MetaLex Naming Conventions and the Semantic Web

Alexander Boer
Leibniz Center for Law
University of Amsterdam
The Netherlands

Abstract. MetaLex XML is an interchange format, a lowest common denominator for other standards, intended not to replace jurisdiction-specific standards and vendor-specific formats in the publications process but to impose a standardized view on legal documents for the purposes of information exchange and interoperability in the context of software development.

To meet these requirements, MetaLex defines a mechanism for schema extension, for adding and extracting metadata, for cross referencing, for constructing compound documents by reference, and for implementation of a naming convention. This paper elaborates on the naming convention mechanism, explaining certain semantics points that are not made explicit in the official 2009 standard proposal, but that are nevertheless of great importance to Semantic Web developers dealing with law.

Keywords. MetaLex, XML, OWL, Semantic Web, Law

Introduction

MetaLex has been confirmed as a CEN/ISSS publicly available specification (CWA15710) in 2006, and is in the process of being updated in late 2009 by the MetaLex CEN/ISSS workshop on an Open XML Interchange Format for Legal and Legislative Resources. I have published about MetaLex, and specific MetaLex issues, in this forum, and closely related fora, several times during my activities as member and former chairman of the technical committee of this workshop [5,4,7,8,9,1,14].

MetaLex XML is an interchange format for legal and legislative resources, a lowest common denominator for other standards, intended not to replace jurisdiction-specific standards and vendor-specific formats in the publications process but to impose a standardized view on legal documents for the purposes of information exchange and interoperability in the context of software development.

To meet these requirements, MetaLex defines a mechanism for schema extension, for adding and extracting metadata, for cross referencing, for constructing compound documents by reference, for legally sound version management, and for implementation
of a naming convention. This paper elaborates on the new naming convention mechanism, explaining certain points not made explicit in the official 2009 standard proposal [3].

In MetaLex, names of legal and legislative bibliographic entities must be related to an identifying IRI reference\(^4\) and play a role in:

1. Self-identification of documents;
2. Citation of other documents;
3. Inclusion of document components.

MetaLex names must be persistent and cover all relevant legal and legislative bibliographic entities, and they must be memorizable and meaningful.

The naming convention requirements are intended to cover common, related standards such as OpenURL – which is for instance successfully used for the identification of academic publications – and representative existing IRI reference-based or metadata-based naming mechanisms for sources of law such as the Juriconnect identifiers of the Netherlands, Norme in Rete (NIR) identifiers in Italy, The Pan-African Akoma Ntoso identifiers, the future European Case Law Identifiers (ECLI), and those in use in the Single Legislation Service\(^5\) (SLS) of the UK.

There are few technical limitations on identifiers acceptable to the new MetaLex standard. The following are examples of IRI reference-based names of naming schemes involved in the MetaLex standardization in some way:


All examples order values from generic to specific, different separators are used, and all except the first are intended to be used as relative references, relative to an implicit base IRI. Other naming schemes, for instance the ones of the Netherlands, are based on key-value pairs, either encoded in proprietary XML structures, in common metadata syntaxes like RDF or HTML’s meta tag, or (optionally) in IRI references as key=value pairs separated by &.

The biggest impediment to MetaLex compliance is conceptual: the naming scheme should be explicit and clear on distinguishing works, expressions, and manifestations. Their names should be different.

MetaLex strictly distinguishes the source of law as a published work from its set of expressions over time, and the expression from its various manifestations, and the items as physical instances of a manifestation, as recommended by the Functional Requirements for Bibliographic Records (FRBR; [12]) and explained in more detail in [9,8]. MetaLex extends the FRBR ontology with a detailed but jurisdiction-independent model of the lifecycle of sources of law.

Mere permanent URLs pointing to digital documents are not sufficient, as section 3 will explain. Moreover, within each of the distinguished levels, each name should be different, and no name should be a descriptive subset of the name of another entity within

\(^4\)I.e. the internationalized resource identifier as defined by RFC 3987, which replaces (and syntactically subsumes) the more commonly known older URL.


that level. Most existing naming schemes fail these criteria, but many do so in ways that are easily resolved on a technical level.

Names must also be also meaningful with respect to these ontological levels. Names should be guessable to some extent, across ontological levels, across document types, and across document components. Given a reference to an expression, for instance, the user should for instance be able to infer the name of the associated work. Also references to instances of the same document type on the same ontological level, and references to different components of the same document at the same ontological level, should use the same pattern.

In this paper, a number of issues with naming convention implementation are discussed. Section 1 gives a general overview of MetaLex names, the MetaLex naming convention mechanism, and its relation with the Web Ontology Language\(^6\) (OWL). In section 2 the OWL semantics of names is explained. Section 3 then refines the notion of a MetaLex name to include the FRBR ontological layers mentioned in this introduction, and MetaLex event descriptions as a conceptual coatrack for descriptive metadata. At this point, in section 4, also the relevance of the naming convention for the Computer Science & Law community is explicitly addressed.

The discussion, in section 5, gives some examples of non-conformance issues in existing naming schemes, and proposes some simple solutions.

This paper leaves some naming requirements undiscussed. There is for instance also a list of metadata that the proposal strongly recommends as constituents of MetaLex names (including such things as the country code, emanating actor, electronic data format, date of publication of the work, date of commencement of the modification producing an expression, etc). This paper therefore does not replace the specification as a guide to implementing a naming convention. This paper does however add some additional detail that will help to avoid or solve common problems in implementation.

1. The MetaLex Naming Convention Mechanism

The MetaLex XML standard requires the association of IRI and of descriptive names to bibliographic entities. The IRI is a supposedly unique identifier. Identifiers are typically based on an injective function; They identify only one entity, but they are not necessarily the only identifier of that entity. A name typically not only identifies just one entity\(^7\), but it is also, within the namespace described by a naming convention, the only identifier of the entity.

MetaLex XML instances must name themselves and identify themselves and their parts by way of IRI. Citations of other bibliographic entities in MetaLex XML instances should also supply both a IRI and a descriptive name of the target bibliographic entity. Also inclusions of bibliographic components by reference in MetaLex XML instances should supply both a IRI and a descriptive name of the target bibliographic component.

There are two methods for supplying name information. The name can be embedded in the IRI reference – this is the transparent IRI reference method – or it can be supplied in the form of RDF metadata triples about an (opaque) IRI.

\(^6\)http://www.w3.org/2004/OWL/

\(^7\)In the real world it may even fail to do that.
The name is essentially a list – an ordered set – of descriptive predicate-object pairs that narrow down the set of described entities to just one individual entity, specified in conformance to a naming convention. Which naming convention is adhered to is declared in the MetaLex XML instance.

The ordering is necessary in transparent IRI references if the predicates are not encoded into the IRI; The implied predicate is derived from the position of the object value. In metadata encodings the ordering is typically not relevant.

A major difference between the transparent IRI reference and the explicitly encoded set of naming predicate-object pairs is that the transparent IRI reference does not allow for incomplete sets of object values – sets that narrow down the set of possible referred entities, but not to a single, unique individual.

Another difference is that the explicit specification of a set of naming predicate-object pairs can be attached to any IRI. This can be conceived of as both an advantage and a disadvantage of explicit naming metadata. If only one transparent IRI is used, no merging of individuals ever needs to take place.

A third difference is that the transparent IRI reference often does not convey information about the datatype of object values (an IRI, a date, a string, etc.). In metadata this would typically (but not necessarily) be clear.

The conforming naming convention must meet certain general criteria for naming conventions, it must specify a GRDDL translator that extracts RDF statements, and it must specify transparent IRI references that contain the same information. GRDDL is an acronym for Gleaning Resource Descriptions from Dialects of Languages. The GRDDL specification introduces XML markup for declaring that an XML document includes data compatible with RDF and for linking to an algorithm (typically an XSLT transformation) for extracting this RDF data from the document. RDF is the common syntax for OWL axioms.

In this paper the criteria, the GRDDL translation, and the OWL interpretation of naming metadata are discussed.

### 1.1. Extracting OWL2 DL from MetaLex Names

Names are serialized into a transparent IRI reference, or into metadata statements about the target of an (opaque) IRI reference. It is important to understand the difference between the IRI and an IRI reference, but this subject is beyond the scope of this paper. A naming convention must implement both methods of serializing names. However, the same method of serializing names – either metadata or serialization into the IRI reference – must be used in all XML documents conforming to the naming convention.

Since MetaLex allows multiple ways to encode metadata into documents, a naming convention is associated to one or more specific ways of encoding metadata into MetaLex documents. The naming convention must be implemented into one or more GRDDL translators used for extracting metadata from MetaLex documents. If N(i) is a set of naming metadata statements about entity i, and n is an IRI reference that encodes the same name as N(i), the translator:

1. extracts naming metadata statements N(n) from naming information serialized into the transparent IRI reference n,

---

8[http://www.w3.org/TR/grddl/](http://www.w3.org/TR/grddl/)

2. extracts the transparent IRI reference n from metadata statements N(i) about some opaque IRI reference i, and
3. Adds OWL2 axioms entailing that the only instance that has type N is identified by IRI reference n (i.e. N is equivalent to \{n\}, hence if N(i) then i=n etc.).

Conformance to a naming convention must guarantee the availability of an OWL2 file for each MetaLex XML file containing both metadata and the canonical transparent IRI references.

For Semantic Web processing the information serialized into the transparent IRI reference should have no direct significance, although the fact that the IRI references are more or less meaningful, memorizable, and guessable is in practice relevant even for the Semantic Web. More often than not RDF files include no human readable labels for entities referred to by IRI references, even though Semantic Web standards do make such labels possible and do not require meaningful IRI or qnames10.

MetaLex-aware name processing may be based on the transparent IRI reference or on the RDF metadata, as preferred.

2. OWL Naming Axioms

The OWL2 file extracted from the MetaLex XML file contains naming axioms. The developer of the translation is free to implement this requirement however he wants in OWL2, as long as the following (as usual in description logic syntax [11]) is entailed.

Let $/v_1/v_2/\ldots/v_n$ be a transparent IRI reference to the entity described by $\exists P_1.\{v_1\} \cap \exists P_2.\{v_2\} \cap \ldots \cap \exists P_n.\{v_n\}$. A naming axiom has the following form:

$$\{v_1/v_2/\ldots/v_n\} \equiv \exists P_1.\{v_1\} \cap \exists P_2.\{v_2\} \cap \ldots \cap \exists P_n.\{v_n\}$$

This axiom which states that there is only one individual that meets a naming description is essential to the concept of a name.

Axioms of this form, which serve a rather trivial purpose, turn out to be especially hard for automated reasoning in OWL for two reasons:

1. Nominal constructs break the traditional TBox-ABox separation. In the presence of nominals, ABox assertions can affect concept satisfiability and TBox classification. Some common ABox reasoning optimizations cannot be applied. As a consequence, the computational cost of inserting new individuals into the knowledge base is high.
2. General Concept Inclusion Axioms (GCIs11) using nominal constructs are not handled by some standard absorption optimizations. GCIs are generally hard to reason with because of the high degree of non-determinism they introduce in tableaux expansion. Without the use of optimizations reasoning may be prohibitively hard.

10 The remainder of the URI when the namespace prefix is subtracted.
11 I.e. subclass axioms with a complex concept expression on the left hand side.
[13] conclude that without special optimizations reasoning with nominals may be entirely unpractical even if only a handful of nominal constructs are present. [13] propose such optimizations, including a nominal GCI absorption technique, for the Pellet reasoner.

The naming axioms will not generally be relevant to practical problem solving. It therefore makes sense to use them only in this form for validation of collections of bibliographic metadata and to leave them out in production environments, or to represent them partially for a task-specific purpose or syntactic species of OWL2.

In production environments we often do not need more information than that individual \( /v_1/v_2/\ldots/v_n \) is an instance of its description, or:

\[
/v_1/v_2/\ldots/v_n : \exists P_1.\{v_1\} \cap \exists P_2.\{v_2\} \cap \ldots \cap \exists P_n.\{v_n\}
\]

If the reasoner used does not support nominal constructs at all, representation in the form of pseudo nominals is possible. Pseudo nominals do not break the traditional TBox-ABox separation and typically speed up consistency and classification. As however pointed out in [13], pseudo nominals are not only unsound, but may in fact degrade realization performance by an order of magnitude in comparison with an implementation that optimizes for nominals. In production environments this will often be too high a price to pay. Pseudo nominals are therefore hardly worth recommending.

### 3. Stratification of Bibliographic Entities

Thus far I have focused on the relation between the MetaLex XML document and its names and identifiers. MetaLex however distinguishes identity of documents on the item, manifestation, expression, and work level, in accordance with the Functional Requirements for Bibliographic Identifiers (FRBR; [12]). The source of law is identified at these levels with different names, although a systematic relationship between names on different ontological levels should exist.

A MetaLex XML document is a standard manifestation of a bibliographic expression of a source of law. Editing the MetaLex XML markup and metadata of the XML document changes the manifestation of an expression. Changing the marked up text changes the expression embodied by the manifestation. Copying an example of the MetaLex XML document creates a new item. The work, as the result of an original act of bibliographic creation, realized by one or more expressions, does not change.

Each bibliographic item exemplifies exactly one manifestation that embodies exactly one expression that realizes exactly one work.

Because all these mappings are functional, i.e. unambiguously map to one entity, item identity can be, and often is in natural language, used as an indirect identifier of the other objects. This is similar to how email addresses have a functional mapping to persons and can be used as an indirect identifier of persons. One can refer to a work by referring to its initial expression in a context where a reference to a work is expected.

The inverse of these relations is often not a function. One can think of the work as an abstraction of one or many expressions, the expression as an abstraction of one or many manifestations, the manifestation as an abstraction of one or many reproducible
items. The manifestation, expression, and work are intentional objects whose existence is conditioned to the existence of at least one item, manifestation, expression, respectively. There is normally speaking no such thing as an expression that is not embodied, a work that is not realized, etc.

Metadata about an item cannot be encoded into an item; it can only be stored outside it. Hyperlinks, as used on the web, identify items by default. Self-identification through IRI generally identifies the manifestation.

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. Every conformant MetaLex document must declare an IRI of the document manifestation it exemplifies using the xml:base attribute, allowing it to include metadata referring to itself. Note that implicit XML base resolution results in an item identifier instead of a manifestation identifier; the mere act of copying would change identity.

There are two distinct methods for uncovering the relationship between manifestation, expression, and work:

1. An explicit encoding of the relation between these IRIs as metadata statements about the IRIs, starting with the manifestation IRI reference.
2. A serialization into IRI references that establishes a systematic hierarchical relationship between transparent IRI references, which allows one to derive the IRI of one from the other.

A 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). A naming convention must also allow for id values to identify document fragments, at all levels.

Every citation in a conforming MetaLex document must supply an IRI reference to and a name, conforming to a naming convention, of the target document, at the work or expression level. Legislation usually cites works, while court decisions cite expressions. Any part, except the top level container, of a standard MetaLex XML document may be implemented as an inclusion reference to an external manifestation component; every inclusion in a conforming MetaLex document should supply a name, conforming to a naming convention, of the target manifestation component.

Let’s examine what the requirements above mean in practice. The distinction between works, expressions, manifestations, and items is based on the modeling device of ontological stratification [10]. We are not dealing with one list of predicate-object pairs \{\{(p_1, v_1), (p_2, v_2), \ldots, (p_n, v_n)\}\}, but three such lists. Each of the predicate object-pairs belongs to either the manifestation \{\{(pm_1, vm_1), (pm_2, vm_2), \ldots, (pm_n, vm_n)\}\} or to the expression \{\{(pe_1, ve_1), (pe_2, ve_2), \ldots, (pe_n, ve_n)\}\} or to the work \{\{(pw_1, vw_1), (pw_2, vw_2), \ldots, (pw_n, vw_n)\}\}.

The required systematic hierarchical relationship usually involves an ordering work/expression/manifestation in the IRI reference. Do note that that the use of / is this document is merely a suggestion; URNs will typically use the : separator, and others are permitted. The transparent IRI of a manifestation /vw_1/vw_2/\ldots/vw_n/ve_1/ve_2/\ldots/ve_n/vm_1/vm_2/\ldots/vm_n includes the name of the expression and work, and the transparent IRI /vw_1/vw_2/\ldots/vw_n/ve_1/ve_2/\ldots/ve_n of an expression includes the name of the work /vw_1/vw_2/\ldots/vw_n.

The OWL2 interpretation of a name must assign an IRI reference to the work, expression, and manifestation, link these using the properties metalex-owl:embodies and
metalex-owl:realizes, and attach the predicate-object pairs to the right ontological level. In addition it should interpret the predicate-object pairs, wherever reasonable, as applying to the event that resulted (metalex-owl:resultOf) in the bibliographic entity.

Let for instance $i_e$ be $/vw_1/vw_2/.../vw_n/ve_1/ve_2/.../ve_n$, the IRI reference of an expression, and $i_w$ be $/vw_1/vw_2/.../vw_n$, the work it realizes. The OWL naming axiom for the expression (1) and work (2) should have the following structure:

1. \{i_e\} \equiv \exists\text{metalex:realizes.} \{i_w\} \cap \exists\text{metalex:resultOf.} \\
   (\exists P e_1.\{ve_1\} \cap \exists P e_2.\{ve_2\} \cap ... \cap \exists P e_n.\{ve_n\})

2. \{i_w\} \equiv \exists\text{metalex:resultOf.} (\exists P w_1.\{vw_1\} \cap \exists P w_2.\{vw_2\} \cap ... \cap \exists P w_n.\{vw_n\})

A worked out example, based on Akoma Ntoso style relative IRI references, is found in section 4.4 of [3], and will not be repeated here.

4. Relevance to Knowledge Representation

MetaLex names are relevant to Computer Science & Law for several reasons.

Firstly, the problem of determining whether legislation is applicable to some case at hand is well recognized in this community, and depends to a large extent on access to descriptive metadata of legislation typically found in the MetaLex naming schemes. This problem is also complex enough (when for instance involving delayed or retroactive applicability, ex tunc versions, etc) to constitute a relevant legal problem domain by itself. In the project reported on in for instance [7] it is indeed the legal problem solving domain at issue. The MetaLex effort addresses aspects of this problem, and makes metadata of interest accessible in a standard format.

Secondly, and more to the point, the distinction between works and expressions in references is also of great importance to knowledge base maintenance. When a source of law is represented in some logical language it is obviously the expression that is being represented. When legislation however refers to other legislation, it refers to a work. A citation (text fragment) $w$ applies to (concept) $C$ should therefore be read as each legal rule that is represented by an expression-level text fragment that realizes work fragment $w$ applies to $C$. The extension of the concept text fragment that realizes $w$ will change over time, and needs to be established dynamically at least for the purposes of impact analysis when legislation changes.

It is only the distinction enforced by MetaLex that makes it possible to express something like this in OWL. This example was expressed in the MetaLex-aware and OWL-based knowledge representation proposed in [2], which will be developed further in the Agile project [6] in order to improve the robustness of traceability from knowledge representation to sources.

Thirdly, the MetaLex preference for interpreting metadata as applying to events happening to the sources of law instead of applying to sources of law directly has been justified with several different arguments in [1], some of which appeal to the pragmatics of knowledge base maintenance and others to the fidelity of the representation of the knowledge domain. The natural coherence between for instance between emanating actor, promulgation data, and promulgation channel information (e.g. state gazette bibliographic information) is apparent to all: all are participants in the publication (promulga-
tion) event. In formal legislation, there is for instance a natural coherence between the old consolidation, the new consolidation, the modifying legislation, the modifying authority, and the modification date. The modification event, if identified explicitly, links together three different but related resources, and interesting metadata about them.

5. Discussion

Naming schemes generally fail the MetaLex criteria in one and the same way: The feature-value pairs used to identify, identify more than one bibliographic entity. The following identification problems commonly occur in naming schemes:

1. The name of an item is equal to that of a manifestation, or the manifestation claims to identify itself as an item;
2. The name of a manifestation is equal to that of an expression or does a bad job of uniquely identifying the manifestation among other manifestations of that expression;
3. The name of the work and the initial expression are the same;
4. The name of a work, expression, or manifestation is a proper subset of the name of another entity within the same level.

For instance, in the SLS the identifier /ukpga/1985/67 names the current in force expression, while /ukpga/1985/67/2003-04-01 names a specific version. An alternative case sometimes encountered is that the shorter identifier identifies the initial expression. Another common problem would be to identify both work and expression currently in force with /ukpga/1985/67; The SLS uses in this case /id/ukpga/1985/67 to distinguish the work, but this is also problematic because it doesn’t add feature values to describe the expression, but subtracts one. Other schemes have similar issues.

Because the context of use of an identifier often makes clear what the appropriate entity should be (inclusion by reference refers to manifestations or items, citation to expressions or works, locators refer to items, etc.), ambiguous names do not necessarily cause practical implementation problems, but they are hard to reconcile with OWL semantics, or any other knowledge representation language based on the open world assumption. The extension of concept \( \exists P_1.\{v_1\} \cap \exists P_2.\{v_2\} \cap \ldots \cap \exists P_n.\{v_n\} \) must be a superset of the extension of \( \exists P_1.\{v_1\} \cap \exists P_2.\{v_2\} \cap \ldots \cap \exists P_n.\{v_n\} \cap \exists P_{n+1}.\{v_{n+1}\} \).

Considering the previous example, the use of an identifier for the current expression is beyond the scope of MetaLex: clearly a reference to a work is intended where an identifier of the current version would be used. Resolution to a current version is merely a useful default behaviour. Because the compliance criterium is only the ability to transform the IRI reference into a set of descriptive metadata, such shortcomings are not always a major problem if one understands how to classify IRI references in context into the right bibliographic level. The GRDDL translator may itself embed knowledge, and may add additional predicate-object pairs to distinguish names on different levels (e.g. adding /initial). The GRDDL translator may also extract a work name from a reference to an expression currently in force.

More difficult is the problem of making sure that names do a good job at uniquely identifying entities within their level, particularly for the manifestation (item 2 in the list).
There is no way to do that for certain without reference to a repository or repositories of used names. MetaLex therefore requires that MetaLex XML instances must include a description of the creation of the manifestation, and at least the (institutional or human) actor involved and the date at which it happened. The responsibility of adding further disambiguating metadata is to the creating actor, who may be presumed to know what MetaLex XML instances he created on that date.

References


