Meta-Metadata: A Semantic Architecture for Multimedia Metadata Definition, Extraction, and Presentation

Andruid Kerne, Sashikanth Damaraju, Bharat Kumar, Andrew Webb
Interface Ecology Lab / Department of Computer Science
Texas A&M University
{andruid, damaraju, bharat, awebb}@cs.tamu.edu

ABSTRACT

We present an overview of an extensible architecture for defining and operating on multimedia metadata. M*eta-metadata* is used to specify the types of metadata for a particular information source. Meta-metadata semantics drive information extraction, information visualization, contextual metadata presentation, editing, and interaction. A pipeline automatically binds meta-metadata XML to strongly typed object instances, compiles metadata subclass definitions from meta-metadata instances, and binds metadata XML to metadata subclass instances. We show how meta-metadata conjoined with metadata drives information visualization in mixed-initiative information composition.

1. ARCHITECTURE

The form and structure of the metadata of multimedia elements varies, depending on their sources and the required operations. We need flexible, easy to author structures for representing how metadata will be derived, used, and represented. To define custom metadata structures in a general and extensible fashion, we introduce *meta-metadata* - authored XML documents, each of which defines the structure, extraction and representation of a set of metadata types. Alterations to the semantic structure, extraction rules, and visual representation are performed without changes to application source code.

The meta-metadata architecture has four main stages (Figure 1), each of which is enabled by the `ecologylab.xml` binding framework [1]. First, `meta_metadata` XML is translated to `MetaMetadata` Java objects. These are translated to `Metadata` class declarations, in Java, which define the structure of the metadata. Like those for meta-metadata, these classes are annotated with `ecologylab.xml`'s metalanguage. Subsequently, the extraction phase utilizes rules specified in meta-metadata for extraction of metadata from particular document sources, creating instances of the classes generated in step one. Next, in the visualization stage, the meta-metadata for each metadata field drives presentation and editing in the mixed-initiative composition space, using instantiated objects of step 2. Finally, the same Java declarations are used to serialize instances of metadata objects.

1.1 Translating Meta-Metadata > Metadata

A language of Java `MetaMetadata` class declarations has been developed as the basis of a vocabulary for expressing the representation of structural of metadata, such as the types of fields, their names, and relationships, and functional components, such as how to extract instances from an HTML or XML document and how to present them to users. `MetaMetadata` instances are one-to-one with `Metadata` class subtypes. Each `MetaMetadata` instance consists of a set of `MetaMetadataFields`, each of which includes type information for the equivalent `MetadataField` subtype. It is represented in XML by an equivalent `meta_metadata_field`. Each `MetaMetadata` instance is, in turn, translated again, to form corresponding `Metadata` class subtype definition. Again, the generated `Metadata` classes, themselves, utilize the `ecologylab.xml` framework, so that instances of `metadata` XML can be translated into Java objects in programs that actually operate on instantiated metadata.

1.2 Structured Metadata Extraction

Metadata is extracted from template information sources, using meta-metadata instances. The appropriate meta-metadata instance is selected from a repository, according to a URL expression for each template source. Extraction of strongly typed metadata from the HTML and XML source documents is directed by a `meta_metadata_field` attribute containing an XPath [3] expression. The structure of the instantiated strongly typed

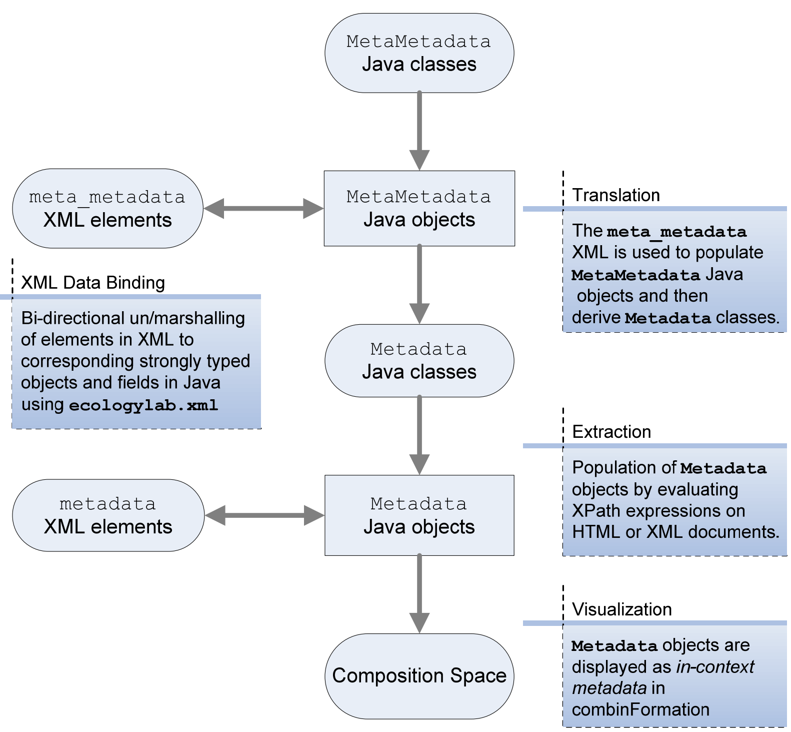

Fig. 1. Stages of the Meta-Metadata architecture.

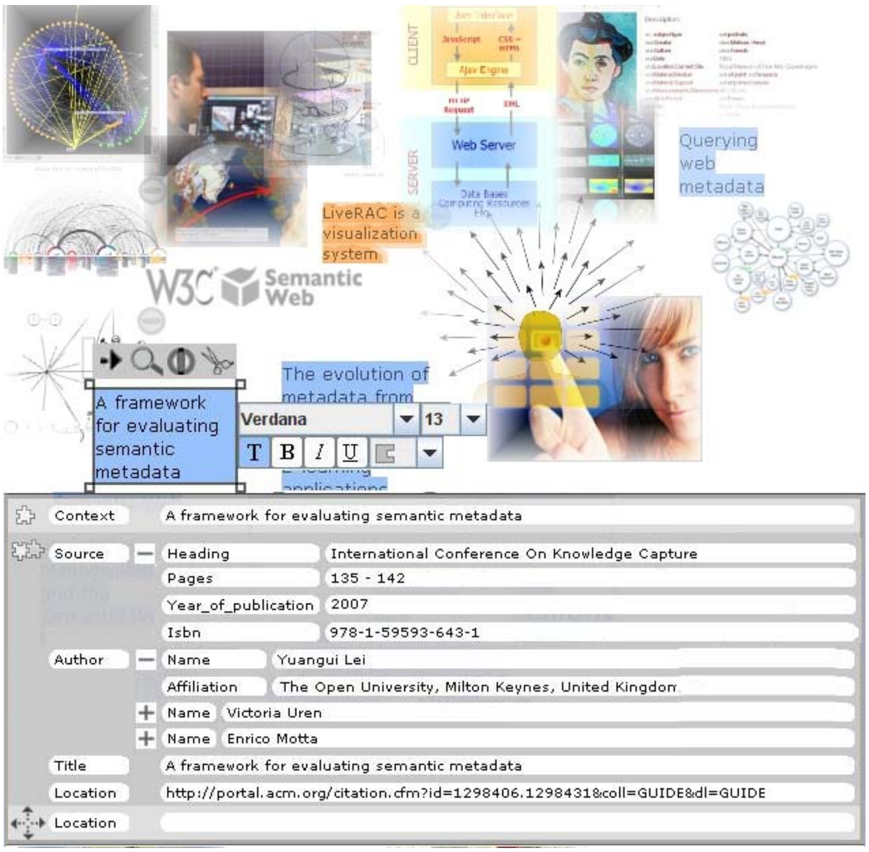

Fig. 2. In combinFormation's composition space for authoring personal collections, the user brushes a surrogate, extracted from a PDF document in the ACM Portal, to show in-context metadata details-on-demand.

metadata classes, along with their equivalent XML representation, is specified within the meta-metadata.

1.3 Composition Space Visualization

Mixed-initiative information composition integrates searching, browsing, and organizing information to support learning and thinking **Error! Reference source not found.**. Composition, an alternative to lists, represents a collection as a whole, using visual design principles. Exposure to diverse relevant information in visual forms optimizes human cognition and helps participants overcome fixation on preconceptions, stimulating innovation.

When the user brushes a surrogate, with mouse rollover, *in-context metadata details-on-demand* are displayed above or below the surrogate, showing the metadata corresponding to the surrogate, its document source, and its hyperlink. This fluid interface is easily activated to promote exploration of details, while minimizing distraction from other collection authoring activities, such as organizing surrogates. It is also used in conjunction with the in-context slider to enable fluid interest expression (relevance feedback) through facets of the metadata, such as author, year, conference, or keywords.

The display of each metadata instance in the in-context details-on-demand is controlled by its declaration in the meta-metadata. The visual nesting of the metadata represents the nesting described by the meta-metadata. The architecture allows the change of visual representation of the metadata by simply altering the meta-metadata. Each extracted metadata field may have various actions assigned to it. For example, metadata fields such as URLs corresponding to an author's page on the ACM portal represent valuable semantic relationships. Instead of displaying such URLs to the user as text, the architecture allows a metadata field to be declared within the meta-metadata as associated with another metadata field, to be represented to the user in the visualization as a navigational affordance. In this example, clicking on an author field would navigate the participant to the author's page.

Figure 2 shows one state of a composition space created in a usage scenario, in which the task was to collect prior work on the topic of information visualization for the semantic web. combinFormation was seeded with the search queries, "information visualization" and "semantic web". The figure shows a text surrogate with in-context metadata details-on-demand amidst an emerging composition.

1.4 XML Information Binding

The use of `ecologylab.xml` as a foundation for the meta-metadata architecture enables seamless translation of live metadata objects in the application to XML. Like other steps, this translation into formatted, nested XML is driven by annotation metalanguage declarations in Java class declarations; in this case, of the metadata objects. This simple, effortless bi-directional XML-Java translation enables the application to easily communicate structured metadata efficiently across a network to other applications, to and from data services, to store it persistently, and to perform retrieval from persistent storage.

2. CONCLUSION

We have presented an architecture for consistently representing, extracting, and presenting multimedia metadata from a particular source. The component-based architecture supports meta-metadata authoring, while minimizing the programming necessary for information visualization and interactive service applications to utilize diverse semantics. Future work will develop full support in combinFormation for citation chaining through facets. We will also offer an open repository of meta-metadata, for use by heterogeneous applications, and authoring tools for community.

3. REFERENCES

[1] Kerne, A., Koh, E., Smith, S.M., Choi, H., Webb, A., Dworaczyk, B., combinFormation: Mixed-Initiative Composition of Image and Text Surrogates Promotes Information Discovery, *ACM Transactions on Information Systems*, in press (Jan 2009).

[2] Kerne, A., Toups, Z.O., Dworaczyk, B., Kumar, B., Webb, A., Concise Object-Oriented XML Binding Framework, *Proc ACM Document Engineering 2008*,

[3] XML Path Language 2.0. http://www.w3.org/TR/xpath20/. 2007.