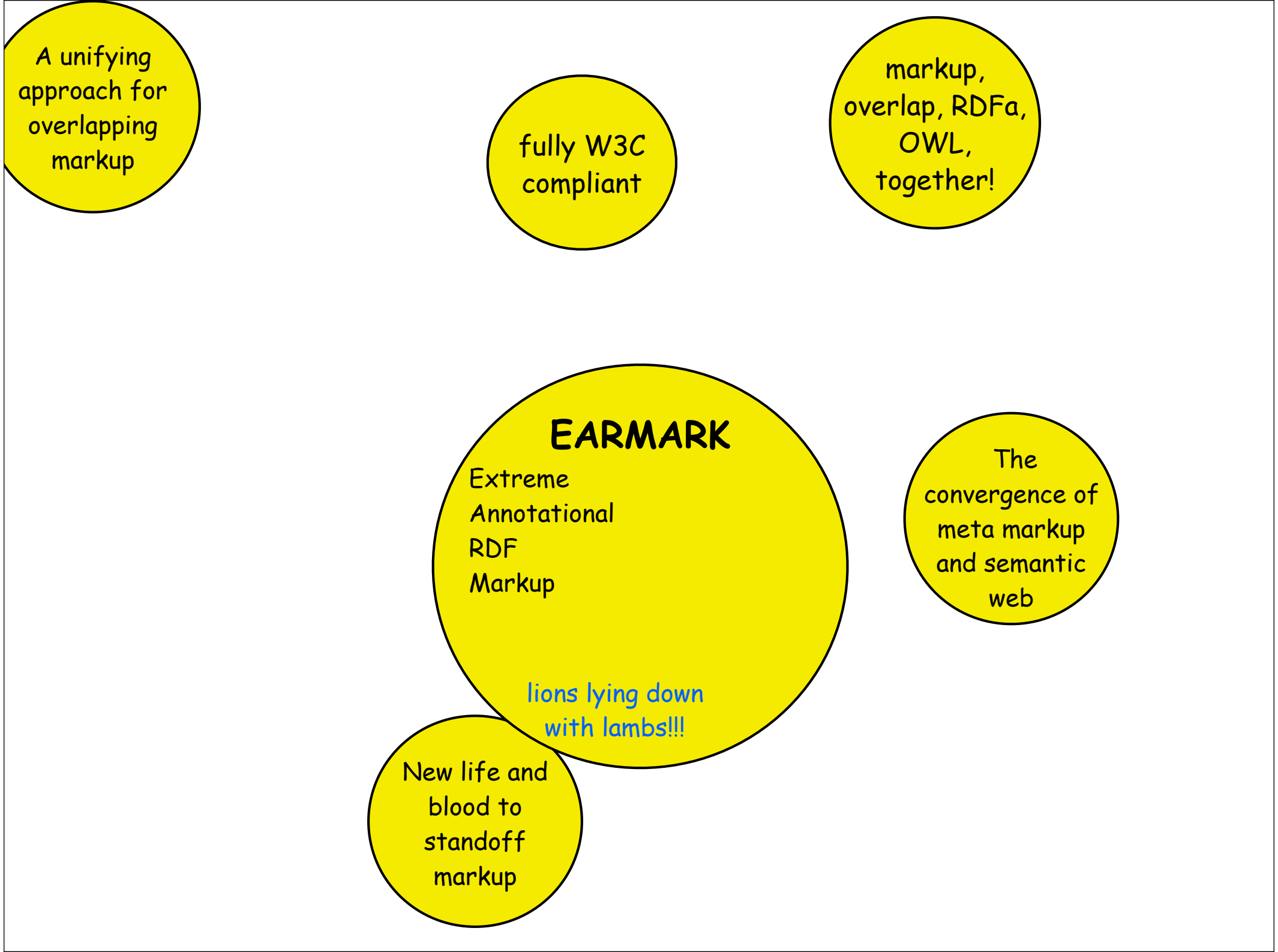
Angelo Di Iorio, Silvio Peroni, ***Fabio Vitali***
Department of Computer Science
University of Bologna


A unifying approach for overlapping markup
fully W3C compliant
markup, overlap, RDFa, OWL, together!
EARMARK
Extreme
Annotational
RDF
Markup
lions lying down with lambs!!!
The convergence of meta markup and semantic web
New life and blood to standoff markup

Angering everybody

- W3C brand of Semantic Web rather than ISO
- Triples rather than quadruples, quintuples, ennuples
- Standoff markup rather than embedded
- Full reliance on existing standards and best practices!!!

Multiple hierarchies on the same content are...

- ... more frequent than we allow ourselves to believe...
- ... actually useful...
- ... dealt with in plenty of available literature...

... but ...

- ... hard to design and understand...
- ... manual to create...
- ...

Some solutions

- **XML approaches**
  - Segmentation
  - Milestones (ECLIX)
  - Flat Milestones (CLIX)
  - Twin Documents
- **Non-XML meta-languages**
  - TexMecs
  - LMNL
  - XConcur
- These solutions are all **embedding** markup within content

Embedding markup considered harmful*

* Ted Nelson, 1997

- Embedded markup only states **one** thing for each fragment

This text is italic and bold

but *really*

- the text is italic,
- the i element is bold, and
- the b element is a paragraph

Embedding markup considered harmful*

* Ted Nelson, 1997

- Embedded markup only states **one** thing for each fragment
- Embedded markup only states things in document order
- Embedded markup only states things about adjacent fragments

<sp who="Hughie">**da de dum de dum...**<l>**gets a new frog**</l></sp>
<sp who="Louis">**Er...**<l>**it's a new pond**</l></sp>
<sp who="Dewey">**Ah...**<l>**When the old pond**</l></sp>

Where is the haiku?

Embedding markup considered harmful*

*Ted Nelson, 1997

- Embedded markup only states **one** thing for each fragment
- Embedded markup only states things in document order
- Embedded markup only states things about adjacent fragments
- Embedded markup does not distinguish between containment and dominance

This text is italic and bold

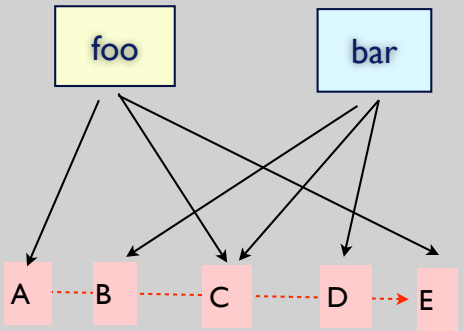
compare with

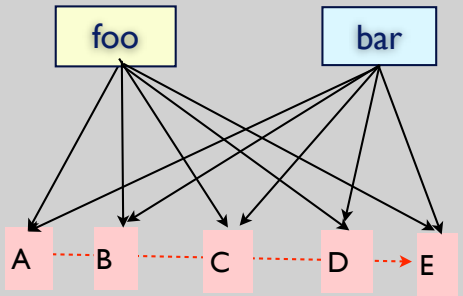
<table><tr><td>***This text is in a cell***</td></tr></table>

- p, b and i all *contain* the text, and p *dominates* b which *dominates* i
- table *contains* tr which *contains* td which *contains* the text

GODDAG and r-GODDAG

- Even GODDAG has no linearization in embedded markup
- r-GODDAG is the best restriction of GODDAG allowing embedded markup (i.e., TexMecs)
  - Contiguity of dominated nodes must be guaranteed
  - Independent domination of the same nodes cannot be allowed





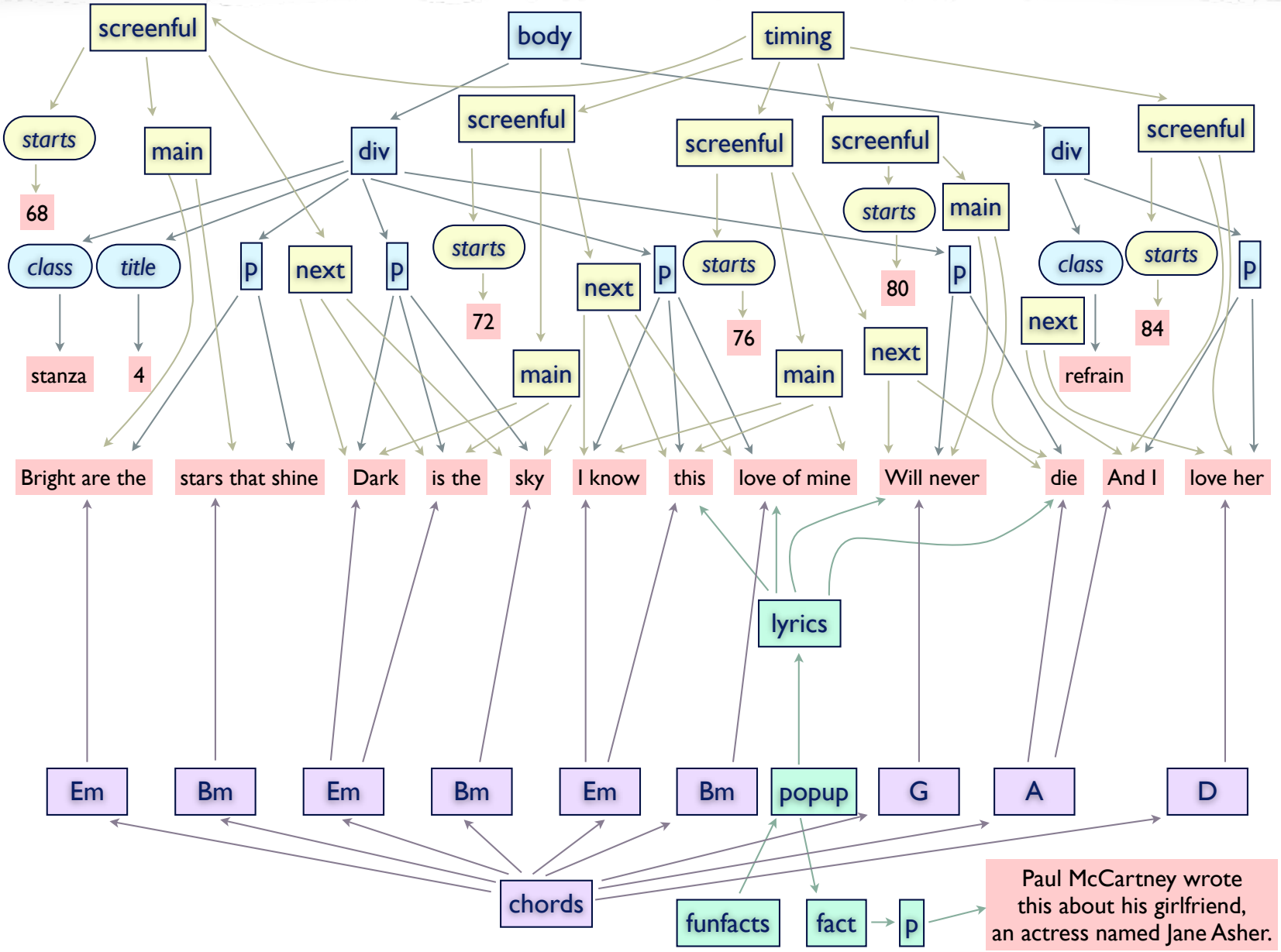
Do we really need the full GODDAG?

A case study: the Karaoke application

- Content to handle:
  - The lyrics of the song
  - The correct timing for two alternate lines appearing at the same time
  - Tne chords of the accompanying guitar (correctly associated to the lyrics)
  - Some fun facts to pop up here and there during the song.
- Problems:
  - Lyrics change according to gender of singer/addressee
  - Lines need to be displayed twice to provide lookahead
  - Chord changes do not match changes in lyrics lines
  - Fun facts pop up independently of the lyric lines

The Karaoke application

The resulting GODDAG



Summarizing

- We need to make fully descriptive assertions (**markup**) on the same content according to many different vocabularies
- We need a full GODDAG to manage all different types of hierarchies over the same content
- Full GODDAGs cannot be linearized via embedded markup

  ... BUT...
- Full GODDAGs **can** be linearized using external annotations
- This is called **standoff markup**. Old stuff. Well known.

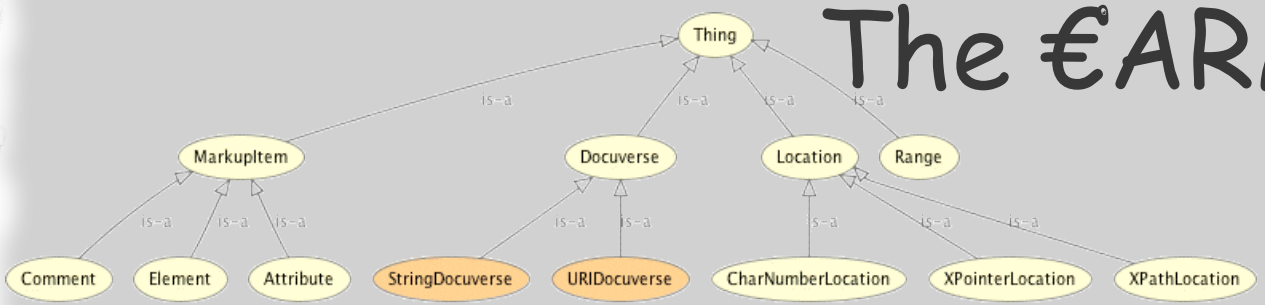
€ARMARK

€xtreme Annotational RDF Markup

- **Basic idea**: markup makes assertions over content. RDF makes assertions over resources. The concepts are close enough.
- Embedded assertions are limited (because of contiguity, document order, confusion of dominance and containment). RDF assertions are not.
- Markup works on text fragments of document, RDF works on resources (needs a URI, in fact).
- Markup intrinsically **makes** assertions over content. RDF needs to be told to, by way of ontologies.

€ARMARK is the OWL ontology that has markup as RDF triples.

The €ARMARK ontology



- Thing
  - Docuverse: *the textual content of the document. A string or an XML document.*
    - StringDocuverse: *for content that is stored with the €ARMARK document*
    - URIDocuverse: *for content that is referenced from other URIs*
  - Location: *a specific location in one of the docuverses...*
    - CharNumberLocation: *... as expressed as a numerical position, ...*
    - XPointerLocation: *... or as an XPointer, ...*
    - XPathLocation: *... or as an XPath.*
  - Range: *the content identified by two locations as boundaries. Locations must point to the same docuverse.*
  - MarkupItem: *artifacts generating markup, i.e., assertions on content or markup.*
    - Element
    - Attribute
    - Comment

"And I love her" in €ARMARK

```
e:lyrics
        a :URIDocuverse ; :has-uri "http://www.essepuntato.it/2009/01/andiloveher.txt"^^xsd:anyURI .
e:funfacts
         a :URIDocuverse ; :has-uri "http://www.songfacts.com/detail.php?id=43"^^xsd:anyURI .
e:attribute_values
         a :StringDocuverse ; :has-text "stanza - refrain - 4"^^xsd:string .
e:time_values
         a :StringDocuverse ; :has-text "68 - 72 - 76 – 80 - 84"^^xsd:string
e:r_refrain_1
         a :Range ; :begins e:location0-lyrics ; :ends e:location6-lyrics .
e:r_refrain_2
        a :Range ; :begins e:location6-lyrics ; :ends e:location14-lyrics .
e:location0-lyrics
        a :XPointerLocation ; :refers-to lyrics ; :at "xpointer(point(.0))"^^xsd:string .
e:location6-lyrics
        a :XPointerLocation ; :refers-to lyrics ; :at "xpointer(point(.6))"^^xsd:string .
e:location14-lyrics
        a :XPointerLocation ; :refers-to lyrics ; :at "xpointer(point(.14))"^^xsd:string .
e:refrain_p
        a :Element , [ a rdf:Seq ; rdf:_1 e:r_refrain_1 ; rdf:_2 e:r_refrain_2 ] ;
        :has-general-identifier "p"^^xsd:string .
```

Generating €ARMARK from XML

- Very easy.
- In fact, it is just one XSLT away.
- No, really, it IS easy.

Embedding €ARMARK as XML

- In case you only have ONE hierarchy, that's easy too.
  - For plain old simple XML, bidirectional conversion from and to €ARMARK is trivial.
- If you have multiple or self-overlapping hierarchies,
  - you must create an XML document with the main hierarchy, and make use of one or more "tricks"
    - segmentation
    - milestones
    - twin document
    - RDFa annotations
    - RDF/XML blocks.
  - Such conversion can be done automatically and parametrically.
  - For free you get also conversion from one trick to another
    - (e.g. convert segments into milestones)

Discussion
1) Why RDF/OWL?

- Because they exist
- Because they are standard
- Because they are W3C
- Because there are tools already
- Because they integrate markup with a lot of other things
  - (e.g., Linked Data)
- Because they allow a lot of new and old features
  - Single hierarchy validation
  - Multiple hierarchy validation
  - Co-constraint checking
  - Pattern-compliance evaluation
  - Inferences, inferences, inferences.

Discussion
2) Why standoff markup?

- Because no other approach can deal with full GODDAGs
- Because they are easy to generate and evaluate separately

  But they are fragile

- No they are not. They are only fragile if you foresee the modification of the content independently of the standoff markup
  - **The Renear/Wickett defense**: documents never change
  - **The Nelson defense**: document changes are the creation of new documents that point to the relevant bits of the old ones
  - **The Wiki/blog/permalink defense**: I will always have access to the original document regardless of the modifications, and I will be able to update my references
  - **The package defense**: if you put the text and the the standoff markup in the same package, as done by zip, tar, Open Office, MS Office, Java, SCORM, etc, you never are with the content and without the standoff markup.

Discussion
3) Why not using XPointers?

- But we **are** using XPointers. They are one way to do locators.
- We don't use XPointers in RDF URI (e.g., as ranges) because
  - the locations become opaque
  - We cannot make inferences on the actual locations
  - We cannot find out where overlap happens
- We prefer to take inspiration from XPointer, and re-implement locations and ranges directly in OWL, so that inferences on them can be performed.

Discussion
4) Extending GODDAGs

- However good GODDAGs are, they are not enough.
- They have problems in handling repetitions of direct dominance of the same nodes.
  - VERY TECHNICAL but problematic in some cases
- We extended GODDAGs into e-GODDAGs for handling multiple dominance arcs out of the same node to the same nodes.
  - For instance, in order to have only one refrain and refer to it several times in different positions of the same song.

Conclusions

- Full GODDAGs are useful. Necessary, even.
- No embedding markup can deal with them. We need standoff
- RDF/OWL are as good candidates as others to do standoff, and are better because
  - it is based on W3C standard
  - we already have tools
  - they serve for much more than standoff
  - they nicely integrate with Linked Data
- €ARMARK provides a reasonable and understandable ontology to do all sorts of markup, including full GODDAG support and beyond.

Lions lying down with lambs!!!

Thank you

More information at http://www.essepuntato.it/earmark

- Full ontologies
- Full "And I love you" €ARMARK example
- Other papers
- Tools