The Annotation Interconnected Strata model: a promising way of using RDF for the Semantic Web

Pierre-Antoine Champin and Yannick Prié

LISI, Université Claude Bernard Lyon1 & INSA de Lyon
43 boulevard du 11 novembre 1918
F-69100 Villeurbanne, France
{champin,yprie}@lisi.univ-lyon1.fr

Abstract Since the expansion of the World Wide Web makes it harder and harder to cope with for users, the content of the Web must become more machine understandable, so that automated agents can provide them an efficient assistance. As RDF is becoming the common language to achieve that, we propose to map the AI-Strata model (initially designed for audiovisual document annotation) to RDF. The aim of this approach is to use the flexible AI-Strata model to describe Web resources, and to use efficient AI-Strata algorithms in the Semantic Web context.

1 Introduction

As a huge information repository, the World Wide Web becomes harder and harder to cope with for users alone. This has reached a point where even the response of a search engine to a request contains an overwhelming amount of data, irrelevant for the most of it. The problem is due to the fact that the Web content is designed by humans for humans; it is all the more poorly structured than the Hypertext Mark-up Language (HTML) has been used in order to make documents well rendered, or inter-browser compatible (or incompatible!), rather than logically structured.

In such an environment, only a limited indexing, based on plain keywords, can be performed by automated agents, but as we pointed out, this is no more a satisfying solution. Until Artificial Intelligence provides human-like behaviored agents, the content of the Web must become more machine-understandable so that users can be automatically assisted in a more efficient way. This is what is now well known as the *Semantic Web* [2].

The first step has been the Extensible Mark-up Language (XML, [5]). XML is in fact a meta-language, which allows to define new vocabulary and new structures by means of Document Type Definitions (DTD) and XML Schemas [19]. That makes it suitable for current as well as future Web applications. Furthermore, a stylesheet mechanism makes it possible to write structured documents independently of rendering issues. A step further is the Resource Description Framework (RDF, [14,7]), which proposes a simple model of assertion. This model provides a means of expressing information about documents (meta-data) or about anything with an identifier.

That alone does not make the Semantic Web a reality. But a common language and a common model are the basis upon which sophisticated mechanisms for description and inference can be built. The word "inference" here must not be taken in its strictest meaning: Knowledge Representation (KR), as it has been studied in Artificial Intelligence, makes many assumptions that do not fit the Web, which is massive, "open world" and dynamic [11]; it is also probably self-contradictory. Owing to those features, classical KR techniques are not suited to the Web, and more adapted techniques must be developed.

Annotation Interconnected Strata (AI-Strata) are a model for audiovisual streams annotation. Since it has no formal semantics, it does not allow strict inferencing; but an efficient algorithm allows "contextual inferencing", based on path matching in a graph. The counterpart is that users are free to use the language in any way that makes sense to them (even in ways that were not intended from the start). Our belief is that many applications of the Semantic Web could benefit from this kind of language. Therefore, the work presented here aims at extending the domain of AI-Strata from audiovisual streams to any kind of resources, by mapping AI-Strata concepts to RDF.

AI-Strata and RDF are first presented in sections 2 and 3, respectively. Then we will show how the former can be mapped to the latter with two examples in section 4. We will then compare our approach to related works, before concluding and discussing future developments.

2 AI-Strata

The model we present here has been designed in the framework of the SESAME project. SESAME means *Système d'Exploration de Séquences Audiovisuelles et Multimédia Enrichi par l'Expérience*: experience enriched audiovisual sequences exploration, and is partially funded by France Télécom. One of the purposes of the project was to study how the use of an audiovisual information system (AVIS) could be enriched with experience of previous sessions.

The first step we proposed was the design of a model called AI-Strata sufficiently general as to be able to express any other AV description model. This model is mainly based on "semantic network modeling" as networks of so-called annotation elements are used to describe AV content. *Annotation elements* (*AE*) are *a minima* terms (which are described in a knowledge container), and annotation graphs can be thought of as terms networks.

The second step of the study focused on interpretation and contextual exploitation of these networks, based on potential graphs, that are tools representing one particular way to search and contextualize terms in the network. We also proposed *description schemes* as means to control how the description should be "written" by the annotator, so how it should be "read" by other users.

We elaborate on these concepts in the next part. For more precision, the interested reader should refer to [17,18].

2.1 An annotation graph

The main interest in AV document modeling is that it consists in modeling something that is usually described at a very low level of conceptualization: mainly the superposition of audio and video streams. Every use of AV document going beyond simple visualization needs to ground on further modeling [1].

Describing and modeling a temporal medium is based on *annotation*, that means attaching a description to a temporally (on a temporal base) situated piece of document. The AI-Strata model is a very general one, based on the notion of annotation element. Annotation elements are objects, named with a term (*e.g.* Mandela, Shot or ZoomIn). They possibly have attributes (e.g. Speech:text, TimeOfDay:date), for more precision, or taking into account pre-calculated image features such as color histograms. Here, we will focus on the names of annotation elements, as terms.

Annotation elements annotate *audiovisual units* (AVU), that are defined by an audiovisual stream and two timestamps, delimiting strata (that may overlap) in the stream. As many annotation elements as necessary can annotate an audiovisual unit (see examples on fig. 1). Of course this notion can be generalized: the MPEG-4 standard [13], for example, audiovisual streams are much more structured. Any element of this structure (*audiovisual objects*)can be considered as an extended AVU.

So as to structure the annotation, and get more expressivity than simple "keyword on the temporal stream", it is possible to set up relations between annotation elements. So as to get homogeneity in the description, we use only one relation type — the *elementary relation* R_e — that can be used to link any two annotation elements. Expressing the semantics of a relation is allowed through the use of one more annotation element, like in $Shot \rightarrow VideoFocus \rightarrow Mandela$ or $Mandela \rightarrow Agt \rightarrow ShakingHands$.

The model is called after these elementary relations whose property is to connect audiovisual units, i.e. AV document strata: annotation interconnected strata, AI-Strata.

Considering the expressivity and the generality of the model, the description it allows can be very rich, and there are of course a lot of possible uses. We actually want to ground every task in an AVIS upon that model.

2.2 A controlled vocabulary as knowledge container

We consider that annotation elements have to be defined in a knowledge container, and cannot be created at will. We define our knowledge container as a network of *abstract annotation elements* (AAE). These elements are at least organized in a specialization hierarchy (no mandatory inheritance relation), and other conceptual relations are allowed. The knowledge container is at least a terminology or a thesaurus, describing the terms (possibly their attributes) that can be used to annotate the stream. *A maxima*, the container could be considered as an ontology, defining with precise relations (allowing inferences) what concepts are defined, and can be used to annotate.

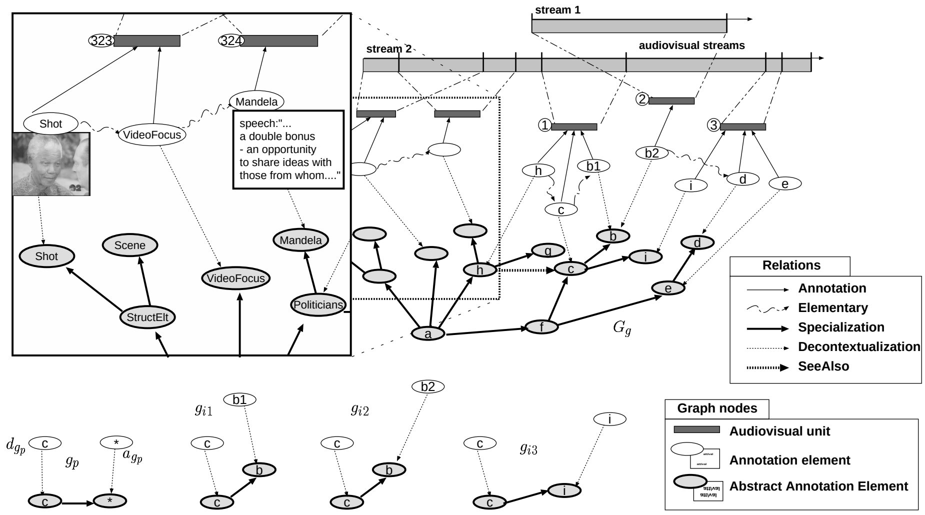

Figure 1. Top: an AI-Strata graph G_g with annotation elements as terms extracted from a knowledge container. Bottom: a potential graph g_p and its three instances in the general graph G_g: g_{i1}, g_{i2} and g_{i3}

Abstract annotation elements are clustered into *analysis dimensions* which represent containers that are actually used to annotate the stream according to one point of view. For instance, the stream could be analyzed along $< AD : Politician >= \{AAE : Clinton, AAE : Mandela, AAE : Chirac\}$ which is dedicated to politician annotations.

2.3 Contextual exploitation as contextual inferences

The audiovisual units, the annotation elements and the abstract annotation elements represent three types of nodes in a general AI-Strata graph.

The idea that was at the origin of the model was to consider that if an audiovisual unit u_1 was at first annotated by its own annotation elements, it was also annotated by annotation elements annotating audiovisual units u_i belonging to its context. This context can be temporal (e.g. inclusion), but also conceptual, if a path exists between two elements of the annotation graph. On figure 1, the AVU 323 is also contextually annotated by $AE : Mandela$ because of the path between AVUs 323 and 324.

By extension, the general notion of *contextual inferencing* consists in putting elements of the graph in the context of others (that are known, e.g. AVU 328), and/or to recognize elements as being in the context of others.

A context can be defined as a *potential graph*, which is an AI-Strata graph with generic nodes. A potential graph is instantiated in the general graph, so as to find subgraphs that are isomorphic to it (considering the generic nodes) [15]. Figure 1 gives example of a potential graph and three instances. Potential graphs can also contain *temporal relations* between annotation elements. Those relations are not matched by actuals arcs of the general graph; instead they are computed from the time-intervals where the annotation elements occur.

As a matter of fact, any exploitation task of the graph can always be thought of as potential graph creation, manipulation and instantiation. Indeed, queries consist in searching elements that belong to the context of known elements (for instance abstract annotation elements, unique by definition, or audiovisual units).

The interpretation of an annotation element (considered as a term) always occurs in the context of the AV stream, and mainly of other annotation elements belonging to the annotation graph. An annotation element is always considered as contextually explained by other elements defined by potential graphs.

As said earlier, we need a means of controlling the annotation, *i.e.* to specify which annotation elements should be used, and how they should be linked together with elementary relations. Analysis dimensions allow us to define clusters of abstract annotation elements that should be used one for another ("paradigmatic" classes).

We define *description schemes* as networks (with extended elementary relations and temporal relations) of analysis dimensions expressing how abstract annotation elements extracted from analysis dimensions should be put into relation ("syntagmatic" relations between paradigmatic classes, as in linguistics analysis).

3 The Resource Description Framework

3.1 Overview

The Resource Description Framework (RDF) is a recommendation of the World Wide Web Consortium [14]. It proposes a standard model and syntax to describe resources, *i.e.* entities identified by a Uniform Resource Identifier (URI, [3]). Most typical examples are of course the resources available on the Web (URLs are a particular kind of URIs).

The RDF model consists in annotating a resource with either another resource or a piece of text (*literal*). Each annotation has a label called *property*; annotations are often called *triples* as they are characterized by three elements: the label, the annotated resource and the annotating element (resource or literal). This produces a directed graph, with its arcs labelled with properties and its nodes labelled with URIs or literals (*cf.* fig. 2). The recommendation proposes a specific syntax based on XML, but other syntaxes could be used to produce an RDF-graph.

It is worth noting that properties are in fact URIs too. This means that a property can be annotated using RDF, like any other resource. Arcs of the graph can be reified, *i.e.* they can be given an URI, thus arcs themselves can be annotated using RDF. It follows that the language can describe any of its elements

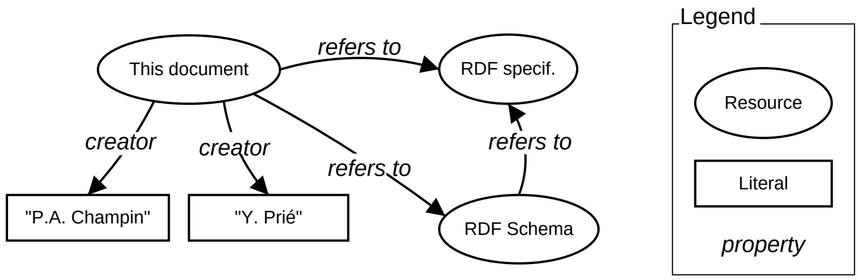

Figure 2. In this RDF graph, the upper-left resource represents this document. It has the property author with two literal values: "P.A. Champin" and "Y. Prié". It also has the property refers to with two resource values: the RDF specification [14] and RDF Schema [6]. As a matter of fact, RDF Schema also refers to the RDF specification. (NB: in the figure, we name each resource with a short text rather than its actual URI, for the sake of readability).

(see fig. 3). RDF Schema [6] offers a mechanism to *define* a new vocabulary (new properties with constraints on them, classes of resources) using RDF itself.

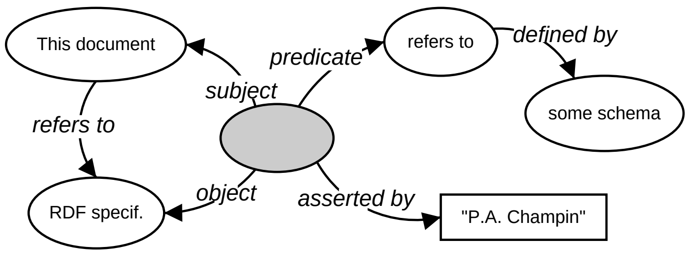

Figure 3. The arc labeled with refers to is reified into the gray resource. This resource has three properties subject, object and predicate with values corresponding to the elements of the triple. Once reified, the arc can be annotated just like any other resource; it has for example the property asserted by with value "P.A. Champin". Note also that the property refers to is actually a resource, so it can be used to annotate another resource (*e.g.* the reified arc) or be annotated itself (property defined by).

3.2 Semantics of resources

RDF has no defined formal semantics: there is no *interpretation function* to map URIs, terms of the language, to a theoretically defined model. It implicitly relies on the intuitive semantics of resources. However, the problem of semantics of resources is not a trivial one, since resources do not have a clear unambiguous meaning.

Let's consider the most common kind of URIs: URLs. They might seem to be the easiest URIs to interpret, as they are *retrievable* through the network (or at least through a single system). The intuitive interpretation for a URL is the retrieved *entity*. But that entity is highly dependent on the retrieving context: the encoding format can depend of a transaction between the browser and the server (in the HTTP protocol), the URL of a weather report will return a different page each day, *etc.* A URI could even be meaningless in some context: a `file://` URL identifies no resource outside its originating system[1].

On the other hand, schemas could provide a formal semantics for the URIs they define (those resources are not retrievable, so there is no retrieving context problem with them). But since the HTML experience, Web users have been known for using well defined standards in ways they were not expected to — yet still meaningful for people. What may have been considered as misuses happens to become the actual web uses. Therefore they should not be treated as "mistakes" in the context of the Semantic Web.

We conclude that URIs (the terms of the RDF language), like terms of natural languages, can have very different meanings depending on the context they are used in. We can not assume that they have a single interpretation, unless RDF assertions are stated with a strictly controlled vocabulary, by a software agent or a trained user. Although that assumption may be fulfilled in some applications, addressed to a limited community, it is much too restrictive for the whole Web (as well as for other domains, as discussed in [16]). Since AI-Strata do not rely on such an assumption, it seems appropriate to map them to RDF.

4 Mapping AI-strata to RDF

AI-strata and RDF both use a directed labeled graph structure, so writing AI-strata into RDF is quite straightforward. After an example taken from the audiovisual domain, we will discuss how each node type and relation type is mapped to the unified RDF model, where everything is a resource; we will see how this uniformization makes AI-Strata usable outside the audiovisual domain, and illustrate this in a second example, involving typical resources of the Semantic Web.

4.1 An example in the audiovisual domain

The AI-Strata version of this example is represented on figure 4. It contains a small knowledge base, an annotated audiovisual stream (with two interconnected AVUs) and a potential graph. RDF translations of these three components are given in figures 5, 6 and 7. We have not addressed analysis dimensions and description schemes so far.

The example is about a single audiovisual stream, probably about a meeting between Bill Clinton and Nelson Mandela. Since both are present during the

[1] Contrary to a common belief, the U in URI does not mean "Universal" but "Uniform"!

first minute, an AVU A with timestamps 0s and 60s has been created, annotated by both terms Clinton and Mandela. There is also an interesting shot between timestamps 10s and 20s, so a second AVU B has been created annotated by Shot. During this shot, the video focus is on Nelson Mandela, so another AE, VideoFocus, is attached to B, and the three AEs Shot, VideoFocus and Mandela are linked.

The potential graph Pg is designed to search an AVU annotated with a Shot AE. This AE must be linked to a VideoFocus AE, which must be linked to any AE. This AE must occur during the occurence of a Clinton AE, and it must be linked to any AAE which specializes the AAE Celebrity. It will be interpreted as *"Search a shot with video focus on any celebrity appearing at the same time as Clinton"*. Such a request would obviously return AVU B.

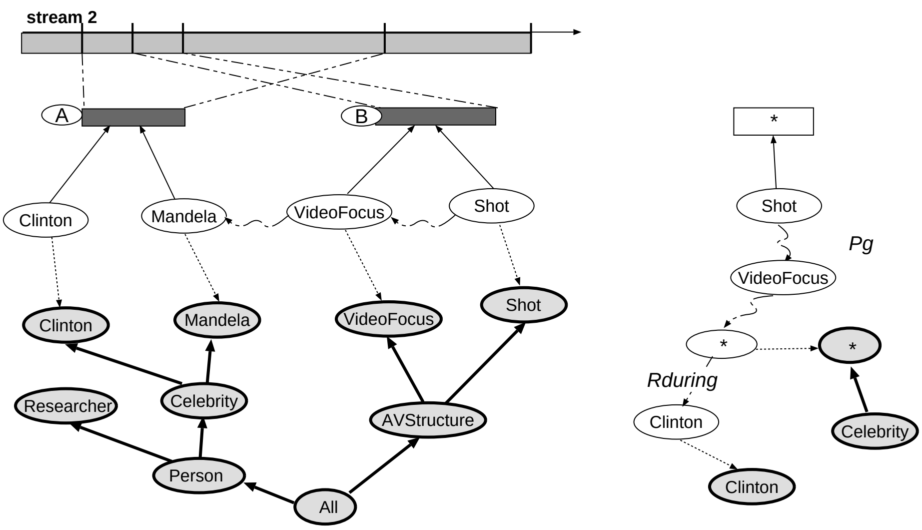

Figure 4. Example: annotated audivisual stream with two AVUs (left), potential graph that instanciates in the general graph (right)

Figure 5 represents a part of the knowledge base definition. It uses two RDF schemas: the core RDF syntax schema (`rdf:`), and our AI-Strata schema (`ais:`). This is not the only way of serializing this graph; actually, RDF would allow much more compact expressions with the same resulting graph, but this one is (intended to be) the most readable.

Figure 6 represents the annotation graph. Clearly, it refers to elements of the knowledge base. AEs are described quite in the same way as AAEs.

Figure 7 represents the potential graph Pg of figure 4. Additionnaly to the schemas mentionned above, it uses a schema defining temporal relations (`aist:`). One might also notice that a special kind of URIs are used (`var:...`); this will be discussed below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE RDF [
 <!ENTITY ais  'http://www710.univ-lyon1.fr/~champin/RDF/AIS/'>
 <!ENTITY rdf  'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
]>
<rdf:RDF xml:lang="en" xmlns:rdf="&rdf;" xmlns:ais="&ais;">

  <!-- AAE definitions -->

  <rdf:Description ID="All"/>
  <rdf:Description ID="Person"/>
  <rdf:Description ID="Celebrity"/>
  <rdf:Description ID="Clinton"/>
  <rdf:Description ID="Mandela"/>
  <rdf:Description ID="Researcher"/>
  <rdf:Description ID="AudiovisualStructure"/>
  <rdf:Description ID="Shot"/>
  <rdf:Description ID="VideoFocus"/>

  <!-- AAE relations -->

  <rdf:Description about="#All">
      <rdf:type rdf:resource="&ais;AEE"/>
      <ais:Rs   rdf:resource="#Person"/>
      <ais:Rs   rdf:resource="#AudiovisualStructure"/> </rdf:Description>
  <Description about="#Person">
      <rdf:type rdf:resource="&ais;AEE"/>
      <ais:Rs   rdf:resource="#Celebrity"/>
      <ais:Rs   rdf:resource="#Researcher"/> </rdf:Description>
  <Description about="#Celebrity">
      <rdf:type rdf:resource="&ais;AEE"/>
      <ais:Rs   rdf:resource="#Clinton"/>
      <ais:Rs   rdf:resource="#Mandela"/> </rdf:Description>
  <Description about="#Clinton">
      <rdf:type rdf:resource="&ais;AEE"/> </rdf:Description>
  <!-- (...) -->
</rdf:RDF>
```

Figure 5. Knowledge base (AAEs) in RDF

4.2 Bringing AI-Strata to the Semantic Web

From audiovisual streams to resources. The AV stream is the support of the AVUs. It has a URL, be it local (file://) or remote (http://), since it has to be retrievable, hence it is a resource of a particular kind.

Considering the AV stream as a resource is not totally neutral, though: as we emphasized in the previous section, the retrieved entity may vary, depending on the retrieving context. We have to assume that those resources are *stable*, as we defined it in [7]: in any context, the resource is the same from the point of view of the annotator (even if technical aspects may change). In conclusion we discuss that stability hypothesis.

From audiovisual units to resource parts. AVU are the objects described by AI-Strata. They are defined as temporal intervals of an AV stream. As we stated that any AV stream has a URI, we can assume that temporal intervals can be described with *fragment identifiers* [3]. For example, AVUs in figure 6 use the following fragment syntax: `#time-interval(`t_1`;`t_2`)`.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE RDF [
 <!ENTITY kb   'example1.kb.rdf#'>
 <!ENTITY ais  'http://www710.univ-lyon1.fr/~champin/RDF/AIS/'>
 <!ENTITY rdf  'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
]>
<rdf:RDF xml:lang="en" xmlns:rdf="&rdf;" xmlns:ais="&ais;">

  <!-- AE definitions -->

  <rdf:Description ID="ea1"/>
  <rdf:Description ID="ea2"/>
  <rdf:Description ID="ea3"/>
  <rdf:Description ID="ea4"/>

  <!-- AE relations -->

  <rdf:Description about="#ea1">
      <rdf:type rdf:resource="&ais;AE"/>
      <ais:Rd   rdf:resource="&kb;Clinton"/>
      <ais:Ra   rdf:resource="file:///movie.mpg#time-interval(0;60)"/> </rdf:Description>
  <rdf:Description about="#ea2">
      <rdf:type rdf:resource="&ais;AE"/>
      <ais:Rd   rdf:resource="&kb;Mandela"/>
      <ais:Ra   rdf:resource="file:///movie.mpg#time-interval(0;60)"/> </rdf:Description>
  <rdf:Description about="#ea3">
      <rdf:type rdf:resource="&ais;AE"/>
      <ais:Rd   rdf:resource="&kb;Shot"/>
      <ais:Ra   rdf:resource="file:///movie.mpg#time-interval(10;20)"/>
      <ais:Re   rdf:resource="#ea4"/> </rdf:Description>
  <rdf:Description about="#ea4">
      <rdf:type rdf:resource="&ais;AE"/>
      <ais:Rd   rdf:resource="&kb;VideoFocus"/>
      <ais:Ra   rdf:resource="file:///movie.mpg#time-interval(10;20)"/>
      <ais:Re   rdf:resource="#ea2"/> </rdf:Description>
</rdf:RDF>
```

Figure 6. AI-Strata annotation in RDF

It will be possible to handle generalized AVUs (any resource part to be annotated) the same way, with other specialized syntaxes for the fragment identifier. Such a syntax exists for XML: XPointer [9]. This is encouraging since XML is expected to be used in a wide range of applications. Furthermore, other standards are following that direction: MPEG-4 as we mentioned it in section 2, JPEG2000[2] which will allow to define zones in images, *etc.*

Annotation elements and abstract annotation elements. Annotation elements (abstract or not) are AI-Strata specific resources. They can be simple terms, or they can be structured objects with many attributes. In the first case, they will be mapped to a non-retrievable URI, defined with the `rdf:ID` attribute of the RDF syntax. In the second case, they will be described in an external XML document. Of course we could also use RDF to describe attributes of AEs and AAEs, but our main interest here is to map the *graph structure* of the AI-Strata to RDF,

[2] `http://www.jpeg.org/`

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE RDF [
 <!ENTITY kb   'example1.kb.rdf#'>
 <!ENTITY ais  'http://www710.univ-lyon1.fr/~champin/RDF/AIS/'>
 <!ENTITY aist 'http://www710.univ-lyon1.fr/~champin/RDF/AIS-Temporal/'>
 <!ENTITY rdf  'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
]>
<rdf:RDF xml:lang="en" xmlns:rdf="&rdf;" xmlns:ais="&ais;" xmlns:aist="&aist;">

  <rdf:Description about="var:1531b97a-592f-463c-b221-48631ac1047d">
      <rdf:type rdf:resource="&ais;AE"/>
      <ais:Rd   rdf:resource="&kb;Shot"/>
      <ais:Re   rdf:resource="var:ed2e5c38-323c-42df-a688-3f6ae9662c88"/>
      <ais:Ra   rdf:resource="var:mark1"/> </rdf:Description>
  <rdf:Description about="var:ed2e5c38-323c-42df-a688-3f6ae9662c88">
      <rdf:type rdf:resource="&ais;AE"/>
      <ais:Rd   rdf:resource="&kb;VideoFocus"/>
      <ais:Re   rdf:resource="var:beb35b45-f1fa-4147-98a6-6bf7a9dcc44c"/> </rdf:Description>
  <rdf:Description about="var:beb35b45-f1fa-4147-98a6-6bf7a9dcc44c">
      <rdf:type    rdf:resource="&ais;AE"/>
      <aist:during rdf:resource="var:ca810b50-e3d5-4d9d-91fc-40a6475dcca0"/>
      <ais:Rd      rdf:resource="var:76ba87b4-9ecd-407f-836d-eaa318051e5a"/> </rdf:Description>
  <rdf:Description about="var:ca810b50-e3d5-4d9d-91fc-40a6475dcca0">
      <rdf:type rdf:resource="&ais;AAE"/>
      <ais:Rd   rdf:resource="&kb;Clinton"/> </Description>
  <rdf:Description about="var:76ba87b4-9ecd-407f-836d-eaa318051e5a">
      <rdf:type rdf:resource="&ais;AAE"/> </Description>
  <rdf:Description about="&kb;Celebrity">
      <ais:Rs   rdf:resource="var:76ba87b4-9ecd-407f-836d-eaa318051e5a"/> </rdf:Description>
</rdf:RDF>
```

Figure 7. Potential graph in RDF

not the internal structure of annotation elements. That's why our examples use the first case and simply define the URIs of annotation elements.

One will also notice in all RDF serializations that every AE and AAE has an additional `rdf:type` property. This is an RDF canonical property, whose values are classes defined in the AI-Strata schema. Actually, this is the only way different node types can be distinguished in RDF.

There is some noticeable difference between the management of AEs and the management of AAEs. To be considered as resources, they must all have an identifier. In the AI-Strata model, nodes only have *names*. Names of AAEs can be considered as identifiers, since AAEs are uniques; but names of AEs are not unique, since AEs always have the same name as the corresponding AAE. It follows that the name is redundant with the R_d relation. Therefore, AEs are given generated URIs (see fig. 6), and only the R_d relation is kept in the RDF graph.

As a remark, we can also stress the fact that AE are not only bound to be linked to a unique AAE, but also to a unique AVU. We could then consider each AE as the reification of a relation between an AVU and an AAE; but the reification syntax of RDF is quite cumbersome; furthermore, RDF could not deal with multiple annotations of an AVU with the same AAE (which AI-Strata can), since it can't deal with multiple identical triples.

Generic node. Generic nodes are used in potential graphs; they are comparable to variables. Although RDF has no variable mechanism, some propositions have been made by people from the W3C[3] to use a dedicated URI family. This proposition is appropriate for modeling AI-Strata's generic nodes in RDF. Such a URI begins with `var:`, either followed by a generated unique identifier, or a conventional name (to mark "interesting" nodes of the potential graph).

It can be pointed out that AEs in potential graphs can be named or not (in which case they are represented with a star, see fig. 4). As we emphasized, the name is redundant with the R_d relation, but that relation may not be mentionned in the potential graph (see nodes Shot and VideoFocus in Gp, fig. 4). Since we do not map names to the RDF graph, we have to explicit every R_d relation.

Relations. There are two kinds of relations in the AI-strata model: static relations, which are explicitly stated (R_a, R_e, R_d, R_s) and dynamic relations, like temporal relations, that are only stated in potential graphs and calculated in the general graph. Every relation is represented as a property in RDF, but the system has to be able to distinguish static properties from dynamic properties, as matching the latter is much less trivial than matching the former. That's why the AI-Strata schema has a meta-schema part, defining two subclasses of Property: Relation (covering AI-Strata's static relations) and CalculatedProp. This allows further dynamic relations to be defined in the future (see the second example).

4.3 An example in the Semantic Web domain

Our second example extends the notion of AVU to the more general notion of resource as it is used in RDF. We believe indeed that AI-Strata can proove useful not only in audiovisual information systems. With RDF, AVUs are identified by URIs with an optional fragment identifier. Figure 8 represents an annotation graph of two web resources.

On the left is a general graph with a knowledge base similar to the one in figure 4. `b.html` is a document about Bill Clinton and contains an image. Hence it is annotated with both AEs Clinton and <img>. As the source of the image in the page is `a.jpg`, the whole image resource is annotated with the AE Photo, and the AE <img> linked to it. We assume that some fragment identifier syntax exists to point to a zone of an image, as we did for temporal intervals in audiovisual streams. A given zone of the image, with URI `a.jpg#zone1`, is annotated with the AE Mandela.

The potential graph on the right echoes the one in figure 4. It looks for a resource annotated by both AEs Clinton and <img>, the latter linked to an AE Photo. Another AE, linked to any AAE specializing the AAE Celebrity, must occur in the annotation scope of the AE Photo. This relation R_{in} is the spatial counterpart of the temporal relations used in audiovisual AI-Strata. That potential graph could be read as "*Search a resource about Clinton, with a photo of*

[3] `http://lists.w3.org/Archives/Public/www-rdf-interest/2000May/0032.html`

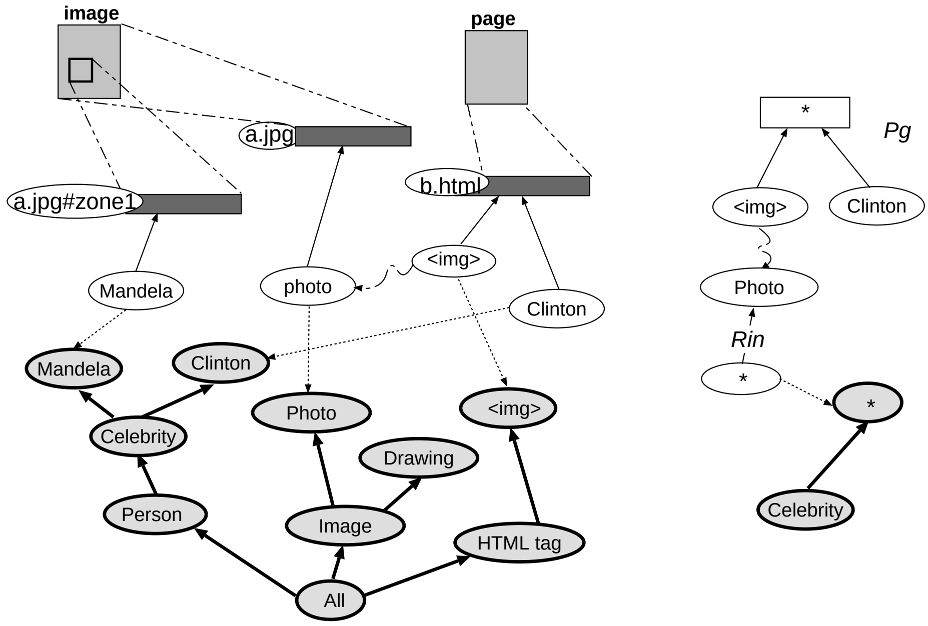

Figure 8. Example: annotated web resources (left), potential graph that instanciates in the general graph (right)

another celebrity in it". It matches `b.html`, since we naturally consider that any zone of an image is *in* the whole image.

We see that AI-Strata general graph and potential graph can easily be used with various kinds of resources, provided that we have a syntax for fragment identifiers to identify parts of the resources, which is the tendency of Web standards.

5 Related works and discussion

The word "semantic" is misleading, in that it has a very strict meaning in the KR community (much stricter that the common sense meaning). Still, we think that the so called Semantic Web can benefit from systems without a formal semantics, as we showed in the previous section. However, most works related to the Semantic Web are based on *ontologies*, systems of strictly organized terms with a frame semantics, despite the common agreement that classical KR techniques can not entirely scale up to the Web.

Ontobroker, for example, is based on F-Logic [8], a first-order logic enriched with frame features (classes, attributes, inheritance, *cf.* [12]). F-Logic assertions can be generated in different ways, including translation from RDF statements. To tackle the "open-wordlness" and self-contradiction of the Web, Ontobroker relies on the notion of *Ontogroup*: a group of users agreeing on the semantics of a specific ontology. This allow to use a close-world subset of the Web, which is supposed to be contradiction-free.

Contrary to Ontobroker, the Simple HTML Ontology Extensions (SHOE, [11]) does not rely on RDF, but only on HTML and XML. It focuses on the definition of ontologies, with mechanisms for versioning, reusing and extending, to fit the dynamic and uncontrolled nature of the Web. Inferencing is based on the original notion of *claim*: each fact keeps track of its claimant (the document it was asserted in), and an infered fact has its claimant computed from the claimants of the premises. Thus, whenever a contradiction arises, facts from a given claimant can be ignored so that the contradiction does not hold anymore.

Approaches mentioned above require that the users agree with the semantics behind the language, although used terms may have a larger meaning in natural languages. [4] propose a technique based on natural language instead: resources are described by a text (which may be the resource itself, for textual documents). This text is then translated into the formal language NKRL, to provide an "intelligent" retrieval. But that translation is not fully automated unless a "controlled natural language" is used, which brings us back to the drawbacks of ontologies.

To conclude, we can say that ontologies are powerful tools, and the AI-Strata knowledge base can be structured as an ontology if needed. But ontologies are also uneasy to use, when easyness is what made the Web so popular. Hence, they should not (and can not) be the only way to the Semantic Web. Our belief is that works related to the Semantic Web focus more on the semantics than on the web, and thus neglect one part of the problem: quite surprisingly, none of them strongly relies on existing *hyperlinks* of the web[4] (though Ontobroker allows to use HTML `href` attributes), since AI-Strata is based on the notion of *interconnection*.

6 Conclusion and future works

In this paper, we presented the AI-Strata, a model based on annotations of audiovisual streams. We also presented the RDF model recommended by the W3C, and discussed semantic issues raised by that model. We then proposed two examples to show that both model can fit, and that AI-Strata is a promising approach, not only for audiovisual information systems, but also for more general applications in the context of the Semantic Web. The originality of our approach, compared to related works, is that AI-Strata knowledge base need not have a formal semantics. That makes AI-Strata easier to use than ontology-based systems, and thus more suited to the Web as we know it.

We already have a prototype [10] allowing to edit general and potential graphs, and to perform subgraph matching. The persistent format used to store graphs is not standard. We are currently working on allowing RDF input and

[4] The recent success of the Google search engine (`http://www.google.org/`) has demonstrated that links pointing to a document are often more relevant that the whole content of the document itself.

output for general graphs as well as potential graphs. Other slight modifications will be needed to allow the user to describe other resources than audiovisual streams.

An RDF representation for analysis dimensions and description schemes is still to be studied. However, potential graphs alone can be used to perform requests in an AI-Strata general graph. Actually, they can be used to perform requests on any RDF graph, as the algorithm is quite generic (it only relies on the fact that the graph is directed and labelled). The AI-Strata prototype could therefore evolve toward a very general RDF tool.

We are also thinking about releasing the *stability assumption* for annotated resources. It would mean that different entities could be annotated under the same URI. This is not an intrinsically bad thing: if those entities have the same URI, they share a common core, and we can not know if the annotations address this common core or specific features of each entity. In the second case, contradiction may arise; managing it could be achieved with a mechanism similar to the claimant mechanism in SHOE.

Finally, there is a key problem in AI-Strata as well as RDF: both languages are meant to be distributed. In the Semantic Web context, where the amount of information will also be huge, it is critical that algorithms optimally rely on the distributed nature of the language, in order to avoid loading massive quantities of potentially useless information. We are currently investigating in this direction.

References

1. Gwendal Auffret and Yannick Prié. Managing full-indexed audiovisual documents: a new perspective for the humanities. *Computer and the Humanities, special issue on Digital Images*, 33(4):319–344, 1999.
2. Tim Berners-Lee. Semantic web road map, oct 1998. `http://www.w3.org/DesignIssues/Semantic.html`.
3. Tim Berners-Lee, R. Fielding, U.C. Irvine, and L. Masinter. Uniform resource identifiers (uri): Generic syntax. RFC 2396, aug 1998. `http://www.ietf.org/rfc/rfc2396.txt`.
4. Elisa Bertino, Barbara Catania, and Gian Piero Zarri. A conceptual annotation approach to indexing in a web-based information systems. *Economic Research and Electronic Networking Journal (NETNOMICS)*, to appear (6p).
5. Tim Bray, Jean Paoli, and C.M. Sperberg-McQueen. Extensible markup language (xml) 1.0. W3C recommendation, feb 1998. `http://www.w3.org/TR/1998/REC-xml-19980210`.
6. Dan Brickley and R.V. Guha. Resource description framework (rdf) schema specification. W3C proposed recommendation, mar 1999. `http://www.w3.org/TR/1999/PR-rdf-schema-19990303`.
7. Pierre-Antoine Champin. Rdf tutorial, mar 2000. `http://www710.univ-lyon1.fr/~champin/rdf-tutorial/`.
8. Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman et al., editor, *Semantic Issues in Multimedia Systems. Proceedings of DS-8*, pages 351–369. Kluwer Academic Publisher, Boston, 1999.

9. Steve DeRose, Ron Daniel, and Eve Maler. Xml pointer language (xpointer). W3C working draft, dec 1999. `http://www.w3.org/TR/1999/WD-xptr-19991206`.
10. Elöd Egyed-Zsigmond, Yannick Prié, Alain Mille, and Jean-Marie Pinon. A graph-based audiovisual document annotation and browing system. In *RIAO'2000*, volume 2, pages 1381–1389, Paris, apr 2000.
11. Jeff Heflin, James Hendler, and Sean Luke. Shoe: A knowledge representation language for internet applications. Technical report, Dept. of Computer Science, University of Maryland at College Park, 1999. `http://www.cs.umd.edu/projects/plus/SHOE/pubs/techrpt99.pdf`.
12. Micheal Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42:741–843, may 1995.
13. Rob Koenen. Mpeg-4: Multimedia for our time. *IEEE Spectrum*, 36(2):26–33, feb 1999.
14. Ora Lassila and Ralph R. Swick. Resource description framework (rdf) model and syntax specification. W3C recommendation, feb 1999. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222`.
15. Yannick Prié, Tahar Limane, and Alain Mille. Subgraph isomorphism for contextual audiovisual information retrieval. In *Reconnaissance de Formes et Intelligence Artificielle, RFIA2000*, volume 1, pages 277–286, feb 2000. In french.
16. Yannick Prié and Alain Mille. Reuse of knowledge containers: a "local semantics" approach. To appear in Workshop on Flexible Strategies for Maintaining Knowledge Containers, 14th European Conference on Artificial Intelligence ECAI 2000, Berlin, aug 2000.
17. Yannick Prié, Alain Mille, and Jean-Marie Pinon. Ai-strata: A user-centered model for content-based description and retrieval of audiovisual sequences. In *Int. Advanced Multimedia Content Processing Conf.*, volume 1554 of *Lecture Notes in Computer Science*, pages 328–343, nov 1998.
18. Yannick Prié, Alain Mille, and Jean-Marie Pinon. A context-based audiovisual representation model for audiovisual information systems. In *International and Interdisciplinary Conference on Modeling and using Context*, volume 1688 of *Lecture Notes in Artificial Intelligence*, pages 296–309, sep 1999.
19. Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. Xml schema part 1: Structures. W3C working draft, nov 1999. `http://www.w3.org/TR/1999/WD-xmlschema-1-19991217`.