International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.1 (2014), pp.265-276
http://dx.doi.org/10.14257/ijmue.2014.9.1.25

Effective Indexed Data for Mobile Service Management

Younky Chung

Department of Computer Engineering, Kyungil University, Republic of Korea
ykchung@kiu.ac.kr

Abstract

Metadata is defined as data about data. In large scale distributed and heterogeneous software systems, metadata is considered as one of the most important components for the architecture and construction of information systems. Metadata is utilized to improve communication between heterogeneous information systems – for the purposes of obtaining and providing information, for communication between user client workstations and information servers, and for electronic business between information systems. Metadata management is much challenging and is a hot topic for research at present. In this paper, we review work that deals with effective indexed data for mobile service management metadata in three different points of view. Firstly, we survey some proposals about applying model management to classical metadata problems. Then we inspect how metadata can help managing multimedia data. Finally, we review some mobile service specific metadata management issues. We survey approaches that are of foundational nature as well as those that are application-oriented. We believe that combining both foundational and practical aspects is important to achieve the goal of managing metadata with the same ease as, for example, database management systems have achieved in the management of classical relational data.

Keywords: *Mobile service management metadata, Heterogeneous databases, Model management for metadata problems, Composition of metadata*

1. Introduction

Metadata are considered as the future of networked information systems. The ubiquity of the mobile services and the increasing need for access to heterogeneous distributed information and the increased use of multilingual and multimedia sources all demand some common representation and understanding of metadata [1]. Metadata is attached to data to aid in its interpretation. Mobile services systems process and interpret the data using the associated metadata which, in more and more cases, are getting very large. Hence the importance of dealing with an appropriate way of managing these large metadata sets.

Model management is a new approach to metadata management that offers a higher level programming interface than current techniques. The main abstractions are models (*e.g.*, schemas, interface definitions) and mappings between models. It treats these abstractions as bulk objects and offers operators for matching metadata objects, merging them, computing the difference between given metadata objects, composing these objects, and generating them. It is a powerful approach to metadata management which is generic in the sense that it not limited to a specific language or application. Generic model management [2, 3] aims at simplifying the development of metadata-intensive applications, such as data integration, software engineering, website management, and network modeling.

ISSN: 1975-0080 IJMUE

Such applications manipulate a variety of models (database schemas, XML schemas, UML / ER diagrams, ontologies, *etc.*) and mappings between models (SQL view definitions, XSLT transformations, XML-to-relational shredding specifications, ER-to-SQL DDL mappings, *etc.*).

Finding correspondences between models is required in many application domains, such as data integration, m-business, data warehousing, web services, and peer-to-peer semantic query processing.

This task is often referred to as matching. In generic model management, matching is embodied in the operator *Match*. The operator takes two models as input and returns a mapping between the models as output. This operator is special as matching typically involves information that is not contained in the input models [4].

As mobile services become more prevalent, tools will be needed to help users find, filter and integrate these services. MSDL (Mobile Services Description Language) is used as the metadata language of web services. Composing existing services to obtain new functionality will prove to be essential for both business-to-business and business-to-consumer applications. The dynamic composition of services is difficult using just the MSDL descriptions. This is essentially due to the hardness of dynamically dealing with metadata management [5].

In this paper, we review work that deals with effective indexed data for mobile service management metadata in three different points of view. Firstly, we survey some proposals about applying model management to classical metadata problems. Then we inspect how metadata can help managing multimedia data. Finally, we review some mobile service specific metadata management issues. We survey approaches that are of foundational nature as well as those that are application-oriented. We believe that combining both foundational and practical aspects is important to achieve the goal of managing metadata with the same ease as, for example, database management systems have achieved in the management of classical relational data.

2. Related Works

2.1. Meta Data: A short Overview

Metadata describe a data source, a particular collection of data (a file or a database or a table in a relational database or a class in an object-oriented database), an instance of data (tuple in a relational database table, object instance in a class within an object-oriented database) or data associated with the values of an attribute within a domain, or the particular value of an attribute in one instance. Metadata can describe data models. Metadata can also be used to describe processes and software. It can describe an overall processing system environment, a processing system, a process, a component of a process. It can describe a suite of software, a program, a subroutine or program fragment, a specification. It can describe an event system, an individual event, a constraint system and an individual constraint. It can describe a process and /or event model. Metadata can describe people and their roles in an Information Technology (IT) system. It can describe an organization, a department, individuals or individuals in a certain role [6].

The topic of Metadata has recently found more limelight than in the past, largely due to a sudden realization of its necessity in making the mobile services usable effectively. Metadata is essential for mobile services to scale up to an astronomical number of users, for finding information of relevance, and for integrating together data and information from heterogeneous sources. Metadata are essential for refining queries so the latter return that what the user intends. It is also essential for understanding the structure of information, its quality and its relevance. Metadata are required for explaining answers from ever more complex information systems. It assists in distilling knowledge from information and data. It assists in multilingualism and in multimedia representations. The engineering of systems from components (data, processes, software, events, and subsystems) is assisted by metadata descriptions of those components. Metadata have been used in information systems engineering for many years, but usually in a specialist, one-off and uncoordinated way.

Commonly, the metadata have been human readable, but not specified sufficiently formally, nor accepted sufficiently widely, to be interpreted unambiguously by IT systems. In addition to information systems such as mobile services (update, retrieval) and systems engineering as described above, metadata are essential for e-business from advertising through catalogue information provision through initial enquiry to contract, purchase, delivery and subsequent guarantee or maintenance.

2.2. Difficulties in Metadata Management

There are three identifiable types of difficulties in metadata management, namely metadata definition and management, technology, and standards [7]. Metadata definition and management is about defining, creating, updating, transforming, and migrating all types of metadata that are relevant and important to a user's objectives. Metadata management technology includes metadata design tools that allow users to model the schema of metadata across all data sources, and metadata repository systems that allow the users to extract metadata from various data sources, search and query metadata, and exchange metadata with other users, etc. Metadata standards include not only those for modeling and exchanging metadata, but also the vocabulary and knowledge ontology. These difficulties have stunted universal adoption of metadata management technologies. Standard knowledge ontology is also needed to organize such types of metadata as content metadata and data usage metadata. With respect to the vocabulary and knowledge ontology, where there are suitable industry standards, the standards may be adopted in full or in part. Appropriate procedures need to be defined and followed within the enterprise in documenting the capture, update, transformation, migration, replication of metadata and relevant transformation rules and business rules, *etc.*, [8].

2.3. Future of Metadata

Metadata have moved centre-stage as one of the most important components of the architecture and construction of modern information systems. The idea of separating the primary information resources from data and processes (metadata system) to provide access to those resources is extremely important. This allows changes of access policy – such as changes in access restrictions for certain kinds of users in certain roles, changes in categorization and classification, and changes in descriptive metadata depending on viewpoints of different authorized users – without accessing the data resource itself [9].

People who are publishing valuable information to the Internet want to be able to create or at least to control the metadata describing their resources. Metadata that are generated not by a single entity such as a search engine, but by many different entities requires some recognized standard metadata formats. Without standard metadata formats and semantics, metadata would be just as unprocessable and unmanageable as the original data. Existing barriers in business, modeling, and technology will have to be addressed for metadata in order for them to play the important role of alleviating barriers between heterogeneous users and applications. Metadata collection has attained a sufficient level of maturity; however, metadata management today is at an elementary phase. In the future, there is a need for an extended ecology of metadata artifacts that will constantly evolve [10]. Before increased automatic metadata management can be readily exploited in enterprise activities, such as e-commerce, education, and government, metadata will require extensible models, richer nuances, and underlying trust mechanisms. The future of the Internet will rely on this evolution.

3. Generic Model Management

Many information system problems involve the design, integration, and maintenance of complex application artifacts, such as application programs, databases, web sites, workflow scripts, formatted messages, and user interfaces. Engineers who perform this work use tools to manipulate formal descriptions, or models, of these artifacts, such as object diagrams, interface definitions, database schemas, web site layouts, control flow diagrams, XML schemas, and form definitions. This manipulation usually involves designing transformations between models, which in turn requires an explicit representation of mappings, which describe how two models are related to each other. Some examples are given in as follows:

- mapping between class definitions and relational schemas to generate object wrappers,
- mapping between XML schemas to drive message translation,
- mapping between data sources and a mediated schema to drive heterogeneous data integration,
- mapping between a database schema and its next release to guide data migration or view evolution, mapping between an entity-relationship (ER) model and a SQL schema to navigate between a database

Today's approach to implementing such applications is to translate the given models into an object-oriented representation and manipulate the models and mappings in that representation. Most of manipulation is programmed using object-at-a-time primitives. In existing works, the authors have proposed to avoid this object-at-a-time programming by treating models and mappings as abstractions that can be manipulated by model-at-a-time and mapping-at-a-time operators. They believe that an implementation of these abstractions and operators, called a model management system, could offer an order-of magnitude improvement in programmer productivity for metadata applications.

3.1. Models and Mappings

Models: Models are defined in as a set of objects, each of which has properties, has-a relationships, and associations. A model is identified by the root object and includes exactly the set of objects reachable from the root by paths of has-a relationships.

Mappings: A mapping between models *M1* and *M2* is a model, *map12*, and two morphisms, one between *map12* and *M1* and another between *map12* and *M2*. Thus, each object *m* in mapping *map12* can relate a set of objects in *M1* to a set of objects in *M2*, namely the objects that are related to *m* via the morphisms. For example, in Figure 1, Map_{ee} is a mapping between models *Emp* and *Employee*, where has-a relationships are represented by solid lines and morphisms by dashed lines.

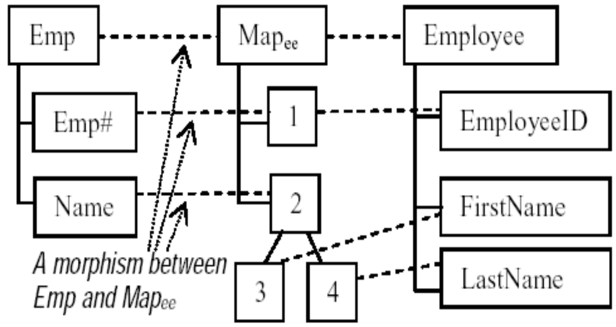

Figure 1. An Example of Mapping [3]

Model Management Operators: In a model management system, models and mappings are syntactic structures. They are expressed in a type system, but do not have additional semantics based on a constraint language or query language. Despite this limited expressiveness, model management operators are powerful enough to avoid most object-at-a-time programming in metadata applications. For a complete solution, metadata problems often require some semantic processing. Summary of the main model management operators are as follows:

Match – takes two models as input and returns a mapping between them

Compose – takes a mapping between models A and B and a mapping between models B and C, and returns a mapping between A and C

Diff – takes a model A and mapping between A and some model B, and returns the submodel of A that does not participate in the mapping

ModelGen – takes a model A, and returns a new model B based on A (typically in a different data model than A's) and a mapping between A and B

Merge – takes two models A and B and a mapping between them, and returns the union C of A and B along with mappings between C and A, and C and B.

Application Scenarios: The operators mentioned in the previous section can be used in various applications like Schema Integration, Schema Evolution, Round-trip Engineering etc. The following example [3] illustrates how the operators might be used to generate data warehouse loading script.

Problem: Given a mapping *map1* from a data source *S1* to a data warehouse *SW*, another mapping is required to be created between a second source *S2* to *SW*, where *S2* is similar to *S1*. Figure 2 depicts the problem.

Solution: The following steps will solve the problem.

Call Match(S1, S2) to obtain a mapping map2 between S1 and S2

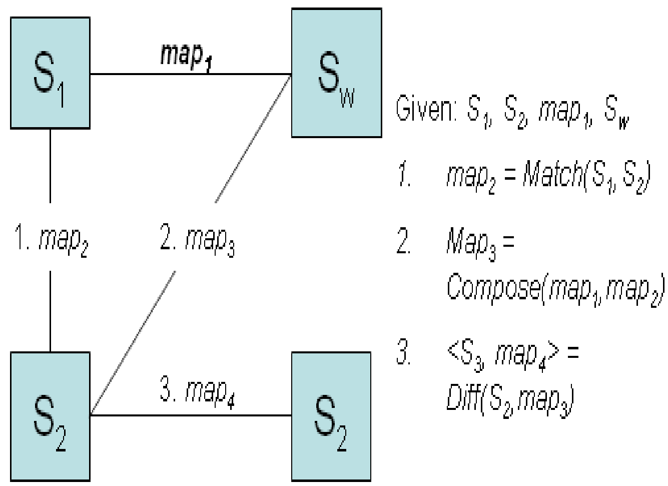

Figure 2. Model Management to Generate a Data

Call Compose(map1, map2) to obtain a mapping map3 between S2 and SW, which maps to SW those objects of S2 that correspond to objects of S1

Call Diff(S2, map3) to find the sub-model S3 of S2 that is not mapped by map3 to SW, and map4 to identify corresponding objects of S2 and S3

Call ModelGen(S3) to generate a warehouse schema for S3 and merge it into SW.

The Figure 3 shows the modeling the mapping model using UML

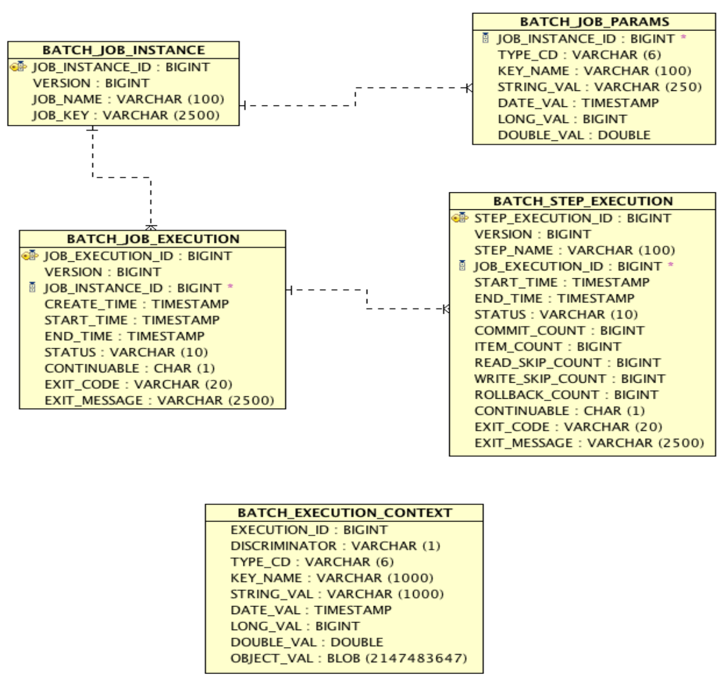

Figure 3. Model Mapping Modeling using UML

3.1.1. Future Scope of Generic Model Management

Developing formal semantics for the operators that combines the state-based and a more structural approach, developing practical materialization algorithms, finding appropriate architectures and techniques for coupling model management applications, tools, and conventional programming languages, developing powerful user interfaces for building model-management solutions and supporting user feedback during script execution, finding mechanisms for deducing equivalence and entailment of scripts, etc. Furthermore, applying model management to practical problems will help validating the algebraic framework exemplified in the work of Melnik [16] in the same way the first practical relational database management systems (such as System R and Postgres) helped exemplify Codd's relational model.

3.2. Ontologies for Metadata Management on Mobile Services

Ontology typically contains a hierarchy of concepts within a domain and describes each concept's crucial properties through an attribute-value mechanism. Further relations between concepts might be described through additional logical sentences. Constants are assigned to one or more concepts in order to assign them their proper type. Ontologies may play a major role in supporting the metadata management over the web.

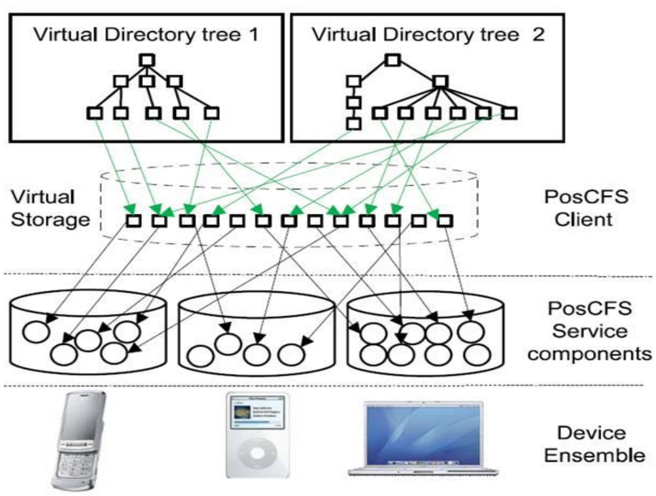

Figure 4. Company Ontology

4. Metadata for Mobile Services

Metadata play a far more important role in managing multimedia data than does the management of traditional (well-) structured data or information retrieval techniques applied to text-only data. The following section highlights the necessity of metadata for digital media

4.1. Concept for Metadata Mobile Services

Various digital media or components of multimedia data involve vary large raw data volume. This has consequences on effective management and retrieval of the digital media. Content-based retrieval on raw data means that the query capabilities are limited to the number of available matching algorithms. Performance is lacking when queries are executed on large data sets. The use of metadata of the digital media seem to be a promising approach to enhance querying and processing and to improve response time as metadata will be of much less than the digital media themselves [5].

We can attach audio, visual, annotation, and content management description tools to the segments to describe them in detail. MPEG-7 Visual description tools include the visual basic structures (such as description tools for grid layout, time series, and spatial coordinates) and visual description tools that let us describe color, texture, shape, motion, localization, and faces. MPEG-7 Audio description tools comprise the audio description framework and high-level audio description tools that let us describe musical instrument timbre, sound recognition, spoken content, and melody. The Semantic description tools that let us describe the content with real-world semantics and conceptual notions: objects, events, abstract concepts, and relationships. We can cross- link the semantic and structure description tools with a set of links.

The MPEG-7 description tools are a library of standardized Descriptors and Description Schemes. This library is presented on the basis of the functionality they provide, but in practice, we can combine them into meaningful sets of description units making use of the Schema tools. Each application builder might want to select a subset of Descriptors and Description Schemes.

MPEG-7 definitions are expressed solely in XML Schema [12]. XML Schema has been ideal for expressing the syntax, structural, cardinality and datatyping constraints required by MPEG-7. In order to make MPEG-7 accessible, re-usable and interoperable with other domains the semantics of the MPEG-7 metadata terms need to be expressed in an ontology using a machine-understandable language. There is scope for building such an ontology represented in more expressive languages.

4.2. Modeling the Mobile Services Metadata

Mobile Service are application components. They communicate using open protocols and are self-contained and self-describing, they can be discovered using UDDI and be used by other applications. XML is the basis for Web services. Basic web services platform elements are SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration), and MSDL (Mobile Services Description Language). Among them, MSDL is used as a metadata description language for web services.

MSDL is an XML-based language for describing Web services and how to access them. It is written in XML. MSDL is used to describe Web services. It also specifies the location of the service and the operations (or methods) the service exposes.

A MSDL document defines a web service using these major elements as Table 1:

Table 1. Mobile Meta Elements and Definition

Element	Defines
<portType>	The operations performed by the Mobile service
<message>	The messages used by the Mobile service
<types>	The data types used by the Mobile service
<binding>	The communication protocols used by the Mobile service

The main structure of a MSDL document looks like the following:

```
<definitions>
<types>
  definition of types........
</types>
<message>
  definition of a message....
</message>
<portType>
  definition of a port.......
</portType>
<binding>
  definition of a binding....
</binding>
</definitions>
```

A MSDL document can also contain other elements, like extension elements and a service element that makes it possible to group together the definitions of several web services in one single MSDL document. The <portType> element is the most important MSDL element. It defines a web service, the operations that can be performed, and the messages that are involved. The <portType> element can be compared to a function library (or a module, or a class) in a traditional programming language. The <message> element defines the data elements of an operation. Each message can consist of one or more parts. The parts can be compared to the parameters of a function call in a traditional programming language. The <types> element defines the data type that are used by the web service. The <binding> element defines the message format and protocol details for each port.

MSDL defines the interface of a Mobile service in terms of what are the messages that are exchanged. A MSDL document also structures the messages into pairs (that correspond to the operations provided by a service

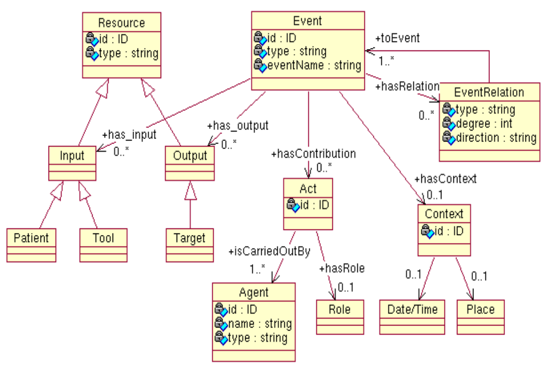

Figure 5. Class Definition for Mobile Services Metadata

4.3. Practical Cases

The use of metadata in information systems is not new. But earlier generations of metadata management systems did not provide adequate facilities for managing metadata and there were no standards for metadata management tasks. Keeping this in mind, we reviewed some metadata related research trends and tried to summarize them in this paper.

```
   <message name="getTermRequest">
      <part name="term" type="xs:string"/>
   </message>

   <message name="getTermResponse">
      <part name="value" type="xs:string"/>
   </message>
   <portTypename="glossaryTerms">
  <operation name="getTerm">
     <input message="getTermRequest"/>
     <output message="getTermResponse"/>
  </operation>
</portType>
   <binding type="glossaryTerms" name="b1">
   <soap:binding style="document"
   transport="http://schemas.xmlsoap.org/soap/http"/>
     <operation>
       <soap:operation
        soapAction="http://example.com/getTerm"/>
       <input>
         <soap:body use="literal"/>
       </input>
       <output>
         <soap:body use="literal"/>
       </output>
     </operation>
</binding>
```

Matching of two semantically similar metadata description (Schema Matching) is the most challenging task of Metadata management. So, we gave special attention to it.

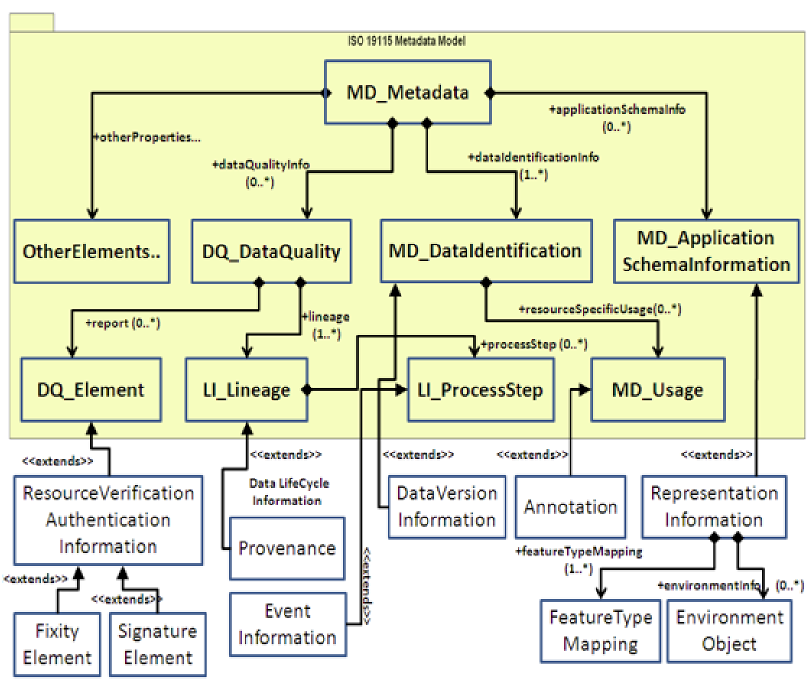

Figure 5. Mapping Hierarchy for Mobile Services Metadata

5. Conclusion and Future Works

This paper has reviewed metadata related research from a certain number of different points of view, namely Model Management for metadata, Multimedia metadata, and Web Service related metadata. There are many lessons that can be learnt from the main trends presented in terms of future directions about the research on metadata. Principled study of metadata: Metadata is an area that has been plagued with ad hoc solutions. The whole intent of work on metadata management has been to remove as much as possible ad hoc solutions from the handling of the main metadata management tasks in favor of a more principled way of handling these tasks. The idea of organizing the main operations for managing metadata into a set which constitutes an algebra amounts to bringing metadata research closer to a foundational approach. Up to date, metadata have mainly been handled by practitioners who deal with heterogeneous data applications. It is obvious that the increasing heterogeneity of today's networked data sources will lead to an increase of the amount of metadata intensive applications in the future. We believe that the combination of foundational and practical aspects is crucial to managing the increasingly complex amount of metadata that are being associated with applications.

Acknowledgement

This study was supported by the Kyungil University Grant.

References

[1] Y. An, A. Brogida and J. Mylopoulos, "Refining Semantic Mappings from Relational Tables to Ontologies", VLDB, Toronto, **(2004)**.
[2] A. Benitez, J. Martinez, H. Rising and P. Salembier, "Description of a Single Multimedia Document", Wiley, **(2002)**.
[3] P. Bernstein, "Applying Model Management to Classical Meta Data Problems", CIDR, **(2003)**.
[4] P. Bernstein, A. Halevy and R. Pottinger, "A Vision of Management of Complex Models", SIGMOD Record vol. 29, no. 4, **(2000)**, pp. 55-63.
[5] S. Boll, W. Klas and A. Sheth, "Overview on Using Metadata to Manage Multimedia Data", DBLP, **(1998)**.
[6] S. Das, E. Chong, G. Eadon and J. Srinivasan, "Supporting Ontology-Based Semantic matching in RDBMS", VLDB, **(2004)**, pp. 1054-1065.
[7] S. Decker, S. Melnik, F. Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann and I. Horrocks, "The Semantic Web: The Roles of XML and RDF", IEEE Internet Computing, vol. 4, no. 5, **(2000)**, pp. 63-74.
[8] R. Fagin, P. Kolaitis, L. Popa and W. Tan, "Composing Schema Mappings: Second-Order Dependencies to the Rescue", PODS, **(2004)**, pp. 83-94.
[9] http://www.w3schools.com/webservices/ default.asp.
[10] http://www.w3schools.com/wsdl/default.asp.
[11] K. Jeffery, "Metadata: An Overview and some Issues", ERCIM, **(1998)**.
[12] W. Kim, "On Metadata Management Technology: Status and Issues", Journal of Object Technology, vol. 4, no. 2, **(2005)**, pp. 41-48.
[13] J. Madhavan, P. Bernstein and E. Rahm, "Generic Schema Matching with Cupid", VLDB, **(2001)**.
[14] J. Madhavan and A. Halevy, "Composing Mappings among Data Sources", VLDB, **(2003)**.
[15] J. Martinez, "MPEG-7 Overview", ISO/IEC JTCI/SC/29/WG11 N6828, **(2004)**.
[16] S. Melnik, "Generic Model Management", Lecture notes in computer Science, Springer, **(2003)**.
[17] S. Melnik, H. Garcia-Molina and E. Rahm, "Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching", ICDE, **(2002)**.
[18] S. Melnik, E. Rahm and P. Bernstein, "Rondo: A Programming Platform for Generic Model Management", ACM SIGMOD, **(2003)**.
[19] F. Nack and A. Lindsay, "Everything You Wanted to Know About MPEG-7: Part 2", IEEE MultiMedia, vol. 6, no. 4, **(1999)**, pp. 64-73.