



The CEN Metalex Naming Convention

Fabio Vitali

University of Bologna



CEN Metalex

- CEN Metalex has been an international effort to create an interchange format between national XML formats for legislation.

<http://www.metalex.eu/>

- It created a conceptual model of documents, an abstract XML vocabulary, a concrete XML vocabulary, an OWL ontology, and a Naming Convention.
- The naming convention can be found in section 6 of the final document "Cen Workshop Agreement", at **<ftp://ftp.cen.eu/CEN/Sectors/List/ICT/CWAs/CWA15710-2010-Metalex2.pdf>**



Basic principles (1/2)

- A name is a list of feature values that uniquely identifies a bibliographic entity, and it is in principle the minimal list of feature values within the naming convention that identifies that bibliographic entity.
- Names *may* be either serialized into an IRI reference, or into metadata statements about the target of an IRI reference.
- A naming convention *must* systematically allow for id attribute identifiers to identify document fragments



Basic principles (2/2)

- The first three FRBR levels *must* be explicitly supported by the naming convention: works, expressions and manifestations *must* all have names and they *must* be different.
- The naming convention *must* explicitly take into consideration the complex structure of a document, and the interrelation between components (e.g., between the main body of a document and its attachments, and the attachments' attachments). In particular, each individual component (including the main body) *must* have an individual name.
- Serialization of names into IRI references *should* use IRI hierarchies whenever possible and appropriate; in particular, hierarchies should be used at least to separate the feature values of the work, expression, and manifestation levels, and of the document components.



Characteristics of a naming convention (1/2)



To allow for the discovery of IRI identifiers, names *must* be:

1. *Persistent* : names at all levels *must* maintain the same form over time regardless of the political, archival and technical events happened since their first generation;
2. *Global*: all relevant documents by all relevant bodies *must* be represented;
3. *Memorable*: names *should* be easy to write down, easy to remember, easy to correct if they were written down wrongly;



Characteristics of a naming convention (1/2)



4. *Meaningful*: names *should* mean something; It *should* be possible to make assumption about the kind, freshness and relevance of a citation by looking only at the document's name;
5. *Guessable across levels*: references to different levels of the same document *must* be similar; e.g., given a reference to an expression a user should be able to deduce the name of the work;
6. *Guessable across document classes*: references to different instances of the same document type *must* be similar;
7. *Guessable across document components*: references to different components of the same document at the same level *must* be similar.



Example

- Given a work-level reference to *act 136/05*, a user should be able to infer the work-level name of *act 76/06*.
- Given an expression-level reference to attachment A of act 136/2005, a user should be able to infer the expression-level name of attachment B of the same act.
- Etc.



Work-level features

1. The **country** emanating the document;
2. The **document type**;
3. Any specification of **document subtype**, if appropriate;
4. The **emanating actor**, who may be implicitly deducible by the **document type**;
5. The **promulgating actor**, who may be implicitly deducible either by the document type or by the emanating actor;
6. Any relevant **creation date** of the work;
7. Any relevant **number** or **disambiguating feature** of the work (possibly including titles).



Expression-level features

1. The **language(s)** associated (could be multiple)
2. The **validity date(s)** associated to actual content (could be multiple)
3. Any content **authoring information** to determine the authoritativeness of the text content. This is separate and independent of the authoring information relative to the metadata and markup, which are among the features of the manifestation.
4. Any **content-specification** date (as opposed to validity dates)



Manifestation-level features

1. The electronic data **format** chosen
2. The **markup authoring information** to determine the authoritativeness of the *markup and metadata*
3. Any relevant **markup-specific date**
4. Any additional **markup-related annotation** (e.g., the existence of multiple versions, of annotations, etc.)



Item-level features

1. The physical **location**
2. The **owner** of the physical location
3. Any additional **service-level annotations** (e.g., authentication, costs, authoritativeness, speed, etc.)



Requirements

- MetaLex documents *must* conform to a naming convention.
- The serialization into IRI reference *may* hide the feature names, which are still explicit property names in RDF.
- Any serialization into IRI references *must* make explicit the protocol and naming convention used.



The Akoma Ntoso Naming Convention

Fabio Vitali

University of Bologna



Akoma Ntoso and FRBR

Akoma Ntoso is heavily based on FRBR concepts:

- The Item is a *physical instance* of a document either as atoms (i.e., in paper form) or as bits (i.e., as specific files on a specific computer)
- The Manifestation is the abstraction of the *form* (including data format) in which different copies of the same document are rendered. Different items of the same manifestation are identical byte-by-byte or page-by-page.
- The Expression is the abstraction of a specific choice of content for a variant of the document. Each time-based, language-based or audience-based version of a document corresponds to a different expression.
- The Work (a "distinct intellectual or artistic creation") is the most general abstraction: it is what keeps together as "the same document" content choices that may appear very different (e.g., different, possibly contrasting versions of the same act, or translation in different languages, etc.)



Legal references in Akoma Ntoso

- Legal references are seldom to physical documents, but rather to abstract conceptualizations of documents: acts refer to acts, not to physical copies of acts. This is particularly true when documents change in time:
 - A static reference points to a version of a document frozen at a specific moment in time.
 - A dynamic reference was created to a document that existed in a given version, but is then expected to move to the most recent version of the document pointed to, or, more precisely, to the *most appropriate* version depending on the moment in time we are interested in examining.



AKN Naming Convention

Work-level URIs: a dynamic reference to a document in its general form

[resolver]/us/act/2010/124Stat119#sect12

Expression-level URIs: static reference to a document in a specific choice of content (e.g., language and time based):

[resolver]/us/act/2010/124Stat119/en@2010-01-24#sect12

Manifestation-level URIs: static reference to a document in a specific choice of format. In Akoma Ntoso this includes metadata and commentaries:

[resolver]/us/act/2010/124Stat119/en@2010-01-24/GPO.html#sect12

Item-level URIs: static reference to a file in a specific location somewhere in the network: Akoma Ntoso does NOT deal with Item-level URIs which are decided by the local authority:

<http://beta.congress.gov/111/plaws/publ148/PLAW-111publ148.html>

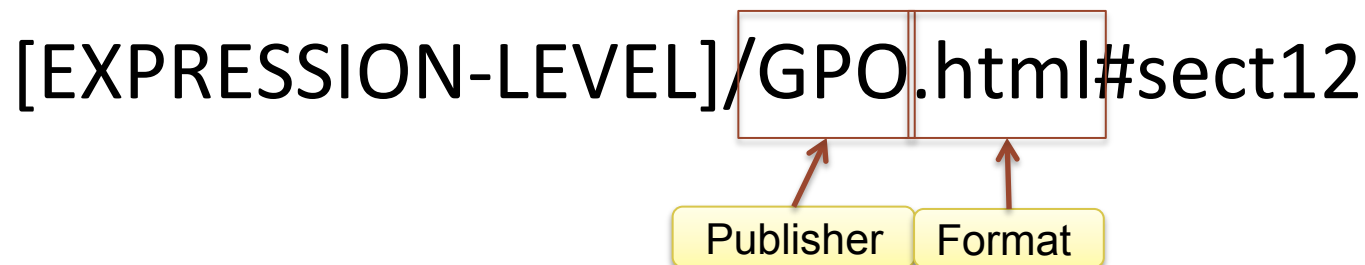
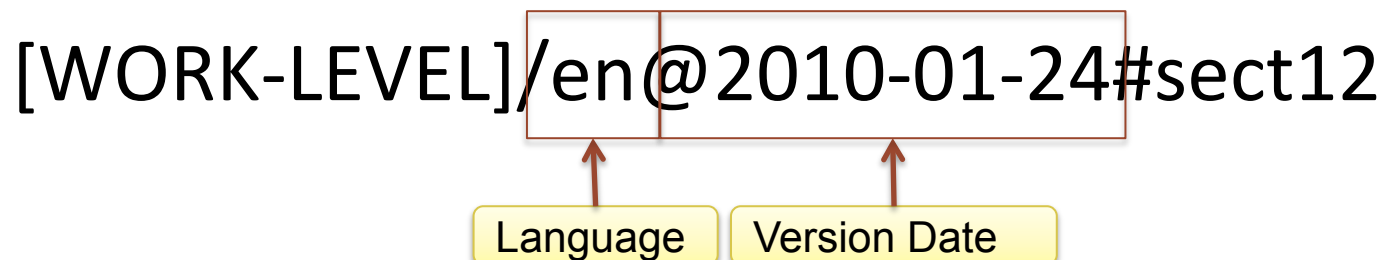
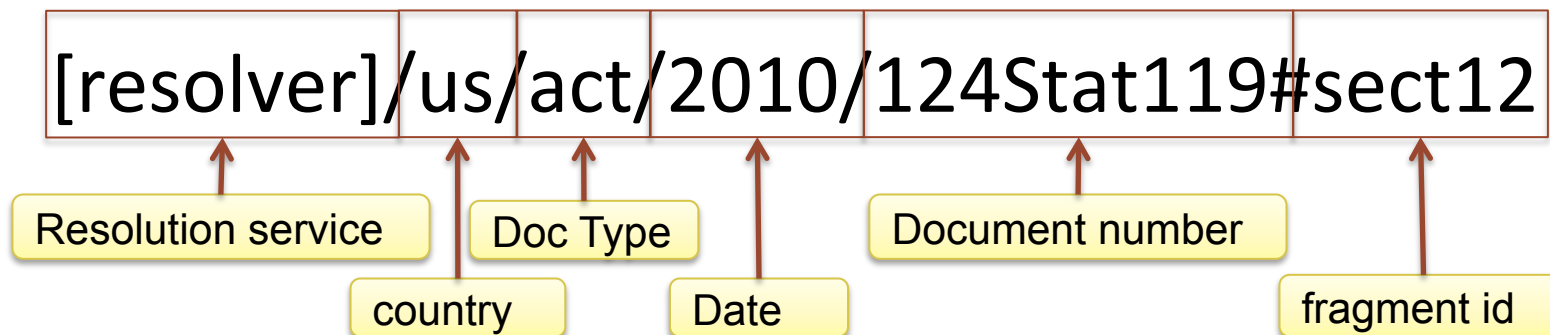


URI elements

- The parsing of an URI must guarantee unique parsing of elements and their unique association to properties, or features of the sought document.
- This is done by checking
 - the order of the pieces of the URI,
 - the vocabulary used in the values, and
 - the separators used.



URI parsing: the features





URI completion

- The **real** resolution of an AKN URIs to a physical URLs is done at the lowest level, the manifestation level.
- Therefore, before you can resolve an AKN URI, you first need to produce the complete Manifestation-level URI by adding missing features.
- This is called *URI completion*, and depends on defaults, user preferences and available versions and variants



URI completion

E.g.: given the Work-level URI

</us/act/2010/124Stat119#sect12>

generating the features:

- Country *us*
- Document type *act*
- Creation date *2010*
- Number *124Stat119*
- Fragment id *sect12*

the completion process may decide:

- Language *eng* user default
- View date *Today* application default
- Publisher *OPC* the only known?
- Format *PDF* the only available?



Dates, dates, dates...

- *Creation date* (Work level): the moment in time in which the document came into existence for the first time
- *Version date* (Expression level): the moment in which this specific selection of content from the work came into existence
- *View date/interval* (Expression level): the moment in time that is interesting for the user's query, which may be the date (or the interval) that is relevant for a case.
- *Manifestation date* (Manifestation level): the moment in time in which this specific representation (e.g., in PDF or XML) of the document has been created and the metadata generated

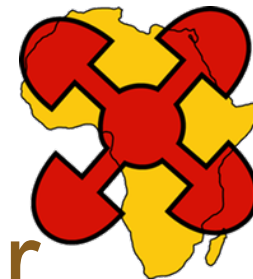


URI resolution

- The *resolution* is the process where the physical URL of the appropriate resource is identified.
- This is done **after the completion**, and it is therefore based on the features of a *manifestation*.
- This is done **on the features**, rather than the URI, so the actual original syntax of the request URI is not particularly relevant, as long as it produces the right set of features.
- Resolution can be done either through pattern matching, if the physical URL uses data from the features, or through one-to-one mapping, if it doesn't



The Akoma Ntoso URI resolver



- Available at <http://akresolver.cs.unibo.it/>
[URItoreolve]
- Documentation (incomplete) at <http://akresolver.cs.unibo.it/admin/documentation.html>



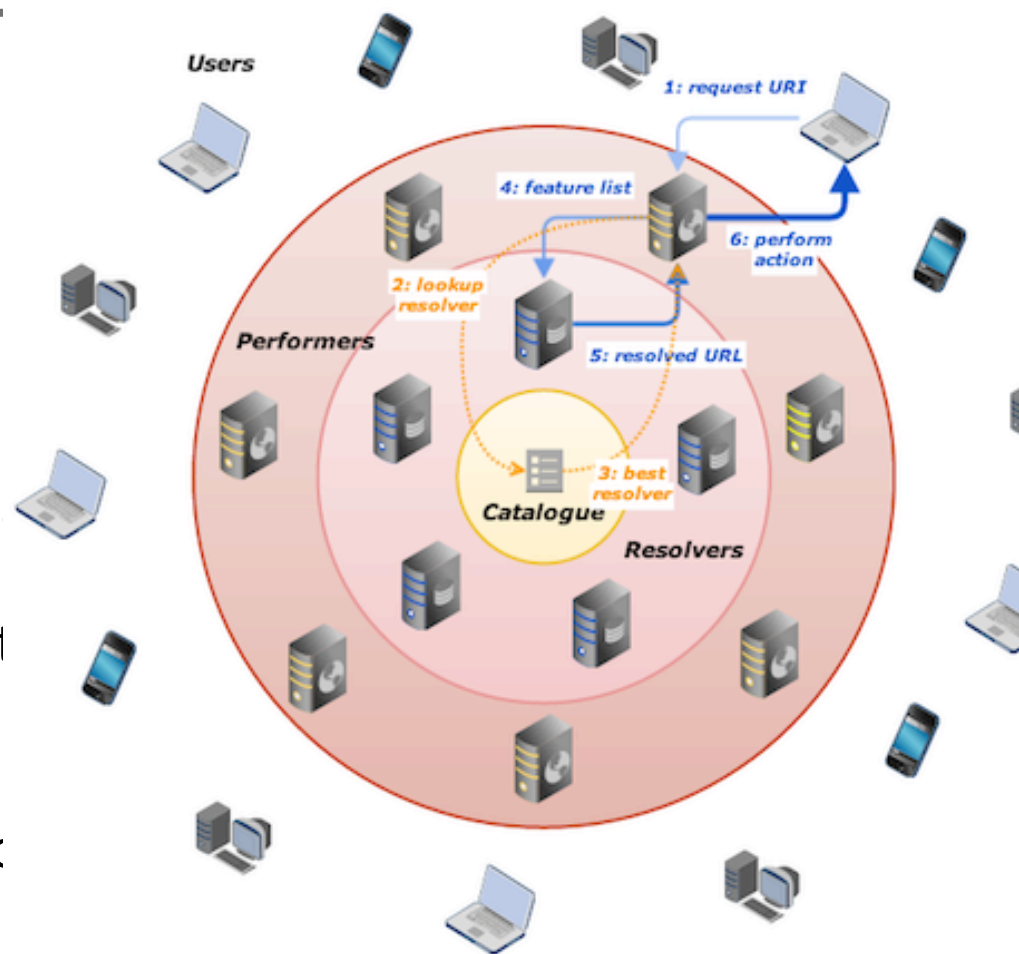
URI action

- There are three possible outputs to a resolution process. The Akoma Ntoso URI resolver does four
 1. *Resolve*: just return a data structure (e.g., JSON or XML providing the data of the resolved document (and further suggestions, maybe).
 2. *Redirect*: return an HTTP 301 code, whereby the browser automatically loads the correct document
 3. *Dereference*: the resolver returns the requested document AS IF it was the origin server
 4. *Wrap*: return a complex document that has features and data and links AND wraps the relevant document.



Architecture of the resolver

- There are two modules in the AKN resolver:
 - The *performer* receives the request, parses it creating the feature list, and calls the resolver
 - The *resolver* receives the feature list, and generates the list of best match and suggestions
 - The *performer* then carries out the requested action





URI resolution via templates

- For instance, in Switzerland the physical URLs of acts are as follows:

– <http://www.admin.ch/opc/de/official-compilation/2007/1.pdf>



- Thus from an AKN manifestation URI such as
 - t/ch/act/ru/2007/1/deu@opc.pdf
- a simple matching template can be created:
 - <http://www.admin.ch/{publisher}/{lang2}/official-compilation/{year}/{number}.{format}>



URI resolution via 1-1 mapping



- This is good because there is a method in physical URLs of Switzerland, but that is not always true.
- Since physical URLs are not under the control of the Akoma Ntoso resolver, this is the best we can do.
- Otherwise, we need to create a thorough mapping between Akoma Ntoso Manifestation URIs and physical URLs on a one-to-one basis.



Resolving and suggesting

- Resolution by all means is NOT a perfect process: there could be NO exact candidate for the manifestation you are looking for.
- The best you can do is find the best matching document. Resolving is therefore more akin to sorting documents according to relevance than to singling out one as the perfect match
- According to this point of view, therefore, there could be a small cohort of documents that might be relevant to the query and are just below the best matching. These could be returned as suggestions



The list of resolvers as the *federation* of performers



- As long as they use the same resolvers, individual performers can be easily interchanged.
- There is a list of available resolvers at a common address. All performers accessing this list will use the same resolvers and will return exactly the same results.
- In fact, it is possible to create specialized performers with different syntaxes always returning the same results.



Demo