Semantic TV

— is the semantic web a part of broadcasting's future?

Jean-Pierre Evain
European Broadcasting Union (EBU), Switzerland

The internet is part of broadcasting's future and might soon be the ultimate opportunity to reach viewers directly. 'Semantic web' and 'ontologies' are not new concepts but they start making sense as a step beyond metadata for providing more efficient search, retrieval and access to 'semantic TV'.

This article is an overview of lessons learnt from developing an ontology for audio-visual programmes and services. The implementation of semantic-web technologies will require a new vision from content providers, including broadcasters. New collaborations will need to be established and, clearly, ambitions should go beyond the use of the narrow-scoped DVB Service Information (DVB-SI).

Introduction

The internet is here and there is no other option but to fully exploit it and its wealth of technologies. The 'semantic web' is one of these internet-specific innovations.

'Ontologies' are one of the essential parts of the 'semantic web'. The use of the word 'ontology' may soon become as normal as 'metadata'. This article briefly introduces *"what an ontology is"* and, based on experience, proposes an implementation for audiovisual programmes and services.

The quantity of information needed to search and retrieve content will continue to grow with the offer. But new models and processes will be required to efficiently parse such a large amount of data. Semantic web is just another step in this evolution ... clearly, a paradigm shift from "DVB-SI present/following".

Semantic web in a nutshell

The semantic web is an extension of the worldwide web (WWW) which combines the web and metadata. But it is actually more than the 'usual' metadata that is being considered here.

The ambition is to shift from a web of static descriptions – exchanged between applications or 'crawled' by search engines and indexed per keywords – to a dynamic connected web of semantically-defined objects (e.g. 'this programme *is provided by* this service'). Machine-interpretable semantic metadata should allow reasoning (more than simple indexing) for faster and more efficient queries, providing more relevant and accurate results. This explains why the three pillars of the semantic web are 'metadata', 'ontologies' plus 'inference and reasoning tools'.

Ontologies: metadata with semantics

What is an ontology?

No doubt that the word ‘ontology’ will sooner or later become as frequently used as metadata. The general definition of ontology is *“the metaphysical study of the nature of being and existence”*. Of course, the intention is not to study the ‘nature of being and existence’ of broadcasting or other forms of content delivery but, rather, the description of its components (e.g. service, schedule, content) following an ontological approach.

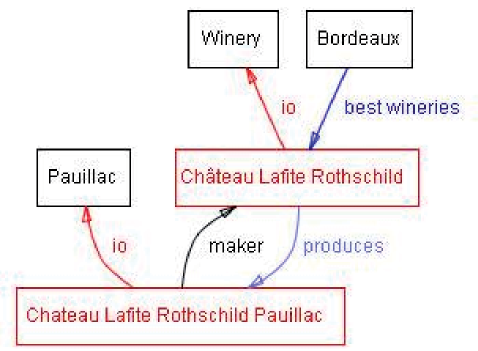

Figure 1
Wine ontology

Ontologies can be very efficiently used to describe and hierarchically classify all the classes or individuals of a category or domain. A typical example is the ‘wine’ ontology from Noy & McGuinness, Stanford University [1], sorting out all associations and restrictions (e.g. “a white wine cannot be associated with a *pinot noir* grape”).

However, ontologies can also be used in a schema-like approach in which semantic links are being defined between abstract classes independently of a particular description instance.

How does it differ from usual metadata?

One option to explain the differences is to look at the different languages currently used to represent these different data structures[1].

The first of these languages is XML (eXtensible Markup Language). This is the most commonly used structured language, allowing metadata designers to define their own tags and custom properties. It also comprises a set of strictly-defined data types to restrict the use of certain values associated with such predefined properties. XML benefits from the existence of many tools to edit, parse and validate the metadata schemas and instances.

Fig. 2 represents a typical example of XML description, which will be seen as a particular instance of the properties of the corresponding web resource. This is the approach followed by most Electronic Programme Guides (EPGs) today. Data is exchanged between applications before being processed. Dedicated servers can be queried, exploiting advanced search criteria.

```
<?xml version="1.0" encoding="UTF-8"?>
<TVProgrammeDescription>
    <Title>Tonight’s Show</Title>
    <Duration timeUnit='minute'>90</Duration>
</TVProgrammeDescription>
```

Figure 2
XML instance snippet

Adding semantic requires the use of new languages like RDF (Resource Description Framework) and OWL (Ontology Web Language). The novelty of these languages consists of replacing a static ‘tree’ of XML properties by a graph-like inter-related series of statements, which can be represented as a natural expression (a so-called *triple*), constituted of a subject, predicate/verb and object. Even more revolutionary, each of these statements and constitutive elements of information become a web resource like the programme itself.

The statement “a TV Programme has for its title *Tonight’s Show*” could be expressed in RDF/OWL as shown in *Fig. 3*.

1. For the purpose of this demonstration, the fact that XML is used to serialise RDF and OWL is left aside. Some escaped characters are also ignored for better readability.

Ontology:

```
<owl:DatatypeProperty rdf:about="&Programme_owl;hasTitle">
    <rdfs:domain rdf:resource="&Programme_owl;Programme"/>
    <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
```

Instance:

```
<rdf:Description rdf:about="#Programme">
    <Programme_owl:hasTitle rdf:datatype="&xsd;string">Tonight's Show
    </Programme_owl:hasTitle>
</rdf:Description>
```

Figure 3
Example of RDF/OWL statement

In this example, 'Duration' and 'timeUnit' would have their own statements respectively *about* #Programme and #Duration (identified by their URI - Unique Resource Identifier).

The difference between this semantic description and the original XML representation is that the predicates/verbs, plus additional semantic properties (like symmetry), are machine-interpretable and can be used by software agents to establish dynamic relationships between resources. *Interpretation results* can be used by applications. Furthermore, statements *about* a resource can be cumulated from different sources if the resource is identified by the same URI. Machines capable of such processing are called 'reasoners'. One of the key questions for the future is to know to what extent such technology will be implemented, e.g. in search engines.

One should also not expect to be able to transform easily an XML schema into an ontology. Some transformation methods have been proposed but it is recommended to re-think the model from an ontology point of view, in particular if the schema makes extensive use of attributes and derived types. Furthermore, the hierarchy of properties in XML trees has nothing in common with a hierarchy of ontology classes.

Why use ontologies?

The main motivation is to provide access to richer information and to augment the probabilities for well-documented content to be found faster and with more relevance to users. More content will require more metadata.

As mentioned above, ontology-based descriptions are machine-interpretable and can be used by software agents to establish dynamic relationships between resources.

It is already foreseen that more complex models will emerge in the future as the quantity of information grows exponentially.

What are the main components of an ontology?

The main components of ontologies are 'classes' (e.g. Programme in *Fig. 4*) and associated 'properties', which can themselves be subdivided into two groups of 'object properties' (e.g. HasSegment in *Fig. 4*) and 'data properties' (e.g. HasTitle in *Fig. 4*). These properties represent the semantic part of an ontology and require the use of new languages such as RDF (Resource Description Framework) and OWL (Ontology Web Language).

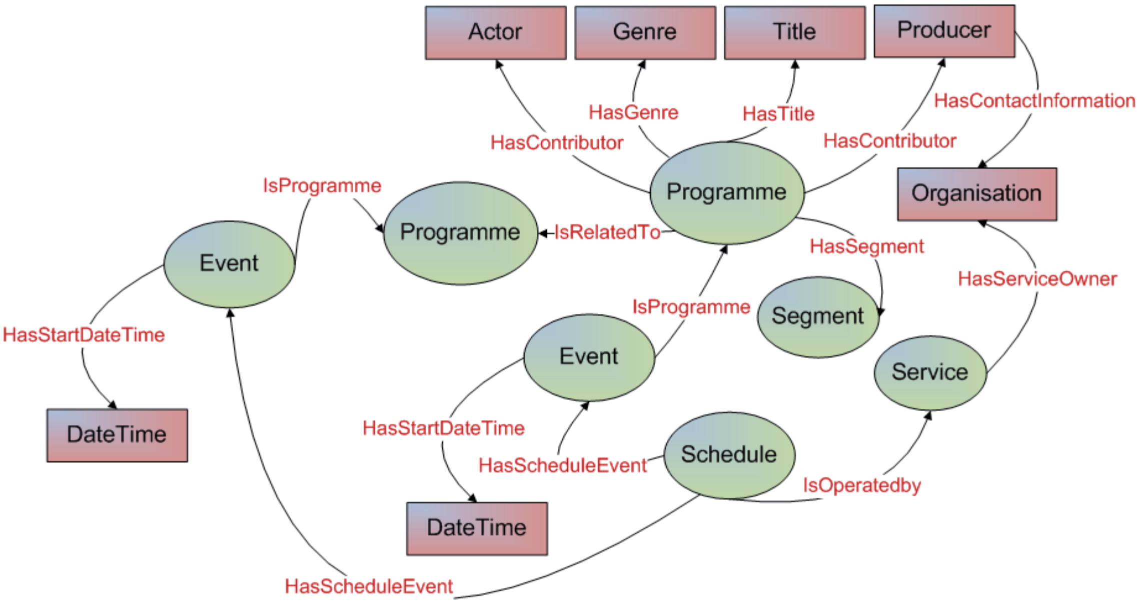

Figure 4
Simplified view of the AV ontology

As mentioned above, the novelty of these languages consists of replacing a static 'tree' of XML properties by a graph-like inter-related series of statements, which can be represented as a natural expression (a so-called *triple*), constituted of a subject, predicate/verb and object.

A 'class' groups together individuals with similar features. It is easy to understand why 'Programme' or 'Service' would represent natural classes of an audiovisual ontology. But the identification of an eligible class and its role in the ontology is not always that simple. As an example, how would this apply to 'segments'?

It would be ontologically valid to declare a 'Segment' as a subclass of 'Programme', as a segment taken individually could be seen as a 'Programme'.

However, it is preferable to define a separate class 'Segment'. Some of the advantages to this approach are:

- The resulting ontology is simpler, which facilitates the generation of instances;
- The class 'Segment' is no longer bound to the class 'Programme' and can be reused in other contexts;
- The very strict relation of a sub-class to a class is replaced by an unlimited set of possible user-defined properties linking a 'Segment' to e.g. a 'Programme'. For example, a 'Segment' can be *part of* a 'Programme'. Similarly, a 'Programme' can be an *aggregation of* 'Segments', etc.;
- It is easier to define data properties intrinsic to each class;
- Defining autonomous classes simplifies the definition of relations between similar classes such as declaring a 'Segment' used in TV-Anytime or MPEG-7 as being *equivalent to* a 'Part' used in NewsML-G2.

There is another type of class that the EBU uses in its ontology, which is a cluster of properties (comparable in functionality to XML complex types). An example is 'Audio and Video Attributes' reused in different contexts such as in association with 'Programme', 'Segment' or 'Part'. In this case, it makes sense to define the 'Audio Attributes' and 'Video Attributes' as sub-classes of the 'Audio and Video Attributes' class.

It could be feared that this adds complexity and requires us to instantiate each class with appropriate identifiers. However, by using Fact++ and Pellet reasoners, it allows us to validate the direct use of sub-properties as shown in *Fig. 5*.

```
Option 1

<EBU_AVAttributes_owl:AVAttributes rdf:about="#AV_Attributes_001">
  <EBU_AVAttributes_owl:hasAudioAttributesCodingName>AAC</EBU_AVAttributes_owl:hasAudioAttributesCodingName>
  <EBU_AVAttributes_owl:hasVideoAttributesCodingName>H264</EBU_AVAttributes_owl:hasVideoAttributesCodingName>
</EBU_AVAttributes_owl:AVAttributes>

<EBU_Programme_owl:Programme rdf:about="#Programme_Class_001">
  <hasProgrammeID>Production_ID_123</hasProgrammeID>
  <EBU_Programme_owl:hasProgrammeAVAttributes rdf:resource="#AV_Attributes_001"/>
  <EBU_Programme_owl:hasProgrammeTitle rdf:resource="#Tonight_Show"/>
</EBU_Programme_owl:Programme>

<EBU_Programme_owl:ProgrammeTitle rdf:about="#Tonight_Show">
  <EBU_Programme_owl:hasProgrammeTitleText>Tonight Show</EBU_Programme_owl:hasProgrammeTitleText>
</EBU_Programme_owl:ProgrammeTitle>

Option 2

<EBU_Programme_owl:Programme rdf:about="#Programme_Class_001">
  <hasProgrammeID>Production_ID_123</hasProgrammeID>
  <EBU_Programme_owl:hasProgrammeTitleText>Tonight Show</EBU_Programme_owl:hasProgrammeTitleText>
  <EBU_AVAttributes_owl:hasVideoAttributesCodingName>H264</EBU_AVAttributes_owl:hasVideoAttributesCodingName>
  <EBU_AVAttributes_owl:hasAudioAttributesCodingName>AAC</EBU_AVAttributes_owl:hasAudioAttributesCodingName>
</EBU_Programme_owl:Programme>

Option 3

<EBU_Programme_owl:Programme rdf:about="#Programme_Class_001">
  <hasProgrammeID>Production_ID_123</hasProgrammeID>
  <EBU_Programme_owl:hasProgrammeTitle rdf:resource="#Tonight_Show"/>
  <EBU_AVAttributes_owl:hasVideoAttributesCodingName>H264</EBU_AVAttributes_owl:hasVideoAttributesCodingName>
  <EBU_AVAttributes_owl:hasAudioAttributesCodingName>AAC</EBU_AVAttributes_owl:hasAudioAttributesCodingName>
</EBU_Programme_owl:Programme>
```

Figure 5
Example implementation options

Why would Option 3 be better than Option 2? Option 3 should be preferred as it abstracts audio and video attributes as data properties but also allows declaring the programme title as a web resource. An example (dummy url) is shown here:

http://www.ebu.ch/metadata/ontologies/EBU/AV/EBU_Programme_owl#Tonight_Show

RDF and OWL support (to a lesser extent than XML) key features such as cardinality, types (e.g. string, integer, anyURI) and enumerated lists. Choices can also be defined through complex conditional property expressions. But these properties are difficult to handle with most popular (affordable) editing tools. Even if the ontology validates itself, errors may reveal themselves when generating description instances. Only the systematic launch of a reasoner will help and prevent facing impossible debugging sessions.

The difficulty of writing an ontology is further complicated by the temptation to benefit from some additional RDF/OWL properties. The use of *disjoint classes*, *functional and symmetric properties* must be handled with great care. Only a well-structured ontology combined with experience will help, as error messages are very succinct.

XML, RDF and OWL — exclusive or complementary?

There do not seem to be clear-cut views on the best approach to implement such ontologies. What follows is an opinion of a possible implementation framework.

To gain full benefit from the advantages of XML schema, it is proposed that the model and instances to be transformed are validated into a less constrained ontology template with e.g. a cardinality set

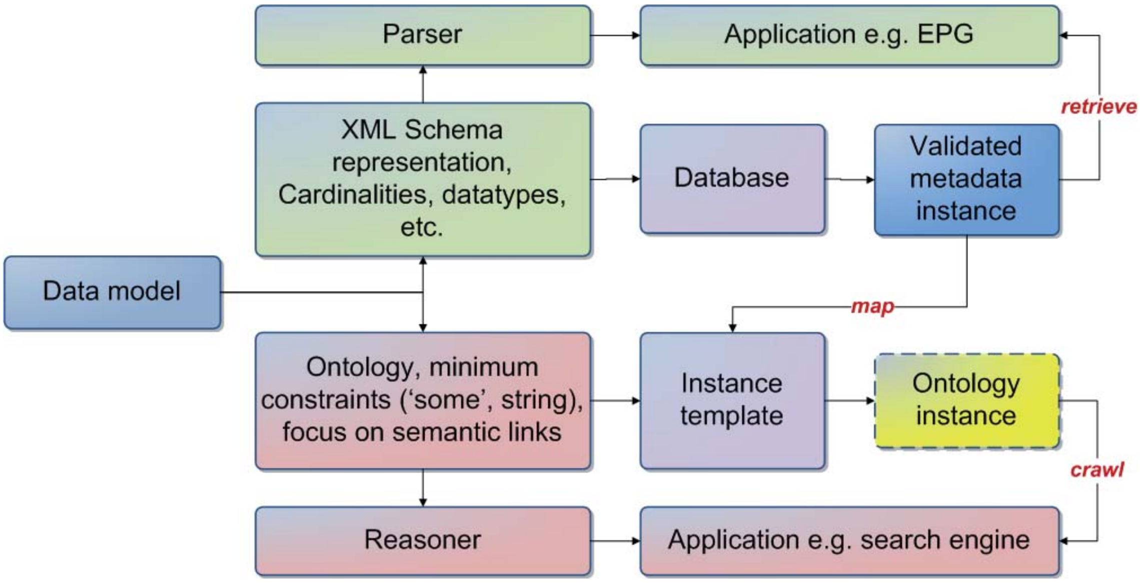

Figure 6
Proposal for an implementation framework

to 'some' and datatype to 'string' by default.

The use of a template should be attractive to users as it hides the complexity of the ontology, which should also allow user-generated descriptions to play a role (e.g. fans of a series helping their community to find content). However, simplifying the templates has an impact on the ontology in which, for instance, *choices* should be avoided.

Generating instance templates for complex ontologies, such as for audio-visual services, is a challenge. Tools like *Protégé*, an advanced free-to-use ontology editor developed by the University of Stanford, do not (yet?) provide a fully-featured automatic instance generator. This is a barrier to implementation.

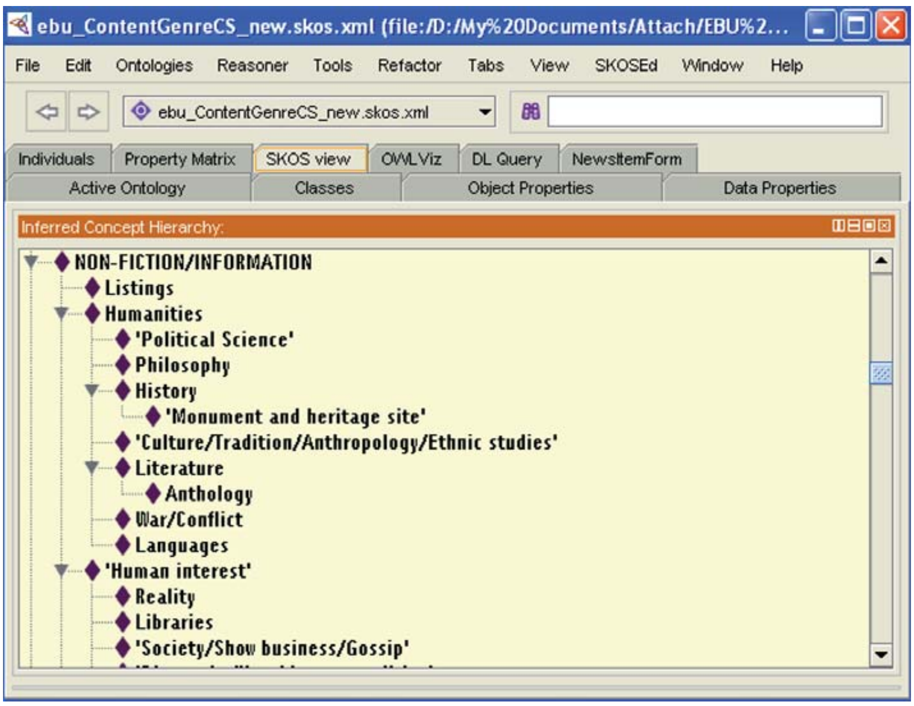

Figure 7
Screenshot of a SKOS view of EBU Content Genre CS using Protégé

SKOS — 'reasoning' applied to thesauruses

SKOS is an ontology that is very convenient for representing classification schemes with object properties like *broader* or *narrower* term, *exactMatch*, *narrowMatch* or *broadMatch*.

An example is given here:

http://www.ebu.ch/metadata/ontologies/skos/ebu_ContentGenreCS.skos.xml

As shown above, each term of a classification scheme (or thesaurus) is independently subject to a series of statements and is no longer part of a hierarchical XML structure such as used by MPEG-7, TV-Anytime or DVB. Nevertheless, the hierarchical structure can be reconstituted by reasoners as shown in *Fig. 5*, in addition to which we get machine-interpretable statements about mapping to other external classification schemes.

We can readily imagine what can be achieved from a series of statements about a programme, its main characteristics and its relations to e.g. a schedule and a service.

Two ontologies on the brink

In order to study the potential value of Semantic Web, EBU TECHNICAL has developed two draft ontologies for:

- NewsML-G2: http://www.ebu.ch/metadata/ontologies/NML2/ and
- audiovisual services:http://www.ebu.ch/metadata/ontologies/AV/.

The use of these resources is governed by Creative Commons usage terms.

The NewsML ontology shown in *Fig. 8* has a clear tree structure. A news item is composed of parts, and both have a certain number of properties. From this perspective, the benefit over traditional structured metadata is questionable.

However, from an ontology perspective, it is important to note that 'Part' (or shot) is a sub-class of 'News Item', which means that taken separately, a 'Part' can be considered fully as a 'NewsItem'. Furthermore, having a news item represented as an ontology will allow us to merge this ontology with the AV ontology shown in *Fig. 2*. What if a news item or part were to become a programme, or if a news item part were to be compared to a programme segment in future?

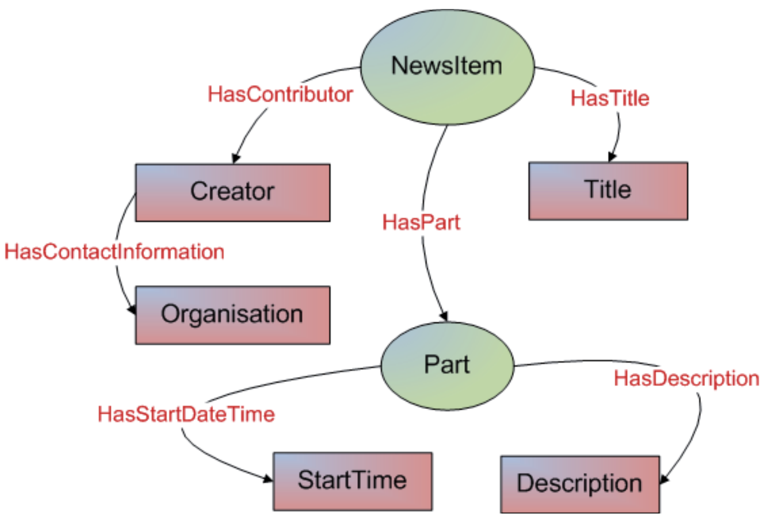

Figure 8
Simplified view of a possible NewsML ontology

The simplified view of an AV programme and service ontology presented in *Fig. 4* shows a more complex relation graph as would be expected. As a reminder, such relational graphs are derived automatically by 'reasoners' from statements about programmes, services, schedules, etc. It is important to note the attempt to harmonise the name of the properties with those used in the news ontology. An expected use is to allow users to search for any of the exposed properties (e.g. a title, an actor, a time slot, a service) and obtain targeted 'hits' like alternative time slots for a repeat of the programme or a similar programme (according to what the relation is between programmes), etc.

There seem to be many possibilities.

The tools

For those readers who would like to get more involved in Semantic Web developments, please use the resources introduced above. Useful information on editing and validating tools is provided on one of the metadata pages of EBU Technical's website (**http://tech.ebu.ch/semanticweb_ebu**).

Conclusions

This article has provided an introduction to the use of Semantic Web for TV programmes and services, which is seen as a unique opportunity for content providers to keep directly in touch with users who have Internet access (broadband, mobile, etc.).

Proposals for collaboration on the improvement and implementation of these ontologies, e.g. in search engines, are welcome. This is just the beginning ... watch this space!

References

[1] Ontology Development 101: **A Guide to Creating Your First Ontology**
Natalya F. Noy and Deborah L. McGuinness, Stanford University, Stanford, CA, 94305
http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.

[2] The Semantic Web wiki, with links to resources and recent news (**http://semanticweb.org/**).

[3] The Semantic Web page of W3C (**http://www.w3.org/2001/sw/**).

[4] Jean-Pierre Evain: **Is Semantic Web Part of the Television Future**
IBC-09, EBU.