DH-CASE 2013 - Sept. 10, 2013, Florence

ANNOTATIONS WITH EARMARK IN PRACTICE: A FAIRY TALE

Gioele Barabucci, Angelo Di Iorio, Silvio Peroni, **Francesco Poggi**, Fabio Vitali

Università di Bologna

Motivations (1/2)

- **Context:** the digitalization of literary texts opens new possibilities since multiple annotations can be added by multiple actors:
  - author(s);
  - contributors to documents production and diffusion (editors, proofreaders, reviewers, etc.);
  - users **without write permissions on the documents**
- **Problems:**
  1. overlapping annotations, i.e. annotations that cannot be expressed inside the strict tree-based hierarchical organization of XML documents;

Motivations (2/2)

2. mechanisms to refer to external entities outside of a document, for example:
   - to associate a person to the role of an annotator on a specific text;
   - to link references to people, characters, places, events, etc. inside texts to their representation on the Web (e.g. DBPedia), opening interesting possibilities for data integration, automatic reasoning and semantic analysis in the Linked Open Data;
   - to state that different text fragments refer to the same univocally identified entity.

- **Goal:** provide a flexible mechanism that supports the creation and sharing of semantic annotations on these documents.

Adding annotations

There are two ways to add annotations to existing digital documents:

- ***embedding* techniques:** annotations are embedded in the document itself
  - **pros:** keep all the information in a single coherent file
  - **cons:** difficult to deal with overlapping markup. Moreover, it is not always possible (or not desirable) to modify the document
- ***standoff* techniques:** annotations are stored in a separate document with references to the parts of the document they refer to
  - **pros:** greatly simplify the problems related to embedded annotations
  - **cons:** serous issues arise when the original document is modified

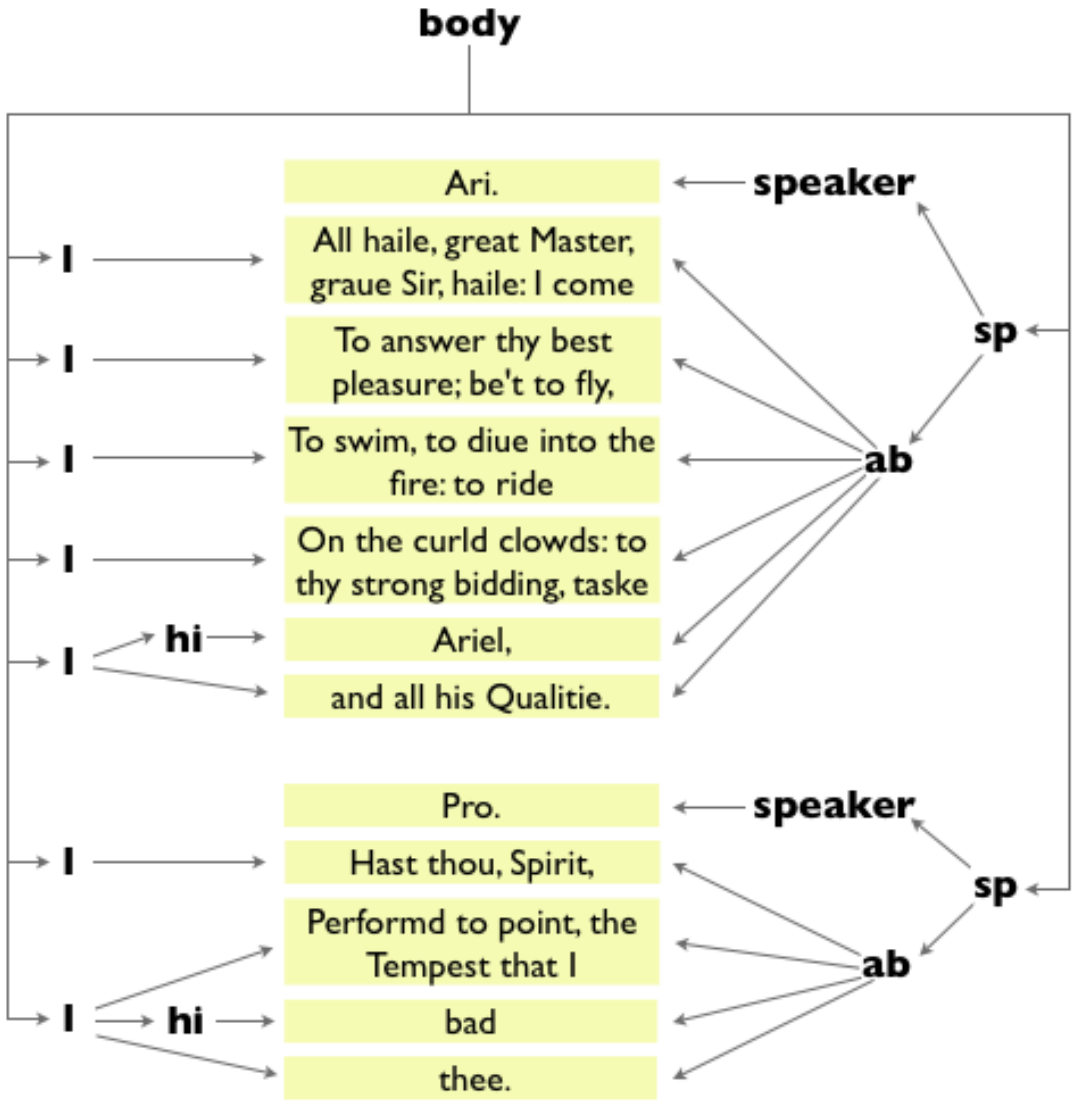
Example 1

```
<body >
    ...
    <sp>
        < speaker rend =" italic "> Ari . </ speaker >
        <ab >
            All haile , great Master , graue Sir , haile : I come < lb n ="301"/ >
            To answer thy best pleasure ; be 't to fly ,<lb n ="302"/ >
            To swim , to diue into the fire : to ride <lb n ="303"/ >
            On the curld clowds : to thy strong bidding , taske < lb n ="304"/ >
            <hi rend =" italic "> Ariel , </ hi > and all his Qualitie .< lb n ="305"/ >
        </ab >
    </sp >
    <sp >
        < speaker rend =" italic "> Pro . </ speaker >
        <ab >
            Hast thou , Spirit ,< lb n ="306"/ >
            Performd to point , the Tempest that I
            < seg type =" homograph "> bad </ seg > thee .<lb n ="307"/ >
        </ab >
    </sp >
    ...
</ body >
```

The Tempest by William Shakespeare

TEI version by the Oxford Text Archive at http://ota.ox.ac.uk/text/5725.xml

Example 1: two different hierarchies



The previous excerpt describes two different hierarchies:

1. the speeches (elements *sp*);
2. the various lines (elements *lb*, line breaks)

Example 1: different meanings for the same string

- Since annotations can be made by different annotators, in annotations environments it is important to keep track of provenance information.
- Let’s suppose that two different users want to annotate the string "Master" (a word pronounced by Ariel and clearly denoting another character of The Tempest, i.e. Prospero) with a slightly different meaning:
  - the first annotator (i.e. Silvio Peroni) wants to connote Prospero as a person: in fact Prospero is a person according to the story;
  - the second annotator (i.e. Gioele Barabucci) wants to refer to Prospero as a fictional character of the Shakespearian's world.

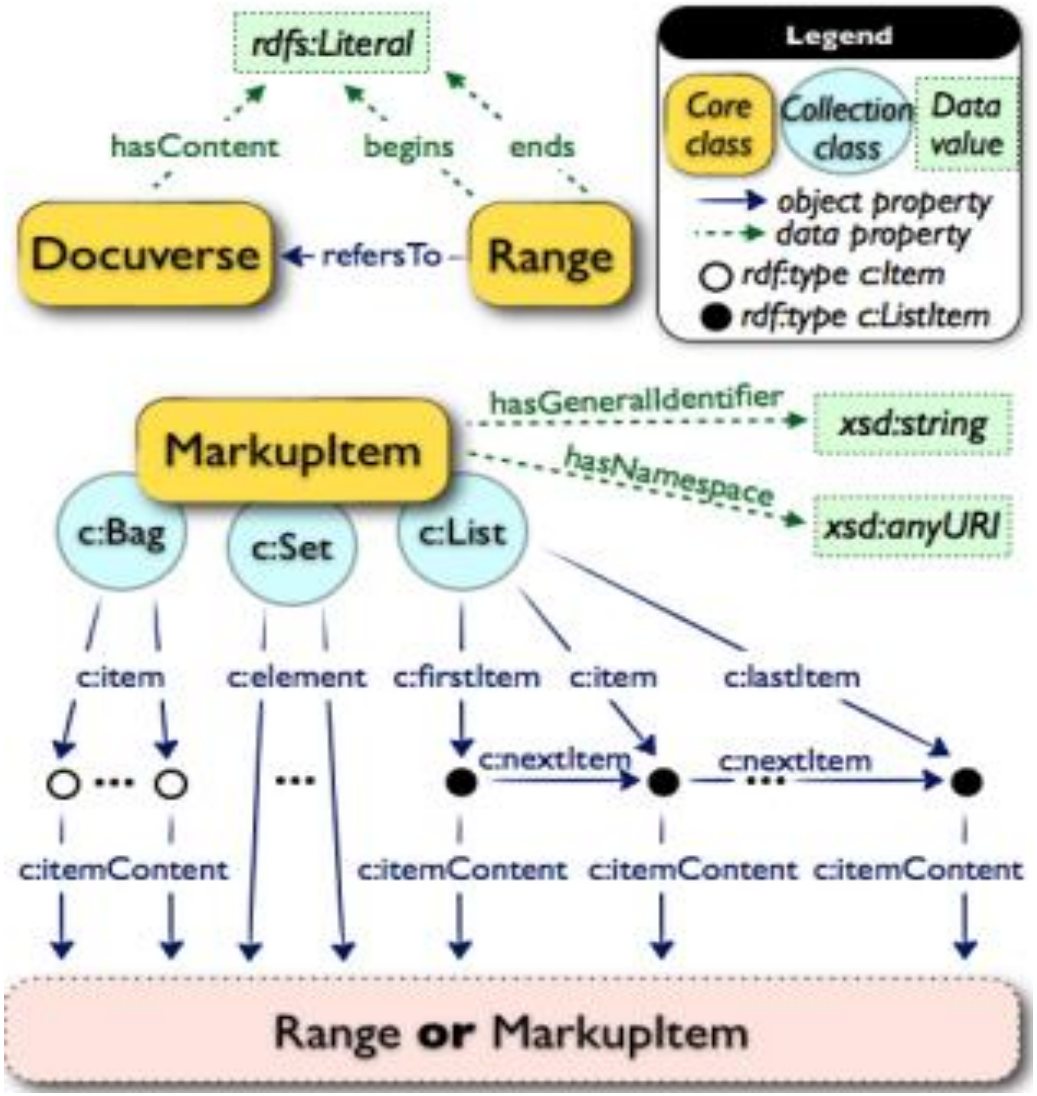
Annotation meets EARMARK

EARMARK: Extremely Annotational RDF Markup

Docuverse: represents the object of discourse, i.e., all the containers of text of an EARMARK Document

Range: any text lying between two locations of a Docuverse

Markup Item: defines artefacts to be interpreted as markup (such as elements and attributes)



Example 1: building blocks

- In the first example we will use:
  - *the Friend Of A Friend (FOAF) ontology to describe people as annotators;*
  - *the PROVenance Ontology (PROV-O) to associate provenance information to our assertions;*
  - the LA-EARMARK ontology, which allows to express semantic characterisations of information objects such as strings, pictures, and the like.

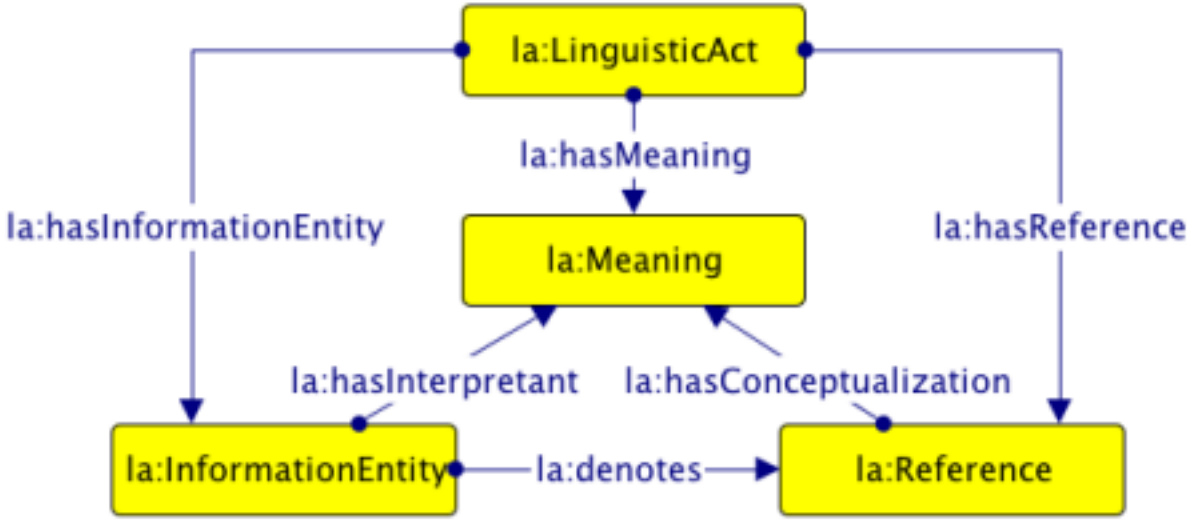
EARMARK: range definition

```
@prefix : <http://www.essepuntato.it/example/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix earmark: <http://www.essepuntato.it/2008/12/earmark#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# The EARMARK Document described as an OWL ontology
< http://www.essepuntato.it/example> a owl:Ontology .

# The textual content of the document to annotate
:content a earmark:URIDocuverse ;
    earmark:hasContent
        "http://ota.ox.ac.uk/text/5725.xml"^^xsd:anyURI .

# The string "Master"
:master-string a earmark:PointerRange;
      earmark:refersTo :content;
      earmark:begins "34023"^^xsd:nonNegativeInteger ;
      earmark:ends "34029"^^xsd:nonNegativeInteger .
```

EARMARK: defining annotators

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:silvio  a  foaf:Person, prov:Agent ;
    foaf:name "Silvio Peroni" .

:gioele  a  foaf:Person, prov:Agent ;
    foaf:name "Gioele Barabucci" .
```

EARMARK: expressing linguistic acts (1/3)



An excerpt of the Linguistic Act module:

- *linguistic acts* are particular communicative situations including information entities (i.e. symbols such as strings, each conveying a meaning or denoting one or more references), meanings (i.e. meta-level objects that explain something, or is intended by something, such as linguistic definitions, logical concepts or relations, etc.) and references (i.e. individuals, sets of individuals, or facts from the world we are describing).
- In addition to linguistic acts, we also added data related to the agent (i.e. prov:Agent), in this case the particular person (i.e. foaf:Person), who made such a linguistic act (i.e. prov:wasAttributedTo) and the time indicating when it has been generated (i.e. prov:generatedAtTime).

EARMARK: expressing linguistic acts (2/3)

```
@prefix la: <http://www.ontologydesignpatterns.org/cp/owl/semiotics.owl#> .
@prefix dbp: <http://dbpedia.org/resource/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix yago: <http://dbpedia.org/class/yago/> .

# Silvio 's interpretation of the string "Master"
:prospero-as-person a la:LinguisticAct , prov:Entity ;
    la:hasInformationEntity :master-string;
    la:hasReference dbp:Prospero ;
    la:hasMeaning foaf:Person ;
    prov:wasAttributedTo :silvio ;
    prov:generatedAtTime
        "2013-06-18T17:23:23Z"^^xsd:dateTime .

# Gioele's interpretation of the string "Master"
:prospero-character a la:LinguisticAct , prov:Entity ;
    la:hasInformationEntity :master-string;
    la:hasReference dbp:Prospero ;
    la:hasMeaning yago:ShakespeareanCharacters ;
    prov:wasAttributedTo :gioele ;
    prov:generatedAtTime
"2013-06-18T17:23:23Z"^^xsd:dateTime .
```

EARMARK: expressing linguistic acts (3/3)

- In the previous example:
  - information entity = the EARMARK range for the string “Master” (both cases)
  - reference = the DBPedia resource identifying *Prospero* (both cases)
  - meaning = a person (first ling. act) and a fictional character (second ling. act)
- In addition to linguistic acts, we also added data related to the agent (i.e. prov:Agent), in this case the particular person (i.e. foaf:Person) who made such a linguistic act (i.e. prov:wasAttributedTo) and the time indicating when it has been generated (i.e. prov:generatedAtTime).

Do it in practice: the EARMARK API

- We have already designed and implemented a framework for the creation, validation and manipulation of EARMARK data structures.
- All the code is written in Java and uses Jena. The implementation of the data structure follows exactly what is defined in the EARMARK ontology, encoding OWL properties as methods of these classes.
- The EARMARK data structure has been maintained as close as possible to a well-known and widely used model for XML documents (the Java DOM implementation).
- The API makes it simple to create/load/store/modify and add assertions to EARMARK documents directly in a Java framework.

EARMARK API: range definition

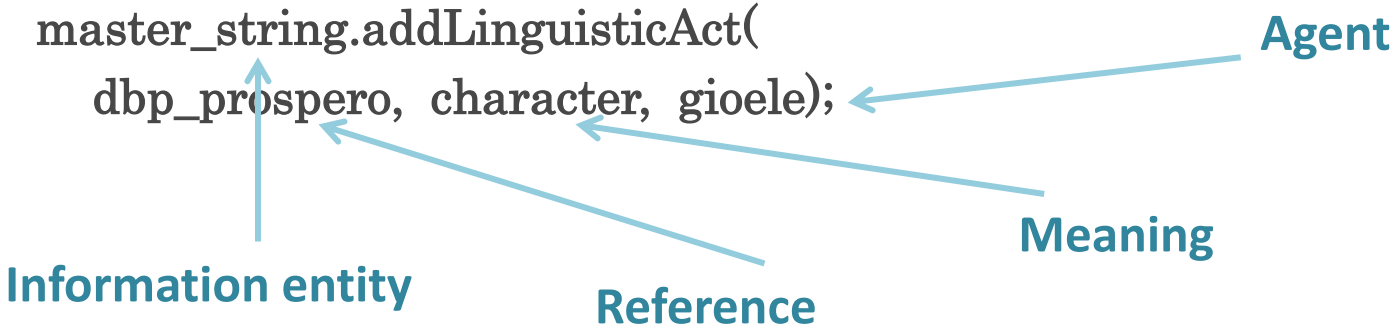
```
String ex  =  "http://www.essepuntato.it/example /";
EARMARKDocument  ed  = new EARMARKDocument(
    URI.create(ex));
Docuverse  doc  = ed.createURIDocuverse ("content",
    URI.create ("http://ota.ox.ac.uk/text/5725.xml"));
Range  master_string  = ed.createPointerRange(
    "master-string", doc, 34023, 34029);
```

EARMARK API: defining annotators

```
String prov = ... /* the same as Turtle prefix */
Model model = ed. getModel ();
Resource prov_Agent = model.getResource(
    prov + " Agent ");

Resource silvio = model.createResource(
    ex + "silvio");
silvio.addProperty(RDF.type, FOAF.Person);
silvio.addProperty(RDF.type, prov_Agent);
silvio.addProperty(FOAF.name, "Silvio Peroni");

Resource gioele = ...
```

EARMARK API: expressing linguistic acts

```
String dbp = ... /* the same as Turtle prefix */
String yago = ... /* the same as Turtle prefix */
Resource dbp_prospero = model.createResource(
    dbp + "Prospero");
Resource character = model.createResource(
    yago +"ShakespeareanCharacters");

master_string.addLinguisticAct(
    dbp_prospero, FOAF.Person, silvio);
master_string.addLinguisticAct(
    dbp_prospero, character, gioele);
```



Example 2: documents do change

```
<sp>
   < speaker rend =" italic "> Ari . </ speaker >
   <ab >
      All haile , great Master , graue Sir , haile : I come <lb n="301"/>
      To answer thy best pleasure ; be 't to fly ,<lb n="302"/>
      To swim , to diue into the fire : to ride <lb n="303"/ >
      On the curld clowds : to thy strong bidding , taske <lb n="304"/>
      <hi rend =" italic "> Ariel , </ hi > and all his Qualitie .<lb n="305"/>
   </ab >
</sp >
```

AUTHOR'S CHANGE

- Add lines (elements *l*) within the *ab* elements
- Spell out all the abbreviations in the *speaker* elements

```
<sp>
   < speaker rend =" italic "> Ari el</ speaker >
   <ab >
      < l>All haile , great Master , graue Sir , haile : I come <l/>
      < l>To answer thy best pleasure ; be 't to fly ,<l/>
      < l>To swim , to diue into the fire : to ride <l/>
      < l>On the curld clowds : to thy strong bidding , taske <l/>
      < l><hi rend =" italic "> Ariel , </ hi > and all his Qualitie .<l/ >
   </ab >
</sp >
```

EARMARK

```
# The string "Ari." (original) and "Ariel" ( modified )
# within the element "speaker"
:ari-string a earmark:XPathPointerRange ;
   earmark:refersTo :content ;
   earmark:hasXPathContext
      "//body/div[2]/sp[43]/speaker" .

# The string "Master " (both original and modified)
:master-string a earmark:XPathPointerRange ;
   earmark:refersTo :content ;
   earmark:hasXPathContext "//body/div[2]/sp[43]/ ab" ;
   earmark:begins "17"^^xsd:nonNegativeInteger ;
   earmark:ends "23"^^ xsd:nonNegativeInteger .
```

EARMARK API

```
Docuverse modified = ed.createURIDocuverse("content",
    URI.create("http:// www.essepuntato.it/2013/dhcase/
        thetempest-modified.xml"));

Range ari_string = ed.createXPathPointerRange(
    "ari-string", modified, null, null,
    "//body/div[2]/sp[43]/speaker");
ed.appendChild(ari_string);

Range master_string = ed.createXPathPointerRange(
    "master-string", modified, 17, 23,
    "//body/div[2]/sp[43]/ab");
ed.appendChild(master_string);
```

Example 3: transclusions

- Since in EARMARK documents all markup items and fragments identified by **URIs**, it possible to refer, from a document, to content belonging to other EARMARK documents without redefining the markup and text of the elements.
- These references can be exploited to create simple inclusions but also to create permanent and live connections between the inclusion and the original source of content, and to build sophisticated applications on top of these connections.
- This is a form of **transclusion**, similar to what has originally been proposed by Ted Nelson for the **Xanadu project**. The original goal of the Xanadu project was to build a global workspace where all users could freely reuse, comment on and link to any piece of content.
- The current EARMARK model already provides everything is needed to express such xanalogical relations between documents, and to mix them with relations to external entities.

Example 3: transclusions

An EARMARK document for the following structure:

```
<div >
   <p>
      This  document  quote , as  follows ,
      the  first  line  of  Ariel  speech
      defined  in  the  excerpts  of  another
      EARMARK  document : </p>
   <blockquote >
      <l> All  haile ,  great  Master ,  graue  Sir ,
      haile :  I  come </l>
   </ blockquote >
</div >
```

With EARMARK, we can transclude the first line spoken by Ariel into an arbitrary document.

EARMARK API: tranclusions (1/2)

```
EARMARKDocument   ed =  new  EARMARKDocument(URI.create(
   "http:// www.essepuntato.it/ another-example"));
Docuverse  doc  =  ed.createStringDocuverse (
   "This  document  quote, as  follows,  ...") ;
Element  div  = ed.createElement(
   another  + "div", "div",  Type.List);
Element  p = ed.createElement(
   another  + "p", "p",  Type.List);
Range  range  = ed.createPointerRange(
   another  + "range", doc, null, null);
Element  blockquote  = ed.createElement(
   another  + "blockquote", "blockquote",  Type.List);
Element  l1 = ed.createElement(
   ex + "l1", "l",  Type.List);
```

EARMARK API: tranclusions (2/2)

```
div.appendChild(p);
p.appendChild(range);
div.appendChild(blockquote);
blockquote.appendChild(l1);

Model  model  = ed.getModel();
model.add(l1,  RDFS.isDefinedBy,  model.createResource(
    "http://www.essepuntato.it/example"));
```

From fairy tales to the cold XML reality

- The proposed approach uses technologies of the Semantic Web while the majority of tools are still based on XML.
- For this reason we developed a tool called FRETTA (From EARMARK To Tag) that can embed arbitrary EARMARK annotations (including non-sequential, non-hierarchical and non-contiguous ones) inside XML documents.

From fairy tales to the cold XML reality

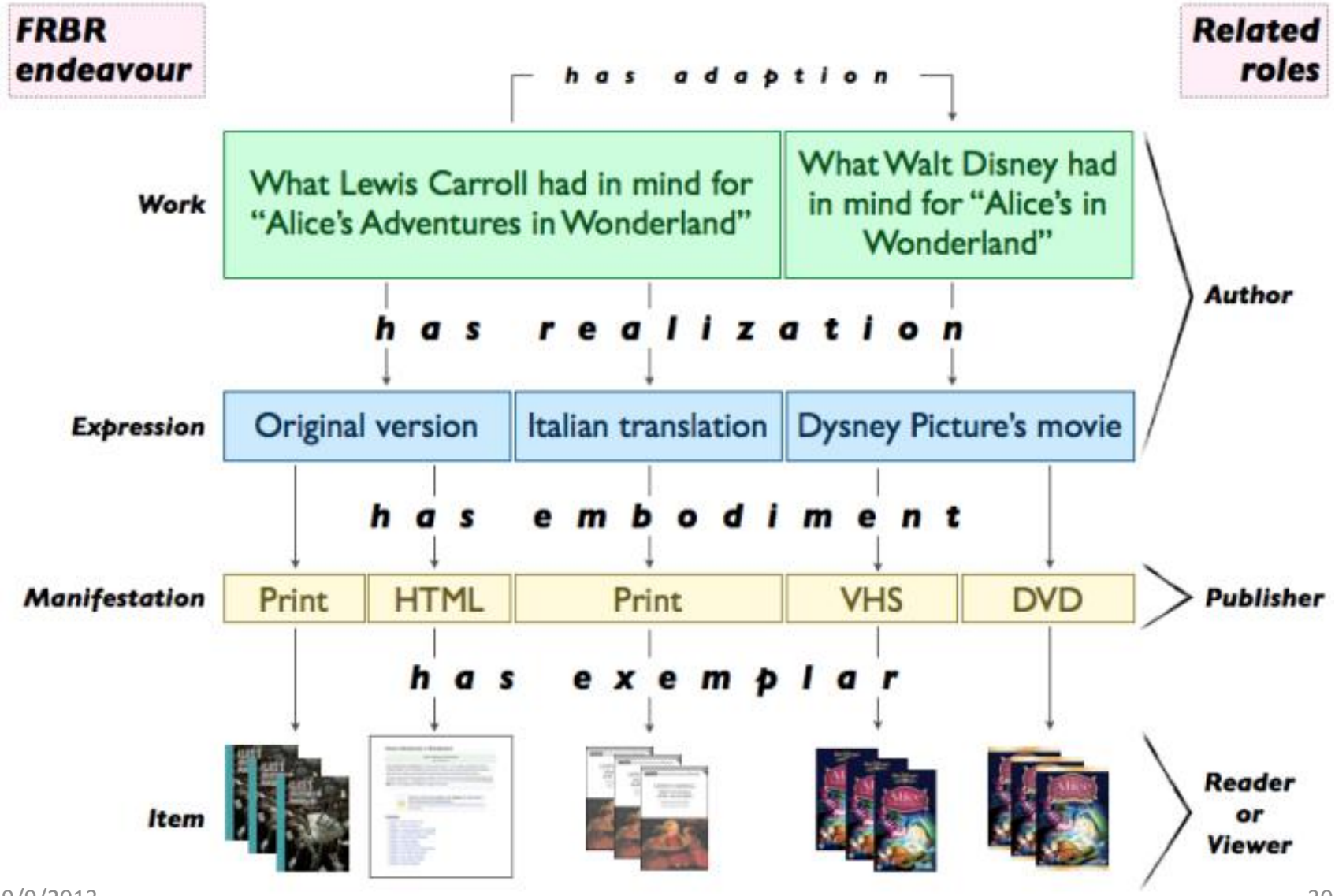
- Main problems addressed by FRETTA:
  1. preservation of the overlapping hierarchies expressed through the structural markup using a half a dozen of the techniques known as “syntactic workarounds” (e.g. milestones, fragmentation, etc.);
  2. the correct embedding of assertions: FRETTA knows different embedding technologies that can be used to deal with annotations (e.g. it is possible to choose to inject them into XML attributes using RDFa, or using solutions based on stand-off markup).

Future works

- Inclusion of direct support for transclusions in the future releases of the EARMARK API;
- To investigate and explore the possibility of integrating transclusions with Semantic Web technologies and sources of data

Thanks for your attention

FRBR


FRBR endeavour
Related roles
has adaption
Work
What Lewis Carroll had in mind for "Alice's Adventures in Wonderland"
What Walt Disney had in mind for "Alice's in Wonderland"
Author
has realization
Expression
Original version
Italian translation
Dysney Picture's movie
has embodiment
Manifestation
Print
HTML
Print
VHS
DVD
Publisher
has exemplar
Item
Reader or Viewer

FRBR

- Work. A FRBR Work is a high-level abstract Platonic concept of the essence of a distinct intellectual or artistic creation, for example the ideas in Lewis Carroll's head concerning Alice's Adventures in Wonderland, independent of any representation of these ideas in a particular form. A Work is realized through one or more Expressions;
- Expression. A FRBR Expression is the realisation of the intellectual or artistic content of a Work. Thus the original text of Alice's Adventures in Wonderland and its Italian translation Le Avventure di Alice nel Paese delle Meraviglie refer to different Expressions of the same Work. An Expression is embodied in one or more Manifestations;
- Manifestation. A FRBR Manifestation of a work defines its particular physical or electronic embodiment, for example, the particular format in which “Alice's Adventures in Wonderland” is stored: as a printed object or in HTML, that are two quite different Manifestations. A Manifestation is exemplified by one or more Items;
- Item. A FRBR Item is a particular physical or electronic copy of Alice's Adventures in Wonderland that a person can own, for example the printed version of that book you have in your bookcase, or the Mobipocket format copy you have downloaded to read on your e-book device.