Preserving object-relational databases for the next generations

Viktor, Herna L.; Wang, Bo; Paquet, Eric; Doyle, Julie

Publisher's version / Version de l'éditeur:
https://doi.org/10.1109/ICDIM.2010.5664663
Conference proceedings - Fifth International Conference on Digital Information Management (ICDIM), pp. 11-18, 2010-06-22

Preserving Object-Relational Databases for the Next Generations

Abstract

Increasingly, resources are "born digital" and their associated formats are short-lived. Subsequently, the development of environments to preserve such digital content over the very long-term (50 years or more) has become a critical issue. To date, however, the preservation of data as contained in object-relational databases has been widely overlooked. Here, the task is inherently complicated by the nature of the data (relational as well as multimedia). Furthermore, the internal structure of the database and the associated applications need to be preserved as it evolves over time. This paper presents an environment to preserve object-relational databases over a very long period of time. We show that our environment is able to host and access multiple databases as they evolve over time.

1. Introduction

Conservation efforts regarding the preservation of electronic digital objects involve organizations such as UNESCO, government agencies and research groups, who are concerned with the threat posed by the so-called "Digital Rosetta Stone"[1] [1]. However, to date, the preservation of data as contained in object-relational databases has been widely overlooked [2, 3, 4]. Here, the task is inherently complicated by the nature of the data itself. An object-relational database contains not only descriptive attributes within a number of normalized tables or relations, but also multimedia content such as images, audio and 3-D objects.

In order to manage a collection of object-relational data, a database management system stores a variety of system metadata, which are contained in a system catalog. The system catalog includes the relevant information to ensure the integrity of the data, grant or revoke access to data, specify the join paths between tables, specify the format of the multimedia content, and so on. The internal structure is thus known (and needs to be preserved) by the users and the DBMS; and this internal structure (or schema) evolves over time. Access to an object-relational database is achieved through a sophisticated graphical user interface (GUI), which then utilizes the so-called logical layer to give access to the physical data, as stored in separate, indexed files. The queries posed through this GUI are inherently "ad hoc" in nature, and change over time. Furthermore, the users range from novices to database administrators, who require very different types of access to, and preservation of, the data [5,6].

This paper describes an environment to preserve such evolving object-relational databases over a very long period of time. To this end, we designed and implemented a multi-agent system, to deal with the scalability and evolution of the data, and the associated database schema [7]. An experimental environment is developed to validate our implementation and to provide a base for further research. We combine theoretical proof and empirical confirmation to illustrate our environment.

This paper is organized as follows. Section 2 provides some background. This is followed by Section 3, which introduces the architecture of our environment and discusses the current implementation. In Section 4, we explain how we evaluated our environment and Section 5 concludes the paper.

2. BACKGROUND

Preserving digital resources over the long term is a critical research area which spans across digital libraries, scientific data repositories, e-government, e-records, and digital cultural heritage. A large number of digital preservation strategies have been proposed, but no single strategy is appropriate for all data types, situations, or institutions. Rather, several strategies are often used in combination.

[1] The Rosetta Stone was written in two languages (Egyptian and Greek), using three scripts (hieroglyphic, demotic and Greek). After centuries of research to decipher the hieroglyphics, Jean-François Champollion succeeded in 1822.

2.1. Preservation strategies

Considering all the preservation strategies, most of the literature regarding preservation strategies refers to five main strategies, which are emulation, migration, encapsulation, technology preservation, and normalization [1, 5]. That is, techniques range from migrating application to preserving obsolete technologies to be used in the future. In our environment, emulation is chosen as the preservation strategy. Within the context of long-term data preservation, emulation is the process of recreating on current hardware (the host environment) the technical environment required to view and use software from earlier times. This is possible through the use of an emulator that runs on an emulation virtual machine. The emulator mimics the original hardware environment, including CPU, memory and peripherals. Along with a saved bitstream of the software application required to display and interact with the document, and the saved database itself, the emulator runs on a virtual machine, hence recreating the original environment.

Our choice of emulation is based on the following motivation. Firstly, we aim to keep the original databases intact and usable as is, together with the underlying metadata, queries and database management system. Secondly, our long-term data preservation environment may contain many databases that require the same "type" of computing environment and it is therefore efficient to use one type of emulator for all these databases. Furthermore, database management systems are strongly dependent on vendor updates which change as new products, or versions, are introduced [2, 3].

2.2. Standards and Reference Models

Preserving databases is a daunting task; and necessitates the adoption of standards and reference models, in order to ensure that no relevant aspects are overlooked [2, 3]. To this end, many excellent practices and reference models have been put into place. Not only does the adoption of such standards have benefit for the preservation of the integrity of, and access to, digital information. The use of standards also helps ensure best practice in the long-term preservation of digital data [8-14]. Resources that are encoded using open standards have a greater chance of remaining accessible after a long time, rather than those resources that are not.

Among all these standards, we adopt the three most relevant and also most widely used [2, 8, 9]. The Open Archival Information System Reference Model (OAIS) is broadly accepted when long-term preservation repositories are implemented, so it is used as the base of our environment [14]. OAIS establishes a common framework of terms and concepts for preservation of information, defines an information model, and identifies the basic functional model of digital archives. Since the OAIS only provides a reference framework without implementation guidance, a metadata schema must also be designed. Further, we would like to establish a long-term data preservation framework, which can interoperate with other digital repositories. Thus, two other standards, namely the Metadata Encoding and Transmission Standard (METS) [10] and Preservation Metadata Implementation Strategies (PREMIS) [11], were used when the metadata were designed.

The METS schema is a standard for encoding descriptive, administrative and structural metadata regarding objects in a digital repository, using XML. It consists of seven major sections, the METS header, descriptive metadata, administrative metadata, file section, structural map, structural links and behavior [10]. PREMIS is a metadata framework especially designed to support the preservation of digital objects, containing a data model of five types of entities involved in digital preservation activities: intellectual entities (e.g. a database table definition), digital objects (e.g. the tuples in a table), events (e.g. Ingest or query), rights and agents (e.g. a person or application software) [11].

3. ARCHITECTURE

This section describes the architecture of our environment for long-term preservation of object-relational databases. Figure 1 depicts the design, which contains five components, namely Users, the Framework Portal, the Business Logic Process System, a multi-agent system (MAS) and the Digital Repositories.

The **Users** component defines the users of the framework, which may be persons, organizations or software systems. There are three categories of users, namely *Administrators*, *Produces* and *Consumers*. All users interact with the Framework through a web-based interface, the Framework Portal. The **Portal** routes users' requests to the proper functional component in the Business Logic Process System.

The **Business Logic** Process System obtains requests from users and passes them to the multi-agent system. For data access, the Business Logic Process delegates the requirements to the Storage Resource Broker System, as will be discussed in Section 3.1. The main functions of the Business Logic Process System are as follows. (1) *Ingest* submissions from the Producer and store the metadata into the Virtual Model. (2) *Administration*, which consists of creating new users, assigning roles, updating and deleting users. (3) *Archiving*, to provide functions for the storage, maintenance and retrieval of the objects as contained in the databases.

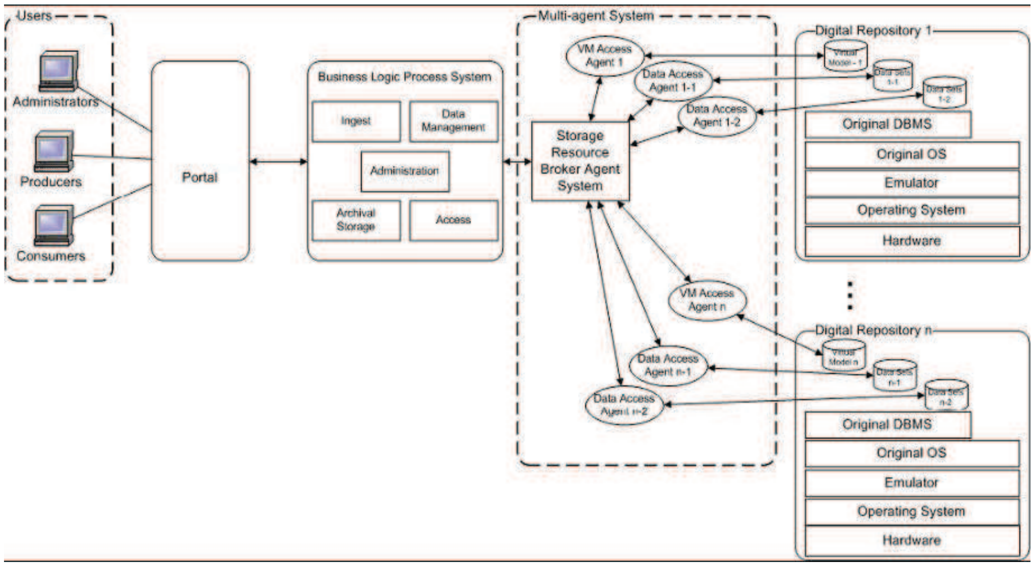

Figure 1: Database Preservation Environment Architecture

(4) *Data management* which creates, updates, reads and deletes the metadata in the Virtual Model and (5) *Access,* which is used to obtain, and return, the results from the Storage Resource Broker Agent.

A **multi-agent system** is created to cope with the scalability and evolution of the data in our environment. A multi-agent system is composed of multiple autonomous agents showing the following characteristics: (a) each component has incomplete capabilities for solving the problem; (b) there is no global system control; (c) data is decentralized; and (d) computation is asynchronous [7].

The most important reason for using a MAS is the need for open and adaptive domains, where different entities with different goals and proprietary information need to fulfill global targets [7]. While parallelism is achieved by assigning different tasks to different agents, reliability is a benefit of a MAS with redundant agents. Another benefit of a MAS is scalability, because it is easier to add new agents to a MAS than to add new capabilities to a monolithic system. Section 3.1 overviews the functionalities of the agents that reside in our MAS.

Preserving a database requires encapsulating (within the **Digital Repository**) the following information [5, 8, 9]:

1. The database to be preserved (saved as a bitstream).
2. The original software environment, including the database management system, operating system, etc. (saved as some executable bitstream).
3. An emulator of the document's original computing hardware platform. All attributes of this hardware platform required to recreate the behavior of the database in its original software environment need to be specified in the emulator.
4. Metadata, explaining to future users how to 'open' and use the preserved environment. Metadata should also include information on the database's origin, details of its software environment and any other information deemed relevant to aid future users in accessing and using the preserved database. Such information should be saved in human-readable format. A **Virtual Model** is used to store our metadata, as depicted in Figure 2. Taking into consideration the special characteristics of databases, the Virtual Model is designed based on the ideas from OAIS, PREMIS, and METS, as introduced in Section 2. Moreover, METS is also used as the mechanism to construct an encapsulation from the datasets and their metadata in the Virtual Model [10, 11].

Note that we use the term "dataset" to refer to different versions of the same database, in order to model the evolution of a database, its schema and its applications [2, 4]. In our environment, each dataset is mapped to an *Information Package* and multiple *Information Packages* co-exist in an *Information Collection*. Since we are interested in long-term preservation, it follows that we need to periodically store such versions for future use, as the databases and their associated schemas evolve.

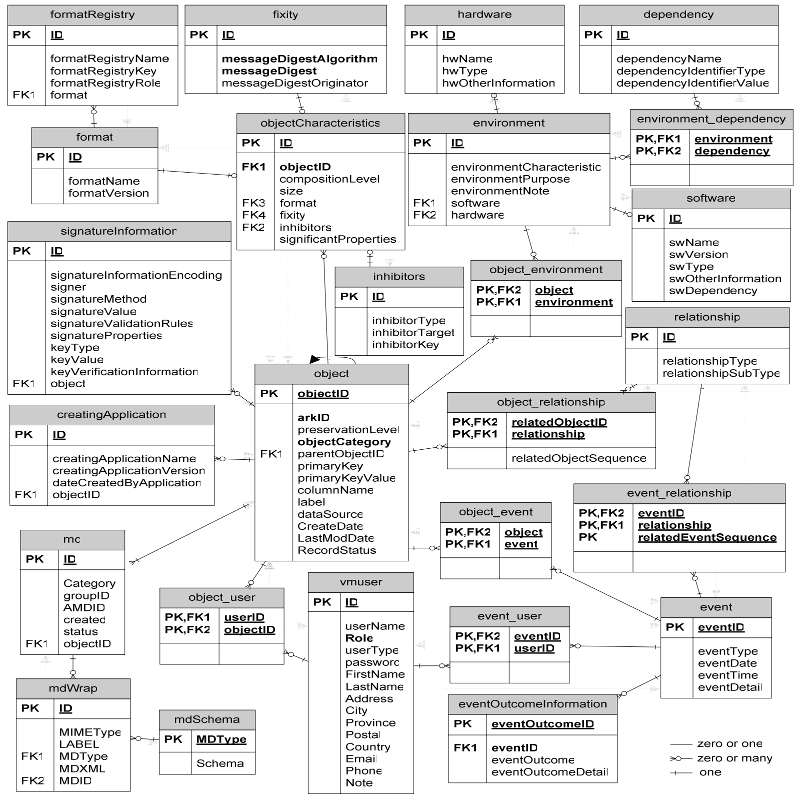

Figure 2: The Virtual Model

3.1. Agent Functionalities

Our MAS implementation follows the FIPA IEEE Computer Society standards, which has been developed to promote the interoperation of heterogeneous agents and the services they can present [12]. The MAS contains the following Agents, as shown in Figure 1.

Each **Data Access Agent** is responsible for accessing the data in one dataset, i.e. for running queries against the correct database tables. A **VM Access Agent** is in charge of the CRUD (Create, Read, Update, and Delete) operations of the metadata in one Virtual Model. The **AMS Agent** acts as the registry of agents and provides naming service, i.e. offering white pages services to other agents. It maintains a directory of all agents, agent identifiers (IDs) and agent states. Only one AMS will exist in the MAS. The AMS maintains a directory of AIDs which contain the "transport address" where the agent may be located (amongst other things). Each agent must register with an AMS in order to get a valid AID.

The **DF Agent** provides the yellow page services as defined in FIPA specifications. In essence, it is a centralized registry of entries which associate service descriptions to agent IDs. In this way, agents are able to search for agents that provide services they are looking for. The DF Agent also allows an agent to deregister or modify its services. That is, the VM Access Agents and Data Access Agents advertise their services to the DF agent. In this way, the SRB Agent is able to be aware of the relevant services and use them.

The **Storage Resource Broker** (SRB) agents connect non-agent environment to the MAS. It receives requests from the Business Logic Process System, passes the requests to suitable Access Agents, obtains results, and returns the results back to the Business Logic Process System. The SRB agent's algorithm consists of two subtasks; namely accessing the Virtual Model and the associated datasets. The metadata related algorithm is responsible for the creation, maintenance and retrieval of metadata in the Virtual Model. The dataset related agent is used to access and query the associated dataset.

Different access agents are created for each specific database (or Virtual Model) in a special computing environment. In this way, the evolution of databases is dealt with by the creation of new access agents. The multi-agent system provides a mechanism for accessing database tables transparently, since the SRB Agent provides a single interface for the interaction with the non-agent environment. The single interface shields the details in the multi-agent system from the Business Logic Process System.

3.2. Current implementation

Our environment is implemented on an IBM eServer Blade HS20 (with two Intel Xeon EM64T 3.6GHz/800 MHz processors), 4GB of memory and 800GB of SAN storage. The current implementation contains two servers, namely the Application Server and the Database Server. In the Application Server, an Apache Tomcat (including an HTTP Server) and a JADE (Java Agent Development Framework) platform are installed. The Portal and the Business Logic Process System are deployed in the Apache Tomcat container. The multi-agent system resides on the JADE platform. The Database Server hosts the Digital Repositories.

In our current implementation, our emulation environment consists of our host (future) computer platform along with VMware Server, which sits atop the host platform. The virtual machine (VM) recreates the hardware architecture of our original platform, which runs the original application software necessary for rendering and interacting with the digital document. Thus, running the VM on the host computer virtually recreates the original environment. In the DB Server, IBM DB2 Version 8.2 is installed and a number of versions of different databases, named "*datasets*", are created. Recall that these datasets model the normal evolution of a database schema, as new attributes, or tables, are added, modified or removed from the database [4]. The users use a browser to access the environment, through the Portal on the Application Server.

4. EVALUATION

As far as we are aware, there are no widely accepted methods to prove the success of a long-term preservation object-relational database repository. However, some standards, such as OAIS, are broadly accepted when long-term data preservation systems are implemented. Furthermore, several general evaluation metrics for trusted digital repositories are proposed in [13], including data integrity, data authenticity, amongst others. To evaluate the ability of our environment to preserve digital data for a long time, we thus decided to use a combination of theoretical proof and empirical confirmation.

The experimental practices were implemented based on multiple versions of the NAMEREMOVED™ anthropometric database, which contained both relational and multimedia content. (Anthropometry is the study of human body measurements (e.g. weight, height, and proportions) and its biochemical characteristics (e.g. stature, and size of body parts) [15]. Anthropometry is used in many application areas, such as the design of clothes, and the design of airplane or bus seats.) The data types in this database are 2D images, 3D objects and relational attributes. The versions of the database differ in content, namely variations of actual tables were included and different subsets of the 3D objects were stored. We also used different naming for the same attributes. We created a METS schema and document, which contains a serialized XML document conforming to the METS schema. The METS schema contains the five data categories namely Information Collection, Information Package (Dataset), Information Unit (Table), Column and Attribute, with the relationships between attributes which may in future be useful for data integration. Throughout, our design and implementation were evaluated by anthropometric experts [5].

4.1. OAIS compliancy

Firstly, the environment was evaluated against the OAIS reference model, which provides a common set of concepts, responsibilities, information models, and processes. A data preservation environment may claim to be OAIS-complaint if it conforms to OAIS responsibilities, the OAIS information model, and the OAIS functional model [14].

The OAIS responsibilities concern the acceptance of appropriate information from information producers, determining which parties should obtain access to the data, documenting policies and procedures, and so on [2]. Our current implementation utilizes the producers' (anthropometric experts') knowledge of the existing data to create the essential metadata within the Virtual Model, including tracking the chain of alteration over time, describing the system from which the data originate, together with the producers and consumers of data and their access privileges. It also provides a web-based unified user interface (the Portal) for creating, reading, updating and deleting both digital data objects and their corresponding metadata.

The OAIS functional model is composed of six functional entities, namely Ingest, Data Management, Archival Storage Access, Administration and Preservation Planning. Table 1 summarizes the fulfillment of these six functional entities in our environment. Our environment is able to Ingest, Archive, Query, Update and Access current and new Information Packages. The current implementation focuses on the technical aspects of preservation, and the Administration and Preservation Planning functions will be realized during a later version. Packaging was achieved using METS, as discussed next.

Table 1. OAIS Functional Entities Implemented (Yes/No/Partial)

Entity	Sub-function		Comments
Ingest	Receive Submission	Y	
	Quality Assurance	Y	
	Generate AIP	Y	
	Generate Descriptive Information	Y	
	Coordinate Update	Y	
Archival Storage	Receive Data	Y	
	Manage Storage Hierarchy	Y	
	Replace Media	N	Replace Media function part of the administration tasks of IT systems.
	Disaster Recovery		Disaster Recovery is used for the business continuity.
	Error Checking	Y	
	Provide Data	Y	
Data Mana.-gement	Administer Database	Y	
	Perform Queries	Y	
	Generate Report	Y	
	Receive Database Updates	Y	
Administration	Negotiate Submission Agreement	N	This function is about the policy and the procedures, not the technical details.
	Manage System Configuration	N	This function is mainly one part of the administration tasks of a computer system and is not in the scope of the core long-term data preservation functions.
	Archival Information Update	Y	
	Physical Access Control	Y	
	Establish Standards and Policies	N	Policies and procedures, such as the preservation strategy and the data ingest procedure, are summarized based on the current technical implementation.
	Audit Submission	Y	
	Activate Requests	N	Instead of event-driven, the access request is responded immediately based on the availability of data now.
	Customer Service	Y	
Preservation Planning (PP)	Monitor Technology and Designated Community, Develop Preservation Strategies and Standards, and Develop Packaging Design and Migration Plans	P	The developing team monitors the up-to-date technology and standards development. The system uses the emulation approach as its preservation strategy. PP will be realized in later phase of the system.
Access	Coordinate Access Activities	Y	
	Generate DIP	Y	
	Deliver Response	P	Only online delivery is implemented.

4.2. Metadata design

One of the main challenges facing digital repositories is the provision of seamless access to assets. METS is commonly used to ensure the interoperability between digital repositories by providing a framework for integrating various types of metadata [10, 15]. The METS schema for our environment contains a XML-based dissemination template of the DIPs (data and its metadata), to be shared, exchanged and searched. Appendix A contains an excerpt of our METS metadata.

The mapping between the Semantic Units in PREMIS and the tables in the Virtual Model indicates that PREMIS is successfully adopted into the Virtual Model. Some properties, which belong to one Semantic Unit in PREMIS, are distributed into several tables in the Virtual Model, for efficiency reasons. This is due to our design requirement that unnecessary duplication of information is minimized. For example, the PREMIS property 'environment' is divided into table environment, table software, table hardware and table dependency, to avoid duplication.

We evaluated the metadata design of our environment by considering the usability, from end users' perspectives [5, 6]. This experimental procedure consists of a usability testing session which was attended by three anthropometric experts. Our aim was to (a) determine whether our emulation environment has successfully preserved the look and feel and behavior and (b) obtain opinions on how to increase the usability of the preserved environment for future end users. In summary, it was the opinion of our expert subjects that the preservation environment was indeed as authentic and usable, given the above criteria, as the original environment. In addition, each subject provided us with highly beneficial ideas on what metadata we should include within our framework to allow future users to easily understand and interpret the database [5, 6].

4.3. Other considerations

One of the central challenges to long-term preservation is the ability to guarantee the interpretability of digital objects. This includes an assurance of integrity, authenticity, and the necessary functionality of the repository. General metrics for trusted digital repositories are given in [13]. The criteria for trusted digital repository evaluation include the evaluation of technology, organizational framework, human resources, amongst others. Since we focus on the technical part of the long-term preservation of databases, we assess three metrics, namely integrity, authenticity, and the necessary functionality of our databases. During our implementation, we ensured that all necessary functionality was included, i.e. we are able to identify all digital objects (databases, relations, queries and metadata), we maintained formal description of the content and structure that are interpretable, and we documented all changes when they occur. Integrity refers to the completeness of the digital objects and to the exclusion of unintended modifications. We tested the integrity of the data as follows. Together with the original data, a message digest of the original data is calculated and kept in the Virtual Model. When integrity needs to be checked, a new message digest is calculated from the current data and is compared with the one in the Virtual Model. If they are same, integrity is ascertained. Digital data are interpreted authentically if they can show their original behavior, functionality, "look and feel", and our experts' evaluations confirmed this [5].

5. CONCLUSION

Most organisations are currently heavily reliant on databases and the associated technologies. An unique aspect of long-term database preservation is its concern with extended periods of time, where 'long-term' may simply mean long enough to be concerned about the obsolescence of technology, or it may mean decades or centuries. When long-term data preservation spans several decades, generations, or centuries, even a minor failure in preservation planning becomes critical.

We created an incremental, expandable multi-agent environment in which to preserve multiple databases as they evolve. multi-agent systems are scalable and our environment is suitable for hosting many more databases. The SRB (Storage Resource Broker) Agent in our multi-agent system is used as a mechanism for both accessing the correct versions of the database and its tables, and to run the associated queries, smoothly and transparently. It provides a single interface to the non-agent environment and makes database access simple and transparent. The Virtual Model, as based on the ideas from OAIS, PREMIS, and METS, forms the cornerstone of our preservation environment. Our implementation contains a web-based portal, and essential functions for preserving databases, which include the abilities to archive, retrieve and query the data.

There still remain many challenges in the long-term database preservation field, in order to ensure durable databases. An important issue which should not be overlooked is "preserving the preservation environment" [5]. We aim to address this challenge in our future research. The appropriateness of emulation needs to be further investigated. We will also research the complex problem of finding mappings between database schemas as they evolve. Research into ensuring that our Digital Repositories are trustworthy is also needed.

REFERENCES

[1] Heminger, A.R., Robertson, S.B.: Digital Rosetta Stone: A Conceptual Model for Maintaining Long Term Access to Static Digital Documents. In: Proceedings of the Hawaii International Conference on System Sciences, Vol. 2, pp. 158-167, 1998.

[2] Buneman, P., Cheney, J., Tan, W., Van Summeren, S., Curated databases. Symposium on Principles of Database Systems (PODS2008), pp.1-12, 2008.

[3] Heuscher, S., Jaermann, S., Keller-Marxer, P, Moehle, F., Providing Authentic Long-term Archival Access to Complex Relational Data, Proceedings PV-2004: Ensuring the Long-Term Preservation and Adding Value to the Scientific and Technical Data, 5-7 October 2004, Noordwijk: European Space Agency, pp.241-261, 2004.

[4] Vassiliadis, P., Papastefanatos, G., Vassilou, Y.. Sellis, T., Management of the Evolution of Database-Centric Information Systems, International Workshop on Database Preservation (PreDB 2007), Edinburgh, March 23, 2007.

[5] reference removed for blind review

[6] reference removed for blind review

[7] Wooldridge, M. An Introduction to Multi-Agent Systems, Second Edition, John Wiley and Sons, London, UK, 2009.

[8] reference removed for blind review

[9] Gladney, H. M., Lorie., R:, Trustworthy 100-year digital objects: durable encoding for when it's too late to ask, ACM Transactions on Information Systems (TOIS), 23 (3), pp. 299-324, 2005.

[10] Metadata Encoding and Transmission Standard (METS). http://www.loc.gov/standards/mets/

[11] PREMIS (PREservation Metadata: Implementation Strategies) Working Group http://www.oclc.org/research/projects/pmwg/

[12] Foundation for Intelligent Physical Agents Specifications, http://www.fipa.org/

[13] Nestor Working Group on Trusted Repositories Certification: Catalogue of Criteria for Trusted Digital Repositories. Version 1. Frankfurt, Germany:. http://edoc.hu-berlin.de/series/nestor-materialien/8en/PDF/8en.pdf., 2006.

[14] Consultative Committee for Space Data Systems (CCSDS): Reference Model for an Open Archival Information System (OAIS). http://public.ccsds.org/ publications/archive/650x0b1.pdf, 2002.

[15] reference removed for blind review

Appendix A

The following METS document contains an excerpt of the metadata for the NAMEREMOVED™ database. (Note that we removed all relevant label names to facilitate blind review.)


```
<Info_Collection>
<Label>NAMEREMOVED</Label>
      <Info_Package>
        <Label>REMOVED1</Label>
        <Info_Unit>
           <Label>REMOVED_MAIN</Label>
           <Column>
           <ColumnName>SUBJECT_NUMBER</ColumnName>
           </Column>
           <Column>
           <ColumnName>ANTHRO_KEY</ColumnName>
           </Column>
           <Column>
           <ColumnName>DEMOGRAPH_KEY</ColumnName>
           </Column>
           ...
        </InfoUnit>
        <Info_Unit>
         <Label>DATA_COLLECTION</Label>
         <Column>
         <ColumnName>SUBJECT_NUMBER</ColumnName>
         </Column>
         <Column>
         <ColumnName>COLLECT_DATE</ColumnName>
             </Column>
             <Column>
             <ColumnName>GET_TIME</ColumnName>
             </Column>
             <Column>
             <ColumnName>RECORDER</ColumnName>
             </Column>
                ...
            </InfoUnit>
         </Info_Package>
         ...
         <Relationships>
             <Relationship>
             <Type>STRUCTURAL</Type>
             <SubType>REFERENCE TABLE</SubType>
             <Member>
                  <Sequence>0</Sequence>
                  <IP_Label>REMOVED1</IP_Label>
                  <IU_Label>REMOVED_CODES</IU_Label>
                  <ColumnName/>
                </Member>
             </Relationship>
             <Relationship>
                <Type>STRUCTURAL</Type>
                <SubType>FOREIGN KEY</SubType>
                <Member>
                   <Sequence>0</Sequence>
                   <IP_Label>REMOVED1</IP_Label>
                   <IU_Label>REMOVED_MN</IU_Label>
                   <ColumnName>ANTHRO_KEY</ColumnName>
                </Member>
                <Member>
                   <Sequence>1</Sequence>
                   <IP_Label>REMOVED1</IP_Label>
                   <IU_Label>ANTHRO_MAIN</IU_Label>
                   <ColumnName>SUBJECT_NUMBER</ColumnName>
                </Member>
             </Relationship>
             ...
           </Relationships>
           ...
          </Info_Collection>
        </Info_Collections
```