A Survey of Techniques for Achieving Metadata Interoperability

BERNHARD HASLHOFER and WOLFGANG KLAS

University of Vienna

Achieving uniform access to media objects in heterogeneous media repositories requires dealing with the problem of metadata interoperability. Currently there exist many interoperability techniques, with quite varying potential for resolving the structural and semantic heterogeneities that can exist between metadata stored in distinct repositories. Besides giving a general overview of the field of metadata interoperability, we provide a categorization of existing interoperability techniques, describe their characteristics, and compare their quality by analyzing their potential for resolving various types of heterogeneities. Based on our work, domain experts and technicians get an overview and categorization of existing metadata interoperability techniques and can select the appropriate approach for their specific metadata integration scenarios. Our analysis explicitly shows that metadata mapping is the appropriate technique in integration scenarios where an agreement on a certain metadata standard is not possible.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*Metadata, Standards, Interoperability*; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems

General Terms: Design, Management, Standardization

Additional Key Words and Phrases: Metadata standards, interoperability, mapping

ACM Reference Format:

Haslhofer, B. and Klas, W. 2010. A survey of techniques for achieving metadata interoperability. ACM Comput. Surv. 42, 2, Article 7 (February 2010), 37 pages.
DOI = 10.1145/1667062.1667064. http://doi.acm.org/10.1145/1667062.1667064

1. INTRODUCTION

Metadata is machine processable data that describes resources, digital or non-digital. While the availability of metadata, as a key for efficient management of such resources in institutional media repositories (e.g., Sheth and Klas [1998]), has been widely required and supported in highly standardized ways, *metadata interoperability*, as a prerequisite for uniform access to media objects in *multiple* autonomous and heterogeneous information systems, calls for further investigation. Since it is not given per default, it must first be established by domain experts before uniform access can be achieved.

Author's address: B. Haslhofer, Department of Distributed and Multimedia Systems, University of Vienna, Liebiggasse 4/3-4, A-1010 Vienna, Austria; email: bernhard.haslhofer@univie.ac.at.

DOI 10.1145/1667062.1667064 http://doi.acm.org/10.1145/1667062.1667064

Regarding the literature from various domains, we can observe that the term *metadata interoperability* has a very broad meaning and entails a variety of problems to be resolved: on a lower technical level, machines must be able to communicate with each other in order to access and exchange metadata. On a higher technical level, one machine must be able to process the metadata information objects received from another. And on a very high, semantic level, we must ensure that machines and humans correctly interpret the intended meanings of metadata. In this article, we focus on the upper two levels and analyze which properties metadata information objects must fulfill in order to be interoperable.

During the last decades a variety of interoperability techniques has been proposed. Introducing a standardized metadata schema is such a technique. But due to strategical or political reasons, in some metadata integration scenarios it is impossible to introduce or adhere to a single standard. In such a situation, the application of global conceptual models, metadata frameworks, or the definition of metadata mappings could solve the interoperability problems. In fact, there exists a wide assortment of possible techniques that aim at establishing metadata interoperability.

By systematically investigating the factors that impede metadata interoperability and analyzing how and to what extent a technique can resolve a specific factor, we can observe differences in the quality of metadata interoperability techniques—some can solve only a narrow, others a broader, spectrum of interoperability problems.

Given the high degree of attention and strong demand for providing uniform access to multiple distributed and autonomous media repositories in domains such as digital libraries or enterprise content management, we believe that it is necessary to analyze the interoperability problem in the context of multimedia metadata and to critically assess available techniques. With this article we want to give technicians and domain experts a comprehensive overview of the field of metadata interoperability, present currently known interoperability techniques, and provide a decision basis for choosing the appropriate approach for solving certain types of heterogeneity.

Our substantial contributions can be summarized as follows. First, we deeply analyze the various notions of metadata found in the literature, identify and characterize the main technical building blocks of metadata, and systematically, for each building block, elaborate on the factors that impede interoperability among metadata descriptions. Second, we give an outline of currently available techniques for achieving metadata interoperability and provide an in-depth explanation of metadata mapping, a very powerful but also technically complex technique. Third, and finally, we compare the quality of these techniques with respect to the extent they can resolve certain types of heterogeneities.

The remainder of this article is organized as follows: in Section 2 we introduce the notion of metadata using illustrative examples, regard its properties and identify the characteristic technical building blocks of metadata. Thereafter, in Section 3, we focus on *metadata interoperability*, and systematically analyze the problems that impede distinct metadata descriptions from being interoperable. In Section 4, we categorize and describe currently known techniques for achieving metadata interoperability with a focus on metadata mapping. We compare the quality of these techniques in Section 5, and conclude with Section 6.

2. UNVEILING THE NOTION OF METADATA

In this section, we briefly introduce the notion of *metadata*, which is essential in order to fully understand the metadata interoperability problem. First, we introduce an example in Section 2.1 that illustrates incompatible metadata descriptions from various institutions. We will refer to this example throughout this work whenever we explain a fact via example. Then, in Section 2.2, we identify the main building blocks of metadata descriptions. Since metadata interoperability must be established not only on the conceptual but also on the technical level, in Section 2.3, we examine how metadata resides or appears in actual information systems.

2.1. Illustrative Example

This example involves three autonomous institutions[1]: the BBC, the Austrian National Library, and the National Library of Australia. Each institution offers different types of content (audio, video, images, documents) about the Olympic Games, and uses its own technical infrastructure to maintain the contents and the adjacent metadata. A journalist, for instance, who is preparing a documentary about the history of the Olympic Games, might want to electronically access these repositories to obtain historical as well as up-to-date multimedia material about this topic. Even if all the repositories were reachable through a single technical infrastructure, uniform access would still be impossible because, as we will see in the following examples, their metadata is not interoperable.

2.1.1. Institution 1: BBC TV-Anytime Service. The BBC offers an online program information service[2] for its TV and radio channels. Via a SOAP Web Service, clients can request various details about the content, including basic program information such as the title or synopsis of a broadcast. The TV-Anytime [ETSI 2006] standard was chosen for representing program information. Its principal idea is to describe the multimedia content of broadcasts such that a user, or an agent on behalf of the user, can understand what content is available, and thus be able to acquire it [McParland 2002]. For representing audio-visual metadata, TV-Anytime encompasses a large number of elements defined by the MPEG-7 [ISO/IEC JTC 1/SC 29 2007b] standard.

Figure 1 depicts a sample TV-Anytime metadata description about a video showing `Ingmar Stenmark's victory in the 1980 Olympic Winter Games at Lake Placid`. The video was created by `John Doe`, belongs to the genre `Sports`, and has been annotated with several terms that further describe the video's contents.

2.1.2. Institution 2: Austrian National Library. The Austrian National Library's image archive[3] is the most important source of digitized historical images in Austria, and also includes well-catalogued images from past Olympic Games. All the images in the archive are described using a proprietary metadata schema, and like many other institutions, the Austrian National Library stores its metadata in a relational database that is accessible via a nonpublic SQL interface.

Figure 2 shows an example description about an image of Willy Bogner who led in the first run of the slalom during the Olympic Games in Squaw Valley in 1960. In July 2003, the Austrian National Library digitized the image, originally taken by Lothar Rübelt. Details about the person `Willy Bogner` are maintained in a so-called authority record that unambiguously identifies him as an entity in order to avoid naming conflicts. Technical features such as the MIME-type or the dimension of the image are also part of the metadata description.

[1] Although these institutions exist in the real world, the samples have been adapted to the needs of this work.

[2] BBC TV-Anytime service: `http://backstage.bbc.co.uk/feeds/tvradio/doc.html`.

[3] The Austrian National Library's image archive portal: `http://www.bildarchiv.at`.

```
<TVAMain "...">
  <ProgramDescription>
    <ProgramInformationTable>
      <ProgramInformation programId="crid://bbc.co.uk/123456789">
        <BasicDescription>
          <Title>Lake Placid 1980, Alpine Skiing, I. Stenmark</Title>
          <Synopsis>Ingmar Stenmark's (SWE-Alpine skiing) victory in
            the Giant Slalom in Lake Placid</Synopsis>
          <Genre href="urn:tva:metadata:cs:ContentCS:2004:3.1.1.9">
            <Name>Sports</Name>
          </Genre>
          <CreditsList>
            <CreditsItem role="urn:mpeg:mpeg7:cs:RoleCS:2001:AUTHOR">
              <PersonName>
                <mpeg7:GivenName>John</mpeg7:GivenName>
                <mpeg7:FamilyName>Doe</mpeg7:FamilyName>
              </PersonName>
            </CreditsItem>
          </CreditsList>
          <CreationCoordinates>
            <CreationLocation>us</CreationLocation>
          </CreationCoordinates>
        </BasicDescription>
      </ProgramInformation>
    </ProgramInformationTable>
  </ProgramDescription>
</TVAMain>
```

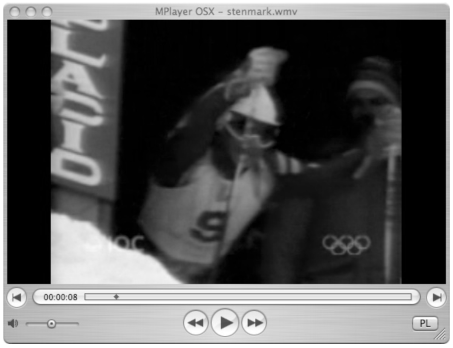
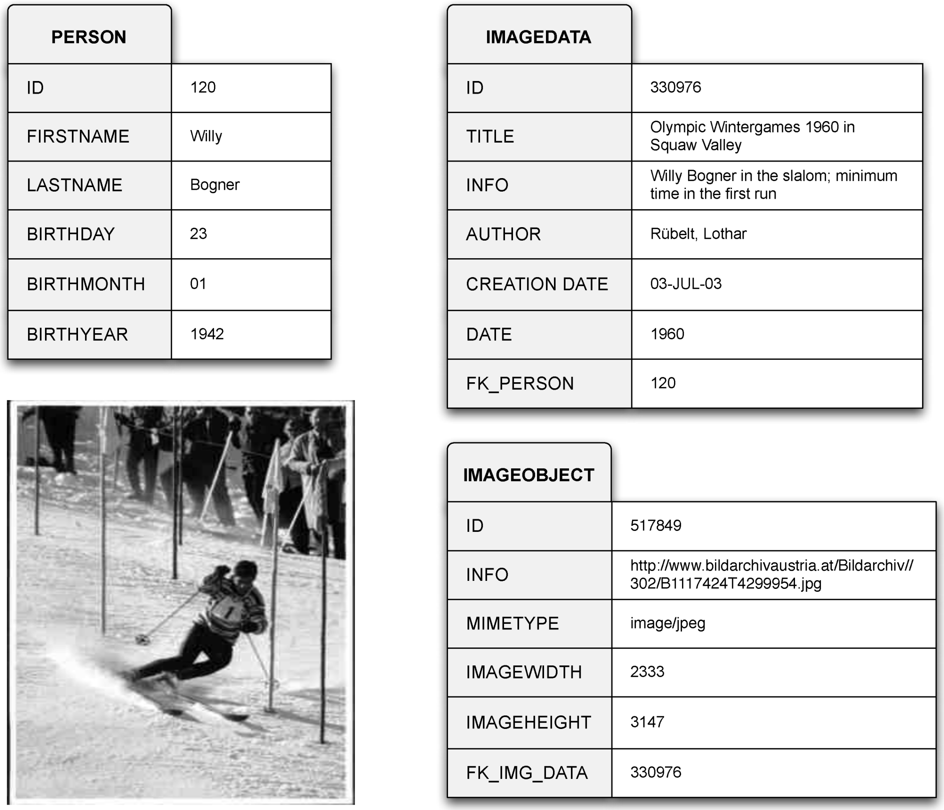


Fig. 1. TV-Anytime metadata describing a video.

2.1.3. Institution 3: National Library of Australia. As a result of the Olympics in Sidney in 2000, the National Library of Australia[4] now maintains a huge image collection from this event. The Dublin Core Element Set [DC 2006] has been chosen to represent the metadata for these images, and all images have been digitized and are now available online. The metadata is exposed via the Open Archives Protocol for Metadata Harvesting (OAI-PMH),[5] an HTTP-based protocol that allows the retrieval of XML metadata descriptions.

[4] The Australian National Library's Web site: `http://www.nla.gov.au/`.

[5] The Australian National Library's OAI-PMH service: `http://www.openarchives.org/OAI/openarchivesprotocol.html`.



PERSON	
ID	120
FIRSTNAME	Willy
LASTNAME	Bogner
BIRTHDAY	23
BIRTHMONTH	01
BIRTHYEAR	1942

IMAGEDATA	
ID	330976
TITLE	Olympic Wintergames 1960 in Squaw Valley
INFO	Willy Bogner in the slalom; minimum time in the first run
AUTHOR	Rübelt, Lothar
CREATION DATE	03-JUL-03
DATE	1960
FK_PERSON	120

IMAGEOBJECT	
ID	517849
INFO	http://www.bildarchivaustria.at/Bildarchiv//302/B1117424T4299954.jpg
MIMETYPE	image/jpeg
IMAGEWIDTH	2333
IMAGEHEIGHT	3147
FK_IMG_DATA	330976

Fig. 2. Proprietary metadata describing a JPEG image.

Figure 3 shows a picture of marathon runners crossing the Sydney Harbour Bridge during the Olympics 2000. The adjacent metadata description gives further details about the photographer, the format of the image, and the date when it was taken. Key words such as `Runners (Sports)` or `Sportsmen and sportswomen` further describe the contents.

2.2. Metadata Building Blocks

Following Gilliand's definition [Gilliland 2005], we conceive metadata as *the sum total of what we can say about any information object at any level of aggregation*, in a machine understandable representation. An information object is *anything that can be addressed and manipulated by a human or a system as a discrete entity*. Defining metadata more abstractly as *any formal schema of resource description, applying to any type of object, digital or nondigital* [NISO 2004] is also appropriate, especially for application scenarios that apply metadata for describing nondigital resources (e.g., persons).

Metadata can be classified according to the functions it is intended to support (descriptive, structural, administrative, rights management, preservation metadata) [NISO 2004; Johnston 2004], or to its level of semantic abstraction (low-level vs. high-level metadata); (e.g., Westermann and Klas [2003]). Low-level technical metadata has less value for end users than high-level, semantically rich metadata that describes semantic entities in a narrative world, such as objects, agent objects, events, concepts,

```
<OAI-PMH "...">
    ...
    <metadata>
        <oai_dc:dc "...">
            <dc:title>Sydney Olympics 2000, marathon runners cross Sydney
                Harbour Bridge [picture] /</dc:title>
            <dc:creator>Mahony, David (David James)</dc:creator>
            <dc:format>1 photograph : gelatin silver ; image 26.9 x 38.4 cm.
                on sheet 30.5 x 40.3 cm.</dc:format>
            <dc:coverage>New South Wales</dc:coverage>
            <dc:date>2000</dc:date>
            <dc:description>Photograph by David Mahony -- On reverse in pencil.;
                Condition: Good. Group of [marathon] runners feature
                eventual Gold Medal Winner Gezahgne Abero of Ethiopia (No.
                1651) [Sydney, N.S.W., September 2000]</dc:description>
            <dc:subject>Runners (Sports) -- Australia -- Portraits.</dc:subject>
            <dc:subject>Sydney Harbour Bridge (Sydney, N.S.W.)</dc:subject>
            <dc:subject>Olympic Games (27th :, 2000 : Sydney, N.S.W.)</dc:subject>
            <dc:subject>Marathon running -- Australia -- Photographs.</dc:subject>
            <dc:subject>Sportsmen and sportswomen.</dc:subject>
            <dc:type>Image</dc:type>
            <dc:identifier>nla.pic-an22842546</dc:identifier>
            <dc:source>Item held by National Library of Australia</dc:source>
            <dc:rights>You may save or print this image for research and study.</dc:rights>
            <dc:identifier>http://nla.gov.au/nla.pic-an22842546</dc:identifier>
        </oai_dc:dc>
    </metadata>
    ...
</OAI-PMH>
```

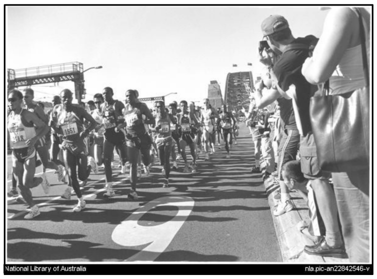


Fig. 3. Dublin Core metadata describing a JPEG image.

semantic states, semantic places, and semantic times, together with their attributes and relations [Benitez et al. 2001]. Hence, the semantic quality and expressiveness of metadata is essential for effectively retrieving digital objects.

To put it simply, metadata is data that describes some resource. In the examples presented in the previous section, the resources are digital images and videos, hence information objects. The adjacent metadata descriptions mainly consist of high-level, semantically rich descriptive information, such as the names of persons (e.g., `Ingmar Stenmark`, `Willy Bogner`) or events (e.g., `Sydney Olympics 2000`). Only the Austrian National Library provides some low-level technical metadata (`mime-type`, `imagewidth`, `imageheight`).

We can identify the following common characteristics: each description is made up of a set of elements (e.g., `title`, `author`, `subject`) and content values (e.g., `Lake Placid`,. . . , `Rübelt Lothar`, `Runners`); the elements are defined as part of a metadata schema, which can be standardized, as is the case for Dublin Core or TV-Anytime, or proprietary; from the fact that two metadata descriptions are expressed in XML and one in terms of relations in a relational database, we can derive that the metadata elements were previously specified using a certain language. For XML, the language is usually XML Schema [W3C 2006], for the case of a relational database, a schema is expressed in terms of tables using SQL-DDL [ISO/IEC JTC 1/SC 32 2003].

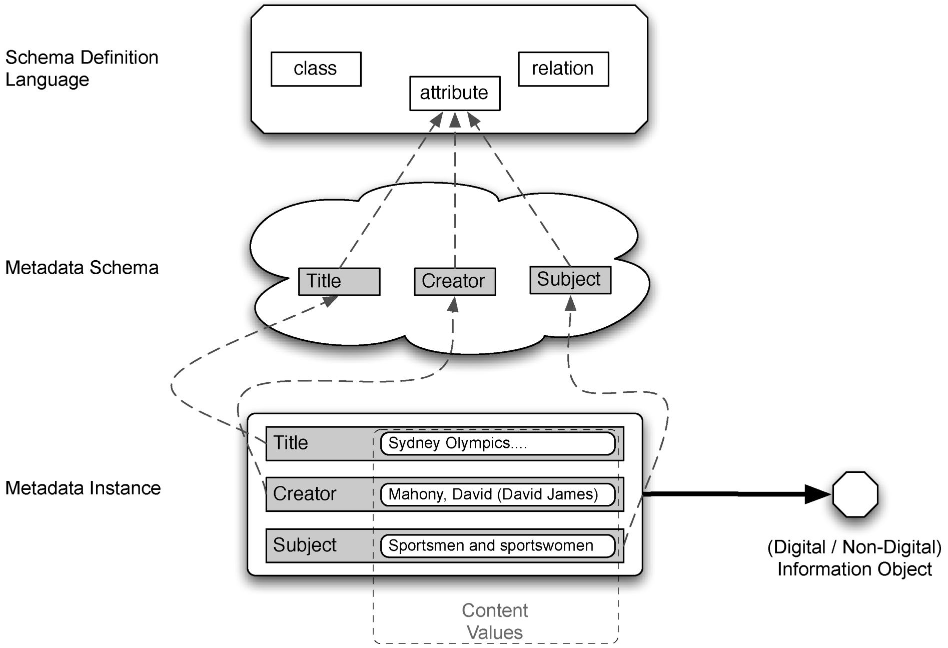


Fig. 4. Overview of the three metadata building blocks.

Based on this observation, we can identify three main metadata building blocks: we denote the set of content values in a metadata description as *metadata instance*, the element definitions as *metadata schema*, and the language for defining metadata schemes as *schema definition language*. Figure 4 illustrates these three building blocks and their dependencies. In the following, we will focus further on each of these building blocks in a reverse order.

2.2.1. Schema Definition Language. An application and domain-specific metadata schema is expressed in a certain schema definition language, whereas each language provides a set of language primitives (e.g., class, attribute, relation). Because machines must *understand* a language, the primitives are not only syntactic constructs but also have a semantic definition.

The semantic definition of the term *language* already implies[6] that, in order to communicate with each other, there must exist an agreement on the meaning of a language's primitives. This is also the case for schema definition languages: usually, there exist language standards or at least some kind of consensus.[7] Sample schema definition languages are XML Schema, SQL-DDL, the RDF Vocabulary Description Language (RDFS) [W3C 2004a], the Web Ontology Language (OWL) [W3C 2004b], or the Unified Modeling Language (UML) [OMG 2007b].

2.2.2. Metadata Schema. Another metadata building block is the element definition, which we denote as metadata schema. A metadata schema is simply a *set of elements*

[6] *Language*: the system of communication used by a particular community or country (*The New Oxford American Dictionary*)

[7] The W3C consortium, for instance, does not publish *standards* but *recommendations*.

with a *precise semantic definition, optionally connected by some structure* [Rahm and Bernstein 2001]. The semantics of a schema is defined by the meanings of its elements. A schema usually defines the names of elements together with their semantics and optionally content rules that define how content values must be formulated (e.g., capitalization, allowable content values). For the encoding of the elements and their values, a schema can define syntax rules. If no such rules are defined, a schema is called *syntax-independent* [NISO 2004].

At this point, we must mention that the notion of *metadata schema*, which originates from the database domain, is often named differently in other contexts or communities. For instance, knowledge workers, librarians or others working in the field of knowledge and information organization tend to use the term *metadata vocabulary* for what we call metadata schema. These communities tend to regard metadata more from a linguistic rather than a technical point of view. Another very common notion is the term *ontology*, which has its origin[8] in the artificial intelligence domain, and is defined as *a specification of a conceptualization* [Gruber 1993]. At its core, an ontology is the same as a schema: a set of elements connected by some structure. Apart from that, Noy and Klein [2004] have identified several features that distinguish ontologies from schemes in a database sense: the first, and probably the most important, one is that ontologies are logical systems that define a set of axioms that enable automated reasoning over a set of given facts. Second, ontologies usually have richer data models involving representation primitives such as inverse properties. Third, ontology development is a much more decentralized and collaborative approach than schema development, which opens the potential for reusing existing ontology definitions. As a consequence of these different perceptions, we can also find different names for the languages used to define metadata elements: the *vocabulary definition language* and *ontology definition language* are examples.

Despite these different perceptions, we believe that the term *metadata schema* is appropriate for the purpose of this work, since it regards metadata interoperability mainly from a technical perspective. Stronger than the term *vocabulary*, it emphasizes that metadata elements have a technical grounding in information systems, which is a significant aspect when metadata interoperability should also be realized on a technical level. Weaker than the term *ontology*, it sets the limits for the underlying technical system. It encompasses logical systems in the AI sense (e.g., first order logic, description logics), but also other, in practice more widespread, systems such as relational database schemes or XML schemes. However, in this work the term *metadata schema* does not refer to traditional database schemes only. Rather, we regard it is a common denominator for all the terms previously mentioned.

2.2.3. Metadata Instance. Continuing with the metadata building blocks, we can identify a third one, the *metadata instance*. A metadata instance holds a set of metadata elements drawn from a metadata schema and adjacent content values. These element-value combinations form a metadata description about a certain information object. As already mentioned, the rules for creating such metadata instances are imposed by the metadata schema it is related to. If there exists such a relation, we say that a metadata instance *corresponds* to a schema.

2.3. The Appearance of Metadata in Information Systems

Metadata is made up of *information objects* that are designed for, persistent in, retrieved from, and exchanged between information systems. The form of appearance of metadata

[8] The term *ontology* actually originates from philosophy; it has its technical origin in the AI domain.

in information systems is defined through information models on various levels. In a typical information system we can identify four such levels: the physical, the logical, the programming/representation, and the conceptual level.

On the lowest level, the *physical level*, metadata consists of bits and bytes that are represented in memory, written to disks and transmitted over wires. The system components that work on this level are mainly concerned with optimizing heap allocation, assigning records to file partitions, and building indices for efficient retrieval. Thus, the information model on this level comprises concepts like format, files, records, or indices.

The physical level is usually hidden from application developers through the *logical level*. This is the level where database management systems are located. Information models such as the Relational Data Model [Codd 1970] provide the basis for creating technically precise and complete information models for a certain target domain. A metadata schema is defined using a data definition language and represented through a set of relation schemes. The metadata instances are described as sets of tuples contained in tables. Besides providing the necessary transparency from the underlying physical model, the logical level organizes metadata efficiently (e.g., through normalization) and introduces important features for application developers, such as data consistency, concurrency control, and transactions. However, metadata is scattered in tables that do not directly reflect the application domain[9] and must be aggregated into higher level, conceptual entities when being processed by applications.

On the *programming/representation level*, the constituents of conceptual schema models are manifested or presented in various forms: metadata schemes can be transformed into code of a certain (object-oriented) programming language and reflect the application domain in terms of classes or types while metadata descriptions become runtime objects. Metadata elements and their contents can also be encoded using a certain mark-up language such as XML or be represented on the Web using HTML. This requires metadata descriptions to be adapted to certain programming and document models (e.g., W3C DOM), and metadata schemes to be modeled using the modeling language (e.g., Java syntax, XML Schema, DTD) imposed by the programming/representation technology.

A metadata schema on the *conceptual level* resembles real-world entities with their properties and relationships among entities. The TV-Anytime standard for instance, which is used in the illustrative example in Section 2.1, defines real-world entities such as `Video` or `Creator` and properties such as `GivenName` or `FirstName`. Common languages for creating conceptual models are the Entity-Relationship Model [Chen 1976] or the Unified Modeling Language [OMG 2007b].

2.3.1. Metadata—A Model Perspective. All these levels have in common that the information elements of a metadata schema, that is, its elements and their relationships, are implemented in terms of a data model. Such a *data model for metadata*, called metadata *model*, encapsulates the defined elements of a metadata schema, represents their semantics (their meaning) in a formal way, and provides a syntax for serializing metadata descriptions. The semantic interpretation of a metadata model is given through the mapping of the model elements to the corresponding entities in a certain application domain. The syntax of a metadata schema, that is, the legal elements and rules for how and what values can be assigned to elements, is defined using a certain notation, which can be symbolic (e.g., words, formulas), textual (e.g., natural language sentences), or graphical (e.g., diagrams).

[9] Also called *domain of discourse* or *universe of discourse*.

Schema definition languages occur on each information level and are used for expressing information models. The core of such a language is its metamodel—we call it metadata *metamodel*—which is the machine-internal representation of the language. It reflects the language primitives (abstract syntax) together with a concrete notation (concrete syntax) and semantics. Further it defines and constrains the allowable structure of models. The semantics or interpretation of a language is the mapping of the metamodel's (abstract syntax) elements to the language's primitives. Semantic definitions of models may feature varying degrees of formality, ranging from natural language descriptions to formal languages or mathematics. Especially from an interoperability point of view, rigid and semantic precise definitions enable consistent interpretation across system boundaries [Seidewitz 2003].

Metadata models and metamodels are arranged on different levels that are orthogonal to the previously mentioned levels of information. On the lowest level we can find metadata (descriptions) that are (valid) instances of a metadata model (e.g., Java classes, UML model, database relations) that reflects the elements of a certain metadata schema. The metadata model itself is a valid instance of a metadata metamodel being part of a certain schema definition language. Due to this abstraction, it is possible to create metamodel representations of metadata (e.g., metadata instances of an UML model can also be represented as instances of the UML meta-model).

The MOF specification [OMG 2006a] offers a definition for these different levels: *M0* is the lowest level, the level of metadata instances (e.g., Title=`Lake Placid 1980, Alpine Skiing, I. Stenmark`). *M1* holds the models for a particular application; that is, metadata schemes (e.g., definition of the field `Title`) are M1 models. Modeling languages reside on level *M2*, their abstract syntax or metamodel can be considered as *model of a particular modeling system* (e.g., definition of the language primitive *attribute*). On the topmost-level, at *M3*, we can find universal modeling languages in which modeling systems are specified (e.g., core constructs, primitive types). Figure 5 illustrates the four levels, their constituents and dependencies.

Regarding these abstraction levels, the remaining question is how the abstract syntax (*meta-metamodel*) of an M3 modeling language is expressed. In general, there are two options for defining a metamodel: either by using a different modeling mechanism (e.g., context-free grammars for programming languages, formal semantics or an explicit set of axioms and deduction rules for logic-based languages) or by using its own modeling language. In the latter case, the elements of a metamodel are expressed in the same language the metamodel is describing. This is called a *reflexive metamodel* [Seidewitz 2003] because the elements of a modeling language can be expressed using a (minimal) set of modeling elements. An example for such an approach is the UML metamodel, which is based on the UML Infrastructure Library [OMG 2006c], which in turn is the minimal reflexive metamodel for UML.

2.3.2. A Selection of Schema Definition Languages. Table I gives an overview of a selected set of schema definition languages which can be assigned to the M2 level. For each language, we describe its *concrete syntax*, that is, how language elements are represented. Further, we outline how the *semantics* is defined for each language, assuming that the available options are: natural language, formal semantics (i.e., precise mathematical definitions), reference implementations, and test suites. Finally, we describe the machine-internal representation of a modeling language: its *metamodel* or *abstract syntax*. We outline the metamodel's type (e.g., object-oriented, semantic network-based) and which model construct categories (structural, behavioral) it supports. Further, we give an overview of its main model elements and sketch out if the model is defined reflexively.

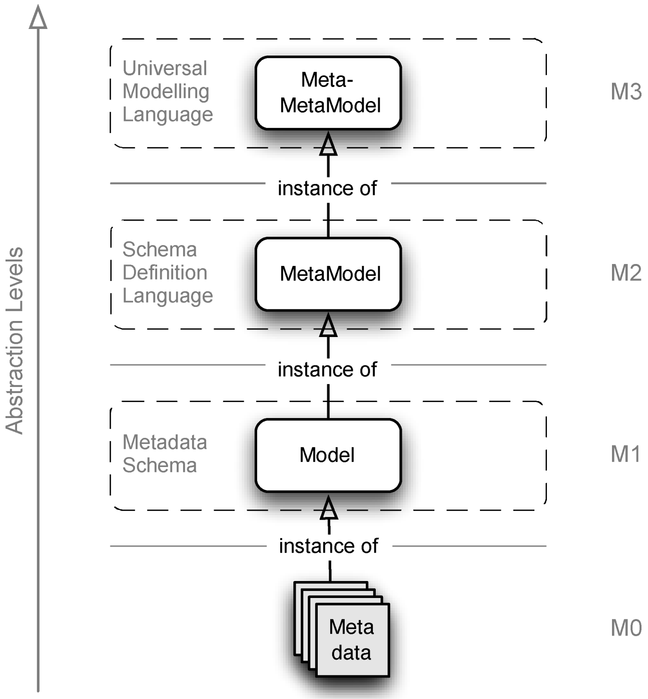


Fig. 5. Metadata building blocks from a model perspective.

The Unified Modeling Language (UML) [OMG 2007b] and the Entity Relationship Model (ER) [Chen 1976] are used for the conceptual modeling of software systems, and therefore employ a graphical syntactic notation. Topic Maps [ISO/IEC JTC 1/SC 34 2006], RDFS [W3C 2004a], and OWL [W3C 2004b] are conceptual languages that allow the modeling of metadata schemes in terms of semantic networks, which requires their machine-internal representation to be graph-based. Other common features are the fact that their semantics is formally defined and that their syntax is XML based, while RDF also provides other syntaxes such as N3 [Berners-Lee 1998] or Triple [Sintek and Decker 2002].

Java is a representative for the programming part of the programming/representation level. Its syntax and semantics are defined in Gosling et al. [2005]; since Java version 6, the Java API also comprises a metamodel [Java Community Process 2006; Bracha and Ungar 2004] that reflects the structural and behavioral primitives of the Java language. Another modeling language for defining metadata schemes on the presentation level is XML Schema. Its semantics in natural language as well as its hierarchical metamodel (abstract syntax) are defined in W3C [2006].

A prominent representative of languages on the logical level is SQL [ISO/IEC JTC 1/SC 32 2003], or more precisely, the SQL Data Definition Language (SQL DDL). Its metamodel is the relational model,[10] which is semantically defined through the relational algebra.

[10] In fact, vendors of database management systems employ their own, system-specific, metamodels. But usually they are still based on the relational model.

Table I. A Selection of Schema Definition Languages and Their Characteristics

		Conceptual					Programming / Representation		Logical		
		UML	ER	Topic Maps	RDF/S	OWL	Java SE 6	XML Schema	SQL-DDL	DL	CL
Abstract Syntax (Meta-Model)	Model Type	Object-Oriented	ER-dedicated graph	Semantic Network based / labelled graph	Semantic Network based / labelled graph	Semantic Network based / labelled graph	Object-Oriented	Hierarchical	Relational Model	Logics	Logics
	Model Constructs Categories	Structural & Behavioral	Structural	Structural	Structural	Structural	Structural & Behavioral	Structural	Structural (Behavioral)	Structural	Structural
	Main Model Elements	Class, Composite Structure, Component, Deployment, Activity, Interaction, State Machine, Use Case	Entity Sets, Attributes, Relation-ships	Topic, Association, Occurrence, TopicMap	Statement, BlankNode, Graph, Property, Literal, Resource, Class, Datatype, Bag, Seq, Alt, List	Ontology, Class, Datatype- & Object-Property, Individual, Restriction	Class, Interface, Field, Method, Annotation, Enum, Package	Data types, Simple Type, Complex Type, Element, Attribute	Relation, Attribute, Schema, Tuple, Domain	Facts, Axioms	Phrases, Terms, Atoms, Sentences,
	Reflexive Definition	Yes (via M3 level)	No	No	Yes	Yes	No	Yes	No	No	No
Concrete Syntax		Graphical Notation	Graphical Notation	XML, HyTime	RDF/XML, N3, N-Triple	RDF/XML, N3, N-Triple	Java Syntax	XML	SQL Syntax	DL Syntax	CLIF, CGIF, XCL
Semantics		Natural language	Formal	Formal	Formal	Formal	Natural language, Reference Implement-ation	Natural language	Formal	Formal	Formal

For creating knowledge bases, logical languages such as Description Logics (DL) [Baader et al. 2003] or Common Logics (CL) [ISO/IEC JTC 1/SC 32 2005] can be applied for defining schemes in terms of knowledge graphs. Naturally, their semantics is defined formally; while the syntax for DL is simply symbolic, CL defines the Common Logic Interchange Format (CLIF), the Conceptual Graph Interchange Format (CGIF), and the Extended Common Logic Markup Language (XCL) for the syntactic representation and exchange of CL schema definitions.

2.4. Basic Observations

Metadata consists of three main building blocks: metadata instances, metadata schemes, and schema definition languages. When applying a technical view on these blocks, we note that a metadata description is in fact an instance of a metadata model, which in turn is an instance of a metadata metamodel. We can conceive metadata as information objects with three abstraction levels: metadata instances reside on the M0 level, metadata models on the M1 level, and metadata metamodels on the M2 level.

We can summarize this section with two main observations. First, the choice of the schema definition language directly affects the appearance of metadata information objects in an information system. This is because there is a direct technical (instance-of) dependency between metadata instances, metadata schemes, and schema definition languages. Second, although some languages overlap in certain aspects (e.g., graph-based metamodel, support for behavioral modeling constructs), there are many discrepancies between their abstract and concrete syntax, and in the way their semantics is defined. This implies that an automatic translation between metadata schemes expressed in different modeling languages is a problematic task. It will, for instance, require human intervention to find a work-around for translating metadata defined in a graph-based model to a hierarchical, tree-like model; a tree is a special kind of a graph, but not vice versa.

3. METADATA INTEROPERABILITY

We have claimed that metadata interoperability is the prerequisite for uniform access to digital media in multiple heterogeneous information systems. Hence, for solutions that aim at establishing uniform access, achieving metadata interoperability is the necessary prerequisite. Before discussing various techniques for achieving interoperability, we first investigate the notion of metadata interoperability in the literature and come up with an appropriate definition in Section 3.1. Thereafter, in Section 3.2, we inspect in detail the various forms of heterogeneities that impede metadata interoperability.

3.1. Uniform Access to Digital Media

In the context of information systems, interoperability literally denotes the *ability of a system to work with or use parts of other systems*.[11] In literature we can also find similar definitions: for the digital libraries domain, Baker et al. [2002] summarize various interoperability viewpoints as *the potential for metadata to cross boundaries between different information contexts*. Other authors from the same domain define interoperability as *being able to exchange metadata between two or more systems without or with minimal loss of information and without any special effort on either system* [NISO 2004; ALCTS CC:DA 2000] or as *the ability to apply a single query syntax over descriptions expressed in multiple descriptive formats* [Hunter and Lagoze 2001].

The notion of interoperability can further be subdivided: for Baker et al. [2002], achieving interoperability is a problem to be resolved on three main levels: the *transport and exchange* level (e.g., protocols), the metadata *representation* level (e.g., syntactic binding, encoding language), and the level of metadata with their *attribute space* (e.g., schema elements) and *value space* (e.g., controlled vocabularies). Based on the perspective of heterogeneities in information systems, Sheth and Larson [1990] and Ouksel and Sheth [1999] present four main classes of interoperability concerns: *system interoperability* dealing with system heterogeneities such as incompatible platforms, *syntactic interoperability* dealing with machine-readable aspects of data representation, *structural interoperability* dealing with data structures and data models, and *semantic interoperability*. Tolk [2006] proposes another view, consisting of six levels: *no interoperability* on the lowest level, *technical interoperability* (communication infrastructure established) on level one, *syntactic interoperability* (common structure to exchange information) on level two, *semantic interoperability* (common information model) on level three, *pragmatic interoperability* (context awareness) on level four, *dynamic interoperability* (ability to comprehend state changes) on level five, and *conceptual interoperability* (fully specified, but implementation-independent model) on level six. Miller [2000] detaches interoperability from the technical level and introduces, alongside *technical* and *semantic* interoperability, several flavors of interoperability: *political / human*, *intercommunity*, *legal*, and *international* interoperability.

Based on its literal meaning, the definitions in literature, and considering the technical characteristics of metadata information objects described in Section 2.3, we define metadata interoperability as follows:

[11] According to the *Merriam-Webster Online Dictionary* (`http://www.m-w.com/`) and the *Oxford Online Reference* (`http://www.oxfordreference.com/`).

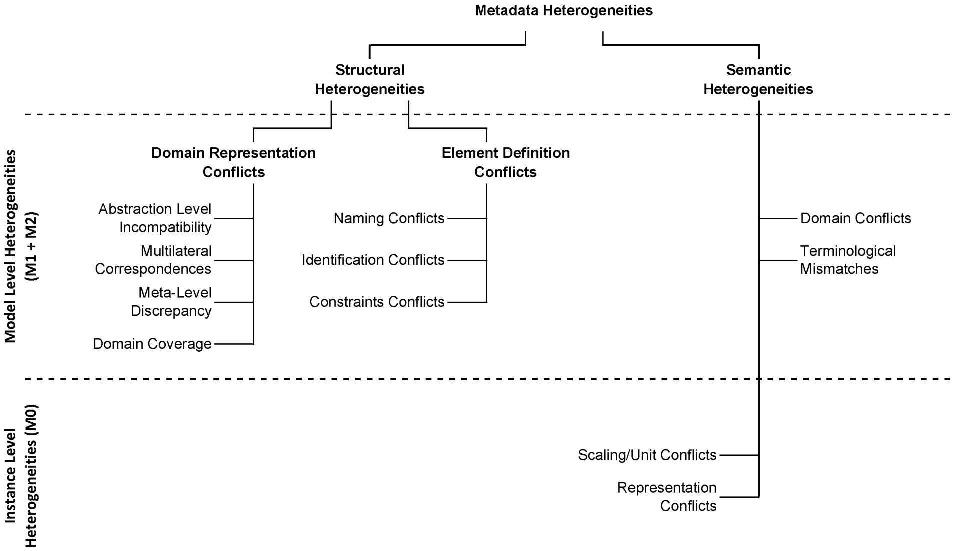

Metadata Heterogeneities
Structural Heterogeneities
Semantic Heterogeneities
Model Level Heterogeneities (M1 + M2)
Domain Representation Conflicts
Element Definition Conflicts
Abstraction Level Incompatibility
Multilateral Correspondences
Meta-Level Discrepancy
Domain Coverage
Naming Conflicts
Identification Conflicts
Constraints Conflicts
Domain Conflicts
Terminological Mismatches
Instance Level Heterogeneities (M0)
Scaling/Unit Conflicts
Representation Conflicts

Fig. 6. Structural and semantic metadata heterogeneities on the model and the instance level.

Definition 1. Metadata interoperability is a qualitative property of metadata information objects that enables systems and applications to work with or use these objects across system boundaries.

With this definition we clearly distinguish our conception of metadata interoperability, which is settled on the information level, from system-level interoperability issues such as communication infrastructures, hardware or software platform incompatibilities.

3.2. Heterogeneities Impeding Interoperability

The heterogeneities to be eliminated in order to provide interoperability have already been identified in the early history of database research. A first in-depth analysis was provided by Sheth and Larson [1990]. Throughout the years they have been investigated more deeply (e.g., Ouksel and Sheth [1999]), and also regained attention in related domains, such as artificial intelligence [Wache 2003; Visser et al. 1997]. In Figure 6, we provide a classification of the predominant heterogeneities mentioned in the literature from a model-centric perspective. We recall that there is an instance-of relationship between metadata instances, metadata schemes, and schema definition languages, that is, metadata are instances of metadata models, and metadata models are instances of metadata metamodels. Generalizing these relationships, we can distinguish between *model-level* and *instance-level* heterogeneities as a first dimension for our classification. For the second dimension we differentiate between two classes of heterogeneities: *structural heterogeneity* caused by the distinct structural properties of models and *semantic heterogeneity* occurring because of conflicts in the intended meaning of model elements or content values in distinct interpretation contexts.

3.2.1. Structural Heterogeneities. Structural heterogeneities on the model level occur because of model incompatibilities. A model mainly consists of its atomic elements (e.g., entities, attributes, relations) and the combination or arrangement of these elements forming a certain structure for representing a particular domain of interest.

That being the case, we can group structural heterogeneities occurring between distinct models into *element definition conflicts*, which are conflicts rooted in the definitions of a model (naming, identification, constraints), and *domain representation conflicts*, which occur because domain experts arrange model elements that reflect the constituents of a certain domain in various ways and detail. In the following we further analyze these two groups of structural heterogeneities.

3.2.1.1. Naming Conflicts. We denote conflicts that occur because model elements representing the same real-world entity are given different names as naming conflicts. On the level of schema definition languages (M2), distinct metamodels assign different names to language primitives that are used to model the same real-world facts. UML, for instance, defines the primitive *Class*, while ER uses *EntitySets* to capture the same kind of real-world concepts. Also on the M1 level, distinct metadata models might assign different names to elements representing the same real-world concepts. In the examples presented in Section 2.1, the model elements that represent the image's descriptions are labeled `Synopsis` in the TV-Anytime, `Description` in the Dublin Core, and `Info` in the proprietary schema.

3.2.1.2. Identification Conflicts. A special case of naming conflicts are those conflicts dealing with unique identification of model elements. On the M2 level, depending on the language used for defining a model, elements are identifiable either by their name only (e.g., ER, SQL-DDL) or by some (fully) qualified identifier (e.g., XMLS, OWL). Identification conflicts can also arise on the M1 level of metadata schemes. In our example, the TV-Anytime schema comprises model elements that are fully qualified via their namespace. The elements in the proprietary model, that is, the tables and columns, are identified by their names only.

3.2.1.3. Constraints Conflicts. Element definition conflicts that occur because distinct models provide different possibilities of defining constraints are denoted as constraints conflicts. An example for an M2 constraint is the ability of a schema to import other schema definitions. This is possible in languages such as XML Schema or OWL but not in ER. Incompatible primary keys, conflicting references or domain constraints could lead to the same type of conflict on the M1 level of metadata schemes.

3.2.1.4. Abstraction-Level Incompatibilities. Abstraction-level incompatibilities belong to the group of domain representation conflicts and turn up when the same real-world entities are arranged in different generalization hierarchies or aggregated differently into model elements. An example for this type of conflict on the M2 level is the ability to define attributes and relations in various languages: while ER (`attribute` and `relation`) and OWL (`datatypeProperty` and `objectProperty`) define primitives for both language features, XML Schema (`attribute`) and Java (`field`) subsume these features under a single primitive. Abstraction-level incompatibilities at the M1 level, the level of metadata models, occur for instance when one metadata model aggregates the creator of a digital resource into single entity `creator`, as is the case with Dublin Core, while other models such as TV-Anytime and also MPEG-7 distinguish between `Persons` and `Organizations`, both being a specialization of the concept `Agent`.

3.2.1.5. Multilateral Correspondences. Another domain representation conflict, which is a direct result of the previously mentioned abstraction-level incompatibilities, are multilateral correspondences. On each level, an element in one model can correspond to multiple elements in another model, and vice versa. In our example there is such a correspondence between the Dublin Core `creator` element and the TV-Anytime elements `GivenName` and `FamilyName`. This is because in the TV-Anytime metadata description, these elements are used in the context of a `CreditsItem` element that is taking the role of an `author`.

3.2.1.6. Meta-Level Discrepancy. Domain representation conflicts that occur because certain model elements do not have any direct correspondences in another model are subsumed under meta-level discrepancy. This, however, does not necessarily mean that the other model cannot capture the same information about a certain domain. Real-world concepts represented as elements in one model (e.g., `author` as attribute) could be modeled differently in another model (e.g., `author` as entity) or even being captured as content of a model element on the instance level. In our example, the Dublin Core and the proprietary schema store the location of the Olympic Games as content value with the field `Title`, while TV-Anytime defines a model element `CreationLocation`, which captures this kind of information. We can distinguish the following kinds of meta-level discrepancies: content-value / attribute, entity / attribute, and content value / entity discrepancy.

3.2.1.7. Domain Coverage. When no correspondences between model elements exists, we speak of domain coverage conflicts. This happens when real-world concepts reflected in one model are left out in the other model, although both models were designed for the same semantic domain. In our example, the TV-Anytime description does not give any evidence about the image's size while the proprietary one does.

3.2.2. Semantic Heterogeneities. Semantic heterogeneities on the model level are conflicts occurring because of the differences in the semantics of models. We recall that a model's semantics is defined by its semantic domain and the semantic mappings (interpretations) from the domain entities to the model elements. The semantic domain provides the meaning for each model element and can contain language expressions, in the case of schema definition languages, or real-world entities in the case of metadata models.

3.2.2.1. Domain Conflicts. When domains overlap, subsume, or aggregate others, or when domains are incompatible, we speak of domain conflicts. An example of such a conflict on the M2 level is the expressiveness of languages; with languages that have a rich domain, that is, an expressive set of language primitives, we are able to model things that are not expressible with other languages having fewer powerful primitives. With OWL, for instance, it is possible to express that two classes are equivalent or that one class is the union of two other classes. Other languages such as XML Schema or Java do not have built-in language constructs to indicate such relationships. Obviously, domain conflicts can also occur among metadata models on the M1 level. If one model reflects the domain of electronic billing and another one the domain of multimedia contents, it is unlikely that there are any meaningful correspondences among these models.

3.2.2.2. Terminological Mismatches. This is another kind of semantic heterogeneity occurring on both model levels: synonym conflicts occur if the same domain concept is mapped to model elements with different names; homonym conflicts exist if different domain concepts are mapped to model elements with the same names.

An example of a homonym conflict on the language level is the polymorphic concept of *overloading* that appears in object-oriented languages like Java. These languages support polymorphic functions whose operands (parameters) can have more than one type. Types can be arranged in a subtype hierarchy and symbols (e.g., field-names, method-signatures) may be *overloaded*, meaning that the same symbol is used to denote semantically different behaviors [Cardelli and Wegner 1985]. In Java, for instance, the *equals* method is usually overwritten and implemented differently for each class, which could lead to unintended behavior during runtime.

An example for a synonym conflict on the schema level is the usage of distinct terms to denote the same semantic concept. In our example, the proprietary schema uses the

Table II. A Categorization of Metadata Interoperability Techniques

	Model Agreement		**Meta-Model Agreement**		**Model Reconciliation**
M2 – Schema Definition Languages	Hybrid Metadata System (e.g. MPEG-7, TV-Anytime)	Standardized Language (e.g. OWL, UML, XML Schema)	Metadata Meta-Meta Model (e.g. MOF)	Abstract Metadata Model (e.g. DCMI Abstract Model)	Language Mapping
M1 – Metadata Schemes		Standardized Metadata Schema (e.g. DC, TEI, MODS)	Global Conceptual Model (e.g. CIDOC-CRM, FRBR)		Metadata Crosswalk, Schema Mapping
			Metadata Framework (e.g. MPEG-21, METS, OAIS)		
		Application Profile (e.g. DC Collection Profile)			
M0 – Metadata		Value Encoding Schema (e.g. ISO-Norms, RFC-Specifications)			Instance Transformation
		Controlled Vocabulary (e.g. LCSH, DCC, MeSH)			
		Authority Record (e.g. LOC Authorities, Deutsche Personennormdatei (PND))			

term `author` and the Dublin Core schema the term `creator` to represent the person who has created the particular image.

3.2.2.3. Scaling/Unit Conflicts. Semantic heterogeneity occurring on the metadata M0 instance level, when different scaling systems are used to measure content values, are called scaling or unit conflicts. In our examples, the dimensions of the described images are represented in pixels in the proprietary schema, and in centimeters in the Dublin Core schema. Even if the images had semantically the same dimension, the content values would be different.

3.2.2.4. Representation Conflicts. Representation conflicts are a result of using different encoding schemes for content values. For instance, two date values, which are semantically the same, could be represented differently in each system (e.g., `date=01.01.2007` or `date=2007/01/01`).

4. TECHNIQUES FOR ACHIEVING METADATA INTEROPERABILITY

Over decades, experts working in the field of metadata interoperability have developed methods and solutions to overcome the previously described heterogeneities. The goal of this section is to set up a framework for categorizing existing techniques according to their common characteristics.

Metadata interoperability can be achieved by eliminating or bridging the structural and semantic heterogeneities at the metadata metamodel (M2), metadata model (M1), and the metadata instance (M0) levels. We can identify three principal ways to attain interoperability among models: (i) agreement on a certain metadata model; (ii) introduction of and agreement on a common metamodel; and (iii) reconciliation of the structural and semantic heterogeneities. Table II provides an overview of a variety of techniques to achieve metadata interoperability and classifies them according to the three previously mentioned categories.

In the following sections, we will focus on each of these techniques and discuss their characteristics. Finally, we have devoted Section 4.4 to *metadata mapping*, an interoperability technique that subsumes model reconciliation on the M1 and M0 levels and, in our opinion, requires a more detailed discussion.

4.1. Model Agreement

Standardization is a strong form of establishing an agreement by means of consensus-building, and is an intuitive, technically effective, and economically well-recognized way to achieve interoperability. It requires accredited institutions (e.g., World Wide Web Consortium (W3C), Object Management Group (OMG), International Standardization Organization (ISO), German Institute for Standardization (DIN)) for building consensus, setting a standard, and eventually assuring its uniform implementation. Regarding the building blocks of metadata, standardization can cover the language level (*standardized language*), the schema level (*standardized metadata schema*), the instance level, or several levels (*hybrid metadata system*).

4.1.1. Standardized Language. In Section 2.3.2, we had already shown a representative selection of various types of schema definition languages ranging from programming languages (e.g., Java), over conceptual modeling (e.g., UML) to logical languages (e.g., Description Logics).

Each schema definition language defines a set of language primitives and, as in natural languages, postulates that multiple parties agree on their semantics. This guarantees interoperability on the level of schema definition languages. Consequently, metadata that is expressed in the same language can be processed by any application that is aware of that language.

Agreement on the M2 level of schema definition languages is typically enforced through various forms of standardization. Programming languages are often specified by companies (e.g., Java by Sun Inc.) or standardized by standards institutes (e.g., ANSI-C by the American National Standards Institute (ANSI)). Languages designed for data representation and exchange are standardized by international consortia (e.g., W3C, OMG).

Regarding our examples in Section 2.1, we can observe that all three metadata schemes have been defined by using a standardized language: the BBC's TV-Anytime metadata are described in TV-Anytime, which in turn is defined in XML Schema. The Austrian National Library's metadata schema is defined in terms of a relational schema expressed in SQL-DDL; and the metadata provided by the National Library of Australia also corresponds to a schema that has been expressed in XML Schema.

4.1.2. Standardized Metadata Schema. If there is an agreement or consensus on a *set of metadata elements* on the M1 level, and this agreement is manifested in a standard, we speak of a standardized metadata schema. In Table III, we present a selection of metadata standards used in various domains. For each standard, we indicate its application domain, which requirements or purpose it fulfills, and to which schema definition languages it is bound in order to also express a metadata schema on a technical level. Furthermore, we show which standardization body maintains a standard, the year of initial publication, and the current version together with the year this version was released.

Most standardized metadata schemes are designed for a specific domain and a certain purpose. The VRA Core Standard [VRA 2007], for instance, is tailored to the cultural heritage community and comprises metadata elements for the description of works of visual culture as well as the images that document them. The Dublin Core Metadata

Table III. A Representative Selection of Metadata Standards

Name	Application Domain	Purpose	M2 Language Bindings	Standardization Body	Current Version	Year of Initial Publication
Dublin Core Element Set (DC)	domain independent	description of a wide range of resources	XMLS, RDF/S	ISO/NISO	1.1 (2008)	1998
Visual Resources Association Core (VRA Core)	cultural heritage	description of works of visual culture and images that document them	XMLS	VRA Data Standards Committee	4.0 (2007)	1996
Guidelines for Elec-tronic Text Encoding and Interchange (TEI)	humanities, social sciences, linguistics	representation of texts in digital form	XMLS	Text Encoding Initiative Consortium	P5 (2007)	1990
Learning Objects and Metadata (LOM)	eLearning	description of digital or non-digital learning objects	XMLS, RDF/S	IEEE	1.0 (2002)	1997
Sharable Content Object Reference Model (SCORM)	eLearning	aggregation, description, and sequencing of learning objects	XMLS	Advanced Distributed Learning Initiative (ADL)	3rd Edition (2004)	2000
Online Information Exchange (ONIX)	publishing / retail (books and serials)	provision of product information to online retailers	DTD, XMLS	EDItEUR group	release 2.1 (2005)	2000
MARC 21 Format for Bibliographic Data	(digital) libraries	exchange of bibliographic data	XMLS (MARCXML)	Network Development & MARC Standards Office	update no. 7 (2006)	1999
Metadata Object Description Schema (MODS)	digital libraries	subset of MARC fields using language-based tags	XMLS	Network Development & MARC Standards Office	3.2 (2006)	2002
Maschinelles Austauschformat für Bibliotheken (MAB)	(digital) libraries in German speaking countries	exchange of bibliographic data	XMLS (MABxml)	Expert group for data formats	2 (2001)	1973
Format for Bibliographic Records (RFC1807)	universities, r&d organizations	description of technical reports	XMLS	Network Working Group	1.0 (revised in 2002)	1995
Geographic Information Metadata (ISO 19115)	geographic information systems (GIS)	documentation of geographic digital resources	XMLS, GML	ISO	1.0 (2003)	2003

Element Set [DC 2006] is an example for a schema that is broad and generic enough to describe a variety of resources across domains.

Besides Dublin Core and VRA, our selection also comprises the Guidelines for Electronic Text Encoding and Interchange (TEI) [TEI 2007], a standard mainly used in the humanities and social sciences for representing texts and data about texts in digital form. In the case of TEI, the standardization body is the consortium that developed this metadata standard.

As representatives for the e-Learning domain, we have selected the Sharable Content Object Reference Model (SCORM) [ADL 2007] and the Learning Objects Metadata (LOM) [IEEE WG-12 2002] standards. While the first standardizes the aggregation and sequencing aspects of learning objects, the latter is mainly concerned with their description. Further, the development of SCORM is driven by the Advanced Distributed Learning Initiative,[12] which embraces several standardization bodies, including IEEE.[13]

The MARC 21 Format for Bibliographic Data [LOC 2007c], a metadata standard in the libraries domain for the purpose of exchanging bibliographic data, is maintained by the Network Development and MARC standardization office. MAB and its successor MAB 2 [DNB 2007a] represent the German counterparts to the family of MARC standards, and were developed by a group of library domain experts. The Metadata Object Description Schema (MODS) [LOC 2007d] is a metadata standard that defines a subset of the MARC fields using language-based instead of numeric-based tags. RFC1807 [NWG 1995] is a very old bibliographic metadata standard mainly used in universities and research organizations for describing technical reports.

Online Information Exchange (ONIX) [EDItEUR 2007] is another metadata standard and is situated at the borderline of bibliographic description and electronic commerce. Its main purpose is to provide product information about books and serials to online retailers. The standard is maintained by an international group of book retailers and vendors, called EDItEUR.

Finally, from the domain of geographic information system, our selection contains ISO 19115 [ISO TC 211 2003], a metadata standard designed for the documentation of geographic digital resources.

The technical implementation of a metadata standard is bound to one or more schema definition languages, which provide the facilities to represent a standard's metadata elements in a machine-readable way. Regarding our selection of metadata standards, we can observe that the majority is bound to XML Schema. Some (e.g., DC and LOM) also provide bindings for RDF/S, and the ISO 19115 standard even defines the Geography Markup Language (GML) [OGC 2004], which is an extension of the XML grammar.

From our examples, two institution follow a standardized metadata schema approach: the BBC provides TV-Anytime and the National Library of Australia features Dublin Core metadata.

4.1.3. Hybrid Metadata Systems. We denote metadata standards that cannot be assigned to a single level but span multiple levels as hybrid metadata systems.

The MPEG-7 [ISO/IEC JTC 1/SC 29 2007b] standard is an important representative of a hybrid metadata system. It spans the M2 and M1 levels and defines a set of metadata schemes (MPEG-7 Description Schemes) for creating multimedia metadata descriptions, as well as a schema definition language called the MPEG-7 Description Definition Language (DDL), which is an extension of XML Schema. This language provides the solid descriptive foundation for users to create their own metadata schemes, compatible with the MPEG-7 standard [Kosch 2003].

The TV-Anytime standard is a representative for a hybrid metadata system that spans the M1 and M0 levels. On the M1 level, it heavily reuses elements defined by the MPEG-7 standard and tailors them to the requirements of the broadcasting domain. Further, it defines a set of classification schemes, which are in fact controlled vocabularies allowing the classification of telecasts along various dimensions. Sample dimensions are a telecast's content (e.g., news, arts, religion/philosophies), its formal structure (e.g., magazine, cartoon, show), or even its atmosphere (e.g., breathtaking, happy, humorous, innovative). The terms listed in the classification schemes are possible content values within metadata descriptions, and can therefore be used as content values in M0-level metadata instances.

[12] Advanced Distributed Learning Initiative: `http://www.adlnet.gov/`

[13] Institute of Electrical and Electronics Engineers (IEEE): `http://www.ieee.org/`

In our BBC metadata example, we can see a TV-Anytime metadata description which makes use of two MPEG-7 elements (`mpeg7:GivenName` and `mpeg7:FamilyName`). Furthermore, the `Genre` element references an M0-level term defined within an MPEG-7 classification schema (`urn:tva:metadata:cs:ContentCS:2004:3.1.1.9`).

4.1.4. Instance-Level Agreement. If several parties agree on a set of possible content values for M0-level metadata descriptions, we denote this as instance-level agreement. In the real world, we can find various forms of agreements or standards on the instance level.

One frequently occurring form is *controlled vocabularies*, such as the Library of Congress Subject Headings (LCSH) [LOC 2007b], the Dewey Decimal Classification System (DDC) [OCLC 2007], or the Medical Subject Headings (MeSH) [NLM 2007]. The main goal of a controlled vocabulary is to support search and retrieval of resources by indexing them with terms taken from a vocabulary that has been designed by domain experts who posses expertise in the subject area. The complexity of a controlled vocabulary can range from a simple list of terms, over a hierarchical arrangement of terms (taxonomy), to systems that defined terms and the semantic relationships between them (thesaurus). In our example, the BBC metadata uses a term taken from a controlled vocabulary to reference the creation location (`us`).

Authority control is another form of instance-level agreement, and is very similar to controlled vocabularies. The goal is to disambiguate identical entities by linking the content values of metadata descriptions to uniquely identifiable authority records maintained and shared by a central authority. In the library domain, authority control is commonly used to relate potentially distinct names of one and the same person with a single uniquely identifiable entity. The Library of Congress Authorities [LOC 2007a] or the German Personennormdatei (PND) [DNB 2007b] are examples for centrally maintained directories of person names. In our examples, the Austrian National Library maintains authority records for authors and persons. Therefore the author of the image described—`Rübelt, Lothar`—is under authority control.

A *value encoding schema* is a form of instance-level agreement, which defines exactly how to encode a certain type of content value. The ISO 8601 [ISO TC 154 2004] standard, for example, provides encoding rules for dates and times; the ISO 3166 [ISO TC 46 2006b] standard defines how to represent country names and their subdivisions. This kind of standardization guarantees that machines can correctly interpret nontextual content values such as dates and times, or abbreviated textual values representing some entity such as country codes. The Austrian National Library metadata provide an example for a nonagreed-upon instance value: the creation dataevalue (`03-JUL-03`) does not follow any standardized value encoding schema.

In theory, if all metadata in all information systems within a certain integration context were instances of a single standardized metadata schema expressed in a single schema definition language, and also if all content values used within the metadata instances were taken from a single controlled vocabulary, all the structural and semantic heterogeneities mentioned in Figure 6 would be resolved, at least technically.

4.2. MetaModel Agreement

In real-world environments, we can observe that often institutions do not adhere to standards. Attempts to find an agreement for a standard often results in semantically weak minimum consensus schemes (e.g., the Dublin Core Element Set) or models with extensive and complex semantic domains (e.g., the CIDOC Conceptual Reference Model (CRM) [ISO TC 46 2006a]). Often, it is not practicable for institutions to agree on a certain model or apply an existing standard because they already have their proprietary solutions in place. In such a case, one possibility for achieving interoperability is not to agree on a model but on a common metamodel. For all existing proprietary models in place, instance-of relationships from a model to the common metamodel are established. Through this relationship, the elements of the proprietary models can then be manipulated as if they were elements of the metamodel. Therefore, metamodel agreement implicitly enables interoperability by creating correspondences between proprietary models via a common metamodel.

4.2.1. Metadata MetaMeta Model. An example for such an approach on the M2 level of schema definition languages is the OMG Meta-Object Facility (MOF), which is a *universal modeling language in which modeling systems can be specified* [OMG 2006a]. It solves language mismatches by introducing the M3 level and by superimposing a metadata meta-metamodel containing a set of elements for modeling language primitives. If the model elements of M2 schema definition languages are aligned with the elements of the M3 MOF model, it is possible to express metadata in terms of the more general MOF model.

Kensche et al. [2007] propose another model that serves as an M3 abstraction for particular metamodels on the M2 level. They relate certain metamodels (e.g., EER, OWL) with a generic meta-metamodel through generalization and introduce *roles* to decorate M3-level elements with M2 specific properties that must be preserved.

4.2.2. Abstract Metadata Model. The specification of an abstract metadata model is another way of achieving interoperability. Such a model resides on the M2 level of schema definition languages and serves as a technical reference for the implementation of metadata schemes in information systems. If there is an agreement on such a metamodel in all information systems and all metadata schemes are expressed in terms of the elements provided by this model, the metadata information objects are interoperable, at least from a structural point of view, because they are technically represented in the same way. The DCMI Abstract Model [Powell et al. 2005] is an example for an abstract metadata model. In a similar manner as RDF/S, it defines an information model for representing Dublin Core metadata in a machine-processable way.

4.2.3. Global Conceptual Model. Introducing a global conceptual model is a way of achieving interoperability on the M1 level of metadata schemes. All information systems to be integrated must align their metadata model elements with the more general elements defined in the global model, which formalizes the notions in a certain domain and defines the concepts that appear in a certain integration context.

The CIDOC CRM is an example for such a model being designed for the Cultural Heritage domain. It defines 81 entities and 132 properties, most of them on a very abstract level (e.g., `physical thing`, `section definition`). Another example for a global conceptual model is the Functional Requirements for Bibliographic Records (FRBR) [IFLA 1997] model, which has been defined by the International Federation of Library Associations and Institutions. With its four key entities (`work`, `expression`, `manifestation`, `item`) it represents a generalized view of the bibliographic universe, independent of any cataloguing standard or implementation [Tillett 2004]. The Suggested Upper Merged Ontology (SUMO)[14] is also a global model that *will promote data interoperability, information search and retrieval, automated inferencing, and natural language processing* [Niles and Pease 2001]. It defines high-level concepts such as `object`, `continuousObject`, `process`, or `quantity`. Another example for a global model

[14] The Suggested Upper Merged Ontology: `http://ontology.teknowledge.com/`.

approach is the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [WonderWeb Consortium 2003].

4.2.4. Metadata Framework. A metadata framework *can be considered as a skeleton upon which various objects are integrated for a given solution* [Chan and Zeng 2006] and is another way of achieving interoperability on the M1 level of metadata schemes. It typically provides a data model consisting of a set of abstract terms, and a description of the syntax and semantics of each model element. Again, the idea is to integrate existing metadata by aligning their model elements to a set of elements defined by the metadata framework. Examples for metadata frameworks are the MPEG-21 Multimedia Framework [ISO/IEC JTC 1/SC 29 2007a], the Metadata Encoding and Transmission Standard (METS) [LOC 2007e], or the Open Archival Information System (OAIS) [CCSDS 2002].

4.2.5. Application Profile. An application profile [Heery and Patel 2000; Baker et al. 2001] is a special kind of model agreement. On the one hand, it is a schema consisting of data elements drawn from one or more standardized schemes, optimized for a particular application domain, whereas its focus is on the reuse of existing, standardized model elements. On the other hand, from a technical point of view, an application profile is a metadata model, which is an extension of a set of agreed-upon metamodels.

Application profiles are created by application developers who declare how to apply standardized schema elements in a certain application context. Within an application profile, we cannot create new model elements that do not exist elsewhere. If this is required, a new metadata schema containing these elements must be created and maintained. Refinement of standard element definitions is allowed. Developers can set the permitted range of values (e.g., special date formats, particular formats for personal names) and narrow or specify the semantic definition of the metadata elements. Application profiles are created with the intent of reuse and are tailored for specific purposes or certain user communities.

Example application profiles are the Dublin Core Collections Application Profile [DC 2007] for describing collections of physical or digital resources, the Eprints Application Profile [Allinson et al. 2007] for describing scholarly publications held in institutional repositories, or the application profiles[15] that were created when METS was introduced in several digital libraries.

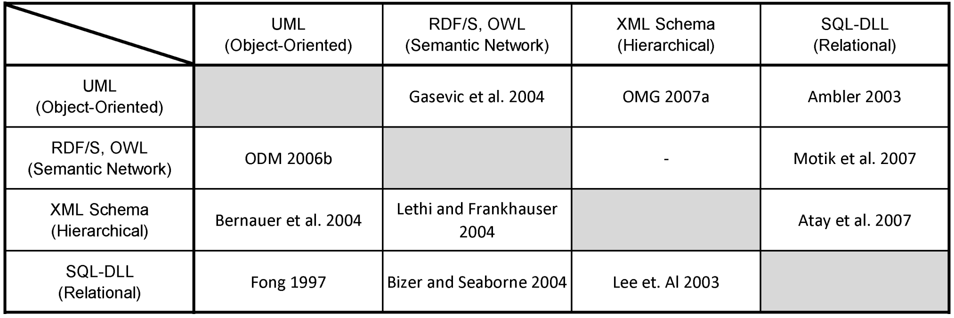
4.3. Model Reconciliation

Often, especially in settings where the incentives for an agreement on standards are weak, neither model nor metamodel agreement are suitable interoperability techniques. The digital libraries domain, for instance, is such a domain: there is no central authority that can impose a metadata standard on all digital libraries. Such settings require other means for reconciling heterogeneities among models.

4.3.1. Language Mapping. If metadata schemes are expressed in different schema definition languages, mappings on the language level are required to transform the instances, that is, the M1 level metadata models, from one linguistic representation to another. Because of the substantial structural and semantic discrepancies among schema definition languages (see Section 2.3), translation from one language into another can cause loss of valuable semantic information. Table IV refers to relevant mapping

[15] A list of METS application profiles is available at `http://www.loc.gov/standards/mets/mets-registered-profiles.html`.

Table IV. A Selection of M2 Language Mapping Approaches



	UML (Object-Oriented)	RDF/S, OWL (Semantic Network)	XML Schema (Hierarchical)	SQL-DLL (Relational)
UML (Object-Oriented)		Gasevic et al. 2004	OMG 2007a	Ambler 2003
RDF/S, OWL (Semantic Network)	ODM 2006b		-	Motik et al. 2007
XML Schema (Hierarchical)	Bernauer et al. 2004	Lethi and Frankhauser 2004		Atay et al. 2007
SQL-DLL (Relational)	Fong 1997	Bizer and Seaborne 2004	Lee et. Al 2003	

literature for a selected set of M2 schema definition languages, each of which is based on a different model type (cp., Table I).

Gasevic et al. [2004] present an approach for automatically generating OWL ontologies from UML models. The OMG XML Metadata Interchange (XMI) specification [OMG 2007a] describes the production of XML documents from instances of MOF models (i.e., UML models) and provides XML schemes for XML validation of these documents. Ambler [2003] discusses the features of the object-oriented and the relational model, analyzes the object-relational impedance mismatch, and proposes mapping strategies between objects and relational databases.

The OMG Ontology Definition Metamodel (ODM) specification [OMG 2006b] offers a set of metamodels and mappings for translating between the UML metamodel, which is based on MOF, and ontology languages such as RDF/S, OWL, Topic Maps, and Common Logics (CL). Mapping and translating from Semantic Web languages such as RDF/S and OWL to XML Schema has, until now, been disregarded in the literature; most work focuses on mapping into the opposite direction, that is, from XML Schema to RDF/S and OWL. Motik et al. [2007] compare OWL and relational databases with respect to their approaches to schema modeling, schema and data reasoning problems, and constraint checking.

Bernauer et al. [2004] discuss various approaches for integrating XML Schemas into UML-based software development processes and compare their quality according to the transformation patterns they apply for generating UML out of models expressed in XML Schema. Lethi and Frankhauser [2004] describe a mapping from XML Schema to OWL ontologies and explore how these ontologies can be used as an abstract modeling layer on top of XML data sources. Atay et al. [2007] analyze the conflicts between the hierarchical, ordered nature of the XML data model and the flat, unordered nature of the relational model. They propose an algorithm to map ordered XML data to relational data.

Fong [1997] describe a mapping of relational schemes into an object-oriented representation and propose a methodology for translating schemes and converting relational instance data. Bizer and Seaborne [2004] describe a mapping from the relational model to RDF/S and present the D2R Server, a wrapper component that exposes data from relational databases as RDF/S. Lee et al. [2003] describe two semantic-based conversion methods that generate XML Schema representations from a given relational input schema.

4.3.2. Schema Mapping. If an agreement on a certain model is not possible, schema mapping is an alternative to deal with heterogeneities among metadata schemes. In the digital libraries domain, *metadata crosswalks* have evolved as a special kind of schema mapping. A crosswalk is *a mapping of the elements, semantics, and syntax from* *one metadata schema to another* [NISO 2004]. The goal of crosswalks is to provide the ability to make elements defined in one metadata standard available to communities using related metadata standards. A complete or fully specified crosswalk consists of the semantic mapping between model elements and a metadata instance transformation specification [Pierre and LaPlant 1998]. In practice, however, crosswalks often define only the semantic mapping on the M1 level, and leave the problem of instance transformation to the application developers.

4.3.3. Instance Transformation. In the context of mappings, instance transformation is the approach for achieving interoperability on the metadata instance level, when there is no agreement on value encoding schemes or other standardization mechanisms. Instance transformations are functions that operate on the content values and perform a specified operation, such as the concatenation of the values of two fields (e.g., `GivenName=John`, `FamilyName=Doe`) into a single field (e.g., `Creator=John Doe`). We will further focus on instance transformations in Section 4.4.

4.4. Metadata Mapping

From all previously mentioned interoperability techniques, those classified as model reconciliation techniques are the most complex. Since heterogeneities can occur on all tree levels, it is necessary to specify mappings for each level: language mappings for the M2 level, schema mappings for the M1 level, and instance transformations for the M0 level. Before a mapping on a certain level can be defined, the heterogeneities on the level above must be reconciled, that is, we must deal with M2 language differences before specifying M1 schema mappings.

Previously, in Section 2.3.2, we have already outlined the characteristics of a representative set of schema definition languages and pointed out the divergence in their abstract and concrete syntax. Because of that, metadata mapping does not deal with heterogeneities on the language level (M2), but assumes that all metadata information objects are expressed in the same schema definition language. This can be achieved by transforming metadata information objects from one language representation into another, which could also entail loss of semantics

Here, we elaborate further on *metadata mapping*, a technique that subsumes *schema mapping* and *instance transformation* as described in Section 4.3. Before discussing its technical details, we define the scope of this technique as follows:

Definition 2. Given two metadata schemes, both settled in the same domain of discourse and expressed in the same schema definition language, we define metadata mapping as a specification that relates their model elements in a way that their schematic structures and semantic interpretation is respected on the metadata model and on the metadata instance level.

From a model perspective, a metadata mapping defines structural and semantic relationships between model elements on the schema level and between content values on the instance level. To represent such relationships, any mapping mechanism requires a set of mapping elements with a well-defined syntax and semantics. From this perspective, we can regard not only metadata schemes but also a mapping between metadata schemes as being a model. Bernstein et al. [2000] as well as Madhavan et al. [2002] have proposed such a perspective. Furthermore, we can denote the total of all mapping relationships contained in a mapping model as *mapping specification*.

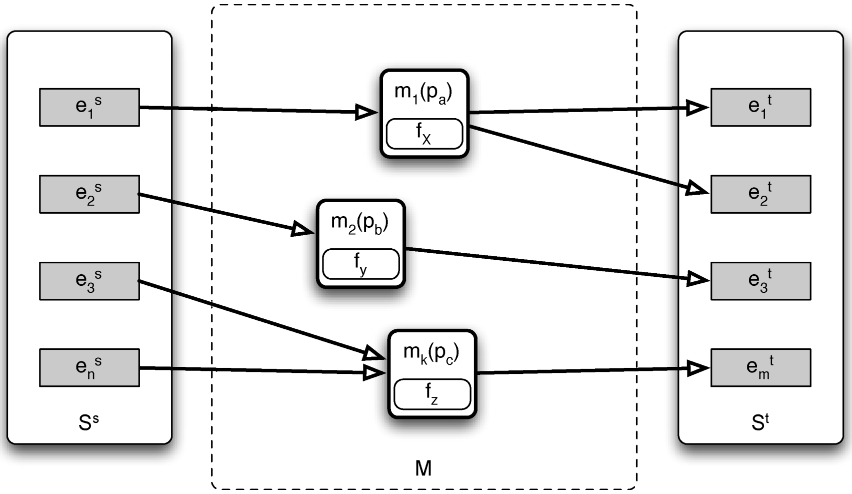


Fig. 7. The main elements of a metadata mapping specification.

4.4.1. Technical Details. From a technical perspective a metadata mapping can formally be defined as follows:

Definition 3. A metadata mapping is defined between a source schema $S^s \in \mathcal{S}$ and a target schema $S^t \in \mathcal{S}$, each consisting of a set of schema elements, $e^s \in S^s$ and $e^t \in S^t$ respectively, which are optionally connected by some structure. A mapping $M \in \mathcal{M}$ is a directional relationship between a set of elements $e_i^s \in S^s$ and a set of elements $e_j^t \in S^t$, where each mapping relationship is represented as a *mapping element* $m \in M$. The semantics of each mapping relationship is described by a *mapping expression* $p \in P$. The cardinality of a mapping element m is determined by the number of incoming and outgoing relations from and to the schema elements. To support heterogeneity reconciliation on the instance level, a mapping element carries an appropriate *instance transformation function* $f \in F$.

Figure 7 illustrates the main elements of a metadata mapping specification. Typically, the cardinality of a single mapping element is either 1:1, 1:n, or n:1, meaning that an element from a source schema is related with one or many elements from the target schema and vice versa. In theory, m:n mappings would also be possible, but in practice they rarely occur because we can model that kind of element correspondence using multiple 1:n or n:1 relationships.

A mapping expression p defines the semantics of a mapping element, that is, it describes how the interpretations of the model elements, denoted $I(e_i^s)$ and $I(e_i^t)$, are related. In its simplest form, such an expression could be *unknown*, stating that two elements are related, without giving any evidence how. A more complex example are mapping expressions that indicate the confidence of a mapping relationship according to a specified metrics, as described in Mena et al. [2000]. We can distinguish between the following types of mapping expressions (e.g., Spaccapietra et al. [1992]).

—Exclude $(I(e_i^s) \cap I(e_j^t) = \emptyset)$. The interpretations of two schema elements have distinct meanings. In the example Section 2, the interpretations of the elements `rights` in the Dublin Core and `birthday` in the proprietary schema exclude each other.

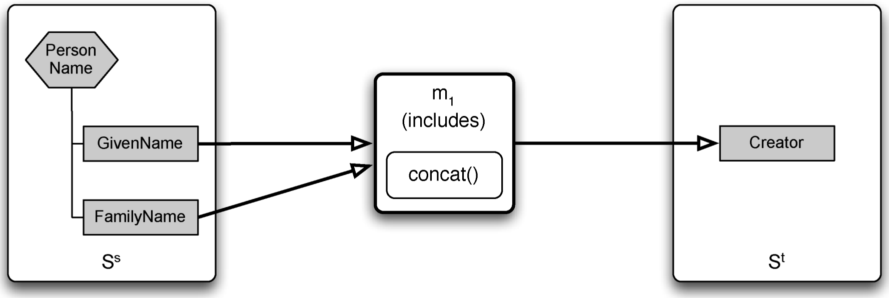


Fig. 8. Achieving metadata interoperability through instance transformation.

- —Equivalent ($I(e_i^s) \equiv I(e_j^t)$). The interpretations of two, possibly lexically different schema elements are equivalent. The element `author` in the proprietary and the element `creator` in the Dublin Core schema are examples for such a relationship.
- —Include ($I(e_i^s) \subseteq I(e_j^t) \vee I(e_j^t) \subseteq I(e_i^s)$). The interpretation of one schema element contains the interpretation of another element. In the context of our example, the interpretation of the Dublin Core element `creator` includes the interpretations of the TV-Anytime elements `GivenName` and `FamilyName` because these elements describe a person in the role of an `author`.
- —Overlap ($I(e_i^s) \cap I(e_j^t) \neq \emptyset \wedge I(e_i^s) \not\subseteq I(e_j^t) \wedge I(e_j^t) \not\subseteq I(e_i^s)$). The interpretations of two schema elements overlap but do not include each other. The elements `description`, `synopsis`, and `info` are examples for elements with overlapping interpretations. A description element usually provides similar information as a synopsis or info element, but in a more comprehensive form.

Instance transformation functions are the mechanisms to cope with the structural and semantic heterogeneities on the instance level. If, for instance, two models (e.g., the TV-Anytime and the DC illustrative samples) are incompatible due to a multilateral correspondences conflict (e.g., `GivenName` and `FamilyName` in the source model and `Creator` in the target model), this can be resolved by relating the elements through a mapping relationship and assigning an instance transformation function `concat`, which concatenates the data values of the respective fields and returns the appropriate result. Figure 8 illustrates the role of mapping expressions and instance transformation functions in metadata mappings.

4.4.2. Mapping Phases. Besides being a mechanism for capturing the semantic and structural relationships between the elements of distinct models, metadata mapping is also a process consisting of a cyclic sequence of phases. As illustrated in Figure 9, we can identify four such phases: (i) *mapping discovery*; (ii) *mapping representation*; (iii) *mapping execution*; and (iv) *mapping maintenance*.

The reason for the cyclic arrangement of the mapping phases is the fact that mapping maintenance is also the key for discovering new mappings from existing ones. If, for instance, there is a mapping between schema A and schema B and another mapping between schema B and schema C, and all this information is available in a registry, the system could derive an additional mapping between schemas A and C, based on their transitive relation.

Being aware that each part of this phase could be the subject of another detailed study, here we merely give a brief outline for each phase. For an in-depth discussion of all four phases we refer to a related survey [Haslhofer 2008].

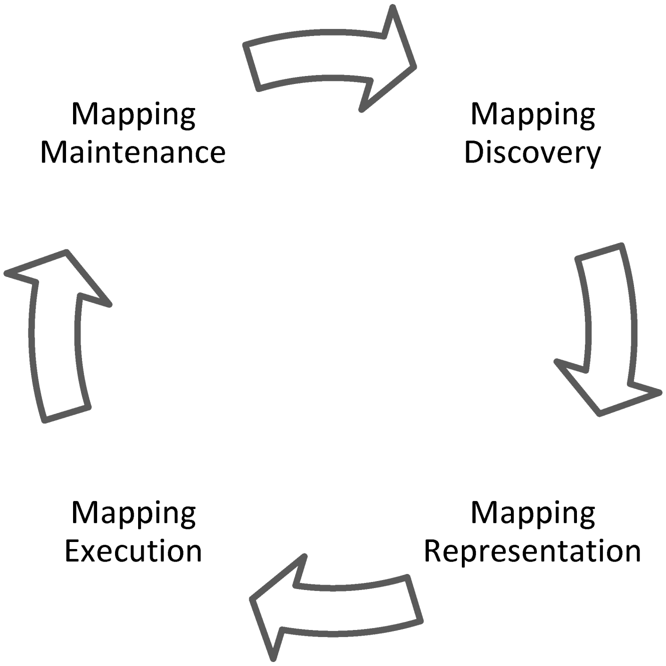


Fig. 9. The four major phases in the metadata mapping cycle.

Mapping discovery is concerned with finding semantic and structural relationships between the elements of two schemes and reconciling the heterogeneities on both the schema and the instance levels. Deep domain knowledge is required to understand the semantics of the elements of the source and target schemes in order to relate their elements on the schema- and the instance-levels. Rahm and Bernstein [2001] as well as Kalfoglou and Schorlemmer [2003] describe a variety of mapping discovery techniques that operate on both levels.

Mapping representation is the second phase of the mapping process and denotes the formal declaration of the mapping relationships between two metadata schemes. Noy [2004] identifies three types of formalisms for representing mappings: (i) representing them as instances of a defined mapping model; (ii) defining bridging axioms or rules to represent transformations; and (iii) using views to define mappings between a source and a target schema.

Mapping execution is the phase for executing mapping specifications at run-time. Mappings can be used for various interoperability-dependent tasks such as metadata transformation, answering queries over a set of metadata sources, or creating software stubs that encapsulate the mappings and provide transparent access to the underlying metadata source. Halevy [2001] gives an overview of view-based mapping approaches.

Mapping maintenance is the last step in an iteration of the metadata mapping phases. Usually, a registry provides the necessary maintenance functionality and keeps track of available metadata schemes and mappings between them. This information allows systems to deal with issues like versioning (e.g., Noy and Musen [2004]), which is required whenever there are changes in the source or target schema of a certain mapping.

Table V. The Quality of Various Interoperability Techniques

		M1 Level Interoperability Techniques						M0 Level Interoperability Techniques			
		Hybrid Metadata System (e.g. MPEG-7, TV-Anytime)	Standardised Metadata Schema (e.g. DC, TEI, MODS)	Global Conceptual Model (e.g. CIDOC-CRM, FRBR)	Metadata Framework (e.g. MPEG-21, METS, OAIS)	Application Profile (e.g. DC Collection Profile)	Metadata Crosswalk, Schema Mapping	Value Encoding Schema (e.g. ISO-Norms, RFC-Specifications)	Controlled Vocabulary (e.g. LCSH, DCC, MeSH)	Authority Record (e.g. LOC Authorities, Deutsche Personennormdatei (PND))	Instance Transformation
M1 Level Heterogeneities	Naming Conflicts	■	■	■	■	■	■	□	□	□	□
	Identification Conflicts	■	■	■	■	■	■	□	□	□	□
	Constraints Conflicts	■	■	□	□	■	■	□	□	□	□
	Abstraction Level Incompatibility	■	■	□	□	■	■	□	□	□	□
	Mulitlateral Correspondences	■	■	□	□	■	■	□	□	□	□
	Meta-Level Discrepancy	■	■	□	□	■	■	□	□	□	□
	Domain Coverage	■	■	□	□	■	■	□	□	□	□
	Domain Conflicts	■	■	■	□	■	■	□	□	□	□
	Terminological Mismatches	■	■	■	■	■	■	□	□	□	□
M0 Level Heterogeneities	Scaling/Unit Conflicts	■	□	□	□	□	□	■	■	■	■
	Representation Conflicts	■	□	□	□	□	□	■	■	■	■

5. ON THE QUALITY OF INTEROPERABILITY TECHNIQUES

In this section, we focus on the quality of the previously mentioned interoperability techniques and analyze to what extent a certain technique can deal with the various kinds of heterogeneities discussed in Section 3.2.

For two reasons we restrict ourselves on techniques that enforce interoperability on the metadata model (M1) and instance level (M0): first, as we have generalized in Section 2.3, we can apply an abstract view on models on various levels and distinguish between models- and instance-level heterogeneities. This is because at the core of both, the schema definition language and the metadata schema, are in fact models. Therefore, we can analyze their potential for dealing with heterogeneities between models and their instances. The second reason is that in practice we can assume that all metadata can be transformed into a uniform language representation.

To determine the quality of an interoperability technique, we have analyzed whether it can resolve a specific heterogeneity type. Table V summarizes the results of this analysis: one dimension shows the various interoperability techniques, the other the heterogeneities grouped by their types. The dotted line separates the model (M1) and the instance level (M0) for both, techniques and heterogeneity types. The grayed areas represent the groups of heterogeneities described in Section 3.2.

5.1. Model Agreement Techniques

Model agreement techniques are an effective means for achieving interoperability. If all proprietary systems adapt their existing information systems in a way that their models fit into a hybrid metadata system, a standardized metadata schema, or an application profile, most heterogeneity problems can be resolved.

A fixed and semantically well-defined set of given metadata elements resolves naming-, identification-, and constraints conflicts. Neither abstraction-level incompatibilities, multilateral correspondences or meta-level discrepancies occur if only one agreed-upon metadata model exists. If all parties involved use the same model, it cannot be that some concepts are not available in a model, which implies that model agreement techniques also resolve domain coverage conflicts. Further, a standardized or agreed-upon schema or application profile can also resolve domain conflicts by fixing the semantic domain (e.g., application profile for the domain of videos or audio material).

The remaining semantic heterogeneity conflicts on the instance level (scaling/unit and representation conflicts) can also be resolved through the combination of constraints on the model level and agreement on the instance level (value encoding, controlled vocabulary, authority record) it is possible to narrow the domain of possible content values within a metadata description to a fixed set of values. Hybrid metadata systems, such as TV-Anytime, also span the M0 level by defining fixed classification schemes, and therefore also provide interoperability on the instance level.

5.2. MetaModel Agreement Techniques

Metamodel agreement techniques such as global conceptual models or metadata frameworks are less powerful than model-agreement techniques. Rather than agreeing on a certain model, their approach is to impose a metamodel and use generalization relationships (e.g., `sub-class` or `sub-property`) to relate the elements of existing proprietary models to the elements of the common metamodel.

These alignment possibilities are very restricted: they cannot deal with instance-level heterogeneities, nor handle structural heterogeneities. Figure 10 illustrates this problem based on the example presented in Section 2.1. It shows the TV-Anytime and the Dublin Core elements for representing the name of a person who has created a certain resource. The TV-Anytime model defines two separated fields, `GivenName` and `FamilyName`, while the Dublin Core model defines only a single field, `Creator`, to capture the same information. A global conceptual model containing the elements `Person` and `Name` has been introduced to bridge this structural heterogeneity conflict. We can see that global conceptual models cannot deal with basic heterogeneity conflicts such as multilateral correspondences. It is not possible to relate the elements `GivenName` and `FamilyName` with the element `Name` in a way that machines can process their instance content values appropriately.

Other types of heterogeneities, which are not resolvable for metamodel agreement techniques, are meta-level discrepancies (e.g., `Name` modeled as entity instead of an attribute) and domain coverage conflicts. Concepts available in the global model may simply not be explicitly available in the proprietary models. The heterogeneities that can be resolved are abstraction-level incompatibilities and, in the case of global conceptual models, domain conflicts when the models' domains are not completely incompatible. Unlike global conceptual models, metadata frameworks are domain-independent and cannot resolve domain conflicts by imposing a certain domain. As we can see in the example, both interoperability approaches could resolve terminological mismatches by aligning terminologically conflicting elements to a common element in the global model (e.g., `Creator` sub-property `Name`).

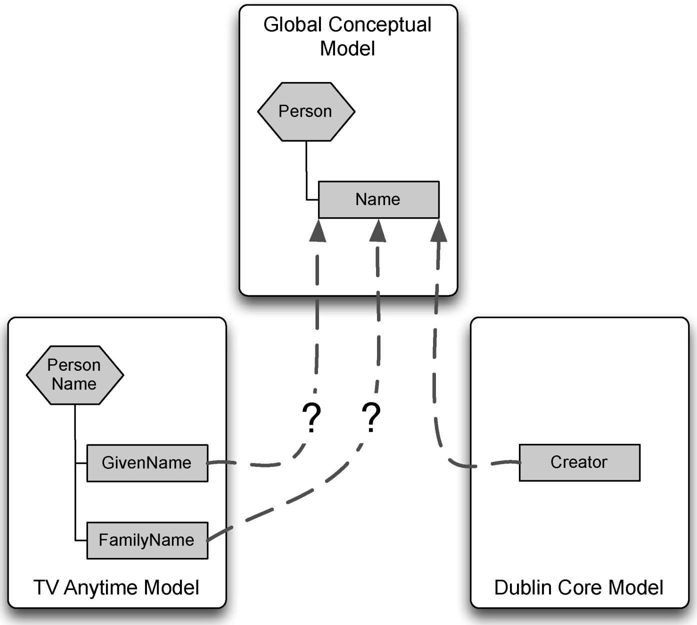


Fig. 10. Example for achieving interoperability via a global conceptual model.

In general, the problems with global conceptual models are manifold: first, it is hard to find a model that covers all possible ontological requirements of all systems in an integration context. Second, the generic nature and complexity of global models can also lead to varying interpretations and inconsistent alignments between the global conceptual models and the metadata schemes in place. Third, conceptual models (e.g., the CIDOC CRM) often lack any technical specifications, with the result that they are implemented differently in distinct systems. Meanwhile the belief in the success of global conceptual model approaches is decreasing: Wache [2003] asserts that *no global model can be defined in such a way that it fulfils all conceptual requirements of all possible information systems that are integrated in a certain domain*. Halevy et al. [2005] argue that *in large-scale environments global models, which should enable interoperability, actually become the bottleneck in the process of achieving interoperability*.

5.3. Model Reconciliation Techniques

Schema mapping (metadata crosswalks) is powerful enough to produce the same interoperability quality as model agreement techniques. Provided that the underlying mapping mechanism is strong enough, schema mapping can deal with all kinds of heterogeneities on the schema level, such as different element names, different ways of identifying these elements, incompatible constraint definitions, and all the remaining conflicts, ranging from abstraction-level incompatibilities to terminological mismatches. In combination with instance transformation, it can also resolve semantic heterogeneities on the instance level, that is, scaling/unit and representation conflicts.

5.4. Observations on the Quality of Interoperability Techniques

Regarding our analysis, we can observe that there exist various options for providing interoperability among heterogeneous metadata information objects. Each option has its special qualities, and can be seen as complementary building block for achieving metadata interoperability. Standardized metadata schemes alone, for instance, cannot deal with instance-level heterogeneities. Hence, they must be combined with value-encoding schemes, controlled vocabularies, or authority records in order to also address the instance level. The same is the case for schema mappings or crosswalks; they operate on the schema level only and must be combined with instance transformation techniques to achieve maximum interoperability. Metamodel agreement techniques such as global conceptual models or metadata frameworks have the disadvantage that they provide only a restricted set of alignment relationships (`sub-class`, `sub-property`). Furthermore, they can hardly be combined with instance-level interoperability techniques.

From a purely technical view, domain experts should envisage the following techniques for establishing metadata interoperability. They have an equally high potential for resolving the various types for semantic and structural heterogeneities:

—*metadata standardization:* hybrid metadata systems or standardized metadata schema in combination with value-encoding schemes, controlled vocabularies, or authority records.

—*metadata mapping:* schema mapping in combination with instance transformation.

In reality, however, the choice of an interoperability technique also depends on other, nontechnical issues: especially on the effort required and the costs of adopting a certain technique will influence such a decision in most organizations. Generally, we can see that any interoperability strategy will increases in cost with an increase in functionality [Lagoze and de Sompel 2001; Arms 2000], due to the growing implementation effort. In the following, we outline a set of potential cost factors that should be considered when choosing between metadata standardization and metadata mapping.

For metadata standardization we can identify the following costs (e.g., Buxmann et al. [1999]).

—*Licensing Costs*. While some metadata standards can be adopted free of charge (e.g., Dublin Core), some standards, especially those that are hosted by accredited standardization organizations, entail costs. The MPEG-7 specification, for instance, currently consists of 11 parts, and each part costs between approximately **€**65 and **€**180.

—*Software Costs*. The creation of standard-conformant metadata often requires licenses for specific software tools or libraries. Extensions or upgrades of the underlying storage infrastructures, which are required to meet the needs of a certain standard (e.g., XML support for MPEG-7, RDF support for Dublin Core, Spatial RDB extensions for ISO 19115), can lead to further investment.

—*Hardware Costs*. The adoption of additional software often entails upgrades to the technical infrastructure.

—*Personnel Costs*. Setting up a project and hiring personnel for implementing a metadata standard is the largest portion of the total cost of standard-adoption. The more complex a standard is, the longer such a project will last, and the more personnel effort will be required. Furthermore, consultancy from experts familiar with the standard might be necessary in order to guarantee a correct standard adoption.

Besides the licensing costs, all previously mentioned costs will also arise if metadata mapping is chosen as the interoperability technique: purchasing mapping solutions causes software costs, additional hardware may be necessary for deploying mappings in the execution phase, and personnel is required to implement the mappings. Additionally, we can identify the following additional, mapping-specific costs.

—*Mapping Discovery Costs.* Determining mapping relationships requires experts who are familiar with both the source and target schema. Mapping discovery can of course be supported by (semi-)automatic matching tools.

Obviously, significant costs may occur for correction of errors in adopting metadata standards or due to incorrect mapping relationships caused by matching tools or the domain experts themselves. The sooner such errors are discovered, the lower the follow-up costs will be.

Although additional costs related to mapping discovery may have to be considered in the context of metadata mapping, they are not always more costly than the adoption of metadata standards. This in fact depends very much on the complexity of the standard to be adopted, and clearly needs further investigation based on the facts of a concrete application. For a more comprehensive discussion on adoption costs, we refer to Karampiperis et al. [2003]. They compare the standardization and mapping costs with regard to the level of standardization and discuss the optimal solution, which is a trade-off between both techniques: if metadata is standardized to a certain extent, less mapping effort is required and vice versa.

6. SUMMARY AND CONCLUSIONS

As we can clearly see in the discussions in this article, metadata interoperability affects all technical levels of metadata: the M2 level of schema definition languages, the M1 level of metadata schemes, and the M0 level of metadata instances. For achieving metadata interoperability, several types of structural and semantic heterogeneities must be resolved on each of these levels. We distinguish between three categories of interoperability techniques: agreement on a certain model, agreement on a certain metamodel, and model reconciliation.

From our analysis, we can observe that model-agreement techniques, such as hybrid metadata systems and standardized metadata schemes, as well as model reconciliation techniques, that is, metadata mapping, cover large parts of possible heterogeneities if they are combined with appropriate techniques on the instance level: metadata standards should be applied in combination with value-encoding schemes, controlled vocabularies, or authority records. Metadata mapping should also consider M0-level heterogeneities and support instance transformation.

Global conceptual models and metadata frameworks rely on restricted means for relating source model elements with those of a global model and do not consider the instance level. Therefore, one outcome of our analysis is that these techniques are less powerful than metadata standardization or mapping. Comparing standardization and mapping, the clear disadvantage of mapping is its technical complexity. However, in open environments having no central standardization authority, metadata mapping is the remaining, more complex, but equally powerful technique.

The Web is such an open environment. It already provides a multitude of autonomous, incompatible media repositories, and it is unlikely that there will ever exist a single agreed-upon metadata schema. Like Franklin et al. [2005], we also believe that, in future, many institutions and organizations relying on a large number of diverse, interrelated media sources will make use of the Web architecture to access available repositories. Since also in the Web context, the metadata information objects in these repositories are not compatible by default, novel mapping techniques that build upon the Web infrastructure and are powerful enough to deal with the heterogeneities we outlined in this article are required.

ACKNOWLEDGMENTS

The authors would like to thank Wolfgang Jochum and Bernhard Schandl as well as our referees for their valuable comments on this article.

REFERENCES

ADL. 2007. Sharable Content Reference Model (SCORM). Advanced Distributed Learning Initiative (ADL). http://www.adlnet.gov/scorm/index.aspx.

ALCTS CC:DA. 2000. Task force on metadata: Final report. Association for Library Collections and Technical Services (ALCTS). http://www.libraries.psu.edu/tas/jca/ccda/tf-meta6.html.

Allinson, J., Johnston, P., and Powell, A. 2007. A Dublin Core application profile for scholarly works. http://www.ariadne.ac.uk/issue50/allinson-et-al/.

Ambler, S. 2003. *Agile Database Techniques*: *Effective Strategies for the Agile Software Developer*. Wiley, New York.

Arms, W. Y. 2000. *Digital Libraries*. MIT Press, Cambridge, MA.

Atay, M., Chebotko, A., Liu, D., Lu, S., and Fotouhi, F. 2007. Efficient schema-based XML to-relational data mapping. *Inform. Syst. 32*, 3, 458–476.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, Cambridge, UK.

Baker, T., Blanchi, C., Brickley, D., Duval, E., Heery, R., Johnston, P., Kalinichenko, L., Neuroth, H., and Sugimoto, S. 2002. Principles of metadata registries. White paper, DELOS Network of Excellence on Digital Libraries.

Baker, T., Dekkers, M., Heery, R., Patel, M., and Salokhe, G. 2001. What terms does your metadata use? Application profiles as machine-understandable narratives. In *Proceedings of the International Conference on Dublin Core and Metadata Applications*. National Institute of Informatics, 151–159.

Benitez, A. B., Zhong, D., Chang, S.-F., and Smith, J. R. 2001. MPEG-7 MDS content description tools and applications. In *Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns (CAIP'01)*. Springer, Berlin, 41–52.

Bernauer, M., Kappel, G., and Kramler, G. 2004. Representing XML schema in UML—A comparison of approaches. In *Web Engineering*, N. Koch et al. eds., Lecture Notes in Computer Sciences, vol. 3140. Springer, Berlin, 440–444.

Berners-Lee, T. 1998. Notation 3. Design note. *World Wide Web Consortium* (W3C). http://www.w3.org/DesignIssues/Notation3.

Bernstein, P. A., Halevy, A. Y., and Pottinger, R. A. 2000. A vision for management of complex models. *SIGMOD Rec. 29*, 4, 55–63.

Bizer, C. and Seaborne, A. 2004. D2RQ—Treating non-RDF databases as virtual RDF graphs. In *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*. http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/.

Bracha, G. and Ungar, D. 2004. Mirrors: Design principles for meta-level facilities of object-oriented programming languages. In *Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'04)*, ACM, New York, 331–344.

Buxmann, P., Weitzel, T., Westarp, F. V., and König, W. 1999. The standardization problem—An economic analysis of standards in information systems. In *Proceedings of the 1st IEEE Conference on Standardization and Innovation in Information Technology (SIIT99)*, IEEE, Los Alamitos, CA, 157–162.

Cardelli, L. and Wegner, P. 1985. On understanding types, data abstraction, and polymorphism. *ACM Comput. Surv. 17*, 4, 471–523.

CCSDS. 2002. Open archival information systems. OAIS. *Council of the Consultative Commitee for Space Data Systems (CCSDS)*. http://public.ccsds.org/publications/archive/650x0b1.pdf.

Chan, L. M. and Zeng, M. L. 2006. Metadata interoperability and standardization: A study of methodology. *D-LIB Magazine 12*, 6. Parts I + II. http://www.dlib.org/dlib/june06/chan/06chan.html.

Chen, P. P.-S. 1976. The entity-relationship model—toward a unified view of data. *ACM Trans. Datab. Syst. 1*, 1, 9–36.

Codd, E. F. 1970. A relational model of data for large shared data banks. *Comm. ACM 13*, 6, 377–387.

DC. 2007. Dublin Core Collections Application Profile. Dublin Core Metadata Initiative (DC). http://dublincore.org/groups/collections/collection-application-profile/.

DC. 2006. Dublin Core Metadata Element Set, Version 1.1. Dublin Core Metadata Initiative. http://dublincore.org/documents/dces/.

DNB. 2007a. Maschinelles Austauschformat für Bibliotheken. German National Library. Expert group for data formats. http://www.d-nb.de/standardisierung/formate/mab.htm.

DNB. 2007b. Personennormdatei (PND). German National Library. http://www.d-nb.de/standardisierung/normdateien/pnd.htm.

EDItEUR. 2007. Online Information Exchange (ONIX). The EDItEUR Group. http://www.editeur.org/onix.html.

ETSI. 2006. TV Anytime: TS 102 822:1-7. European Telecommunications Standards Instititute (ETSI). http://www.etsi.org/etsisite/website/technologies/tvanytime.aspx.

Fong, J. 1997. Converting relational to object-oriented databases. *SIGMOD Rec. 26*, 1, 53–58.

Franklin, M., Halevy, A., and Maier, D. 2005. From databases to dataspaces: A new abstraction for information management. *SIGMOD Rec. 34*, 4, 27–33.

Gasevic, D., Djuric, D., Devedzic, V., and Damjanovi, V. 2004. Converting UML to OWL ontologies. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters (WWW Alt.'04)*. ACM, New York, 488–489.

Gilliland, A. J. 2005. Introduction to metadata: Pathways to digital information. http://www.getty.edu/research/conducting_research/standards/intrometadata/ index.html.

Gosling, J., Joy, B., Steele, G., and Bracha, G. 2005. *The Java Language Specification* (3rd Ed.). Addison-Wesley, Reading, MA.

Gruber, T. 1993. A translation approach to portable ontology specifications. *Knowl. Acquisit. 5*, 199–220.

Halevy, A. Y. 2001. Answering queries using views: A survey. *VLDB J. 10*, 4, 270–294.

Halevy, Y., Ives, G., Suciu, D., and Tatarinov, I. 2005. Schema mediation for large-scale semantic data sharing. *VLDB J. 14*, 1, 68–83.

Haslhofer, B. 2008. A comparative study of mapping solutions. Tech. Rep., University of Vienna. http://www.cs.univie.ac.at/publication.php?pid=3886.

Heery, R. and Patel, M. 2000. Application profiles: Mixing and matching metadata schemas. http://www.ariadne.ac.uk/issue25/app-profiles/.

Hunter, J. and Lagoze, C. 2001. Combining RDF and XML schemas to enhance interoperability between metadata application profiles. In *Proceedings of the 10th International Conference on the World Wide Web (WWW'01)*, ACM, New York, 457–466.

IEEE WG-12. 2002. IEEE standard for learning object metadata: 1484.12.1-2002. IEEE, Washington, D.C. http://ltsc.ieee.org/wg12.

IFLA. 1997. Functional requirements for bibliographic records. Study Group on the Functional Requirements for Bibliographic Records, International Federation of Library Associations (IFLA). http://www.ifla.org/VII/s13/frbr/frbr.htm.

ISO TC 154. 2004. Data elements and interchange formats–Information exchange–Representation of dates and times. ISO 8601:2004. International Standardization Organization (ISO). http://www.iso.org/iso/catalogue_detail?csnumber=40874.

ISO TC 211. 2003. Geographic information metadata. ISO 19115:2003. International Standardization Organization (ISO). http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26020.

ISO TC 46. 2006a. CIDOC conceptual reference model (CRM). ISO 21127:2006. International Standardization Organization (ISO).

ISO TC 46. 2006b. Codes for the representation of names of countries and their subdivisions. Part 1: Country codes. ISO 3166-1:2006. International Standardization Organization (ISO). http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm? csnumber=39719.

ISO/IEC JTC 1/SC 29. 2007a. MPEG-21 multimedia framework. ISO 21000-17:2003-2007. International Standardization Organization (ISO).

ISO/IEC JTC 1/SC 29. 2007b. MPEG-7 multimedia content description interface. ISO 15938-1-11:2002-2007. International Standardization Organization (ISO).

ISO/IEC JTC 1/SC 32. 2003. SQL - ISO/IEC 9075-1:2003. International Standardization Organization (ISO).

ISO/IEC JTC 1/SC 32. 2005. Common logic (CL). A framework for a family of logic-based languages. ISO/IEC 24707:2007. International Standardization Organization (ISO).

ISO/IEC JTC 1/SC 34. 2006. Topic maps. Part 2: Data model. ISO/IEC 13250-2:2006. International Standardization Organization (ISO).

Java Community Process. 2006. JSR 269: Pluggable annotation processing API. http://jcp.org/en/jsr/detail?id=269.

JOHNSTON, P. 2004. Minerva—Technical guidelines for digital cultural content creation programs. Tech. Rep., UKOLN, University of Bath. MLA The Council for Museums, Libraries and Archives. http://www.minervaeurope.org/structure/workinggroups/servprov/ documents/techguid1_0.pdf.

KALFOGLOU, Y. AND SCHORLEMMER, M. 2003. Ontology mapping: the state of the art. *Knowl. Eng. Rev. 18*, 1, 1–31.

KARAMPIPERIS, P., KASTRADAS, K., AND SAMPSON, D. 2003. A schema-mapping algorithm for educational metadata interoperability. In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA'03)*. D. Lassner and C. McNaught (Eds.), Association for the Advancement of Computing in Education (AACE).

KENSCHE, D., QUIX, C., CHATTI, M. A., AND JARKE, M. 2007. Gerome: A generic role-based metamodel for model management. *J. Data Semantics 8*, 82–117.

KOSCH, H. 2003. *Distributed Multimedia Database Technologies Supported by MPEG-7 and MPEG-21.* CRC Press LLC, Boca Raton, FL.

LAGOZE, C. AND DE SOMPEL, H. V. 2001. The open archives initiative: Building a low-barrier interoperability framework. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries.* ACM, New York, 54–62.

LEE, D., MANI, M., AND CHU, W. W. 2003. Schema conversion methods between xml and relational models. In *Knowledge Transformation for the Semantic Web*, IOS. Press, 1–17.

LETHI, P. AND FRANKHAUSER, P. 2004. XML data integration with OWL: Experiences and Challenges. In *Proceedings of the Symposium on Applications and the Internet (SAINT'04)*. IEEE, Los Alamitos, CA, 160–170.

LOC. 2007a. Library of Congress Authorities. Library of Congress (LOC). http://authorities.loc.gov/.

LOC. 2007b. Library of Congress Subject Headings (LCSH). Library of Congress. http://www.loc.gov/aba/cataloging/subject/.

LOC. 2007c. MARC 21 concise format for bibliographic metadata. Library of Congress (LOC) Network Development and MARC Standards Office. http://www.loc.gov/marc/bibliographic/ecbdhome.html.

LOC. 2007d. Metadata object description schema. Library of Congress (LOC) Network Development and MARC Standards Office. http://www.loc.gov/standards/mods/.

LOC. 2007e. METS (Metadata Encoding and Transmission Standard). Library of Congress (LOC) Network Development and MARC Standards Office. http://www.loc.gov/standards/mets.

MADHAVAN, J., BERNSTEIN, P. A., DOMINGOS, P., AND HALEVY, A. Y. 2002. Representing and reasoning about mappings between domain models. In *Proceedings of the 18th National Conference on Artificial intelligence*. American Association for Artificial Intelligence, 80–86.

MCPARLAND, A. 2002. TV-Anytime: Using all that extra data. Tech. rep., BBC. http://www.bbc.co.uk/rd/pubs/whp/whp-pdf-files/WHP050.pdf.

MENA, E., ILLARRAMENDI, A., KASHYAP, V., AND SHETH, A. P. 2000. Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distrib. Para. Datab. 8*, 2, 223–271.

MILLER, P. 2000. Interoperability. What is it and why should i want it? http://www.ariadne.ac.uk/issue24/interoperability/intro.html.

MOTIK, B., HORROCKS, I., AND SATTLER, U. 2007. Bridging the gap between OWL and relational databases. In *Proceedings of the 16th International Conference on the World Wide Web (WWW'07).* ACM, New York, 807–816.

NILES, I. AND PEASE, A. 2001. Towards a standard upper ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS'01)*, ACM, New York, 2–9.

NISO. 2004. Understanding metadata. National Information Standards Organization (NISO). http://www.niso.org/standards/resources/UnderstandingMetadata.pdf.

NLM. 2007. Medical subject headings. U.S. National Library of Medicine (NLM). http://www.nlm.nih.gov/mesh/.

NOY, N. F. 2004. Semantic integration: A survey of ontology-based approaches. *SIGMOD Rec. 33*, 4, 65–70.

NOY, N. F. AND KLEIN, M. 2004. Ontology evolution: Not the same as schema evolution. *Knowl. Inform. Syst. 6*, 4, 428–440.

NOY, N. F. AND MUSEN, M. A. 2004. Ontology versioning in an ontology management framework. *IEEE Intell. Syst. 19*, 4, 6–13.

NWG. 1995. A format for bibliographic records (RFC 1807). Network Working Group (NWG). http://rfc.net/rfc1807.html.

OCLC. 2007. Dewey decimal classification (DDC). Online Computer Library Center (OCLC). http://www.oclc.org/dewey/.

OGC. 2004. Geography markup language. Tech. rep., Open Geospatial Consortium (OGC). http://portal.opengeospatial.org/files/?artifact_id=4700.

OMG. 2006a. Meta Object Facility (MOF) core specification—Version 2.0. Object Management Group (OMG). http://www.omg.org/cgi-bin/apps/doc?formal/06-01-01.pdf.

OMG. 2006b. Ontology definition metamodel specification (ODM). Object Management Group (OMG). http://www.omg.org/docs/ptc/06-10-11.pdf.

OMG. 2006c. UML 2.0: Infrastructure specification. Object Management Group (OMG). http://www.omg.org/docs/ptc/03-09-15.pdf.

OMG. 2007a. MOF 2.0 / XMI Mapping Specification, V2.1.1. Object Management Group (OMG).

OMG. 2007b. Unified Modelling Language (UML). Object Management Group (OMG). http://www.uml.org/.

Ouksel, A. M. and Sheth, A. 1999. Semantic interoperability in global information systems. *SIGMOD Rec. 28*, 1, 5–12.

Pierre, M. S. and LaPlant, W. P. 1998. Issues in crosswalking content metadata standards. Tech. rep., National Information Standards Organization (NISO). http://www.niso.org/press/whitepapers/crsswalk.html.

Powell, A., Nilsson, M., Naeve, A., and Johnston, P. 2005. DCMI abstract model. Dublin Core Metadata Initiative (DC). http://dublincore.org/documents/abstract-model/.

Rahm, E. and Bernstein, P. A. 2001. A survey of approaches to automatic schema matching. *VLDB J. 10*, 4, 334–350.

Seidewitz, E. 2003. What models mean. *IEEE Softw. 20*, 5, 26–32.

Sheth, A. and Klas, W. 1998. *Multimedia Data Management: Using Metadata to Integrate and Apply Digital Media.* McGraw-Hill, New York.

Sheth, A. P. and Larson, J. A. 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv. 22*, 3, 183–236.

Sintek, M. and Decker, S. 2002. Triple—A query, inference, and transformation language for the semantic web. In *Proceedings of the International Semantic Web Conference*. I. Horrocks and J. A. Hendler (Eds.), Lecture Notes in Computer Science, vol. 2342. Springer, Berlin, 364–378.

Spaccapietra, S., Parent, C., and Dupont, Y. 1992. Model independent assertions for integration of heterogeneous schemas. *VLDB Journal 1*, 1, 81–126.

TEI. 2007. TEI P5: Guidelines for electronic text encoding and interchange. TEI Consortium. http://www.tei-c.org/Guidelines/P5/.

Tillett, B. 2004. What is FRBR: A conceptual model for the bibliographic universe. http://www.loc.gov/cds/FRBR.html.

Tolk, A. 2006. What comes after the semantic web: PADS implications for the dynamic web. In *Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06)*, IEEE, Los Alamitos, CA, 55.

Visser, P. R. S., Jones, D. M., Bench-Capon, T. J. M., and Shave, M. J. R. 1997. An analysis of ontological mismatches: Heterogeneity versus interoperability. In *AAAI Spring Symposium on Ontological Engineering*.

VRA. 2007. VRA Core 4.0. Visual Resources Association's (VRA) Data Standards Committee. http://www.vraweb.org/projects/vracore4/index.html.

W3C. 2004a. RDF vocabulary description language 1.0: RDF schema. W3C semantic web activity—RDF Core Working Group. http://www.w3.org/TR/rdf-schema/.

W3C. 2004b. Web Ontology Language (OWL). W3C semantic web activity. Web Ontology Working Group. http://www.w3.org/2004/OWL/.

W3C. 2006. XML schema 1.1 Part 1: Structure. W3C XML Core Working Group. http://www.w3.org/TR/xmlschema11-1/.

Wache, H. 2003. Semantische mediation für heterogene informationsquellen. Ph.D. thesis, University of Bremen.

Westermann, U. and Klas, W. 2003. An analysis of XML database solutions for the management of MPEG-7 media descriptions. *ACM Comput. Surv. 35*, 4, 331–373.

WonderWeb Consortium. 2003. DOLCE: A descriptive ontology for linguistic and cognitive engineering. http://www.loa-cnr.it/DOLCE.html.

Received March 2008; revised August 2008; accepted October 2008