred to the tag `<volume>` as the reference of `<isbn>`, if both are present;
- If only one is present, that is used as the reference;
- However, when both `<issue>` and `<volume>` tags are absent, the XSLT creates a new anonymous manifestation having the ISBN identifier specified in the `<isbn>` tag specified, with the article under consideration declared to be a part of this.

- **No contextual data.** Some markup items can have multiple and ambiguous interpretation in the JATS spec, and there is no clue as to the correct or best interpretation. For instance, the attribute @journal-id-type can be used to specify a particular identifier for a journal according to a particular identifier scheme or to specify a local identifier of the journal attributed to its publisher. In the current specification, there is no way to understand when to use the former interpretation and when to use the latter, when the value of the attribute given is not recognisable as belonging to one of the known identifier schemes. The solution we implemented in the XSLT transform for @journal-id-type is as follows: when its value is not recognised, it will be defined as an identifier of an unknown identification scheme.

- **Structured vs. unstructured tags.** Some tags can appear either in very structured contexts that define the sources to which the metadata described by those tags apply, or alternatively in quite ambiguous contexts that require further information from the creator of those metadata. For instance, the tag `<name>` can appear either within the tag `<contrib>` to indicate the name of a particular contributor of the article under consideration, or as a direct child of the tag `<element-citation>` or `<mixed-citation>`, in which context the name might be that of an author of the cited work, or of an editor of that work, who
Conclusion: What the JATS2RDF mapping makes possible

Nowadays, publishers are exploring new ways of making available and sharing their bibliographic data, such as semantic publishing (10). In brief, semantic publishing is the use of Web and Semantic Web technologies to enhance the meaning of a published document, to define its metadata and to publish them as Open Linked Data. This approach seems to be a concrete aspect of the current business model of modern publishers - inasmuch that recently both Nature Publishing Group and the American Association for the Advancement of Science have agreed to open their articles’ reference lists and to publish them as Open Linked Data **,** **,**.

By choosing to map JATS to RDF, from among all possible XML DTDs, we hope to promote the use of JATS to a wider community. The additional functionality provided by this mapping will permit JATS metadata to become part of the web of linked data. It will, for example, permit the ingest into the Oxford DataBank, a semantically aware institutional data repository, of bibliographic metadata encoded in RDF that describe JATS-encoded journal articles related to the research datasets stored within the Oxford DataBank.

In the future, we plan to increase the set of JATS elements described in our mapping document (e.g. including the document structural elements sec, fig, table), so as to address the structural content of the document and its rhetorical organization.

Supplementary material

The current version of the JATS To SPAR XSLT Transform.
Download file (148K)

The RDF conversion of our paper obtained through the XSLT Transform we developed.
Download file (65K)

The PDF of the mapping document.
Download PDF (587K)

The Microsoft Word file of the mapping document.
Download MS Word (238K)

Acknowledgments

The mapping work undertaken by S. Peroni, D. Shotton, and D. A. Lapeyre that is described in this paper was made possible by the provision of financial support from the JISC (Joint Information Systems Committee) in the form of a grant to DS.

Footnotes

* The RDF statements in this paper are given in Turtle notation (http://www.w3.org/TeamSubmission/turtle). The symbol "*" in "bibliographic-entity", and in similar expressions used in this paper, is an abbreviation for the fictional namespace http://www.example.org/resource/ used to describe these generic resources.

** An alternative mapping could be to interpret the element <person-group> so as to represent a text string of personal names that is a textual component part of a bibliographic reference. For this reason, the mapping of <person-group> to RDF can be given in two parts: first to identify the text string of names :person-group-text within the reference, and then to assert that this text string of names :person-group-text denotes the group of real people .person-group, thus applying mechanisms introduced in 5:

```
<element-citation>
...
<person-group>YYY</person-group>
...
</element-citation>
```

is mapped as follows (where :ref-XXX is a bibliographic reference in a reference list):

```
:person-group-text literal:hasLiteralValue "YYY" ;
  lmm:denotes :person-group .
:person-group a foaf:Group .
```

However, we think what we actually proposed in the mapping document (while less precise) is easier and more comprehensive than the aforementioned one, at least in this stage.

*** http://opencitations.wordpress.com/2012/06/11/nature-to-open-its-reference-list-data/

Fig. 1
A diagrammatic description of the Time-indexed Value in Context Pattern

The authors assert themselves as the copyright holders of this paper.

The copyright holders grant the U.S. National Library of Medicine permission to archive and post a copy of this paper on the Journal Article Tag Suite Conference proceedings website.

The copyright holders license this paper for open access under a Creative Commons Attribution 3.0 Unported License (CC BY 3.0).