uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Bo Wang

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M. (Computer Science)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A Framework for Long-term Preservation of Multiple Relational Databases

TITRE DE LA THÈSE / TITLE OF THESIS

Herna Viktor

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Liam Peyton

Michael Weiss

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

A Framework for Long-term Preservation of Multiple Relational Databases

Bo Wang

Thesis

Submitted to the Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the degree of Master of Computer Science

Ottawa-Carleton Institute for Computer Science

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ontario, Canada, 2007

Your file Votre référence
ISBN: 978-0-494-50934-0
Our file Notre référence
ISBN: 978-0-494-50934-0

Abstract

With the advancement of information technology, more and more information is stored in digital format. Along with the evolution of computer science and its related technologies, new hardware and software emerge every day. At the same time, old hardware and software become obsolete and the data on them become inaccessible. Therefore, the development of environments to preserve digital data for the long-term (such as 50 years) has become a critical issue.

This thesis focuses on how to preserve data in multiple databases for a very long time. A framework is proposed to provide a platform for archiving and retrieving the data in persistent databases. Within this framework, a multi-agent system is used to deal with the scalability and evolution of the data. An experimental environment is developed to validate the framework and to provide a base for further research. Since there is a lack of universally accepted methods to prove the success of a long-term data preservation repository, we combine theoretical proof and empirical confirmation together to address this issue.

Acknowledgment

I would like to express my gratitude to many people who helped and supported me during my master's study at University of Ottawa.

First and foremost, I would like to thank my advisor, Prof. Herna L Viktor, for introducing me to long-term data preservation and multi-agent systems. This thesis would not have happened without her support, guidance, patience, and penetrating insight.

I owe the deepest gratitude to my parents for their deep love. As a son, I am really profoundly indebted to them. I hope I will make them proud of my achievements, as I am proud of them.

I am indebted to my sister and her family for their great support throughout all stages of my master program.

I would also like to thank Yun Li for her love and support through challenging times.

Finally, I would like to extend my thanks and gratitude to all others who helped me to succeed.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
ACL	Agent Communication Language
ACLMessage	Agent Communication Language Message
AID	Agent Identifier
AIP	Archival Information Package
AMS	Agent Management System
CRUD	Create, Read, Update, Delete
DAO	Data Access Object
DBMS	Database Management System
DF	Directory Facilitator
DIP	Dissemination Information Package
EIS	Enterprise Information System
ETL	Extract, Transform, Load (data)
FIPA	the Foundation for Intelligent Physical Agent
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IC	Information Collection
IP	Information Package
IU	Information Unit
JDBC	Java Database Connectivity
J2EE	Java 2 Enterprise Edition
JSP	JavaServer Pages
MAS	Multi-Agent System
MABD	Multi-Agent Behavior Description
MASD	Multi-Agent Structure Description

METS	Metadata Encoding and Transmission Standard
MTS	Message Transport Service
NISO MIX	NISO Metadata for Images in XML
OAIS	Open Archival Information System Reference Model
OLAP	Online Analytical Processing
PREMIS	Preservation Metadata: Implementation Strategies
SABD	Single Agent Behavior Description
SASD	Single Agent Structure Description
SIP	Submission Information Package
SQL	Structured Query Language
SRB	Storage Resource Broker
UML	Unified Modeling Language
XML	Extensible Markup Language

Terminology

Agent: Active persistent software component that perceives, reasons, acts, and communicates.

Consumer: The role played by those persons or client systems, which interact with OAIS services to find preserved information of interest and to access that information in detail.

Digital Object: An object composed of a set of bit sequences (e.g., a PDF file, a WORD file).

Virtual Model: The data schema in the IDeAL Framework, containing the metadata of Digital Objects.

Designated Community: An identified group of potential Consumers who should be able to understand a particular set of information.

Ingest: A function entity that provides the services and functions to accept submission from Producers and prepares the contents for storage and management within digital archives.

Independently Understandable: A characteristic of information that has sufficient documentation to allow the information to be understood and used by the Designated Community without having to resort to special resources not widely available, including named individuals.

Knowledge Base: A set of information, incorporated by a person or system, which allows that person or system to understand received information.

Producer: The role played by those persons or client systems, which provide the information to be preserved.

Representation Information: The information that maps a Data Object into more meaningful concepts. An example is the ASCII definition that describes how a sequence of bits (i.e., a Data Object) is mapped into a symbol.

Chapter 1. Introduction

We are living in the "Information Age", in which more and more data is created and stored in digital format. With the advancement of information technology, the upgrading or replacement of hardware and software occurs very frequently. According to the Moore's Law [Moore, 1998], the number of transistors that can be inexpensively placed on an integrated circuit is increasing exponentially, doubling approximately every two years. Further, almost every measure of the capabilities of computing devices is linked to Moore's Law: processing speed, memory capacity, even the resolution of LCD screens. All of these are improving at roughly exponential rates as well. However, when the old hardware and software become obsolete, the data depending on them becomes inaccessible too [Heminger, & Robertson, 1998]. How to preserve such data, especially over a very long time, has become a crucial problem [Hedstrom, 2001]. As Jeff Rothenberg [1999] says, the goal of preservation is "... to allow future users to retrieve, access, decipher, view, interpret, understand, and experience documents, data, and records in meaningful and valid ... ways." It is relatively simple to preserve digital data over a short time, but it is quite difficult to preserve it over a period of a few decades.

Challenges and issues associated with preserving digital resources over the long term are a critical research area. The coverage of these concerns spans across digital libraries [Rothenberg, 1999], scientific data repositories [Hedstrom, Brandt, & Campbell, 2002], e-government [Neuroth, & Strathmann, 2005], e-records, and digital cultural heritage

[Lorie, 2004; Viktor, & Paquet, 2005]. Several approaches have been proposed as the solutions to the long-term preservation of digital data, such as emulation [Rothenberg, 1999], encapsulation [Waugh, Wilkonson, & Hills, 2000], migration [Wheatley, 2001], amongst others. Furthermore, some best practices and regulations have been standardized in OAIS (Open Archival Information System Reference Model) [Lavoie, 2000; CCSDS, 2002], PREMIS (Preservation Metadata Implementation Strategies) [OCLC, RLG, 2007], and METS (Metadata Encoding and Transmission Standard) [DLF, 2007].

1.1. Motivations

Although the long-term data preservation has been researched during the past several years, there are still many unsolved issues. This is due to the complexity of the problem, such as vast volumes of data, the large number of data formats, and heterogeneous data in one digital collection, amongst others [Hedstrom et al., 2002]. Among all these issues, we are particularly interested in the long-term preservation of data in multiple databases. These databases may contain various data in the e-society, including the e-health, e-government, and e-commerce domains. Typically, such databases contain both relational data and multimedia data. In this study, special focus is put on how to archive, maintain, and retrieve the software-dependent data (e-data) in these databases for over 50 years.

To better understand the intrinsic subtleties of these issues, our goal is to create a conceptual framework for the long-term preservation of data in multiple databases. An experimental environment is established for validating the framework, studying implementation issues, and providing a testing environment for future research. The design of the frame-

work aims to cope with some important concerns, including the scalability of digital archives, the retrieval from heterogeneous data sources, the data evolution, as well as the upgrade of hardware and software.

1.2. Thesis Contributions

The major contributions of this thesis are summarized as follows.

First of all, we design a framework for addressing the problem of long-term data preservation in multiple databases. The framework is name the IDeAL Framework, where IDeAL stands for "Intelligent Decision Analysis and Data Lab". Metadata is the cornerstone of digital preservation, without which a digital object may be irretrievable, incomprehensible, and unusable. Based on some established standards [CCSDS, 2002; OCLC, RLG, 2007; DLF, 2007], we establish a specific metadata schema, the Virtual Model, in the framework. As far as the author is aware, there is no other framework that integrates OAIS, PREMIS, and METS. As such, we provide a new set of guidelines or "checklists" for long term data preservation environment. This framework can be used as an approach for addressing the problems of archiving, maintaining, and retrieving the data in persistent databases over a very long time.

Secondly, a multi-agent system is used in the IDeAL Framework to cope with the scalability and evolution of the data therein. Agents are active persistent software components that perceive, reason, act, and communicate [Huhns, & Singh, 1998]. Agents provide a way to describe independent components of a distributed system. Multi-agent sys-

tems are the systems composed of multiple autonomous agents [Sycara, 1998]. The adoption of the multi-agent system addresses the scalability of the IDeAL Framework. This is due to the fact that it is easier to add new agents to a multi-agent system than it is to add new capabilities to a monolithic system [Stone, & Veloso, 2000]. Further, a storage resource broker agent in the multi-agent system is used as a mechanism for accessing the data smoothly and transparently.

We also thirdly develop an implementation, based on the conceptual framework, which provides a base for the validation of the IDeAL Framework. The implementation contains a web-based portal, two similar data sets and their related metadata databases, and essential functions of digital repositories, which include archiving, retrieving and analyzing data in these repositories.

Moreover, we lastly use a novel evaluation method of combining theoretical proof and empirical confirmation. This evaluation method may be a base for the establishment of widely accepted evaluation criteria and methodologies. Our evaluation method is chosen based on the following observations. Neither theoretical proof nor empirical confirmation can individually be used to confidently prove digital repositories as trustworthy and sustainable long-term data preservation systems. The digital preservation systems should first be validated against some pervasively used standards that are based on best practices. If confirmed as valid, the concepts in these systems can be thought as complete from the theoretical perspective. Then, the detailed processes and functions may be assessed in

practical tests. These tests are used to prove that the above concepts are successfully realized in the digital preservation systems.

1.3. Thesis Outline

This thesis contains six chapters and is organized as follows. Chapter 2 reviews the current literature pertaining to long-term data preservation and multi-agent systems. In Chapter 3, we propose a framework, the IDeAL Framework, to address the issues mentioned in the motivations. The framework is clarified from several aspects: the overall design, the design from the perspective of long-term data preservation, the design from the view of multi-agent systems, and the design from the angle of J2EE (Java 2 Enterprise Edition). Chapter 4 presents the current deployment environment and the implementation details of the IDeAL Framework. In Chapter 5, two kinds of methods are used to appraise the IDeAL Framework based on the current implementation. The compliance of the IDeAL Framework with some proven standards is assessed first. Then, the framework is evaluated against general metrics for trusted digital repositories. Chapter 6 concludes this thesis by some final remarks, summary of contributions and directions for future research.

Chapter 2. Background

This chapter introduces the background of this thesis research from two main aspects: long-term data preservation, and agent technology. The challenges and issues of the long-term preservation of digital data are presented first, and agent technology is introduced.

2.1. Long-term Preservation of Digital Data

2.1.1 Overview

One of the most obvious characteristics of information technology is the continuous improvement in computer processor, memory, and storage performance and their simultaneous drop in cost. Large organizations routinely upgrade their computer systems with terabytes of storage, and more and more people own personal computers with tens of gigabytes of storage. Therefore, more and more information is stored in digital format. With the advancement of computer science and its related technologies, new hardware and software emerge every day. At the same time, old hardware and software become obsolete and the data on those computers become inaccessible [Heminger et al., 1998]. Therefore, the development of environments to preserve digital data for the long-term (such as 50 years) has become a critical issue [Hedstrom, 2001]. To this end, from the 1980's, people have begun investigating this issue in order to create viable approaches and methodologies for the long-term preservation of digital data. One unique aspect of long-term data 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 of the preservation of digital objects becomes critical. Recently, ISO (International Organization for Standardization) has announced the *Open Archival Information System (OAIS) Reference Model* as its standard ISO 14721:2003 [ISO, 2003]. This standard provides a common conceptual framework for the environment, functional components, and information objects of a long-term data preservation system. However, there are still many challenges in the long-term data preservation field. These involve establishing corresponding metadata and preservation standards, storing vast heterogeneous digital data, designing affordable and reliable tools, constructing full-fledged preservation architectures, amongst others [Day, 2006; Doyle, Viktor, & Paquet, 2007; Hedstrom et al., 2002; Neuroth et al., 2005; Rothenberg, 1999; and Verdegem, 2003]. Among all the challenges, eight major ones are the following.

The national preservation policy is the declared statement to preserve digital heritage nation wide, which includes an overall preservation policy to give out the strategic guidance, advice, and standards [Neuroth et al., 2005]. There are no countries that have announced a national preservation policy, to which individual preservation practice needs to be compliant for future interoperability and information exchange.

Secondly, organizational, social, economic, and legal issues must be taken into consideration. Incentives for long-term preservation of digital information are required. Further,

the metrics to qualify all aspects of digital archiving are needed. Legal issues and intellectual property rights must also be considered when acquiring and maintaining digital data [Hedstrom, & Ross, 2003]. For example, the preservation and modification of some data may be prevented by copyrights.

Thirdly, the growth of the vast and heterogeneous digital collections outpaces our ability to manage and preserve them. People need the models and metrics for selecting complex digital objects. After choosing suitable digital objects, proper methods are needed for aggregating digital objects into collections. Even after having data and methods on hands, more and more digital content is available and worth preserving. However, most people increasingly depend on digital resources and do not realize the needs to preserve those [Hedstrom et al., 2002].

Fourthly, there are no mature architectures for long-term data preservation archives. The attributes of digital archives determines that a single architecture approach will unlikely satisfy all the digital preservation needs of various organizations and individuals. These attributes include the security framework of the archives, the possible changed purpose of the archives, the temporary changes of data, and the evolution of data, amongst others. Currently, only one reference model, the previously introduced Reference Model for an Open Archival Information System, is widely used to build persistent archives. Yet, the applicability, effectiveness, and efficiency of OAIS as a universal architectural framework need more practice and assessment [Hedstrom et al., 2002].

Fifthly, new tools and technologies are needed for the long-term preservation of digital data, such as automatic acquisition of data and metadata, preservation decision support, interoperability, amongst others [Hedstrom et al., 2002; Hedstrom et al., 2003].

The sixth challenge is web archiving. Web information has become a major part of our digital heritage and has become a new dimension of social cultures. According to Day [Day, 2006], web archiving initiatives exist to collect ephemeral web content for use by current and future generations of users. Until now, most such initiatives have concentrated on the development of strategies and software tools for the collection of web content and for the provision of access interfaces. Although it is difficult to get accurate and up-to-date statistics on web page longevity, several researches have indicated that the longevity of web pages is fairly short. It is estimated by Lawrence [2001] that pages disappeared on average after 75 days after they are published. Longitudinal studies of web page persistence by Koehler found that only 33.8 percent of a sample set of pages selected in December 1996 persisted at their original URLs by May 2003 [Koehler, 2004]. The short longevity is a specific feature of web content as digital data and makes the preservation tasks urgent. Further, the most apparent nature of web content is that the massive and growing amounts of digital information is now being generated combined with a proliferation of format types. Moreover, the unpredictable remote changes can affect the web content, which should be reflected in the archive of this web content. All these above issues make the long-term preservation of web archiving a challenging task, requiring specific monitoring, collecting and preserving strategies, procedures, and tools [Day, 2006].

Seventhly, the preservation of databases possesses unique problems compared to other digital data (such as an image file or a Microsoft Word file) [Verdegem, 2003]. A database has an internal structure that is understandable by both people and programs, and changes over time. Many databases contain information with considerable value and their preservation is a matter of priority. Each database is unique in its data, application and management. For example, the application to change or use or view the data is usually not widely available and is generally database specific. Furthermore, many databases are being produced as parts of business transactions or scientific procedures, and are valuable both for the information they contain and the evidential value they provide. The database will be useless if only the database is preserved. Thus, specific metadata, procedures, and methods are needed for database preservation [Verdegem, 2003].

Lastly, it is crucial to consider the end users' needs with respect to the preserved digital data when a long-term data preservation system is designed and implemented, because they are the ultimate users of the data [Doyle et al., 2007]. Actually, this is a matter of data usability in the future. Such considerations aid in determining exactly what information should be preserved along with the digital data. However, we can not predict everything an end user wants to do with preserved data in the future. Further, there is a lack of study in the research literature regarding the needs of the future end users.

2.1.2 Preservation Strategies

Many digital preservation strategies have been proposed, but no single strategy is appropriate for all data types, situations, or institutions. For one digital archive, several strategies may be used in combination. Considering all the preservation strategies, most of the literature regarding preservation strategies refers to five main strategies, which are introduced as follows.

The first strategy is **emulation**, which combines software and hardware to reproduce all essential characteristics of the original computing environment, allowing programs designed for the original environment to operate in a newer environment. Emulation requires the creation of emulators, programs that duplicate the functions of one system within a different system. There are three options in this strategy: application emulation, operating system emulation, and hardware emulation [Rothenberg, 1999]. One of the benefits of this strategy is that the original data is kept intact. Only the emulation will change with time. Moreover, this strategy is applicable to all kinds of digital objects. Another advantage of implementing emulation is its possible efficiency. It can be a one-time effort for large groups of digital objects. Once the data is archived with appropriate metadata and software, no other action is required apart from media refreshing until access is desired by consumers. One emulator can also be used as a solution for many data objects requiring the same operating environment. Certainly, this strategy has some disadvantages, such as the technical difficulties of creating emulators, users lacking knowledge of the obsolete software, and intellectual property rights violations [Rothenberg, 1999].

The second strategy is **migration**, which refers to the "periodic transfer of digital materials from one hardware/software configuration to another, or from one generation of computer technology to a subsequent generation" [Waters, & Garett, 1996]. Migration includes refreshing and copying digital information without changing it [Waters et al., 1996]. However, while refreshing will overcome the media obsolescence, it is sometimes not enough to cope with the technological extinction. Migration to new computing environment often means that the copy is not exactly the same as the original data. This strategy has some advantages [Waters et al., 1996]: the conversion functionality is usually supplied with software, the migration result has a format that is familiar to the user, and the new format may have some new functionality. Nevertheless, the migration strategy has several disadvantages, including appearance changes, conversion errors, changed meaning, inefficiency of migrating everything.

The third strategy is **encapsulation**, a technique of grouping together a digital object and the information necessary to provide access to that object (i.e. the information used to interpret the bits appropriately for access and the provenance to describe the source of the object). Obviously, the grouping process lessens the likelihood that any critical component necessary to decode and render a digital object will be lost. Appropriate types of metadata to be encapsulated with a digital object include reference, representation, provenance, fixity, and context information [Cornell University Library, 2003; Rothenberg, 1999]. The main benefit of this strategy is that it preserves digital objects in their native format and can include useful metadata as much as possible. However, updating metadata is difficult.

The fourth strategy is **technology preservation**, which preserves obsolete information systems to ensure access to digital data. This strategy is often called "computer museum". It needs to save everything: files, software and hardware and to keep them alive. It offers the potential of coping with media obsolescence, assuming the media hasn't decayed beyond readability. It can extend the window of accessing obsolete media and file formats, but it is ultimately a dead end, since no obsolete technology can be kept functional indefinitely [Cornell University Library, 2003].

The fifth strategy is **normalization**, which refers to converting all objects to one or more chosen preservation formats. Within an archival repository, all digital objects of a particular type (e.g., colour images, structured text) are converted into a single chosen file format that is thought to embody the best overall characteristics such as functionality and longevity. It assumes that such chosen format will endure and that problems of compatibility resulting from the evolution of the computing environment (applications software and operating systems) will be handled by the continuing need to accommodate the standard within the new environment [Cornell University Library, 2003]. With this strategy, only a limited number of formats need to be preserved and maintained, and the formats chosen have a higher chance of surviving longer than most of the original formats. However, this strategy has some drawbacks: the meaning of data can be changed after normalization, the choice of formats can be wrong, and the strategy is not flexible.

To compare these strategies clearly, we use —Table 1 to clarify their advantages and disadvantages, as discussed next.

Strategy Name	Advantages	Disadvantages
emulation	The original data is kept intact. It is applicable to every sort of digital data. It may be efficiency. It can be a one-time effort for large groups of digital data.	The disadvantages include the technical difficulties of creating emulators, users lacking knowledge of the obsolete software, and intellectual property rights violations.
migration	It overcomes the media obsolescence. The conversion function is usually supplied with software. The migration result has a format that is familiar to the user. The new format may have some new functionality.	The disadvantages include appearance changes, conversion errors, changed meaning, inefficiency of migrating everything.
encapsulation	It preserves digital objects in their native format and can include useful metadata as much as possible.	The update of the metadata is difficult.
technology preservation	It offers the potential of coping with media obsolescence. It can extend the window of access for obsolete media and file formats.	No obsolete technology can be kept functional indefinitely.
normalization	Only a limited number of formats need to be preserved and maintained.	The meaning of data can be changed after normalization. The choice of formats can be wrong, the strategy is not flexible.

Table 1 Comparison of Preservation Strategies

The migration strategy may overcome media obsolescence and software obsolescence, but the migration of the data in a database may be a task with a huge workload. This is especially true in cases when there are many data types in the database and the database contains a large amount of data. Moreover, the database potentially has to be migrated every few years. The encapsulation strategy allows the digital objects to be preserved in its original format, together with some useful metadata. However, updating the metadata

in an encapsulation, that was created decades ago, can be very complicated. The technology preservation strategy cannot be a practical strategy, even though it provides the benefit for preserving the original hardware and software; because no obsolete hardware can be kept functional indefinitely. Since the meaning of data can be changed, and the data type of a data object may be modified so that it cannot be stored in its original database table, the normalization strategy is not suitable for the preservation of data in databases. Emulation is chosen as the primary strategy of our proposal for the long-term preservation of multiple databases, because of two main reasons [Doyle et al., 2007]. Firstly, we aim to keep the original databases intact. Secondly, our long-term data preservation system may contain many databases that require the same type of computing environment and it is therefore efficient to use one emulator for all these databases.

2.1.3 Related Standards

In the digital preservation field, many excellent practices and reference models have been put into place. The adoption of such standards has 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. For example, 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. Furthermore, the activity of adopting standards increases the level of cooperation and will ultimately result in enhanced interoperability among digital repositories [Hodge, & Frangakis, 2004]. Among all these standards, the three most widely used ones (OAIS, PREMIS, and METS) are adopted in the IDeAL Framework [Lavoie, & Gartner, 2005].

OAIS is broadly accepted when long term preservation repositories are implemented, so it is used as the base of the IDeAL Framework. Since the OAIS only provides a reference framework without implementation guidance, a metadata schema must be designed for the IDeAL Framework. Further, we would like to establish a long-term data preservation framework, which can interoperate with other digital repositories. Thus, some other standards, such as PREMIS and METS, are brought into consideration when the metadata in the IDeAL Framework was designed as discussed next. PREMIS is a metadata framework to support the preservation of digital objects, which contains a set of preservation metadata elements. These metadata elements construct an essential and minimal metadata set [OCLC, RLG, 2007]. METS is a flexible and software independent platform, which can be used for interoperability between digital repositories by providing a framework for integrating various types of metadata [DLF, 2007]. The METS standard provides a method to combine a digital object and its diverse metadata as a whole, which can be shared, exchanged, and searched. The three standards are introduced as follows.

Open Archival Information System Reference Model (OAIS RM)

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 [CCSDS, 2002]. It is a conceptual framework, not a blueprint for system design. However, it gives out some hints of the construction of preservation metadata, the design of system architectures, and the development of systems and components. Currently, it is widely used by current digital archives [Lavoie et al., 2005]. The simple model shown in Figure 1 depicts the environment surrounding an OAIS.

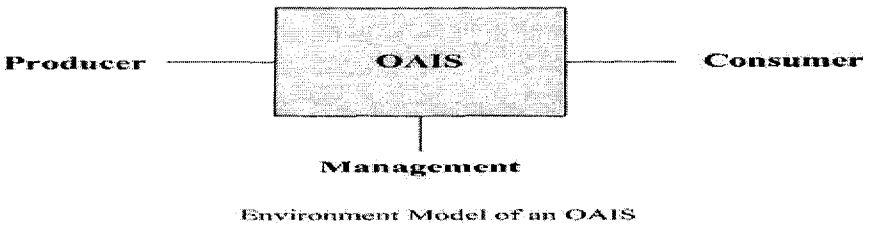


Figure 1 OAIS Environmental Model [CCSDS, 2002]

Outside the OAIS are Producers, Consumers, and Management. Producer is the role played by those persons or client systems which provide the information to be preserved. Consumer is the role played by those persons or client systems that interact with OAIS services to find and gain preserved information of interest. Management is the role played by those who set overall OAIS policy in a broader policy domain [CCSDS, 2002].

As depicted in Figure 2, the OAIS information model defines the broad types of information that would be required, in order to preserve and access an information object stored in a repository [CCSDS, 2002].

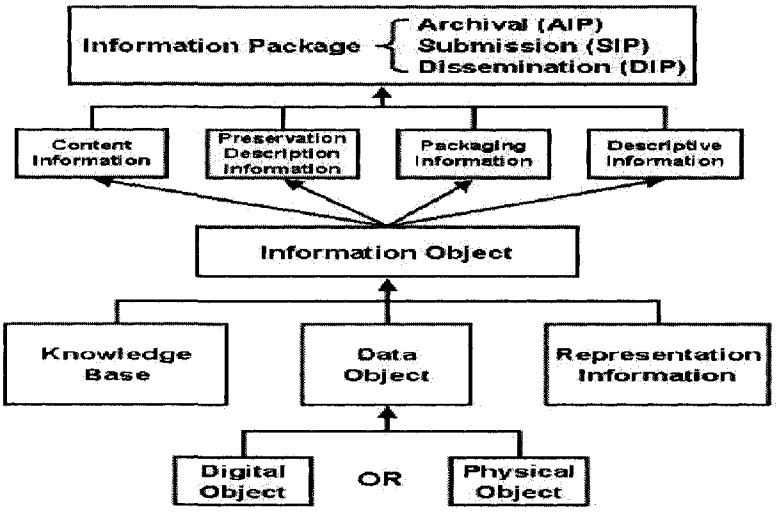


Figure 2 OAIS Information Model [CCSDS, 2002]

In the digital preservation areas, information means knowledge communicated or received concerning a particular fact, such as a PDF file. In the context of the OAIS, infor-

mation can exist in two forms: either as a physical object (e.g., a paper document, a gas sample), or as a digital object (e.g., a PDF file, a WORD file). Altogether, these two types may be referred to as the data object. Interpretation of the data object as meaningful information is achieved through the combination of the Designated Community's knowledge base and the representation information associated with the data object. The knowledge base of the Designated Community is not always enough to be used to fully understand the archived information. In this event, the data object must be supplemented by representation information. The composition of the data object, the Designated Community's knowledge base, and the representation information results in an information object, depicting understood information to the Designated Community [CCSDS, 2002].

The core concept of OAIS, the information package, is comprised of four types of information objects: Content Information, Preservation Description Information, Packaging Information and Descriptive Information. Content Information is the primary information of interest - the data object and its associated representation information. Preservation Description Information (PDI) contains information essential to sufficiently preserve the Content Information with which it is associated. Normally, PDI contains four sections: provenance information, context information, reference information, and fixity information. Packaging Information combines the components of the information package into an identifiable entity, while Descriptive Information facilitates access to the information package [CCSDS, 2002].

Within the OAIS model, three types of information package are identified: the Submission Information Package (SIP), which is received from the Producer; the Archive Information Package (AIP), which is the information package actually stored in the archive; and the Dissemination Information Package (DIP), which is the information package transferred from the archive in response to a request by a Consumer [CCSDS, 2002].

As depicted in Figure 3, the OAIS defines the core set of functional entities for digital archives in its functional model, which is composed of six functional entities: Ingest, Data Management, Archival Storage, Access, Administration, and Preservation Planning. To preserve information over a long time, a digital archive can adopt these functional entities as parts of its system. The OAIS functional model provides a collection of six high level services that achieve the goals of OAIS to preserve and provide access to the information. In Figure 3, the lines connecting entities identify communication paths over which information flows in directions. The lines to Administration and Preservation Planning are dashed only to reduce diagram clutter [CCSDS, 2002].

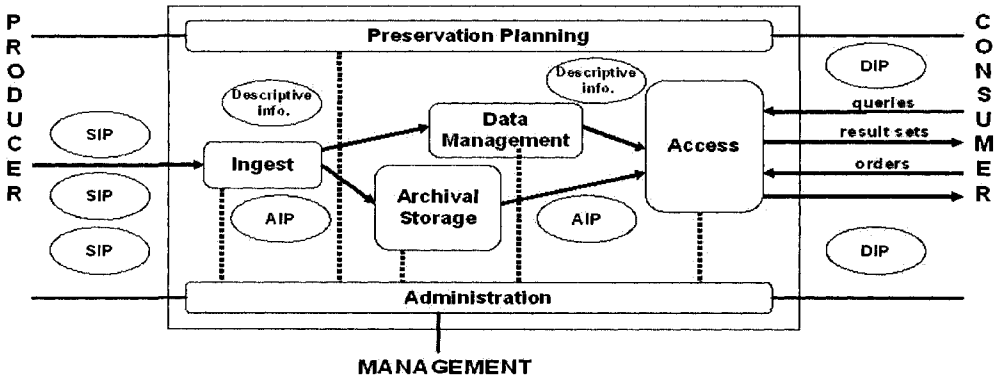


Figure 3 OAIS Functional Model [CCSDS, 2002]

The OAIS standard also establishes mandatory responsibilities that an organization must discharge in order to operate an OAIS archive [CCSDS, 2002]. As defined in [CCSDS, 2002], the OAIS archive must fulfil these responsibilities: (1) negotiate for and accept appropriate information from Producers; (2) obtain sufficient control of the information provided to the level needed to ensure long-term data preservation; (3) determine, either by itself or in conjunction with other parties, which communities should become the Designated Community and, therefore, should be able to understand the information provided; (4) Ensure that the information to be preserved is Independently Understandable to the Designated Community; (5) follow documented policies and procedures which ensure that the information is preserved against all reasonable contingencies, and which enable the information to be disseminated as authenticated copies of the original; and (6) make the preserved information available to the Designated Community.

PREservation Metadata Implementation Strategies (PREMIS)

The second major standard, PREMIS, is a metadata framework to support the preservation of digital objects, which is claimed to contain a set of preservation metadata elements. These metadata elements construct an essential and minimal metadata set. Moreover, the metadata elements can be implemented easily in digital repositories, which can support the viability, understandability, identity, and authenticity of digital objects over time [OCLC, RLG, 2007].

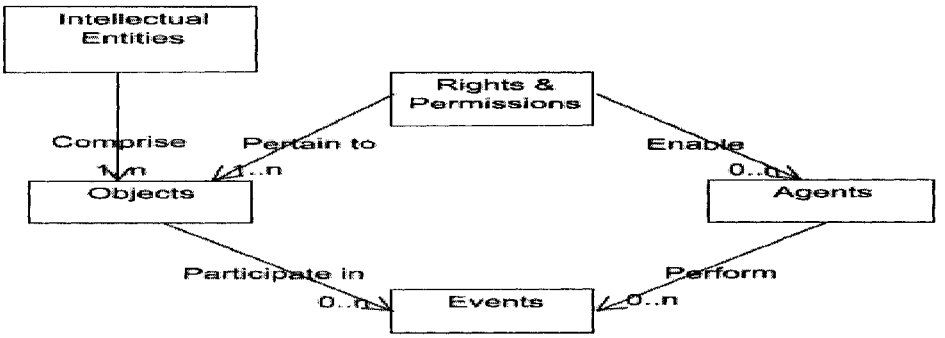


Figure 4 PREMIS Data Model

As shown in Figure 4, PREMIS contains a data model of five types of entities involved in digital preservation activities: Intellectual Entities, Objects, Events, Rights, and Agents. Among all the five types of entities, an Object, or Digital Object, is a discrete unit of information in digital form. An Intellectual Entity is a coherent set of content that is reasonably described as a unit, which can include other Intellectual Entities. For example, a database table can include a column, and that column can include an attribute. An Event is an action that involves at least one Object or Agent known to the preservation repository. An Agent is a person, organization, or software program associated with preservation events in the life of an object. Rights are assertions of one or more privileges or permissions pertaining to an object and/or agent [OCLC, RLG, 2007]. All of the five types of entities have their own properties, named Semantic Units. The preservation metadata is actually these properties. Moreover, PREMIS also provides guidance for local implementations. Many digital preservation projects have used PREMIS in their systems, including FEDORA [Gewirtz, & Gano, 2006], APSR [Lee, Clifton, & Langley, 2006], MathArc [Enders, Kehoe, & Smith, 2006], amongst others.

Metadata Encoding and Transmission Standard (METS)

The METS schema is a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, using XML. It is a data encoding and transmission specification that provides the means to convey the metadata necessary for both the management of digital objects within a digital repository and exchange of such objects between digital repositories. A METS document refers to the serialized XML document conforming to the METS schema. A METS document consists of seven major sections, METS Header, Descriptive Metadata, Administrative Metadata, File Section, Structural Map, Structural Links, and Behavior. These sections may contain a variety of elements and attributes as specified in the METS schema. The METS Header contains metadata describing the METS document itself. The Descriptive Metadata may point to descriptive metadata outside the METS document or involve internally embedded descriptive metadata, or both. The Administrative Metadata section represents the management information for the object, such as the date it was created, rights information, amongst others. The File Section lists all files containing content that comprise the digital object. The Structural Map outlines a hierarchical structure for the digital object and links the elements of that structure to content files and metadata that pertain to each element. The Structural Links section records hyperlinks between nodes in the hierarchy outlined in the Structural Map. The Behavior section may be used to associate executable behaviours with the content object [DLF, 2007].

2.1.4 Data Integration

Integrating disparate data has always been a difficult task, and given the data explosion occurring in most organizations, this task is not getting any easier [White, 2005]. Over 69% of respondents to the survey in [White, 2005] rated data integration issues as either a *very high* or *high* inhibitor to implementing new applications. The three main data integration issues listed by respondents were data quality and security, lack of a business case and inadequate funding, and a poor data integration infrastructure.

From the theoretical perspective, data integration refers to the problem of combining data residing at different sources, and providing the user with a unified view of this data [Lenzerini, 2002]. From the practical perspective, data integration involves a framework of applications, products, technologies, and techniques for providing a unified and consistent view of enterprise business data [White, 2005]. The applications are custom-built and vendor-developed solutions that utilize one or more data integration products. Products are off-the-shelf commercial solutions that support one or more data integration technologies. Technologies implement one or more data integration techniques, which are technology-independent approaches for doing data integration.

There are three main techniques used for integrating data, namely consolidation, federation, and propagation [White, 2005]. Data consolidation captures data from multiple source systems and integrates it into a single persistent data store. This data store may be used for reporting and analysis, as in data warehousing, or it can act as a source of data for downstream applications, as in an operational data store. Data federation provides a single virtual view of one or more source data files. When a business application issues a

query against this virtual view, a data federation engine retrieves data from the appropriate source data stores, integrates it to match the virtual view and query definition, and sends the results to the requesting business application. Data propagation applications copy data from one location to another. Data propagation may be used for the real-time or near real-time movement of data, workload balancing, backup and recovery, and disaster recovery [White, 2005].

Data integration is relevant in our long-term data preservation system in order to reflect the evolution of data in digital archives, and to extract useful information from multiple related data sources. The data integration systems can be characterized by an architecture based on a global schema and a set of data sources. The sources contain the real data, while the global schema provides a reconciled, integrated, and virtual view of the underlying sources. How to model the relation between the sources and the global schema is therefore a crucial aspect of data integration, from our perspective. Two basic approaches have been proposed to this purpose. The first approach, called global-as-view (GAV), requires that the global schema is expressed in terms of the data sources [Lenzerini, 2002]. The second approach, called local-as-view (LAV), requires the global schema to be specified independently from the sources, and the relationships between the global schema and the sources are established by defining every source as a view over the global schema [Lenzerini, 2002].

One basic service provided by the data integration system is to answer queries posed in terms of the global schema. Given the architecture of the system, query processing in data

integration requires a reformulation step: the query over the global schema has to be reformulated in terms of a set of queries over the sources. Since sources are in general autonomous, in many real-world applications the problem arises of mutually inconsistent data sources. In practice, this problem is generally dealt with by means of suitable transformation and cleaning procedures applied to the data, as retrieved from the sources. The theory of query processing in data integration systems is commonly expressed using conjunctive queries [Levy, 2001]. One can loosely think of a conjunctive query as a logical function applied to the relations of a database such as "$f(A,B)$ where $A < B$". If a column or a set of columns are substituted into the rule and satisfies it (makes it true), then that column is considered as part of the set of answers in the query.

The LAV and the GAV approaches are compared in [Ullman, 1997], from the point of view of query processing. Processing queries in the LAV approach is a difficult task. In this method, the only knowledge we have about the data in the global schema is through the views representing the sources, and such views provide only partial information about the data. Since the mapping associates, to each source, a view over the global schema, it is not immediate to infer how to use the sources in order to answer queries expressed over the global schema. Thus, query processing is easier in the GAV approach, since the mapping directly specifies which source queries corresponds to the elements of the global schema. From the point of view of modelling the data integration system, the GAV approach provides a specification mechanism that has a more procedural benefit with respect to the LAV approach. While in LAV the designer may concentrate on declaratively specifying that the content of the sources in terms of global schema, in GAV, the de-

signer needs to specify how to get the data of the global schema by means of queries over the sources. In this study, we adopt a modified GAV approach, as will be discussed in Section 3.3.3.

In this thesis, we consider data integration from the point of view of integrating different versions of the same database, over a long period of time. This will be discussed further in Section 3.3.3.

2.2. Agent Technology

A multi-agent system is used in the IDeAL Framework to cope with the scalability and evolution of the data in the framework. JADE (Java Agent Development Framework) is adopted for implementing the multi-agent system in the IDeAL Framework. JADE is a software environment to build agent systems in compliance with the FIPA (the Foundation for Intelligent Physical Agent) specifications, so the FIPA Standard is introduced too.

2.2.1 What is an Agent

Agents are active persistent software components that perceive, reason, act, and communicate [Huhns et al., 1998]. Agents provide a way to describe independent components of a distributed system. Agent-based computing is a multidisciplinary field which combines artificial intelligence and distributed computing technologies. Thus, multi-agent systems can provide excellent mechanisms for simulating distributed, complex and dynamic real-world environments. For example, simulation of economies, societies and biological environments are typical application areas [Luck, McBurney, Shehory, & Willmott, 2005].

As defined in [Huhns et al., 1998], an agent generally has four necessary properties: the first property is autonomy, which means that an agent can acting independent from the user intervention and self-govern its functions; the second property is reactiveness, which means that an agent can monitor its environment and respond effectively to changes in the environment; the third property is proactiveness, which means that an agent has goals that direct behaviour over longer periods of time towards achieving complex tasks; the last property is socialability, which means one agent must have the ability to interact and communicate with other agents. This is because the agent operates in dynamic and open environments with many other agents.

Agents provide software developers with a way of designing applications based on autonomous, communicative components [Luck et al., 2005]. They offer a new and appropriate approach to the development of complex software systems, especially in distributed and dynamic environments. Most traditional software systems involve static relationships between software entities sharing the same preferences and goals, in a system with a single thread of control. While agent systems contain dynamic relationships between autonomous software entities with different preferences or goals, and there is no central point of control [Luck et al., 2005; Jennings, 2001].

Agent technologies involve a variety of specific techniques and algorithms for dealing with interactions in distributed environments, which address issues such as controlling actions in individual agent architectures, communicating with other agents in the environment, initiating and carrying out actions upon user preferences, cooperating with other agents, and finding appropriate means of forming and managing coalitions [Luck et al.,

2005]. Many applications involving multiple individuals or organizations must take into account the relationships between participants and these individual agents may also need to be aware of these relationships in order to make appropriate decisions [Luck et al., 2005; Jennings, 2001].

2.2.2 Agent Architectures

As defined in [Maes, 1991], the agent architecture is a particular methodology for building agents, which specifies how the agent can be decomposed into the construction of a set of components and how these components should be made to interact. The architecture is a mechanism encompassing techniques and algorithms that support this methodology. Further, there are mainly three major agent architectures: Reactive Agent Architecture, Deliberative Agent Architecture, and Hybrid Agent Architecture, which are briefly introduced as follows.

Reactive agent systems act by means of stimulus-response rules and do not symbolically represent their environment. Agents can respond to events in the environment in a timely and responsive manner. From this aspect, agent's behaviours are directly coupled with the environment; the environment provides a stimulus that causes a rule to fire and the agent to respond in a predefined way [Luck, ASHRI, & D'INVERNO, 2004; Vidal, Buhler, & Huhns, 2001].

Deliberative agent systems symbolically model their environment and manipulate these symbols in order to act [Luck et al., 2004; Vidal et al., 2001]. Agent systems capable of maintaining and manipulating representations of the environment, without stimulus-

response rules of the kind described above, are called deliberative agents. In order to model rational or intentional agency in these kinds of agents, mental attitudes are used to describe and characterize behavior. These attitudes include beliefs, goals, desires, knowledge, plans, motivations, and intentions, which are commonly grouped into three categories: informative, motivational, and deliberative. The first category refers to that which a system consider to be true about the environment, involving knowledge, beliefs, and assumptions; the second refers to the wants of a system, including goals, desires, and motivations; and the third concerns how an agent's behavior is directed and includes plans and intentions. The BDI (Belief-Desire-Intention) architecture is one of the typical deliberative agent architectures [Vidal et al., 2001]. In the BDI model, agents continually monitor their environments and act to change them, based on the three mental attitudes of belief, desire, and intention, representing informational, motivational, and decision-making capabilities. Architectures based on the BDI model explicitly represent beliefs, desires, and intentions as data structures, which determine the operation of the agent [Luck et al., 2004; Vidal et al., 2001].

Hybrid agent systems can act both deliberatively and reactively. In general, agents are neither totally deliberative nor totally reactive. If they are only reactive, they cannot reason about their actions and will not be able to achieve any sophisticated behavior; if they are just deliberative they may never be able to act in time [Luck et al., 2004; Nahm, & Ishikawa, 2005].

2.2.3 Multi-agent Systems

Multi-agent systems (MAS) are systems composed of multiple autonomous components showing the following characteristics: (1) each component has incomplete capabilities for addressing the problem; (2) there is no global system control; (3) data is decentralized; and (4) computation is asynchronous [Sycara, 1998]. The most important reason to use MAS is motivated by the need of open and adaptive domains where different entities with different goals and proprietary information need to fulfil global targets [Stone et al., 2000; Sycara, 1998]. While parallelism is achieved by assigning different tasks to different agents, reliability is a benefit of MAS with redundant agents [Stone et al., 2000]. Another benefit of MAS is scalability, because it is easier to add new agents to MAS than it is to add new capabilities to a monolithic system [Stone et al., 2000].

2.2.4 FIPA Standard

FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies. FIPA specifications represent a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that they can represent. The core specifications in FIPA include FIPA Abstract Architecture, Agent Management, Message Transport, Message Structure, Inter-agent Interaction Protocols, Ontology, and Security. The details of Agent Management, Message Transport, and Inter-agent Interaction Protocols are as follows [FIPA, 2002].

Agent Management provides the normative framework within which FIPA agents exist and operate. This establishes the logical reference model for the creation, registration,

location, communication, migration and retirement of agents. As shown in Figure 5, the reference model includes some agent platforms, each of which contains agents, a Directory Facilitator (DF), an Agent Management System (AMS), and a Message Transport Service (MTS). The DF, if implemented, provides yellow pages services to other agents. Yellow page is the term for a directory which maps services to agents that provide these services. Agents may register their services with the DF or query the DF to find out what services are offered by other agents. The AMS supervise the access and use of the agent platform. The AMS maintains a directory of Agent Identifier (AID) which contains transport addresses for agents registered with the agent platform. The AMS offers white pages services to other agents. Just like telephone books, white pages map agent names to their location (i.e. a network address, an identifier in a software system, amongst others). Each agent must register with the AMS to get a valid AID. The Message Transport Service (MTS) is the communication method between agents on different agent platforms [FIPA, 2002].

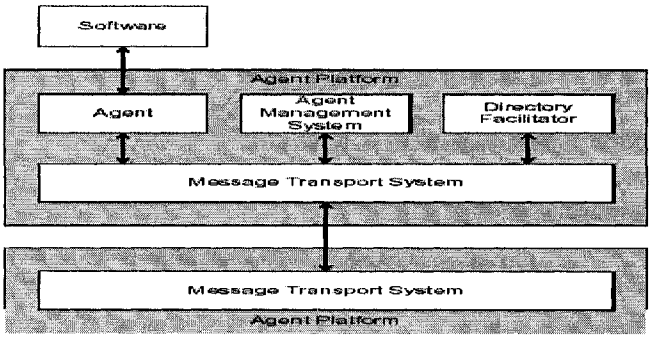


Figure 5 FIPA Agent Management Reference Model [FIPA, 2002]

Message Transport in FIPA contains three levels as depicted in Figure 6. The first level, Message Transport Protocol (MTP), is used to carry out the physical transfer of message between two agent communication channels. The second level, Message Transport Service (MTS), supports the transportation of FIPA ACL (Agent Communication Language) Messages between agents. The topmost level, the ACL, represents the payload of the messages carried by both the MTS and MTP.

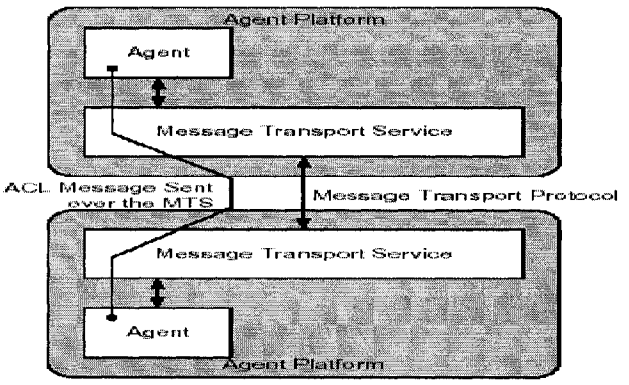


Figure 6 FIPA Message Transport Reference Model [FIPA, 2002]

Inter-agent Interaction Protocols are the critical part of the FIPA specifications. In FIPA, various interaction protocols are defined, which indicate certain message sequences exchanged among agents in typical conversations. Figure 7 presents an example of one of the Inter-agent Interaction Protocols – the FIPA Request Interaction Protocol.

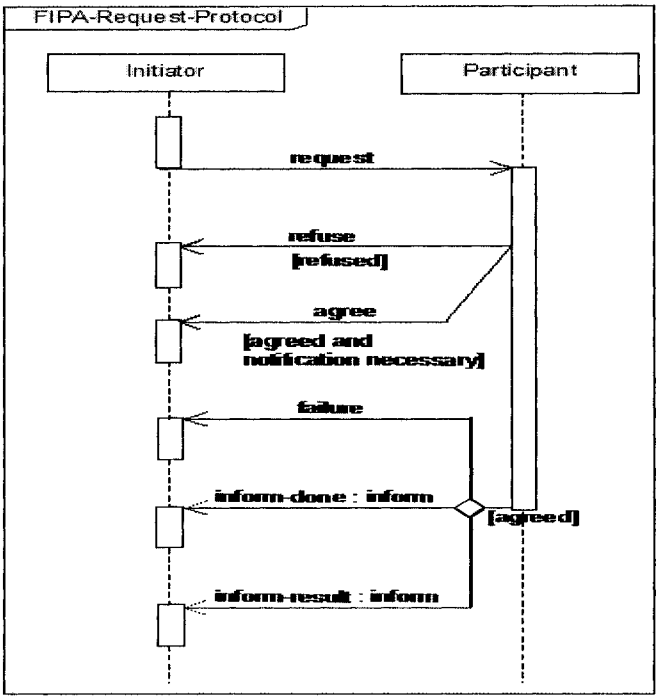


Figure 7 FIPA Request Interaction Protocol [FIPA, 2002]

The FIPA Request Interaction Protocol allows one agent to request another to perform some action. The Participant processes the request and makes a decision whether to accept or refuse the request. In case of agreement, the agent must communicate either a "failure" if it cannot fulfil the request or an "inform-done"/"inform-result" to indicate that it successfully complete the request and to notify the result.

2.3. Summary

In this chapter, the background literature of long-term data preservation and agent technology was introduced. The challenges, preservation strategies, and related standards of the long-term preservation of digital data were brought out one by one. Eight major chal-

lenges were discussed, such as national preservation policy, social and legal issues, amongst others. Five main preservation strategies, to which most of the literature regarding preservation strategies refers, were briefly introduced. The strategies include emulation, migration, encapsulation, technology preservation, and normalization. After their advantages and disadvantages were compared, emulation was chosen as the primary strategy of the IDeAL Framework. This is due to the fact that the strategy can keep the original databases intact and it is efficient to use one emulator for all the databases that require the same type of computing environment.

OAIS, PREMIS, and METS are noted as the three major standards and to be employed in this research. OAIS is broadly accepted when long term preservation repositories are implemented, and it is used as the base of the IDeAL Framework. Since the OAIS only provides a reference framework without implementation guidance, a metadata schema must be designed for the IDeAL Framework. PREMIS is a metadata framework that supports the preservation of digital objects, and it contains an essential and minimal metadata set. For this reason, it is adopted in the IDeAL Framework. In order to interoperate with other digital repositories, METS is used to provide a mechanism for combining a digital object and its diverse metadata as a whole, which can be shared, exchanged, and searched. Moreover, we use data integration technologies to deal with the problem of data evolution. A multi-agent system is used to cope with the scalability and evolution of the data in the IDeAL Framework

The next chapter introduces the IDeAL Framework.

Chapter 3. A Long-term Data Preservation Framework – IDeAL Framework

As stated in Chapter 1, we are particularly interested in the long-term preservation of data in multiple databases. Among all the major challenges in the long-term data preservation field, four are addressed in this thesis: (1) establishing a mature architecture for long-term preservation of multiple databases, (2) preserving and querying multiple heterogeneous databases, (3) dealing with the large volume, the scalability, and the evolution of the preserved digital data, and (4) taking the future end user's needs into consideration. After comparing the preservation strategies in Section 2.1.2, emulation is chosen as the primary strategy in our research. The strategy benefits the end users in that the end users can view and interact with the preserved data in the same way the original users would. The IDeAL Framework is proposed to achieve these goals. In this chapter, we clarify the detailed objectives of the IDeAL Framework first, and then present an overview of the architecture of the IDeAL Framework. Moreover, varied perspectives of the framework are elaborated.

3.1. Objectives of the IDeAL Framework

The IDeAL Framework has the following four objectives.

Firstly, the IDeAL should be able to preserve the digital data in databases for a very long time. Thus, the IDeAL Framework should be OAIS-compliant, because OAIS is a widely accepted standard for the long-term preservation of digital data [CCSDS, 2002]. According to the OAIS standard and its implementation recommendations, the IDeAL Framework needs to provide enough essential metadata and functions for preserving digital data for a long period. Moreover, we require it to meet the common long-term data preservation functional requirements as introduced in Chapter 2, such as acquiring data and metadata from the Producer, archiving data into repositories, and providing access services to the Consumer, amongst others.

Secondly, it should be able to provide a uniform web-based access interface to the users. Through the interface, these users create and manage metadata, and retrieve information from the data sets. The interface consists of a set of built-in functions to access the digital information and its common characteristics, such as the key metadata of the digital information.

Thirdly, it should be able to cope with the scalability and evolution of the preserved data. Data in the persistent databases may change over time, so multiple data sets with similar content may exist. Although they contain analogous content, these data sets may reside on different platforms. Further, more and more data sets will be added to the IDeAL Framework. In the future, the IDeAL Framework may consist of hundreds of persistent data sets. Thus, the IDeAL Framework needs a mechanism to access these data sets smoothly and transparently. A multi-agent system is employed to cope with the require-

ments, since the multi-agent system can easily grow with the expansion of the IDeAL Framework. One dedicated access agent can be designed to handle the access to individual data set; the problems of accessing the new data sets in the framework are transformed to increase the number of the access agents in the multi-agent system. Further, an SRB (Storage Resource Broker) agent is used in the multi-agent system to provide a single interface to the non-agent environment and to make the database access operation simple and transparent for the logic process component of the framework. Moreover, the multi-agent system may be upgraded without influences on the data sets in the IDeAL Framework, because the access agent for each data set can be kept intact.

Last but not least, the framework should be able to retrieve and manipulate data from heterogeneous data sources (different hardware, operating systems, DBMS, amongst others), and to aggregate and integrate data from multiple sources. A comprehensive data integration mechanism should be included to track the evolution of data sets and present the varied versions of the same data sets. Since the data sets in the IDeAL Framework may evolve over time and new data sets may be added into the IDeAL Framework, a good mechanism is required to access these data sets.

3.2. Overview of the Architecture of the IDeAL Framework

Figure 8 shows an overview of the architecture of the IDeAL Framework, which contains five components, namely Users, the IDeAL Framework Portal, the Business Logic Process System, the Multi-agent System (including Storage Resource Broker Agent System, VM Access Agents, and Data Access Agents), and Digital Repositories.

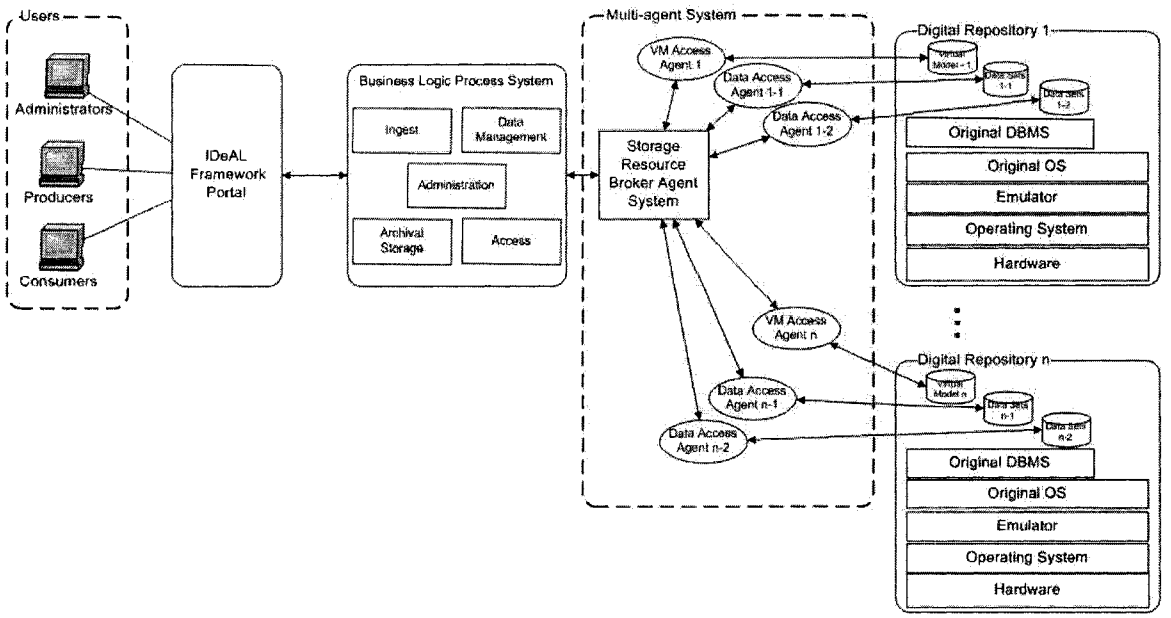


Figure 8 Architecture of the IDeAL Framework

The Users component defines the users of the IDeAL framework. All users interact with the IDeAL Framework through a web-based interface, the IDeAL Framework Portal. The portal routes users' requests to the proper functional component in the Business Logic Process System. If database access needs to occur, a SRB (Storage Resource Broker) agent is instantiated in the Multi-agent System, and the database access request is delegated to this agent. Then, a specific access agent is chosen to handle the database access request. In the IDeAL Framework, a data set must have its metadata stored in one metadata database, named the Virtual Model. All data sets and their Virtual Models reside in Digital Repositories. In order to address the problem of preserving digital data over a period of several decades and ensure their usability over a very long time, the metadata in the Virtual Model is designed based on the consideration of the characteristics of data-

bases, and on the concepts in OAIS, PREMIS, and METS. As indicated in the beginning of this chapter, emulation is chosen as the primary preservation strategy. All the data sets and their Virtual Models in the IDeAL Framework are kept in their original state within their original computing environments. These computing environments are deployed on top of emulators in Digital Repositories. Recall that in order to query multiple databases transparently and to cope with the scalability and the evolution of the preserved data, a Multi-agent System is used. Among the Multi-agent System, each data set and each Virtual Model has its own access agent. Different access agent can be created for each specific data set or Virtual Model in a special computing environment. Thus, the evolution of data sets can be dealt with by the creation of special access agents. Moreover, one Multi-agent System may contain thousands of access agents, which means that it can connect to thousands of data sets and Virtual Models [Chmiel, &Gawinecki, 2005]. Furthermore, the Multi-agent System provides a mechanism for accessing digital objects transparently, because 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. In the next several sections, the functions of the components in Figure 8 are explained in detail.

3.2.1 Users

Firstly, the users can be persons, organizations, or software systems. Secondly, as illustrated in Figure 8, the users of the IDeAL Framework can be divided into three categories as based on the OAIS – Administrator, Producer, and Consumer. Each user must at least have one role; however, a user can play several roles. The role Administrator mainly focuses on the administration of the users, including creating new users, assigning roles,

updating users, and deleting users. The role Producer provides digital objects in data sets for preservation. Moreover, it also takes the responsibility for creating and maintaining the metadata in the Virtual Model. The role Consumer interacts with the IDeAL Framework to find and acquire specific digital objects of interest.

3.2.2 The IDeAL Framework Portal

As shown in Figure 8, the framework portal is a unified web-based user interface that is used to fulfil the objective of providing a consistent "look and feel" with access control and procedures for multiple applications. It is a centralized application that has access to the Business Logic Process System, which consists of multiple applications. Moreover, various users with different roles using different applications may have a single access point over the Internet. Firstly, Users access the framework portal from a web browser; secondly, the IDeAL Framework fulfils their requirements through backend applications; and finally, the portal presents result in a uniform display.

3.2.3 The Business Logic Process System

As depicted in Figure 8, the Business Logic Process System is where the main logic of the IDeAL Framework is processed. It obtains requests from users through the framework portal and fulfils the corresponding requirements. For the data access part of the requirements, the Business Logic Process System delegates the requirements to the Storage Resource Broker System. The main functions of this system are listed as following: (1) Ingest receives the submission from the Producer and stores the metadata into the Virtual Model; (2) Administration fulfils the functions of the role Administrator, which consist of creating new users, assigning roles, updating users, and deleting users; (3) Archi-

val Storage provides the functions for the storage, maintenance and retrieval of the digital objects; (4) Data Management creates, updates, reads, and deletes the metadata in the Virtual Model; and (5) Access obtains results from the related data sets based on the user's requirement, and then integrates the results.

3.2.4 Mutli-agent System

As illustrated in Figure 8, the Multi-agent System contains the SRB (Storage Resource Broker) Agent System, the VM Access Agents, and the Data Access Agents. The Multi-agent System acts as the bridge between the Business Logic Process System and the Digital Repositories. As shown in Figure 9, the Multi-agent System presents a simple mechanism to access distributed data, which is stored in Virtual Models or data sets. Not only does it have abilities to support the CRUD (Create, Read, Update, and Delete) operations of the metadata in the Virtual Models, it also underpins the retrieval and integration of the data in the data sets. Moreover, the multi-agent system is the crucial component of the IDeAL Framework for addressing the issues of scalability and evolution of data sets. Each data set and each Virtual Model has its own access agent. The data sets and the Virtual Models may reside in different computing environments during the very long life cycle of a system based on the IDeAL Framework. Different access agent can be created for each specific computing environment. Thus, the evolution of data sets can be handled by creating special access agents for them. Except for a few SRB Agents, one Multi-agent System may contain thousands of access agents which are mapped to thousands of data sets and Virtual Models [Chmiel et al., 2005]. Moreover, the Multi-agent System provides a mechanism for accessing digital objects transparently, because it uses the SRB

Agent as a single interface to the non-agent environment and the Business Logic Process System does not need to know the details in the Multi-agent System.

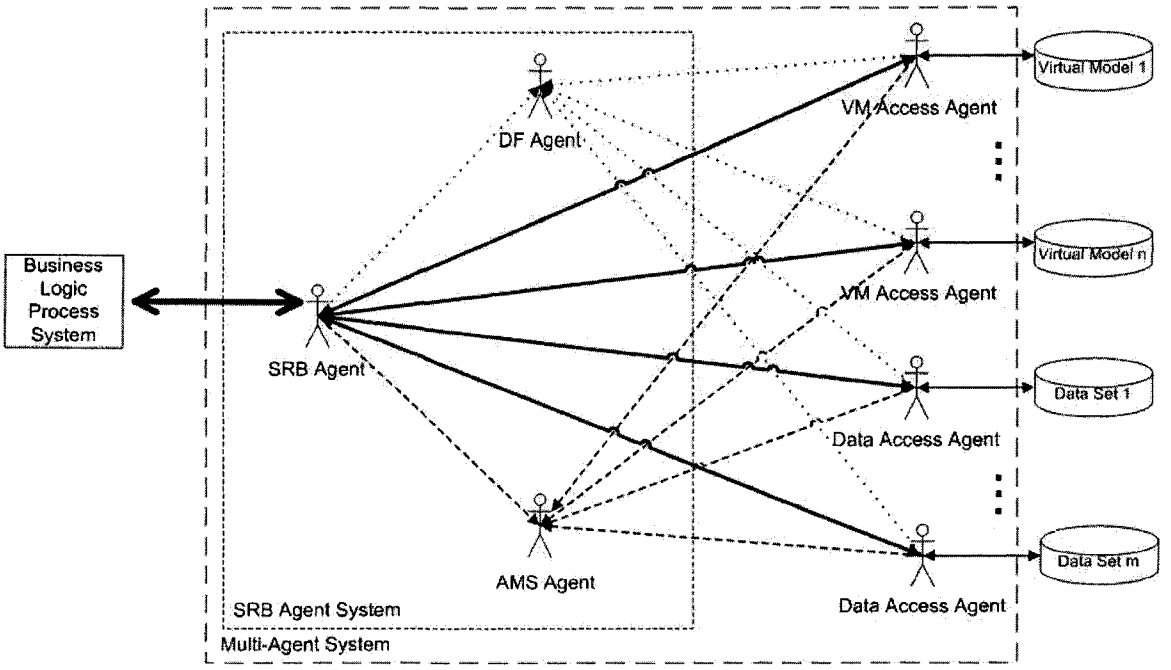


Figure 9 Architecture of the MAS

3.2.5 Digital Repositories

As shown in Figure 8, the IDeAL Framework may consist of many digital repositories, which may reside on different platforms. One digital repository may contain some data sets and some Virtual Models, which reside in their original computing environment on top of an emulator. To preserve the original computing environment, the data sets themselves, their original rendering software, and their related DBMS, their related operating system are all saved.

In the IDeAL Framework, one data set must have its metadata stored in a related Virtual Model. Although one digital repository may involve a combination of numerous data sets and Virtual Models, one specific data set would not necessarily be located in the same digital repository where its corresponding Virtual Model resides. For each data set or Virtual Model, one access agent is created, which provides data access service to the SRB Agent System.

3.3. Design from the Perspective of Long-term Data Preservation

In order to preserve digital data over a very long time, first, we need to identify the responsibilities, components and major functions of a feasible archival system. During the past several years, OAIS has become the most widely adopted standard as the reference model for the digital preservation [Lavoie et al., 2005]. Thus, we would like to establish an OAIS-compliant system as the IDeAL Framework. Second, OAIS only provides a conceptual structure for a digital preservation system. We create our own metadata schema, the Virtual Model, which is an incorporation of the concepts in OAIS and some metadata standards related to digital preservation. Finally, the mapping information in the Virtual Model and the SRB agent system is used to achieve the data integration mechanism, and this mechanism is used to retrieve data from various versions of same data sets in heterogeneous data sources [CCSDS, 2002].

3.3.1 Architecture Design from the Perspective of OAIS

Figure 10 shows the architecture of the IDeAL Framework from the perspective of OAIS.

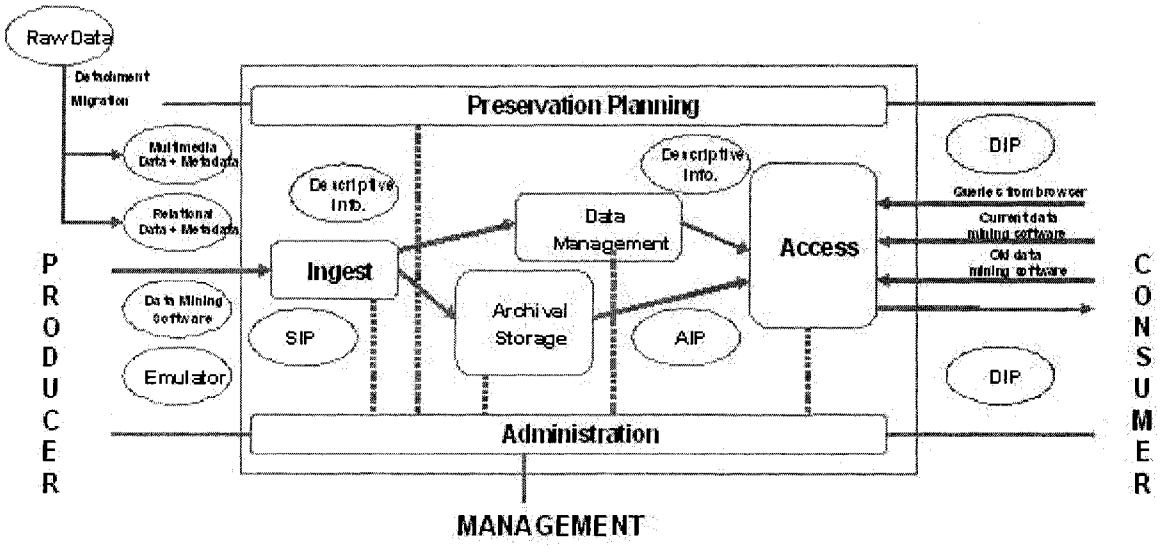


Figure 10 Architecture from the Perspective of OAIS

Figure 10 depicts the architecture of the IDeAL Framework from the perspective of how it corresponds to OAIS. A special characteristic of the IDeAL Framework, which only archives data already in databases, makes the IDeAL Framework employ the conceptual information structure in OAIS with some variances [CCSDS, 2002].

In OAIS, the conceptual structure for supporting long-term preservation of information is the Information Package (IP). Two specializations of the IP are defined in OAIS as Information Collection (IC) and Information Unit (IU). While, in the IDeAL Framework, we extend them to five specializations: Information Collection (IC), Information Package (IP), Information Unit (IU), Column, and Attribute. Correspondingly, the data in the digital repositories of the IDeAL Framework are distributed into five categories that reflect the logical relationship among them, which are Data Collection, Data Set, Table, Column and Attribute. Data Collection is a logical concept that contains logically related Data

Sets. A data set corresponds to a database. Further, these data sets involve varied versions of the same data, the data in which are not exactly the same because the data evolves over time. Table and Column correspond to the concepts of table and column in databases. Attribute represents the value of one column at a specific row [CCSDS, 2002].

Figure 11 is a UML diagram illustrating the relationships among these specializations and the associations among the digital data in the digital repositories. One Information Collection (Data Collection) may contain multiple Information Packages (Data Sets). One Information Package (Data Set) may include some Information Units (Tables). One Information Unit (Table) may involve some Columns (Columns). One Column may contain multiple Attributes.

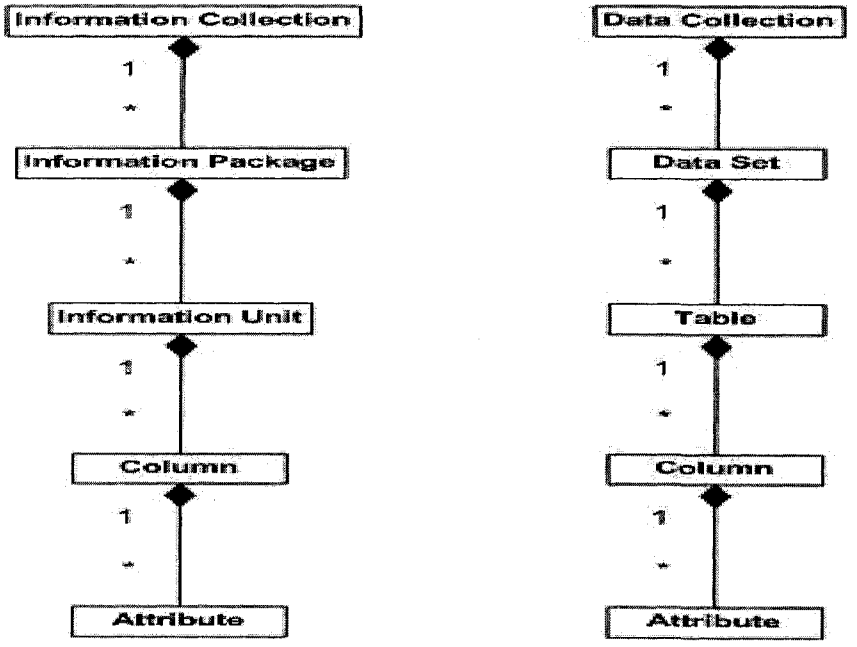


Figure 11 Architecture from the Aspect of OAIS

Figure 12 portrays the relationship between the IP Specializations and the digital data in the IDeAL Framework. Each IP Specialization maps to one category of Digital Data, such as that one Information Package maps to one Data Set.

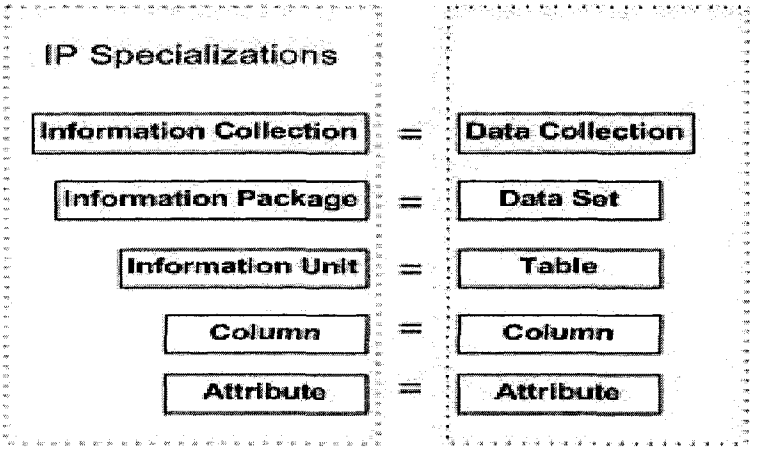


Figure 12 Relationships between IP Specializations and Digital Data

As shown in Figure 10, the major functions of the OAIS are implemented in the IDeAL Framework. The Producer determines what are to be archived, prepares necessary meta-data, collects application software, and acquires the digital data into the IDeAL Framework. All of these actions make up the functions of Ingest.

The Archival Storage provides the services and functions for the storage, maintenance and retrieval of AIPs [CCSDS, 2002]. Among the functions of the Archival Storage, Receive Data and Provide Data are implemented. There are two kinds of data in the Digital Repositories: digital object and its metadata. The digital object is stored directly into the Digital Repositories and can be retrieved directly or through the SRB Agent System. While, the metadata is received from the Ingest functions, processed, then stored into the

Digital Repositories via the SRB Agent System. Meanwhile, the access to the metadata is also through the SRB Agent System.

The Data Management provides the services and functions for populating, maintaining, and accessing both Descriptive Information which identifies and documents archive holdings and administrative data used to manage the archive [CCSDS, 2002].

The functions of Access are realized in three ways. First, the metadata in the Virtual Model can be obtained via the SRB Agent System. Second, the digital objects may be presented on the users' web browser by the IDeAL Framework. Third, the digital objects can be accessed directly by applications such as data mining tools and data retrieval software.

The functions of Administration and Preservation Planning mainly focus on planning, monitoring, and controlling the daily operations of the IDeAL Framework.

3.3.2 Architecture Design from the Perspective of Metadata

Metadata is the cornerstone of digital preservation, without which a digital object may be irretrievable, incomprehensible, and unusable [Hodge, 2002]. In the IDeAL Framework, all essential metadata is incorporated in the Virtual Model. With the considerations of the special characteristics of databases, the Virtual Model is designed based on the ideas from OAIS, PREMIS, and METS. Moreover, the METS is also used as the mechanism to construct an encapsulation from the data sets and their metadata in the Virtual Model

[DLF, 2007; OCLC, RLG, 2007]. The encapsulation can be used to share, transfer, or exchange data with other long-term data preservation systems.

Figure 13 depicts the entity-relationship model of the Virtual Model, whose explanation is the following:

- Entity object, entity relationship and entity object_relationship are used to achieve the function of the Structural Map section of METS. The Structural Map is the core of a METS document, which outlines a hierarchical structure for the digital object. In addition, this function is extended to include manifold relationships among the elements of the digital objects in the IDeAL Framework, such as the relationships among the IP specializations [DLF, 2007].
- Entity md, entity mdWrap and entity mdSchema are both for embedding the Descriptive Metadata section and the Administrative section of METS and for adopting other useful metadata. For instance, MIX (Metadata for Images in XML Schema) can be accepted for the detailed description of still images [DLF, 2007].
- The core set of the PREMIS metadata elements, including Objects, Events, Agents, Rights, and Relationships, are involved in the Virtual Model [OCLC, RLG, 2007].

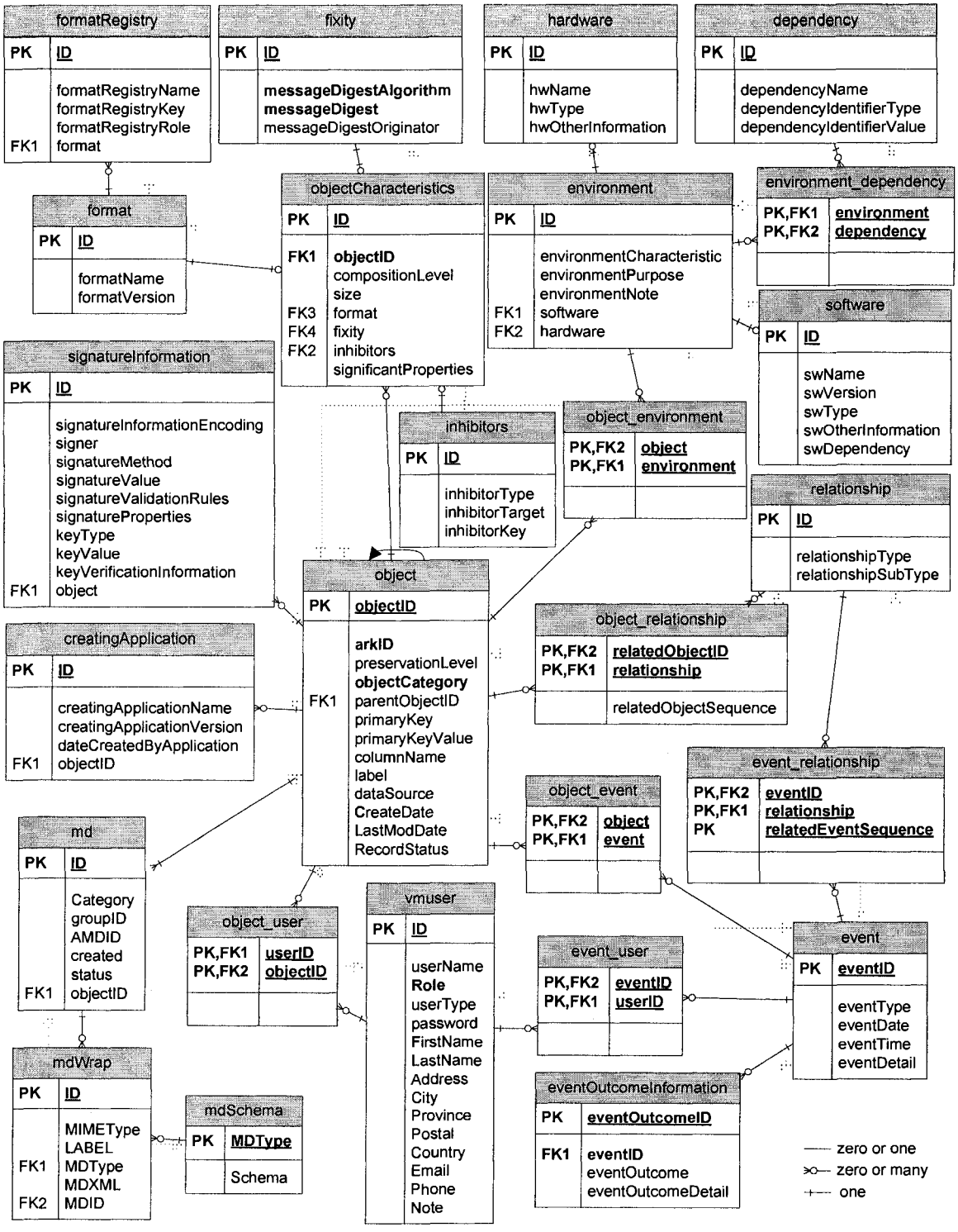


Figure 13 The Virtual Model

3.3.3 The Data Integration Mechanism

The data integration mechanism is an important part of the IDeAL Framework, because it is common to retrieve and analyze various versions of the same data sets during the long lifecycle of the IDeAL Framework. As introduced in Section 2.1.4, the Data federation technique is used in the current implementation of the IDeAL Framework. Recall that this technique provides us a single virtual view of all data. Moreover, an alternative approach of the global-as-view (GAV) approach is adopted. Instead of generating the global schema from all related data sets, when a user would like to retrieve data in the IDeAL Framework, he/she defines the query criteria based on the information of one data set. Since a data set is mapped to one IP (Information Package) in the IDeAL Framework and multiple IPs may coexist in one IC (Information Collection), a list of similar IPs can be obtained. There are four steps in the data integration mechanism. Firstly, a user chooses an IC (named IC-A) that may contain multiple IPs. Then, he/she chooses one of these IPs (named IP-A) and defines query specification based on the information of the IP. For example, the query specification of IP-A includes the column 'name' of the table 'employee' as the result column and 'people.age > 25' as the predicate condition. Secondly, the mapping information in the Virtual Model is used for defining the query criteria of *other* IPs in IC-A based on the query specification of IP-A. For instance, suppose that IC-A contains another IP (IP-B). The column 'name' of the table 'employee' of IP-A maps to the column 'employeeName' of table 'emp' of IP-B. The predicate condition also has a corresponding mapping. Thirdly, the query specification, the internal relationships within an IP, and the specific characteristics of the DBMS on which an IP reside are used to formulate the query statement for each IP in IC-A. In IP-A, the query statement is 'select employee.name from employee, people where (people.age > 25) and (people.id = em-

ployee.peopleID)'. However, the query statement in IP-B is 'select emp.employeeName from emp, people where (people.age > 25) and (people.id = emp.peopleIdentifier)'. Fourthly, the reformulated query statements are executed by corresponding Data Access Agents, based on the available database schemas. Then, query results of IPs in IC-A are synthesized into one web page.

3.4. Architecture Design from the Perspective of Multi-agent Systems

First, the overview of the multi-agent system (MAS) in the IDeAL Framework is explained. Then the algorithms of the SRB (Storage Resource Broker) Agent in the MAS are described in detail.

3.4.1 Overview of the Multi-agent System

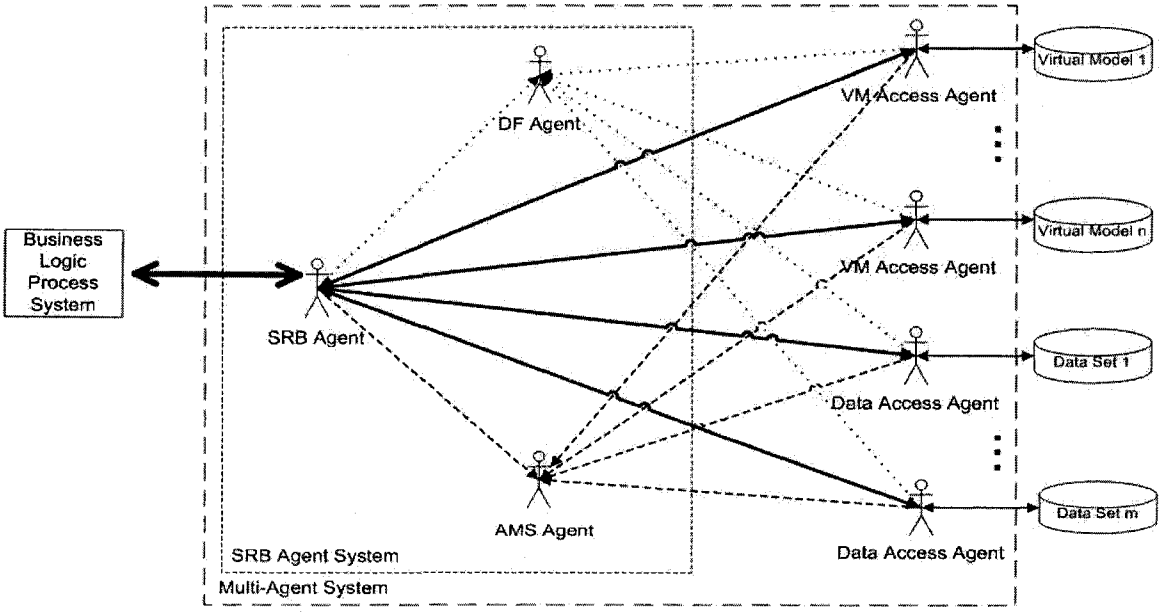


Figure 14 Architecture of the MAS

Figure 14 illustrates the architecture of the MAS. The MAS contains one DF Agent, one AMS Agent, one SRB Agent, several VM Access Agents, and some Data Access Agents. As introduced in Section 2.2.4, DF Agent and AMS Agent are two FIPA specified mandatory agents [FIPA, 2002]. The AMS Agent acts as the registry of agents and provides the naming service. It maintains a directory of the agent, agent identifier and agent state. All agents in the MAS must register with the AMS agent. As introduced in Section 2.2.4, the DF Agent provides the yellow page service defined in FIPA specifications. It is the registry of services, through which agents can register their services and obtain service information of others. VM Access Agents and Data Access Agents advertise their services to the DF agent. Then, the SRB Agent can find the relevant services and use them. Additionally, each VM Access Agent is in charge of the CRUD operations of the metadata in one Virtual Model. Each Data Access Agent is responsible for accessing data in one data set. The SRB Agent is the key player of the MAS, which connects non-agent environment to the MAS [Bellifemine, & Caire, 2007]. It gets the requests from the Business Logic Process System, passes the requests to suitable Access Agents, gains results, and returns the results back to the Business Logic Process System. Further, SRB Agent, AMS Agent, and DF Agent reside in the SRB Agent System component of the IDeAL Framework.

3.4.2 Algorithms of the SRB

There are mainly two kinds of SRB (Storage Resource Broker) algorithms in the MAS: the metadata related SRB algorithm and the data set related SRB algorithm. The metadata

related algorithm is responsible for the creation, maintenance, and retrieval of metadata in Virtual Model. The data set related algorithm provides the service of accessing digital objects in a data set [Bellifemine et al., 2007].

Metadata Related SRB Algorithm

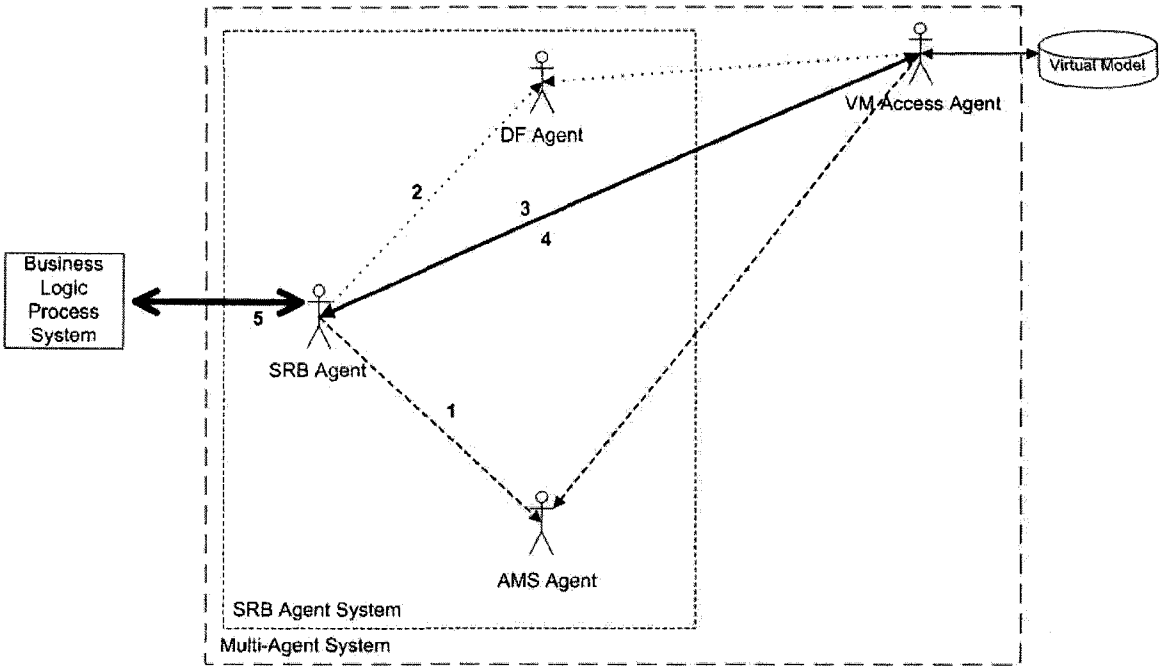


Figure 15 Metadata Related SRB Algorithm

Figure 15 illustrates the metadata related SRB Algorithm, whose pseudocode steps are detailed in the following figure.

Input: mEnvelope from the Business Logic Process System
Output: resultObject (database retrieve or manipulation result)

```
1. JadeGateway.init(VMBrokerAgent) //initiate a Broker Agent
   if ( mEnvelope instance of MessageEnvelope) {
   queryResult = mEnvelope.getVMResult(); //get database command
   aid = queryResult.getAid();  //get agent id if exists

2. if (aid == null) {          //if no agent id in mEnvelope
         create DFAgentDescription as template; //create searching template
         create ServiceDescription as tempSD in template;
         tempSD.type = "VM_AccessAgent";
         tempSD.language = FIPANames.ContentLanguage.FIPA_SL;

         //get possible agents based on the template
         get DFAgentDescription_list based on template;
         aid = DFAgentDescription_list[0];

   }

3. //send database command to a specific agent identified by aid
   ACLMessage queryFromBroker = ACLMessage(ACLMessage.CFP);
   queryFromBroker.setContentObject(queryResult);
   queryFromBroker.addReceiver(aid);
   send(queryFromBroker);

4. //use a behaviour to receive the response from agent identified by aid
   addBehaviour( new Behaviour {
         MessageTemplate mTemplate =
         MessageTemplate.MatchPerformative(ACLMessage.PROPSE) &&
         MessageTemplate.MatchSender(aid);

         ACLMessage message = blockingReceive(mTemplate);
         if (message != null) {
               resultObject = message.getContentObject();
         }
5.       return resultObject;  //return output result
   });

   }
```

Figure 16 Data Set Related SRB Algorithm

The steps of the algorithm may be illustrated by the following example. Suppose that the Business Logic Process System needs to create an environment metadata in a Virtual Model, VirtualModel-A. It sends a request to the SRB Agent System. The request contains the creation command and the agent identifier of the VM Access Agent for VirtualModel-A, namely A-AID.

1. A SRB Agent instantiates and registers with the AMS Agent. It obtains the creation command and A-AID from the Business Logic Process System.
2. If the database access command does not provide a VM Access Agent identifier, the SRB Agent searches for the suitable VM Access Agent from the DF Agent. However, in our example the agent identifier, A-AID, has already been obtained. Therefore, this step is skipped.
3. The SRB Agent sends the creation command to the chosen VM Access Agent, identified by A-AID. Then, the VM Access Agent creates the corresponding environment metadata in the Virtual Model VirtualModel-A. The result of the creation operation is returned to the SRB Agent.
4. The SRB Agent receives the result from the VM Access Agent.
5. The SRB Agent returns the result to the Business Logic Process System.

Data Set Related SRB Algorithm

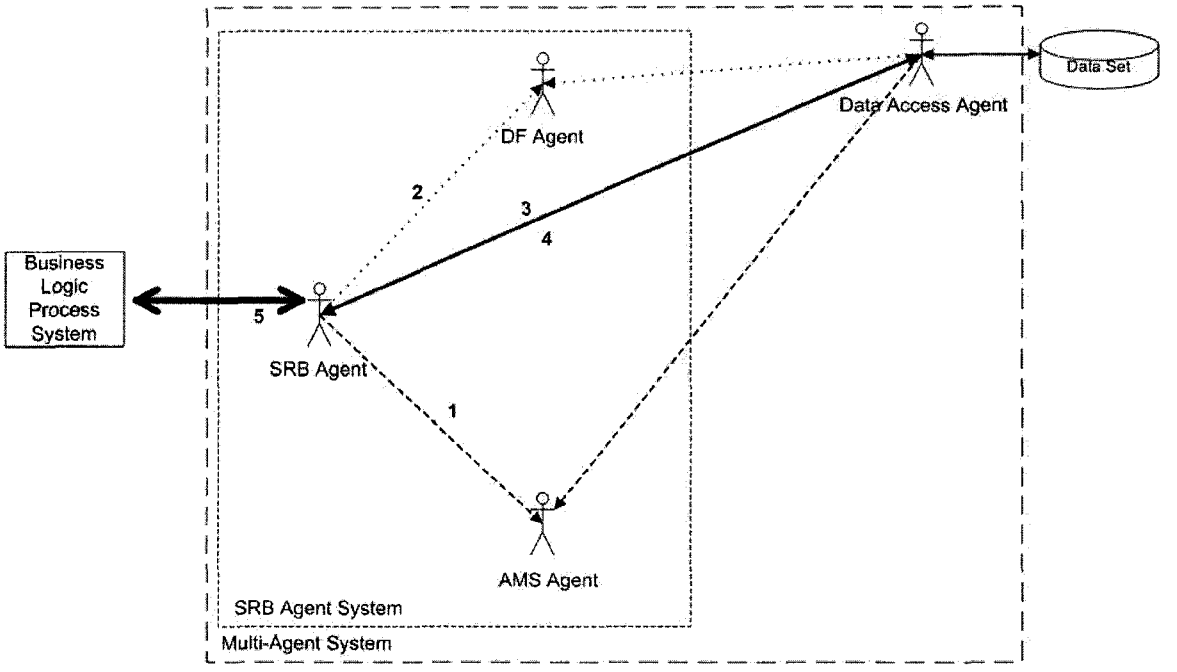


Figure 17 Data Set Related SRB Algorithm

Figure 17 illustrates the data set related SRB Algorithm, whose steps are similar as the metadata related SRB Algorithm except for "tempSD.type = "DS_AccessAgent" instead of "tempSD.type = "VM_AccessAgent". That is, it accesses the digital objects already in the archive.

3.5. Architecture Design from the Perspective of J2EE

In order to build a web-based framework, J2EE (Java 2 Enterprise Edition) is adopted in the current implementation of the IDeAL Framework. J2EE describes the overall architecture for designing, developing, and deploying component-based, enterprise-wide applications. J2EE is widely used in web applications. Moreover, J2EE promotes Java-centric computing and all components deployed into a J2EE deployment must be written in JAVA. This matches with programming language (JAVA) chosen in the implementa-

tion of the IDeAL Framework. Therefore, J2EE is chosen to build the web application part of the IDeAL Framework.

A common implementation approach of J2EE applications is software layering, among which each layer provides one section of the functions of the J2EE system. Layering has proven itself in the operating system domain; moreover, the same benefits are available when applied to web applications. Layered architectures have become essential in supporting the iterative development process by promoting reusability, scalability, and maintainability [Alur, Crup, & Malks, 2003; Bodoff et al., 2002]. Figure 18 illustrates the layered J2EE architecture, which contains the Presentation Layer, Controller Layer, Domain Layer, and EIS (Enterprise Information System) Layer.

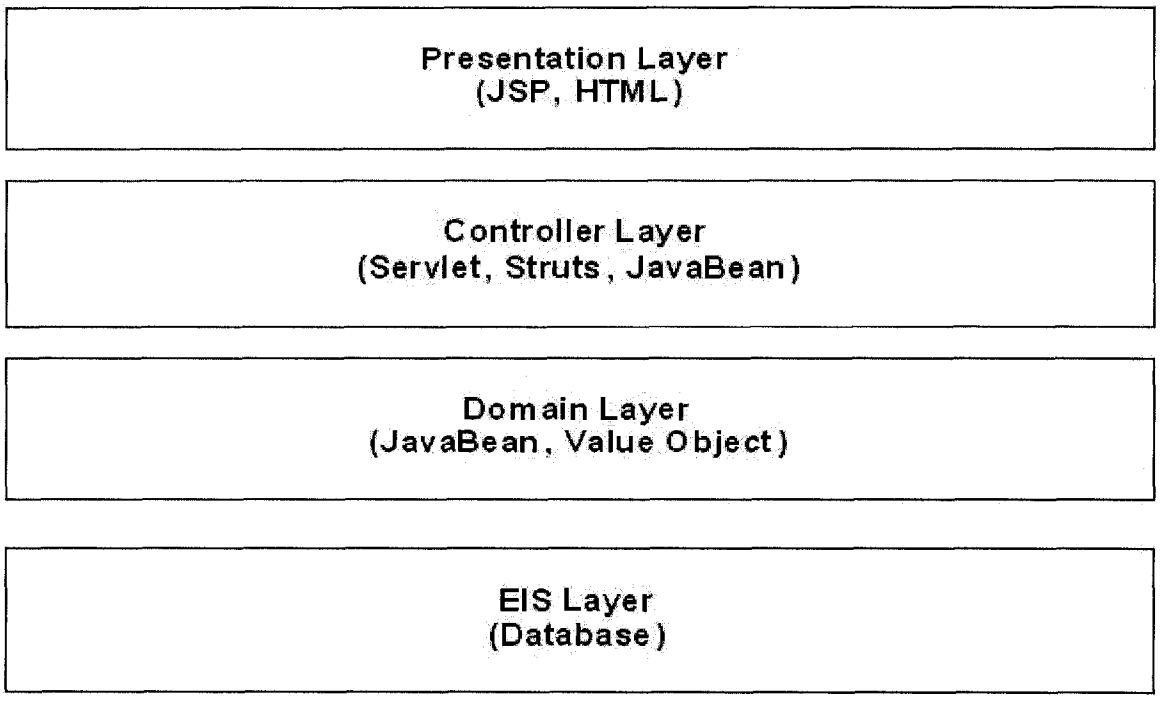


Figure 18 J2EE Architecture

The Presentation Layer consists of objects defined to accept user input and display application outputs. The presentation technologies that are used are JSP, HTML, and JavaScript. Under the Presentation Layer is the Controller Layer. The Controller Layer is used to dispatch requests from the Presentation Layer to the components in the Domain Layer and to return responses back. Further, the business logic is processed in the Domain Layer. If a business logic needs to access databases, the Domain Layer with interact with the EIS Layer. The EIS layer manages access to persistent databases, Virtual Models and data sets, in the digital repositories [Singh, Stearns, & Johnson, 2002].

3.6. Summary

The IDeAL Framework is proposed for coping with the long-term preservation of data in multiple databases. We aim to use the IDeAL Framework as a mature architecture for long-term preservation of multiple databases. Heterogeneous databases can be archived in the framework. The Multi-agent System is used to deal with the problems caused by the large volume, the scalability, and the evolution of the preserved digital data. In order to preserve digital data over a very long time and to ensure their usability at the same time, the metadata in the Virtual Model of the IDeAL Framework is designed based on the characteristics of databases, and on the ideas in OAIS, PREMIS, and METS. Emulation is chosen as the primary preservation strategy. All the data sets and their Virtual Models in the IDeAL Framework are kept in their original state within their original computing environments, which are deployed on top of emulators in the Digital Repositories. A Multi-agent System is included for querying multiple databases transparently and coping with the scalability and the evolution of the preserved data. Within the Multi-agent System, each data set or Virtual Model has its access agent. Different access agent

can be created for each special data set or Virtual Model. Then, the problem of the evolution of data sets can be addressed through creating special access agents for them. Moreover, one Multi-agent System may contain thousands of access agents, which means that it can possibly connect to thousands of data sets and Virtual Models [Chmiel et al., 2005]. Furthermore, the SRB Agent in the Multi-agent System is used as a single interface to the non-agent environment and the details in the Multi-agent System is hidden from the non-agent systems.

Chapter 4. Implementation of the IDeAL Framework

This chapter presents the implementation of the proposed framework for evaluating the concepts in the IDeAL Framework. Section 4.1 describes the implementation environment, including hardware, software, and topology. Section 4.2 introduces the adopted software. Section 4.3 explains the main components of the framework. Especially, for the implementation of agents, the Agent Implementation Model of PASSI (a Process for Agent Societies Specification and Implementation) is adopted to describe the multi-agent system. PASSI is a software engineering methodology for designing and developing multi-agent systems using UML notations [Cossentino, & Potts, 2002]. It focuses on agent platforms that conform to the FIPA Specializations [Bernon, Cossentino, Gleizes, & Turci, 2004]. PASSI is suitable for depicting the multi-agent system in the IDeAL Framework, because a FIPA-compliant agent development framework, JADE (Java Agent Development Framework), is used establish the Multi-Agent System in the IDeAL Framework.

4.1. Implementation Environment

Figure 19 is a UML deployment diagram, which shows the topology, the hardware environment, and the software artefacts in the current implementation. The implementation contains two servers: the ApplicationServer and the DBserver. In the ApplicationServer, an Apache Tomcat (including an HTTP Server) and a JADE platform are installed. The

IDeAL Framework Portal and the Business Logic Process System are deployed in the Apache Tomcat container. The Multi-agent System resides on the JADE platform. The DBServer can be treated as a Digital Repository. Currently, no emulators have been developed, though the two computers' hardware can be considered as two hardware emulators. In the DBServer, a IBM DB2 Version 8.2 is installed and four databases are created as data sets or Virtual Models. The users of the IDeAL Framework use a browser to access the IDeAL Framework, through the IDeAL Portal on the ApplicationServer.

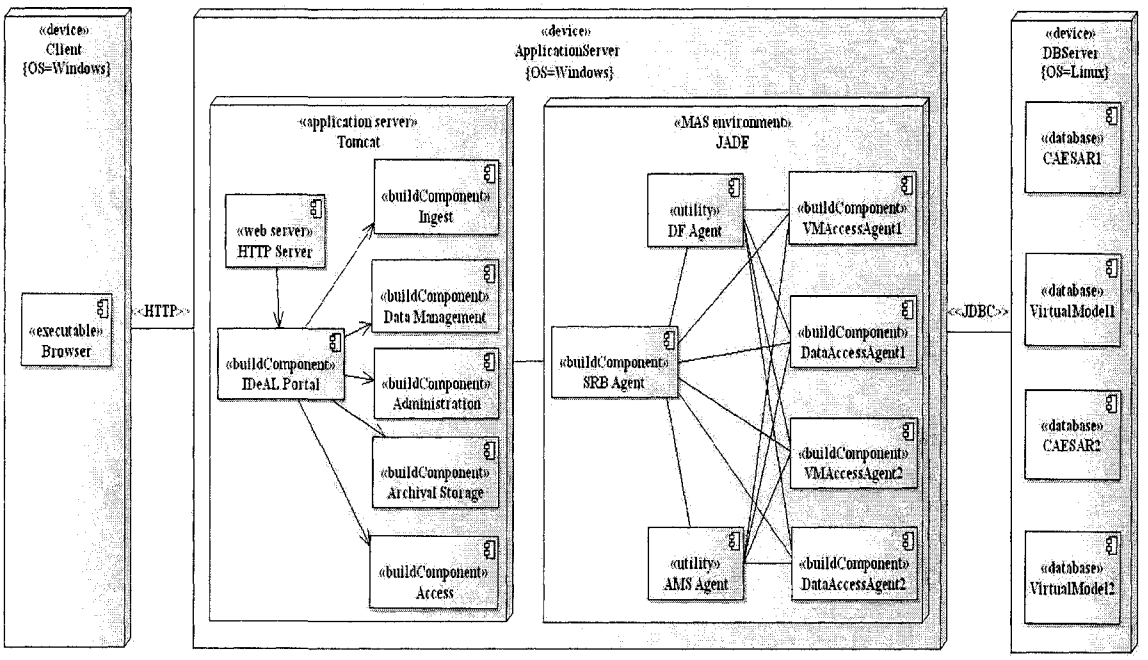


Figure 19 IDeAL Framework Deployment Diagram

4.2. Software Tools

4.2.1 Programming Languages

JAVA and JavaServer Page (JSP) were used for programming. The adopted JAVA is JAVA Standard Edition 1.5 and the JSP is JSP 2.0. The benefits of using them are: (1)

JAVA is object-oriented and allows programmers to create modular programs and reusable codes; (2) JAVA is platform-independent and is capable of moving easily from one platform to another; (3) JAVA is high performance, robust, and secure; (4) JSP is a Java technology that allows software developers to dynamically generate HTML, XML or other types of documents in response to a Web client request [Bergsten, 2003]; (5) The adopted agent framework JADE is a platform developed in JAVA; and (6) Apache Tomcat is the application server of the IDeAL Framework, implementing the servlet and JSP specifications, providing an environment for Java code to run in cooperation with a web server [Eckel, 2004]. Thus, the IDeAL Framework is completely based on JAVA technologies.

4.2.2 Application Server and Web Server

Apache Tomcat is a web container developed at the Apache Software Foundation (ASF). Apache Tomcat version 5.5.23 is adopted in the IDeAL Framework, which implements the Servlet 2.4 and JavaServer Pages 2.0 specifications from the Java Community Process and contains many additional features that make it a great platform for developing and deploying web applications [Moczar, 2004]. Apache Tomcat can be easily configured and deployed. It is very stable. Moreover, Tomcat can act as an application server and an HTTP server of the current implementation, since it includes its own internal HTTP server. A standalone HTTP Server is not used in this implementation, because the IDeAL Framework Portal mainly contains dynamic content. The advantages of a standalone HTTP Server are to serve static content fast and provide some other features (e.g. Perl, PHP, etc.) that are not needed in the current IDeAL Framework.

4.2.3 Apache Struts

Apache Struts is an open-source web application framework for developing Java EE web applications. One of the most significant advantages of Struts is to encourage developers to adopt Model-View-Controller (MVC) architecture in web design. Moreover, Apache Struts is adopted in the implementation and its advantages is listed as the following: (1) it is very stable and mature; (2) rather than hard coding information into JAVA programs, many Struts values are put in centralized configuration files; (3) the Form Bean feature simplifies the processing of request parameters; (4) the custom tags let programmers easily output the properties of JavaBeans components; and (5) it has a very large user base, which proves its effectiveness and provides great programming aids [Husted, & Dumoulin, 2003]. Therefore, Apache Struts 1.2.7 is adopted as the skeleton of the IDeAL Framework Portal, which handles the requests from browsers, determines processing actions, routes page links, and returns the results to users.

4.2.4 JADE Framework

As previously introduced, JADE (Java Agent Development Framework) is a software environment to build agent systems in compliance with the FIPA specifications for multi-agent systems. The agent platform can be distributed across platforms and the configuration can be controlled via a remote GUI. JADE incorporates the three basic management agents defined by FIPA: the Director Facilitator (DF), the Agent Management System (AMS), and the Remote Monitoring Agent (RMA) that manages a GUI of the JADE platform and all registered agents. The full FIPA communication model has been implemented and its components have been fully integrated. JADE includes both the libraries of JAVA classes required to develop agents and the run-time environment. A JADE

multi-agent application is composed of the FIPA standard agents, provided by the JADE platform, and of a set of application dependent agents realized by the application developer. Agents are implemented through a JAVA class containing a set of inner classes that realize the different behaviours of the agent. Agent behaviours can be composed of other behaviours and can be executed either a single time (one-shot behaviour) or different times (cyclic behaviours) [Bellifemine et al., 2007]. Currently, JADE is one of the most used and promising Java-based agent development framework [Vrba, 2003]. Thus, it is adopted into the IDeAL Framework implementation.

4.2.5 IBM DB2

IBM DB2 UDB Version 8.2 is the DBMS used in the current Digital Repositories. It is adopted for the following reasons: (1) DB2 can scale to meet all business requirements on most hardware platforms, from single processor and mid-range multiprocessor systems to large-scale mainframe and clustered environments, which, together with the scalability of JADE Framework, provides an excellent base for the expansion of the Multi-agent System in the IDeAL Framework; (2) DB2 can ensure minimal downtime and zero data loss, which is critical for the purpose of long-term data preservation; (3) DB2 offers all the features and functionality required to secure and protect all important data, so it can fulfil the security requirements of the Digital Repositories in the IDeAL Framework; and (4) The high performance of DB2 is the critical base of the high performance IDeAL Framework, because the most time-consuming operations happen in the Digital Repositories [Baklarz, & Wong, 2002].

4.3. Implementation Details

4.3.1 J2EE Implementation in the IDeAL Framework

The following figure depicts the J2EE implementation in the IDeAL Framework.

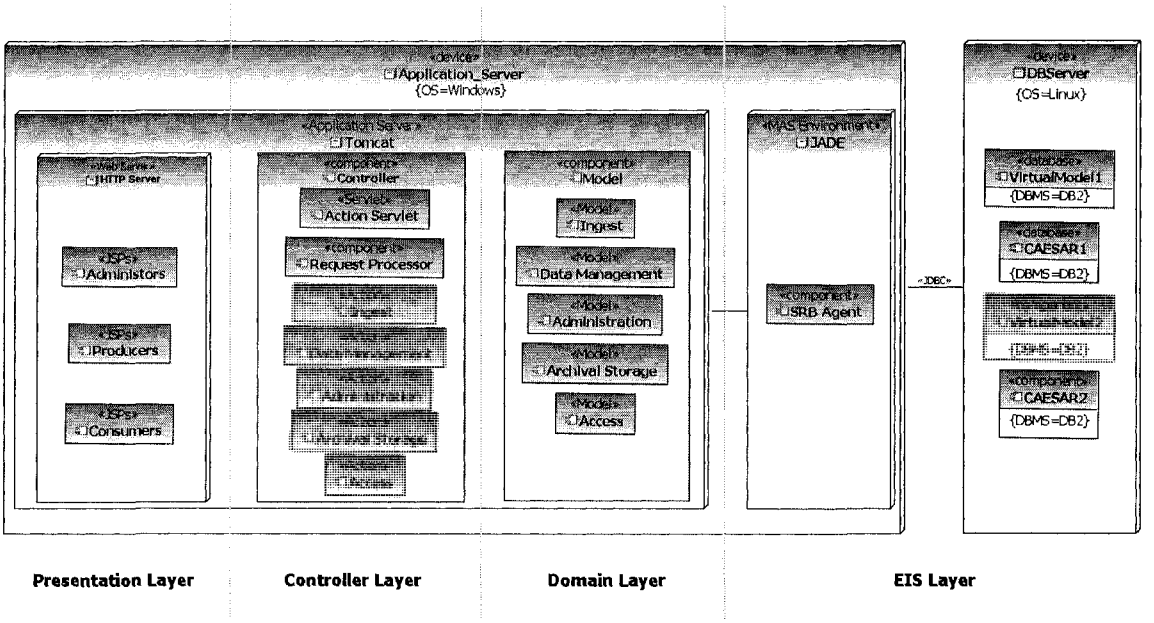


Figure 20 J2EE Implementation

The J2EE implementation is based on Model-View-Controller (MVC) design paradigm, which is achieved by Apache Struts in Apache Tomcat. As introduced in Section 3.5, the layered J2EE architecture contains the Presentation Layer, Controller Layer, Domain Layer, and EIS (Enterprise Information System) Layer. JSPs, the View components of the MVC model, are used in the Presentation Layer, which are divided into three groups corresponding to three types of users: Administrators, Producers, and Consumers. The Controller of the MVC Model, mapping to the Controller Layer, involves ActionServlet, RequestProcessor, and actions specific for the IDeAL Framework. Among them, Action-Servlet and RequestProcessor are provided by the Apache Struts framework. They re-

ceive the HttpServletRequest, automatically populate a JavaBean from the request parameters, invoke proper action component, get the response from the action component, and forward it to suitable JSP. The action components act as a bridge between user-invoked commands and business methods, and delegate these commands to the Domain Layer, the Model component of the MVC model. JavaBeans are used as the Model components in the IDeAL Framework to process the business logic. The interaction between the Domain Layer and the EIS Layer is accomplished through the SRB Agent of the MAS in JADE platform. The actual database access is through the JDBC connection between access agents in MAS and the databases in the DBServer.

4.3.2 The Multi-agent System Implementation

The PASSI methodology integrates design models and concepts from both object-oriented software engineering and artificial intelligence approaches using the UML notation with some extensions. Its Agent Implementation Model describes the agents in terms of both structure and behavior. At the multi-agent level, the Agent Implementation Model use MASD (Multi-Agent Structure Definition) diagram and MABD (Multi-Agent Behavior Description) diagram to describe the agent system. MASD diagram focuses on the architecture of the multi-agent system, in which agents are represented as classes with their behaviours in the operation compartments and attributes specifying the agent knowledge. MABD diagram is a UML activity diagram used to illustrate the flow of events at the multi-agent level [Cossentino et al., 2002].

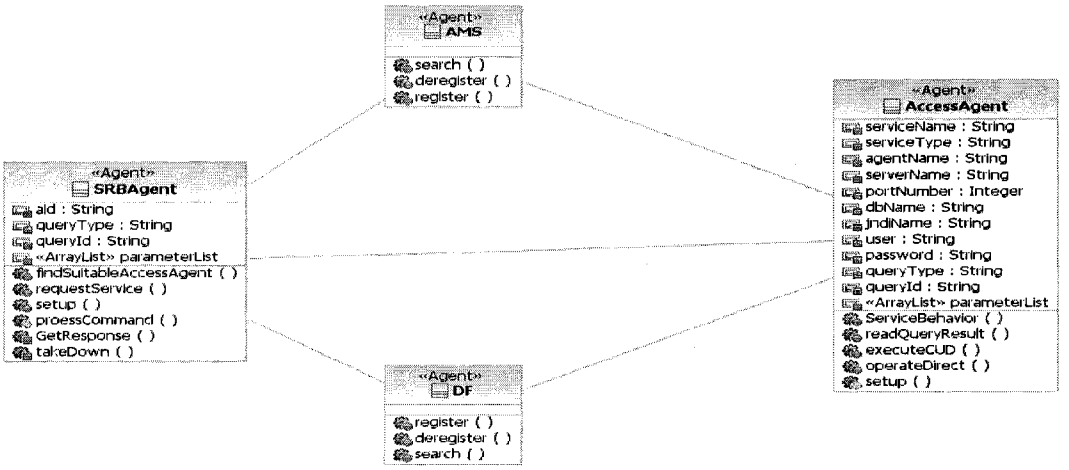


Figure 21 MAS Multi-Agent Structure Description Diagram

The above figure shows the MASD diagram of the Multi-Agent System. Among the four agent classes, the AMS and DF are the agents provided by the JADE platform. The AMS agent is used to manage the agents in the JADE platform, including managing agent life-cycle, registering agents, deregistering agents, searching for agents, amongst others. Providing the yellow pages service in accordance to the FIPA specifications, the DF registers, deregisters, and searches agent services in its catalogue. The SRBAgent (Storage Resource Broker Agent) and the AccessAgent (either VMAccessAgent or DSAccessAgent) extend the Agent super-class in JADE. Both of them contain the setup method, which is invoked by the JADE platform and used to begin the agent activities. After instantiated, an agent is assigned a universal identifier called AID (Agent Identifier). Other methods in these two agent classes are used to achieve their specific functions. The attributes of the SRBAgent are used to compose an agent communication message. The attributes of the AccessAgent are for establishing a JDBC connection, registering its agent service with the DF agent, and preparing agent communication messages.

Both the SRBAgent and the AccessAgent register themselves with the AMS agent when they instantiate. Then the AccessAgent registers its agent service with the DF agent. Later, the SRBAgent search for suitable AccessAgents through the DF agent. After obtaining the suitable agent identifier, the SRBAgent directly interacts with the AccessAgent to fulfil the database access tasks that the SRBAgent receives from outside the Multi-Agent System. Figure 22 depicts the Multi-Agent System MABD diagram.

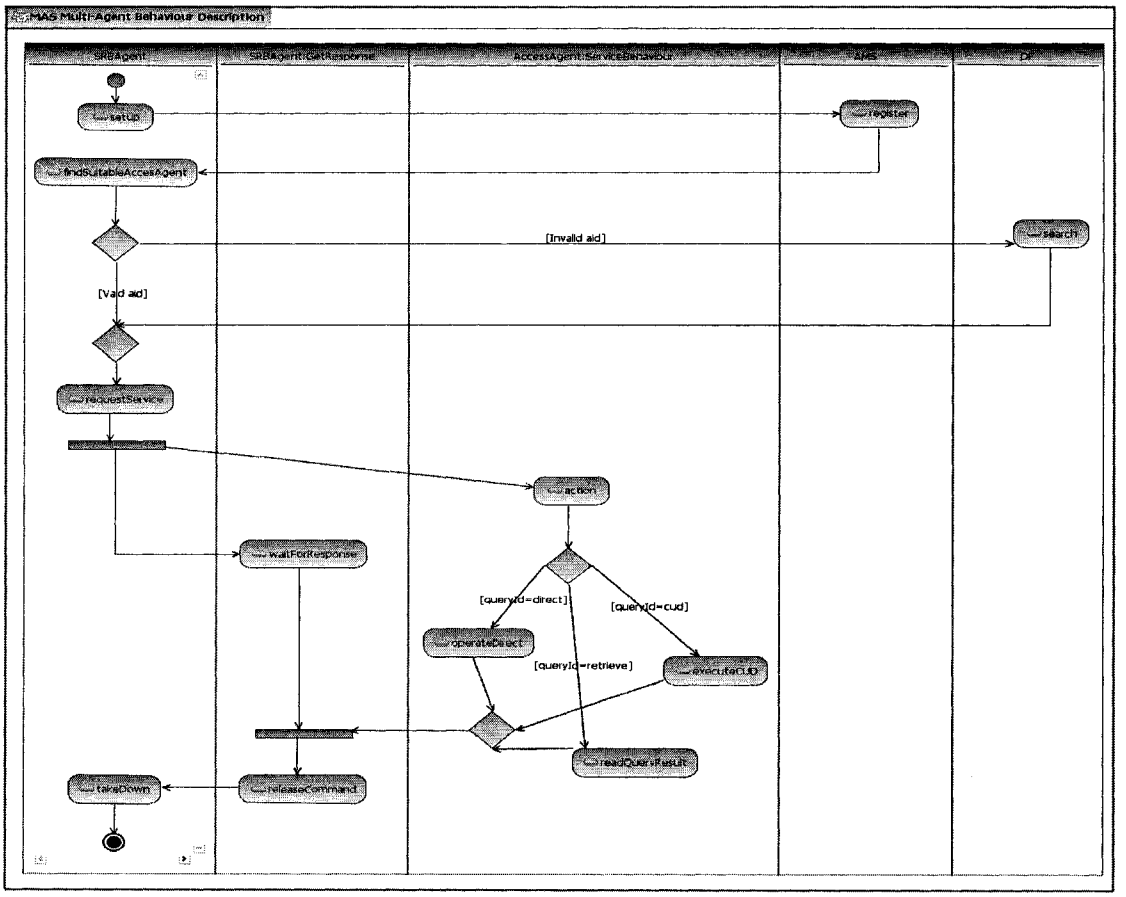


Figure 22 MAS Multi-Agent Behaivor Description Diagram

In Figure 22, we depict one swimlane for one agent or one task, and the activities inside the swimlanes indicate the methods of the related class. To better understand this diagram, Section 4.3.3 and Section 4.3.4 should be used as a reference for the task

"SRBAgent:GetResponse" and the task "AccessAgent:ServiceBehavior", which are the tasks in their corresponding agent's SASD (Single Agent Structure Description).

When a SRBAgent is instantiated, it first registers itself with the AMS agent. Then, it try to get a suitable AccessAgent identifier either through searching with the DF agent or from the transfer object that is transferred to the SRBAgent from the Business Logic Process System. After this, the SRBAgent prepares and sends an ACLMessage to request the service from an AccessAgent. The SRBAgent will instantiate a task "SRBAgent:GetResponse" to wait for the reply from the AccessAgent after sending out the ACLMessage. The AccessAgent that receives the request ACLMessage from the SRBAgent will firstly check the content of the ACLMessage. Then, three optional methods can be chosen based on the value of the attribute "queryId" obtained from the content. Upon the completion of the optional method, the task "AccessAgent:ServiceBehavior" will assemble and send back a reply ACLMessage. Receiving the reply, the SRBAgent extract the result and return it.

In section 4.3.3 and section 4.3.4, two representational single-agent implementations are presented. At the single-agent level, the Agent Implementation Model use SASD (Single Agent Structure Description) diagram and SABD (Single Agent Behavior Description) diagram to details the agent design. The SASD diagram is created for individual agent in order to explore its internal composition and its tasks in detail. This diagram is a UML class diagram with the agent main class and each agent task as a class. Further, we use

UML Activity Diagram as SABD diagram to describe the internal operations of each agent [Cossentino et al., 2002].

4.3.3 The Storage Resource Broker Agent Implementation

Figure 23 shows the SASD of the Storage Resource Broker Agent (SRBAgent).

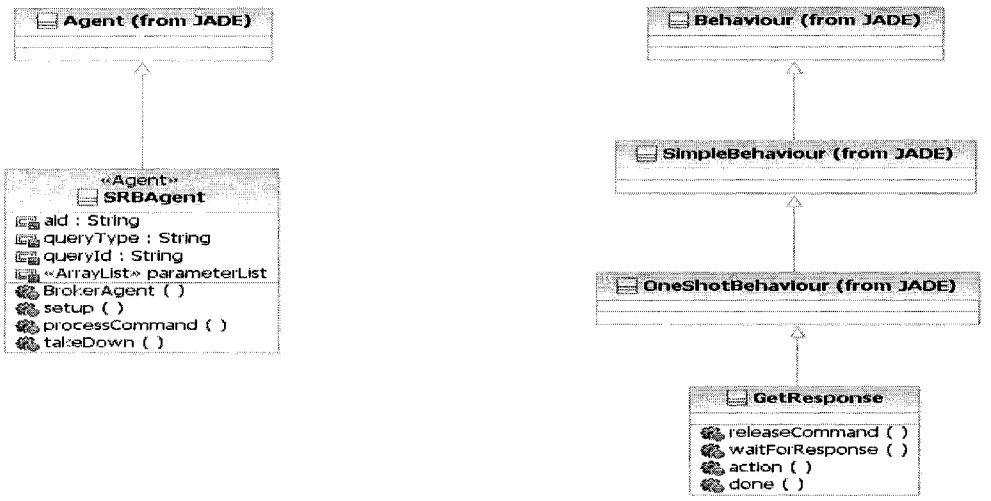


Figure 23 SRBAgent Single Agent Structure Description Diagram

Besides its main class, the SRBAgent has one task, the class GetResponse, which extends the OneShotBehaviour class of JADE platform. The SRBAgent receives request from its invoker, obtains a suitable AccessAgent, prepares and sends out a request ACLMessage, and finally takedown itself. The GetResponse task is instantiated after the SRBAgent sends out the ACLMessage. It waits for the response and processes it. Figure 24 shows the SABD of the SRBAgent.

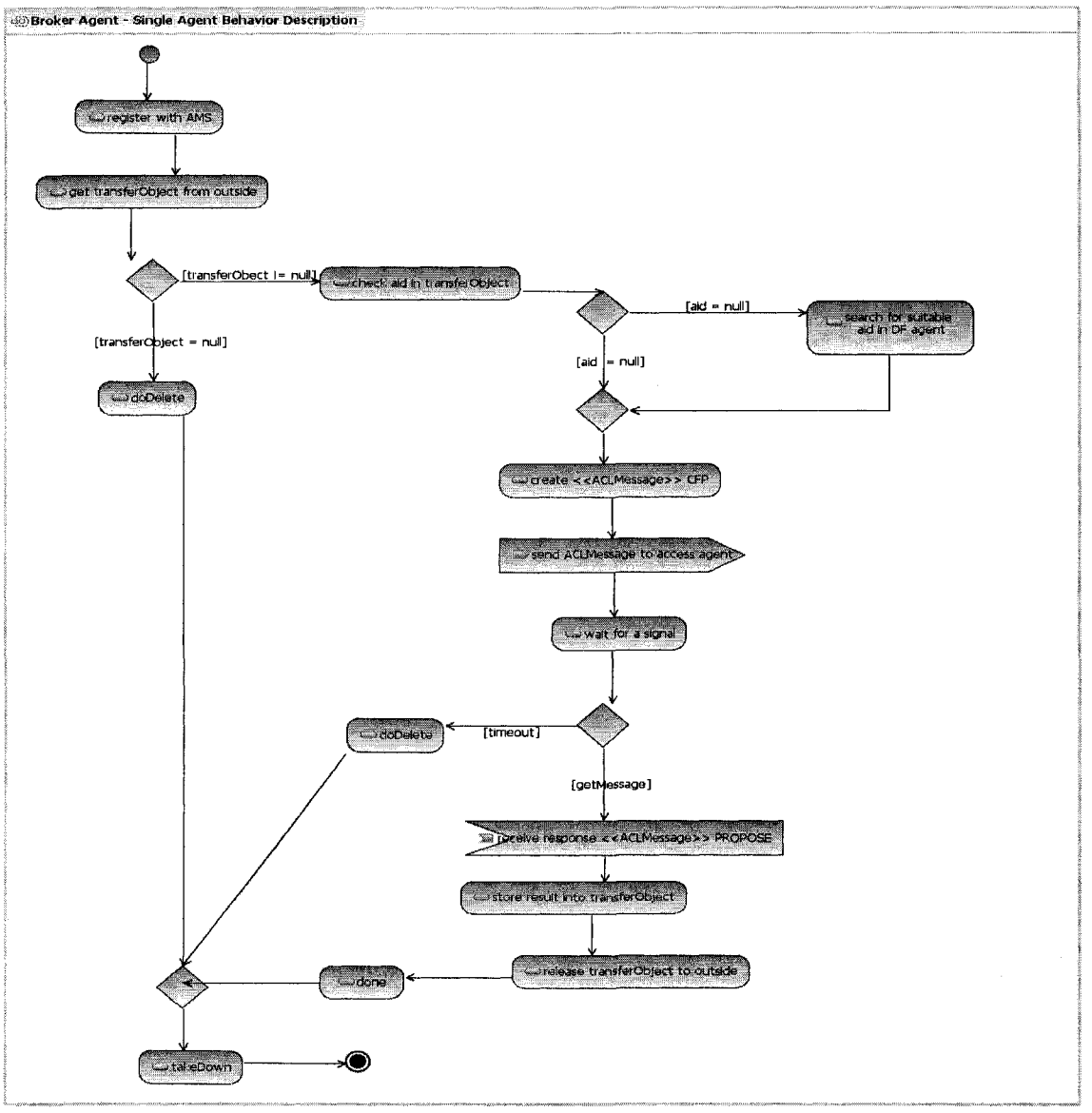


Figure 24 SRBAgent Single Agent Behavior Description Diagram

As illustrated in Figure 24, a SRBAgent begins its activities with registering with the AMS agent. Then it receives and checks the transfer object from outside. If the transfer object is invalid, the SRBAgent will terminate. Otherwise, it continues to check whether the AID (Agent Identifier) in the transfer object is valid. If not, the SRBAgent will search in the DF catalogue for suitable AccessAgent. Then, the SRBAgent will assemble and

send out an ACLMessage based on the content of the transfer object and the valid aid (Agent Identifier). Next, the task GetResponse is instantiated and will wait for the reply from the AccessAgent for a specific time. If no proper reply is received when time is out, the SRBAgent terminates. If not, result will be extracted from the reply message and return to the invoker of the SRBAgent.

4.3.4 The Access Agent Implementation

The following Figure 25 shows the SASD diagram of the AccessAgent.

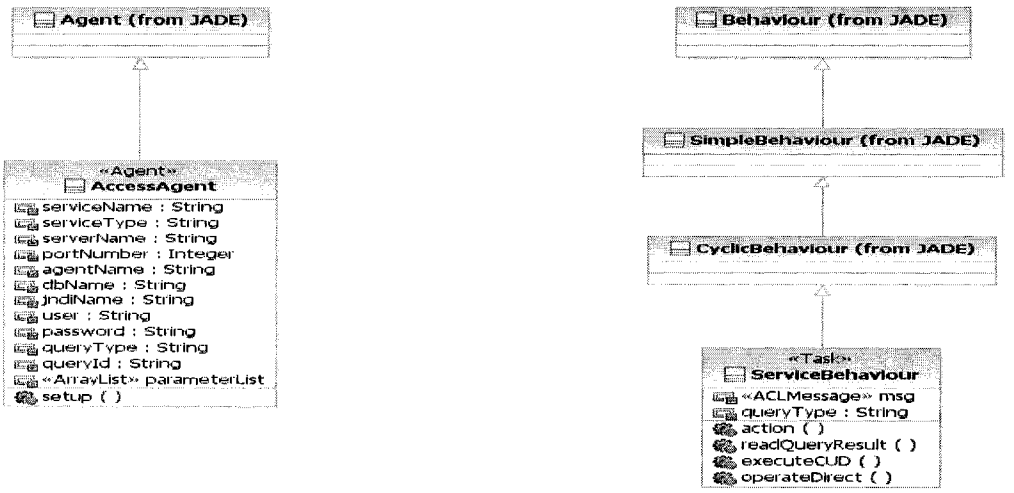


Figure 25 AccessAgent Single Agent Structure Description Diagram

Aside from its main class, the AccessAgent has one task, the class ServiceBehaviour, which extends the OneShotBehaviour class of JADE platform and keeps executing continuously. The AccessAgent initiates its JDBC connection, registers its service with DF agent, and instantiates the ServiceBehaviour. The ServiceBehaiour task is a cyclic one. Upon receiving the request ACLMessage, it responses it. Figure 26 shows the SABD of the AccessAgent.

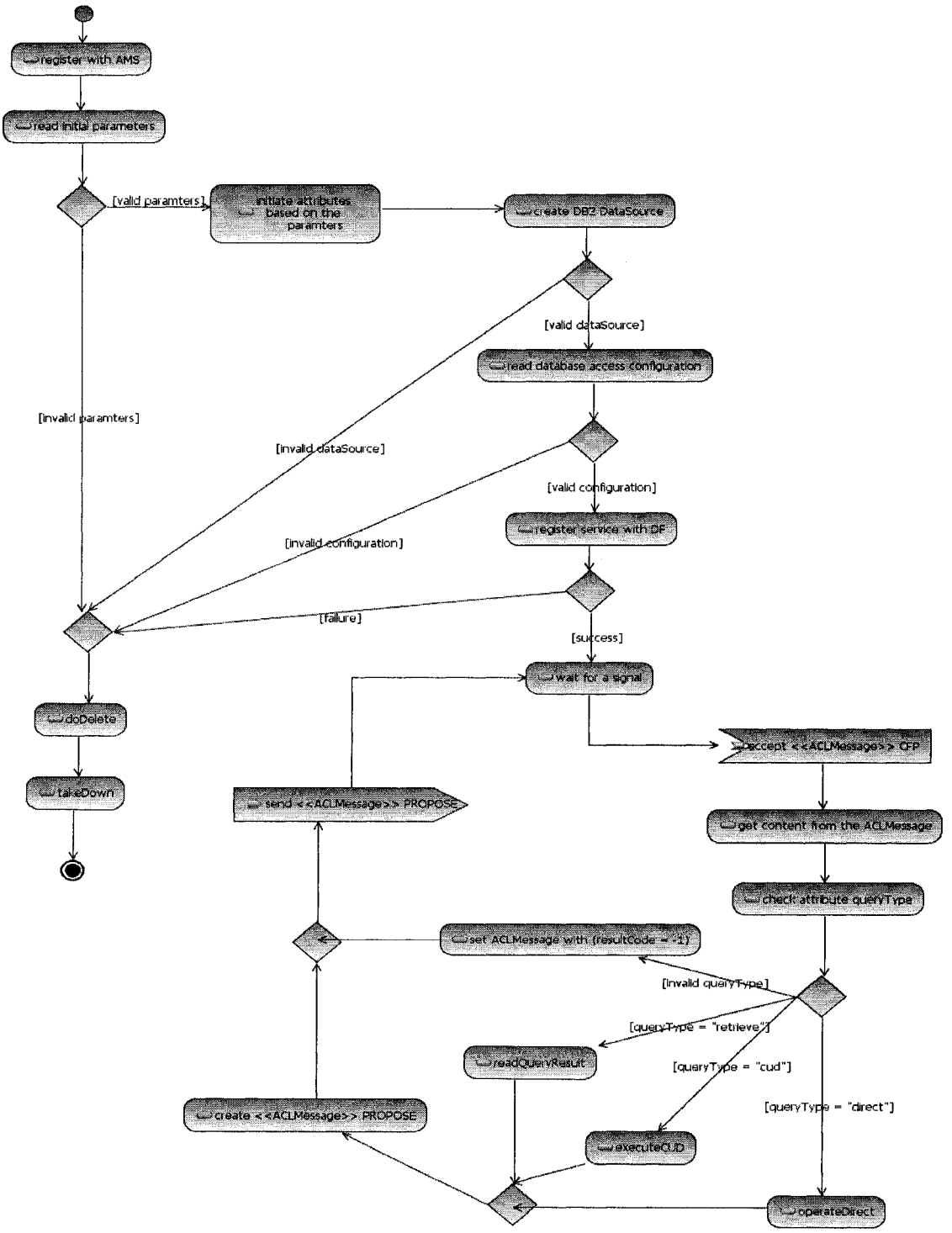


Figure 26 AccessAgent Single Agent Behavior Description Diagram

As depicted in Figure 26, an AccessAgent registers itself with the AMS agent at the beginning. Then it reads and checks its parameters. If the parameters are invalid, the AccessAgent will terminate. Otherwise, it tries to create a DB2 JDBC data source. If the operation fails, it terminates. If successful, it continues to register its service with the DF agent. Then it instantiates the cyclic task, ServiceBehavior. Firstly, the task is blocked and waits for request signals. Upon the arrival of proper request messages, the task extracts useful information from the message and directs the operation flow to one of the three optional methods based on the value of the "queryType". Subsequently, the returned result from the optional method is incorporated into the reply ACLMessage. If anything is wrong during the above steps of the task, a specific reply ACLMessage is created. After sending out the reply ACLMessage, the task backs to its blocked and waiting status.

4.4. Summary

In this chapter, the implementation environment of the IDeAL Framework is introduced. We provide an overview of the whole system, including the hardware, the software, and their relationships. Then, the software tools we use are described. JAVA is used as the programming language due to its platform-independence, high performance, and robustness. Apache Tomcat is chosen as the application server and web server due to its stableness and easy deployment. Further, Apache Struts is used to construct the MVC (Model-View-Controller) skeleton of the implementation. JADE is chosen as the multi-agent system development platform since it has JAVA-based feature and is effectiveness as an agent development framework. IBM DB2 is used as the DBMS based on its high performance, security, and availability.

The J2EE implementation of the IDeAL Framework is explained in detail. The PASSI method is used to describe the Multi-Agent System. It is suitable because it is a step-by-step requirement-to-code methodology for designing and developing Multi-Agent Systems. The architecture of the Multi-Agent System is depicted with MASD (Multi-Agent Structure Diagram). Also, one MABD (Multi-Agent Behavior Diagram) is used to illustrate the flow of events inside the Multi-Agent System. Using the SASD (Single Agent Structure Diagram), the internal compositions and the tasks of the two major types of agents, namely the SRBAgent and the AccessAgent are detailed. The SABD (Single Agent Behavior Diagram) is used to explain the activities of the two types of agents.

All of the above provide a workable platform for the experiments, as discussed in the next chapter.

Chapter 5. Experiments

5.1. Evaluation Methodologies Overview

As far as the author is aware, there are no widely accepted methods to prove the success of a long term preservation repository. However, some standards, such as OAIS, are broadly accepted when long term preservation repositories are implemented. Further, several general evaluation metrics for trusted digital repositories are proposed in [Nestor, 2006], including data integrity, data authenticity, the necessary functionality of digital repositories, amongst others.

In order to evaluate the ability of the IDeAL Framework to preserve digital data for a long time, we decided to use a combination of theoretical proof and empirical confirmation. The compliance of the IDeAL Framework with OAIS, PREMIS and METS was assessed first. Then, the framework was evaluated against the general metrics for trusted digital repositories.

Furthermore, the achievement of other objectives of the IDeAL Framework was appraised. These include providing uniform web-based access interfaces, coping with the scalability and evolution of the digital data, and retrieving data from heterogeneous data sources. These objectives were implied by the specific characteristics of the IDeAL Framework and their functions were clearly defined and described, the corresponding experiments were implemented.

All of the experimental practices were implemented based on two similar data sets: CAESAR1 and CAESAR2. The two data sets contained various data types, such as integer, string, date, time, 2D image, amongst others. Moreover, CAESAR2 could be considered as an evolution from CAESAR1. The detail of the two data sets will be introduced later.

5.2. Assessment of the Compliance of the IDeAL Framework with OAIS

First, we will briefly introduce the meaning of OAIS Compliant. Then, the compliance of the IDeAL Framework will be validated from three aspects: compliance with OAIS responsibilities, conformance to OAIS functional entities, and conformity with OAIS information model [CCSDS, 2002].

5.2.1 What Does It Mean to Be OAIS Compliant?

The OAIS reference model offers a common set of concepts, responsibilities, information models, and processes. The OAIS standard claims to be a basis for compliance [CCSDS, 2002]. Firstly, a standard or other document that claims to be conformant to the OAIS Reference Model should use the terms and concepts defined in the OAIS Reference Model in the same manner. For example, the same terminology is used. Secondly, the OAIS-compliant archive implementation should support the information model as introduced in Section 2.1.3, although the OAIS Reference Model does not define or require any particular method of implementation of these concepts. Thirdly, the OAIS-compliant archive should fulfil the OAIS responsibilities as noted in Section 2.1.3. Finally, the

OAIS reference model is recommended as a guide, but no assumption is made for any specific computing platform, system environment, system design paradigm, system development methodology, database management system, database design paradigm, data definition language, command language, system interface, user interface, technology, or media required for implementation [CCSDS, 2002]. Additionally, an OAIS-compliant archive may provide additional services to users that are beyond those required of an OAIS. Therefore, a data preservation system can claim to be OAIS-compliant if it conforms to OAIS responsibilities, OAIS information model, and OAIS functional model.

5.2.2 Compliance with OAIS Responsibilities

As introduced in Section 2.1.3, the OAIS standard establishes six mandatory responsibilities that an organization must discharge in order to operate an OAIS archive [CCSDS, 2002]. The OAIS archive must:

- Negotiate for and acceptance of appropriate information from Producers.
- Obtain sufficient control of the information provided to the level needed to ensure long-term data preservation.
- Determine, either by itself or in conjunction with other parties, which communities should become the Designated Community and, therefore, should be able to understand the information provided.
- Ensure that the information to be preserved is Independently Understandable to the Designated Community.
- Follow documented policies and procedures which ensure that the information is preserved against all reasonable contingencies, and which enable the information to be disseminated as authenticated copies of the original, or as traceable to the original.
- Make the preserved information available to the Designated Community.

These responsibilities are normally carried out by digital archives. Detail of the compliance to these responsibilities is described as follows.

Negotiate for and Accept Information from Producer

"An organization operating an OAIS will have established some criteria that aids in determining the types of information that it is willing to, or it is required to, accept." [CCSDS, 2002]. The OAIS standard requires that the OAIS archive should extract, or otherwise obtain, sufficient descriptive information from the data producers.

It is normal for digital archives to set certain criteria on quality of the data they accept from depositors. This practice has become a standard for long-term data preservation archives and it can fulfil this responsibility only on the condition that the deposited material meets certain criteria [Beedham, Missen, & Palmer, 2004]. The IDeAL Framework collects data from databases, which contain data in the e-society fields, including e-health, e-government and e-business domains. These databases may involve not only relational data but also multimedia. For example, the experimental data set CAESAR1 is a case of e-health database, which at least contains a mixture of textual and numeric data as well as images. Currently, the IDeAL Framework depends on the Producer's knowledge of the IDeAL Framework and of their data sets to fulfil this responsibility. In future, essential metadata that accompany a data submission and general guidance on depositing can be strictly defined and enforced. This is due to the fact that the IDeAL Framework not only consists of all requisite metadata but also provides a web-based unified user interface (IDeAL Framework Portal) for creating, reading, updating and deleting both digital data objects and their corresponding metadata.

Obtain Sufficient Control for Preservation

The OAIS standard recommends that, when acquiring digital data from any other Producer or entity, the OAIS archive must ensure that there is a legally valid transfer agreement that either transfers intellectual property rights to the OAIS, or clearly specifies the rights granted to the OAIS and any limitations imposed by the rights holder. Further, the OAIS must ensure that its subsequent actions to preserve the information and to make it available conform to these rights and limitations. At the same time, the OAIS archive must assume sufficient control over the objects and their metadata [CCSDS, 2002].

This responsibility is mainly about copyright, intellectual property, and other legal restrictions, which can be accomplished through signing essential legal transfer documents with the depositor for the data that is to be stored in the IDeAL Framework.

Determine Designated Consumer Community

The OAIS standard requires that the Designated Community is identified when digital data is submitted or planned to be submitted, because this is necessary in order to determine if the digital data will be understandable to that specific community [CCSDS, 2002].

Since the IDeAL Framework may contain data from various domains in the e-society, the user community is broad and one specific community is largely determined by the digital data characteristics and its contractual obligations with the depositor.

Ensure the Information is Independently Understandable

The OAIS archive must be capable to preserve the usability and maintain the understand ability of its data, so that the preserved digital data has sufficient metadata to allow its information to be understood and used by the Designated Community without having to resort to special resources not widely available, including named individuals [CCSDS, 2002].

The IDeAL Framework is adopting the emulation strategy as its primary preservation strategy. The strategy benefits the end users in the way that the end users can view and interact with the preserved data in the same way the original users could. Further, the IDeAL Framework may contain many same types of databases and it can be efficient to use one emulator for all the same type of databases. Moreover, the IDeAL Framework selects open standard for information representation of common data types, ensuring that the data remain manageable over the time. Extensive structural and descriptive metadata is captured during Ingest and subsequent storage archival processing stages. Thus, in the IDeAL Framework, original bit streams of data are held in perpetuity with the original preservation manifestation and full metadata, ensuring that the data information is independently understandable in the far future.

Follow Established Preservation Policies and Procedures

It is essential for an OAIS archive to have documented policies and procedures for preserving its collections, and it should follow those procedures. The Producer and Consumer communities should be provided with submission and dissemination standards,

policies, and procedures to support the preservation objectives of the OAIS [CCSDS, 2002].

Our current research on long-term data preservation mainly focuses on the technical implementation of the IDeAL Framework. The necessary policies and procedures may be documented in near future, because many policies and procedures, such as the preservation strategy and the data ingest procedure, can be summarized based on the current technical implementation. Actually, the Virtual Model contains various metadata, which support the digital preservation processes, including tracking the chain of alterations over time, describing the environment from which the digital data originated, and describing technical details of the digital object, amongst others.

Make the Information Available

By definition, an OAIS makes its archives visible and available to its Designated Communities. Multiple views of its holdings, supported by various search aids, may be provided. The expectations of OAIS users (Consumers) regarding access services will vary widely among archives and over time as technology evolves. Pressures for more effective access must be balanced with the requirements for preservation under the available resource constraints. Moreover, some collections may have restricted access and therefore may only be disseminated to consumers who meet access requirements [CCSDS, 2002].

Current fulfilment of the IDeAL Framework uses the user role to restrict access. The role Consumer can access the metadata of the digital data through the IDeAL Framework Portal. Further, the role Consumer can retrieve digital data through customized database

query statement. In future, the relationship metadata between the digital data and the user can be used to control the data access in a more precise manner. The IDeAL Framework Portal will be improved as the technology evolves.

Conclusions of the Compliance with OAIS Responsibilities

The compliance testing of the IDeAL Framework against the OAIS mandatory responsibilities shows that the IDeAL Framework can fulfil the the technical aspects of these responsibilities. The biggest discrepancy with the OAIS is revealed to be the definition and documentation of the policies and procedures, which falls outside the scope of this thesis.

5.2.3 Compliance with OAIS Functional Entities

As introduced in Section 2.1.3, the OAIS reference model defines six functional entities: Ingest, Data Management, Archival Storage, Access, Administration, and Preservation Planning. The assessments of the compliance with the OAIS functional entities are described as follows.

Ingest

This entity provides the services and functions to accept Submission Information Packages (SIPs) from Producers and prepare the contents for storage and management within the archive. As illustrated in Figure 27, Ingest functions include five services: receiving SIPs, performing quality assurance on SIPs, generating an Archival Information Package (AIP) which complies with the archive's data formatting and documentation standards, extracting Descriptive Information from the AIPs for inclusion in the archive database, and coordinating updates to Archival Storage and Data Management. The five services are introduced as follows [CCSDS, 2002]:

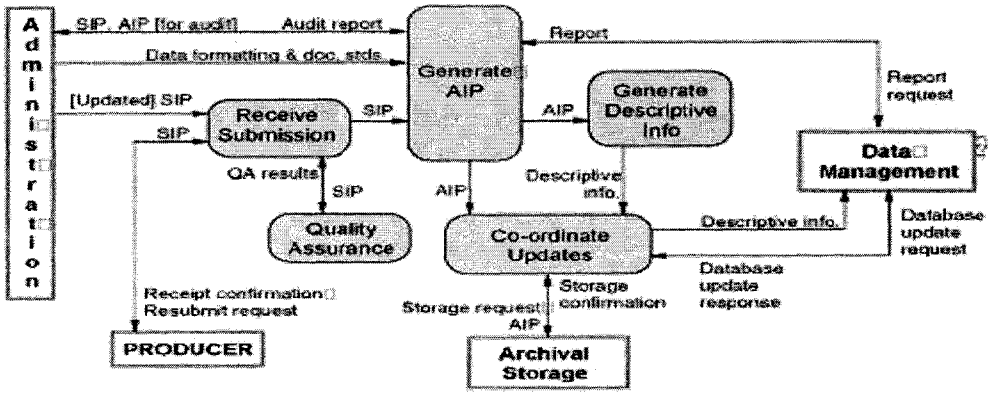


Figure 27 Functions of Ingest [CCSDS, 2002]

The first service, the Receive Submission function, provides the appropriate storage capability or devices to receive a SIP from the Producer [CCSDS, 2002]. The IDeAL Framework receives the SIP from the Producer in twofold: one is the data set in a database; the other is the corresponding metadata, which is input through the IDeAL Framework Portal and is stored in the Virtual Model. Since the IDeAL Framework focuses on the long-term preservation of data in databases, a critical Ingest function is to receive the metadata of the tables in databases. The Figure 28 shows the page for creating metadata of a table in the Virtual Model.

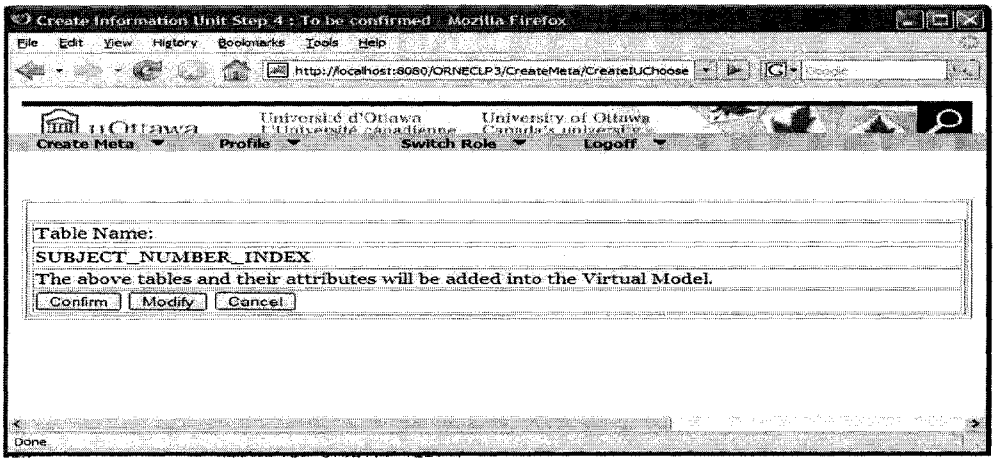


Figure 28 Create Metadata for a Table

The second service, the Quality Assurance function, validates the successful transfer of the SIP to the staging area [CCSDS, 2002]. Currently, the Producer of the IDeAL Framework takes the responsibility to assure the quality of the SIP through validating the content of the data set, assuring the completion of the data, checking the metadata, amongst others.

The third service, the Generate AIP function, transforms one or more SIPs into one or more AIPs that conform to the archive's data formatting and documentation standards [CCSDS, 2002]. In the current IDeAL Framework, all AIPs are the combination of one digital object in a data set and its corresponding metadata in a Virtual Model.

The fourth service, the Generate Descriptive Information function, extracts Descriptive Information from the AIPs and collects Descriptive Information from other sources to provide to Coordinate Updates, and ultimately Data Management [CCSDS, 2002]. In the IDeAL Framework, the descriptive information is first collected and generated during the creation of the metadata of a data set. Later, the descriptive information can be modified manually through the IDeAL Framework Portal. The following Figure 29 is an index page for manually create or modify metadata in the Virtual Model.

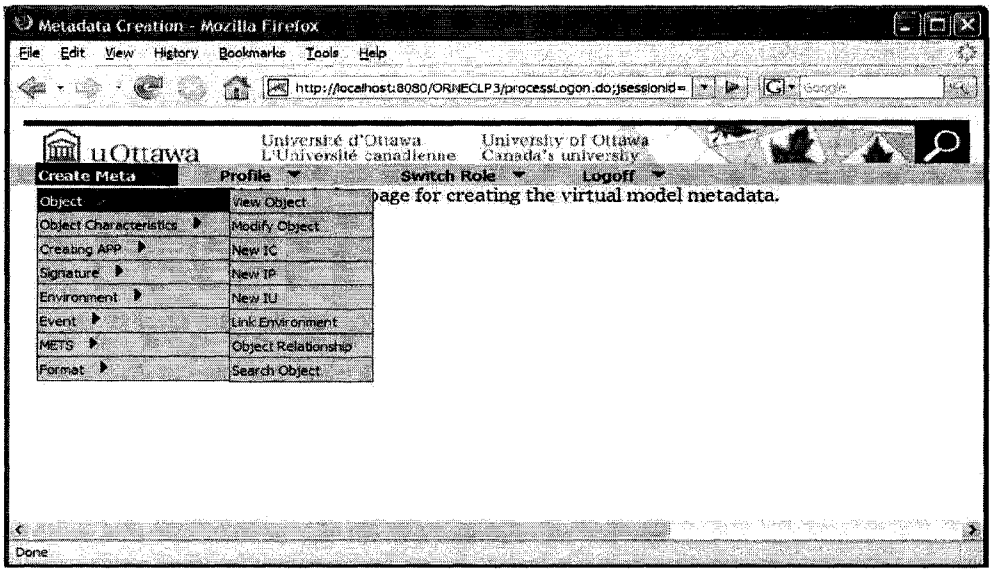


Figure 29 Functions of Archival Storage [CCSDS, 2002]

The fifth service, the Coordinate Updates function, is responsible for transferring the AIPs to Archival Storage and the Descriptive Information to Data Management [CCSDS, 2002]. In the current IDeAL Framework, after the Receive Submission function, the data set is transferred to Archival Storage and the metadata including Descriptive Information is stored in the Virtual Model of the Framework.

Archival Storage

This entity provides the services and functions for the storage, maintenance and retrieval of AIPs. As depicted in Figure 30, Archival Storage functions include five services: receiving AIPs from Ingest and adding them to permanent storage, managing the storage hierarchy, refreshing the media on which archive holdings are stored, performing routine and special error checking, providing disaster recovery capabilities, and providing AIPs to Access to fulfill orders [CCSDS, 2002].

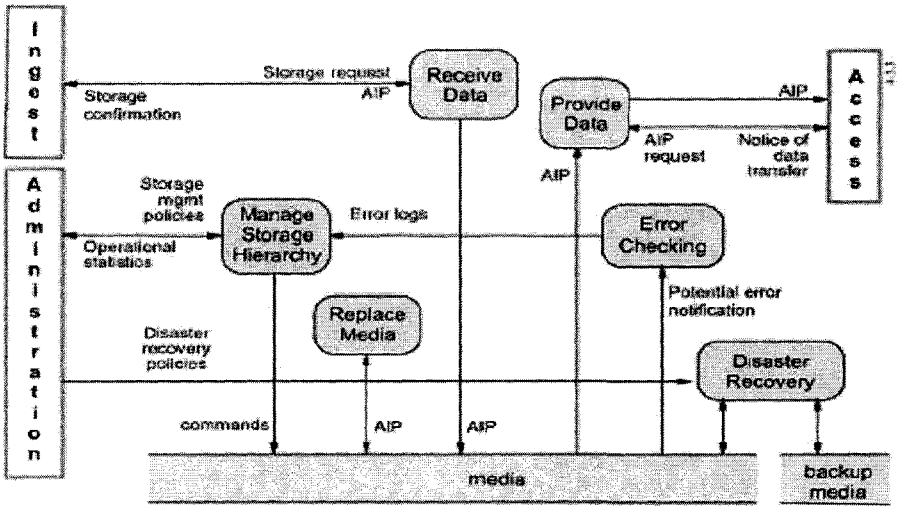


Figure 30 Functions of Archival Storage [CCSDS, 2002]

The IDeAL Framework is implemented on an ultra-fast fibre-optic mass storage in University of Ottawa. A tape-based backup system is used to backup the data in the IDeAL Framework. The Archival Storage function is achieved mainly through the Business Logic Process System, the Multi-agent System, and the Digital Repositories in the IDeAL Framework.

The Receive Data function receives a storage request and an AIP from Ingest and moves the AIP to permanent storage within the archive. This function will select the media type, prepare the devices or volumes, and perform the physical transfer to the Archival Storage volumes [CCSDS, 2002]. In the current implementation, the Producer takes the responsibility to store the data into specific databases on the designated computer platform. Through the IDeAL Portal, the Producer can determine where to store the metadata.

The Manage Storage Hierarchy function positions the contents of the AIPs on the appropriate media based on storage management policies, operational statistics, or directions from Ingest via the storage request [CCSDS, 2002]. In the IDeAL Framework, the Producer stores the data set into a specified database on a designated media. Then the location information is input into the Virtual Model of the data set. However, there are no storage management policies and operational statistics in current IDeAL Framework. The IDeAL Framework depends on its Multi-agent System to determine the positions of a data set and its related Virtual Model. Figure 31 depicts the location information obtained from the Multi-agent System.

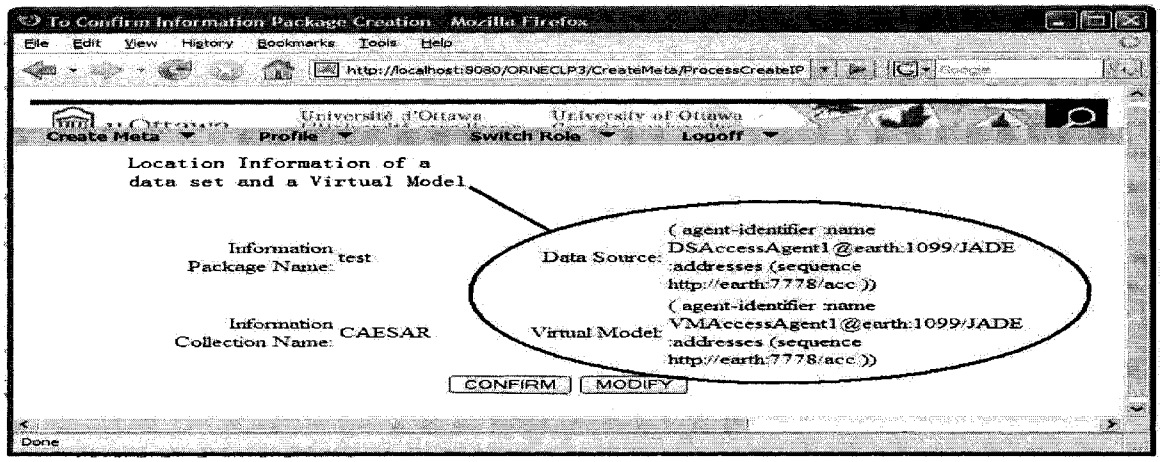


Figure 31 Location Information of a Data Set and a Virtual Model

The Error Checking function provides statistically acceptable assurance that no components of the AIP are corrupted during any internal Archival Storage data transfer [CCSDS, 2002]. The IDeAL Framework bases this function on the error checking functions of its implementation infrastructure, including those in DB2, operating system (Windows and Linux), fibre-optic mass storage, amongst others.

The Replace Media function is used to supersede the obsolete media with new ones. The Disaster Recovery function provides a mechanism for duplicating the digital contents of the archive collection and storing the duplicate in a physically separate facility [CCSDS, 2002]. The Replace Media function is mainly one part of the administration tasks of a computer system. The Disaster Recovery is mainly for the business continuity and is considered as an individual research area. Although they benefit the preservation of digital data, they are not in the scope of this research. Moreover, they can be incorporated with the IDeAL Framework easily in the future as add-on functions.

The Provide Data function provides copies of stored AIPs to Access. This function receives an AIP request that identifies the requested AIP(s) and provides them on the requested media type or transfers them to a staging area [CCSDS, 2002]. Currently, the IDeAL Framework can fulfil this function through its Business Logic Process System and Multi-agent System. Specific data can be retrieved by defining customized database query statement in the IDeAL Framework Portal and its metadata can also be obtained in the similar way.

Data Management

This entity provides the services and functions for populating, maintaining, and accessing both Descriptive Information which identifies and documents archive holdings and administrative data used to manage the archive. Figure 32 depicts that Data Management functions include four services: administering the archive database functions, performing database updates, performing queries on the data management data to generate result sets, and producing reports from these result sets [CCSDS, 2002]. An OAIS archive requires

abundant metadata to assure the usability and understandability of the digital data for a long time. Further, an efficient metadata management system is needed to manage those complex metadata. In the IDeAL Framework, all metadata is stored in the Virtual Model and the manipulation functions of these metadata are the important part of the framework.

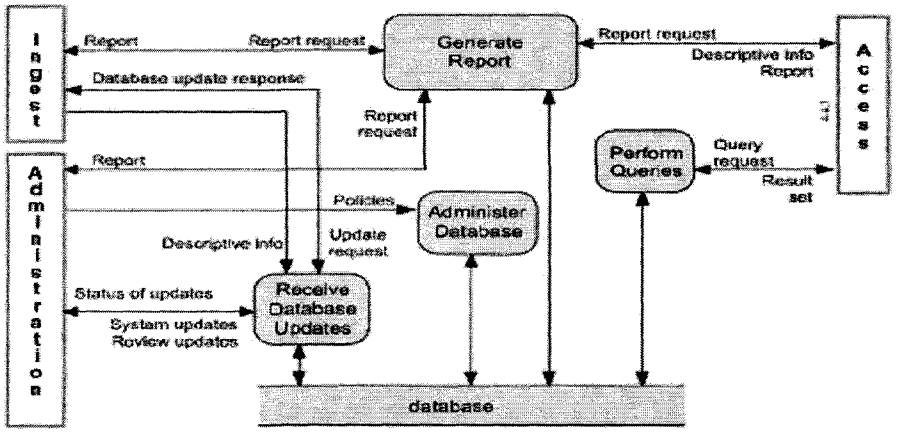


Figure 32 Functions of Data Management [CCSDS, 2002]

The Administer Database function is responsible for maintaining the integrity of the Data Management database, which contains both Descriptive Information and system information [CCSDS, 2002]. This function is fulfilled by IBM DB2 in the current IDeAL Framework.

The Perform Queries function receives a query request from Access and executes the query to generate a result set that is transmitted to the requester [CCSDS, 2002]. This function is carried out through the Multi-agent System.

The Generate Report function receives a report request from Ingest, Access or Administration and executes any queries or other processes necessary to generate the report that it supplies to the requester. [CCSDS, 2002] The current IDeAL Framework focuses on

the core long-term preservation tasks, such as metadata design, and data archival and retrieval. The report functions can be added in the future.

The Receive Database Updates function adds, modifies or deletes information in the Data Management persistent storage [CCSDS, 2002]. This function is achieved through the Multi-agent System.

Administration

As presented in Figure 33, the Administration entity provides the services and functions for the overall operation of the archive system. Administration functions include soliciting and negotiating submission agreements with Producers, auditing submissions to ensure that they meet archive standards, and maintaining configuration management of system hardware and software. It also provides system engineering functions to monitor and improve archive operations, and to inventory, report on, and migrate/update the contents of the archive. It is also responsible for establishing and maintaining archive standards and policies, providing customer support, and activating stored requests [CCSDS, 2002].

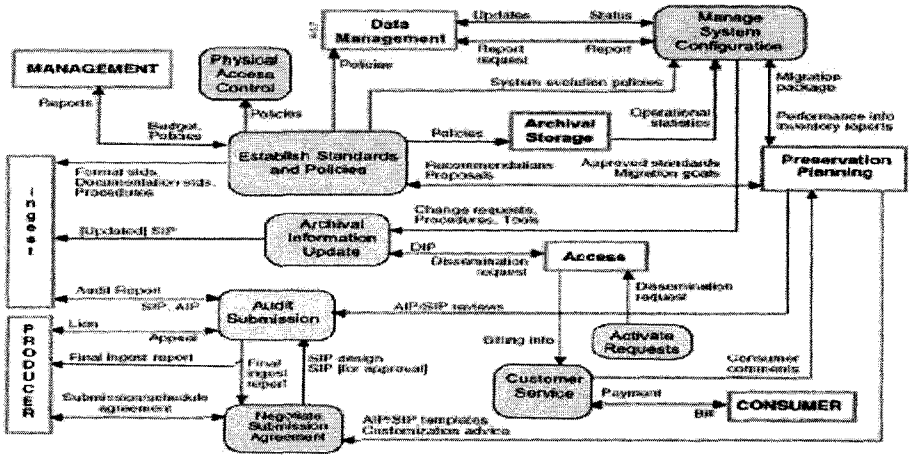


Figure 33 Functions of Administration [CCSDS, 2002]

The Negotiate Submission Agreement function solicits desirable archival information for the OAIS and negotiates Submission Agreements with Producers [CCSDS, 2002]. This function is about the policy and the procedures, so it will be considered in the later phase of the IDeAL Framework lifecycle.

The Manage System Configuration function provides system engineering for the archive system to continuously monitor the functionality of the entire archive system and systematically control changes to the configuration [CCSDS, 2002]. 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. Subsequently, it can be added into the IDeAL Framework later.

The Archival Information Update function provides a mechanism for updating the contents of the archive [CCSDS, 2002]. The update to the data in the IDeAL Framework is through the IDeAL Portal interface. The update operations to the Virtual Model are achieved through the access agents in the Multi-agent System.

The Physical Access Control function provides mechanisms to restrict or allow physical access (doors, locks, guards) to elements of the archive [CCSDS, 2002]. In current IDeAL Framework, this function is provided by University of Ottawa, because the IDeAL Framework is implemented on the computing facilities in the computing lab of University of Ottawa.

The Establish Standards and Policies function is responsible for establishing and maintaining the archive system standards and policies [CCSDS, 2002]. Our current researches

on long-term data preservation mainly focus on the technical implementation of the IDeAL Framework. This function may be fulfilled in near future, because many policies and procedures, such as the preservation strategy and the data Ingest procedure, can be summarized based on the current technical implementation.

The Audit Submission function will verify that submissions (SIP or AIP) meet the specifications of the Submission Agreement [CCSDS, 2002]. In current implementation of the IDeAL Framework, the Producer takes the responsibility to ensure the authenticity and the integrity of the submission before the submission is ingested into the IDeAL Framework.

The Activate Requests function maintains a record of event-driven requests and periodically compares it to the contents of the archive to determine if all needed data is available [OAIS]. This function is not carried out in current implementation of the IDeAL Framework, because the access request is responded immediately based on the availability of data now.

The Customer Service function will create, maintain and delete Consumer accounts. It will collect billing information from Access and will send bills and collect payment from Consumers for the utilization of archive system resources [CCSDS, 2002]. This function is partially implemented. The Consumer accounts can be manipulated through the IDeAL Framework Portal as shown in Figure 34, but currently there are no policies and procedures for other services, such as consumer billing, system utilization analysis, amongst others.

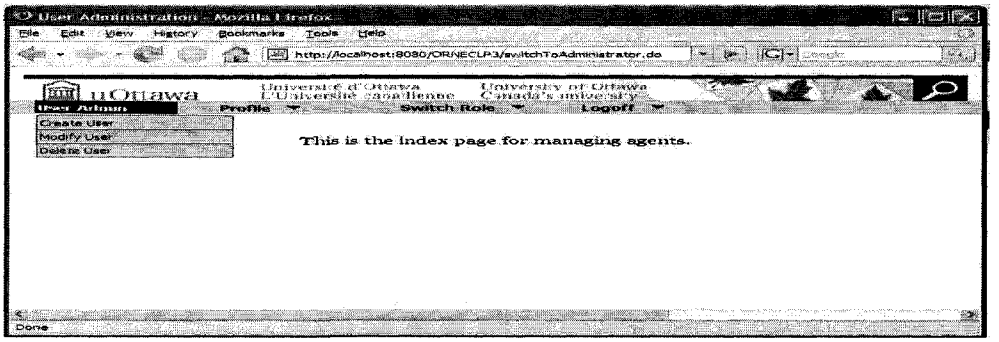


Figure 34 User Administration Index Page

Preservation Planning

This entity provides the services and functions for monitoring the environment of the OAIS and providing recommendations to ensure that the information stored in the OAIS remains accessible to the Designated User Community over the long term, even if the original computing environment becomes obsolete [CCSDS, 2002]. As depicted in Figure 35, Preservation Planning functions include evaluating the contents of the archive and periodically recommending archival information updates to migrate current archive holdings, developing recommendations for archive standards and policies, and monitoring changes in the technology environment and in the Designated Community's service requirements and Knowledge Base.

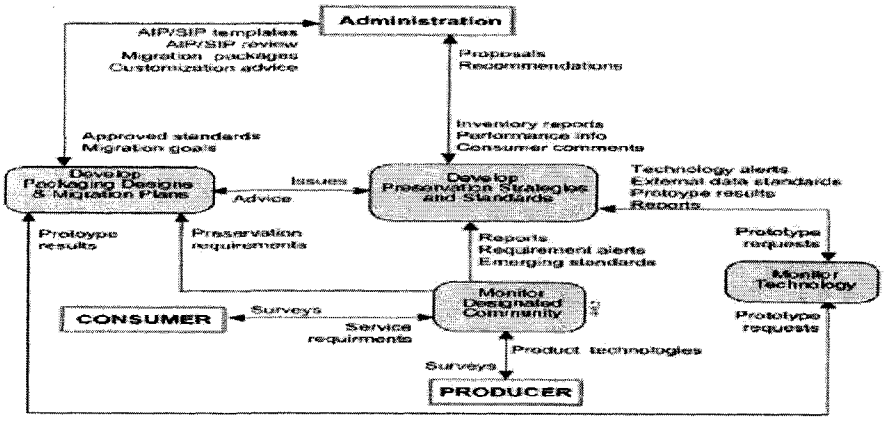


Figure 35 Functions of Preservation Planning [CCSDS, 2002]

The IDeAL Framework developing team monitors the up-to-date technology and standards development. The IDeAL Framework uses the emulation approach as its preservation strategy. Since the Preservation Planning function is not technology-oriented, it can be realized in later phase of the IDeAL Framework lifecycle.

Access

This entity provides the services and functions that support Consumers in determining the existence, description, location and availability of information stored in the OAIS, and allowing Consumers to request and receive information products. As shown in Figure 36, Access functions include communicating with Consumers to receive requests, applying controls to limit access to specially protected information, coordinating the execution of requests to successful completion, generating responses (Dissemination Information Packages, result sets, reports) and delivering the responses to Consumers [CCSDS, 2002].

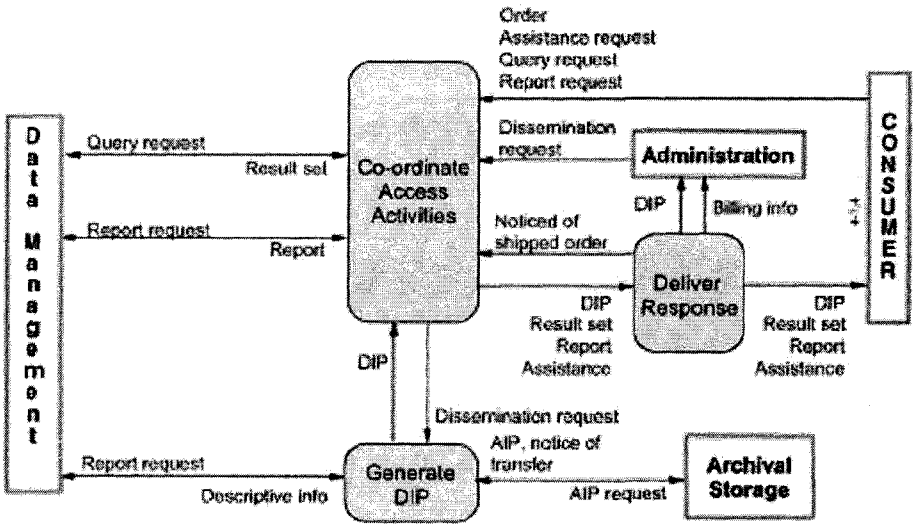


Figure 36 Functions of Access [CCSDS, 2002]

The Coordinate Access Activities function provides a single user interface to the information holdings of the archive. Three categories of Consumer requests are distinguished: query requests, which are executed in Data Management and return immediate result sets for presentation to the user; report requests, which may require a number of queries and produce formatted reports for delivery to the Consumer; and orders, which may access either or both Data Management and Archival Storage to prepare a formal Dissemination Information Package (DIP) for on- or off-line delivery [CCSDS, 2002]. The current IDeAL Framework Portal is implemented as a single user interface. Among the three categories of requests, the service of the query requests is achieved and other two services can be achieved in future implementation of the IDeAL Framework. The following Figure 37 shows the index page of the user interface in the IDeAL Framework.

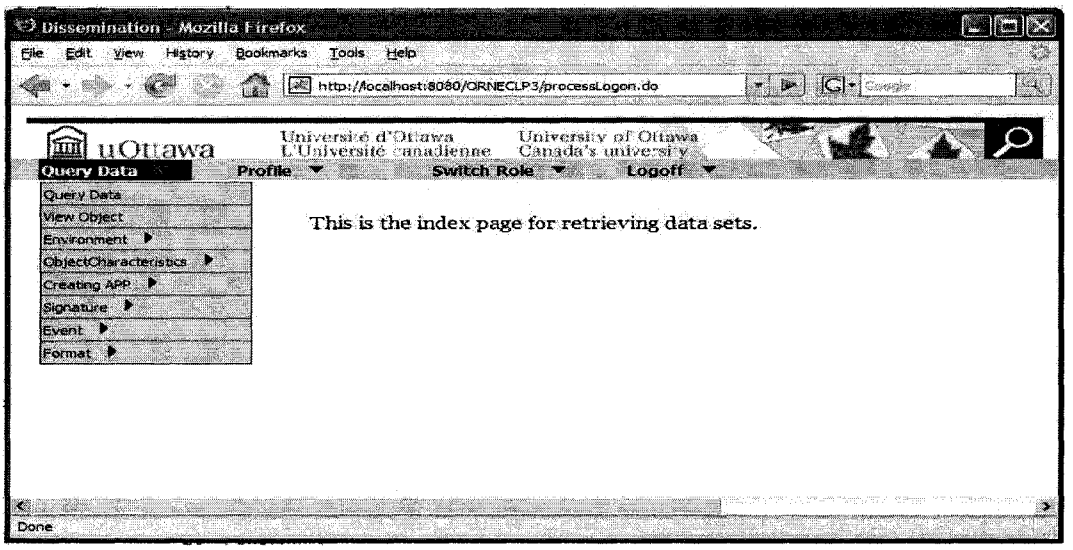


Figure 37 Dissemination Index Page

The Generate DIP function accepts a dissemination request, retrieves the AIP from Archival Storage, and moves a copy of the data to a staging area for further processing [CCSDS, 2002]. The digital data and its metadata in the Virtual Model are used to construct the DIP.

The Deliver Response function handles both on-line and off-line deliveries of responses (DIPs, result sets, reports and assistance) to Consumers [CCSDS, 2002]. The current IDeAL Framework provides on-line delivery of responses through the IDeAL Framework Portal.

Conclusions of the Compliance with OAIS Functional Entities

The current IDeAL Framework focuses on the technology-oriented functions in the functional model of OAIS. Within all six functional entities, the Preservation Planning function entity and the Administration function entity are barely realized, because they are mainly concerned with polices, procedures, agreements, and responsibilities. Overall, the core functions of a long-term data preservation system are implemented. The following table is used to summarize the fulfilment of these functional entities in the IDeAL Framework.

OAIS Functional Entity	Sub-function	Is it fulfilled in the IDeAL Framework ?	Comments
Ingest	Receive Submission	Yes	
	Quality Assurance	Yes	
	Generate AIP	Yes	
	Generate Descriptive Information	Yes	
	Coordinate Update	Yes	
Archival Storage	Receive Data	Yes	
	Manage Storage Hierarchy	Yes	
	Replace Media	No	The Replace Media function is mainly one part of the administration tasks of a computer system. The Disaster Recovery is mainly for the business continuity. They are not in the scope of the research.
	Disaster Recovery		
	Error Checking	Yes	
	Provide Data	Yes	
Data Management	Administer Database	Yes	
	Perform Queries	Yes	
	Generate Report	No	The current IDeAL Framework focuses on the core long-term preservation tasks, such as metadata design, and data archival and retrieval. The report functions can be added in the future.
	Receive Database Updates	Yes	
Administration	Negotiate Submission Agreement	No	This function is about the policy and the procedures, so it will be considered in the later phase of the IDeAL Framework lifecycle.
	Manage System Configuration	No	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	Yes	
	Physical Access Control	Yes	
	Establish Standards and Policies	No	This function may be fulfilled in near future, because many policies and procedures, such as the preservation strategy and the data ingest procedure, can be summarized based on the current technical implementation.
	Audit Submission	Yes	
	Activate Requests	No	Instead of event-driven, the access request is responded immediately based on the availability of data now.
	Customer Service	Yes	
Preservation Planning	Monitor technology, Monitor Designated Community, Develop Preservation Strategies and Standards, and Develop Packaging Design and Migration Plans	Partially	The IDeAL Framework developing team monitors the up-to-date technology and standards development. The IDeAL Framework uses the emulation approach as its preservation strategy. Since the Preservation Planning function is not technology-oriented, it can be realized in later phase of the IDeAL Framework lifecycle.
Access	Coordinate Access Activities	Yes	
	Generate DIP	Yes	
	Deliver Response	Partially	Only online delivery is implemented.

Table 2 OAIS Functional Entities Implemented in the IDeAL Framework

5.2.4 Compliance with OAIS Information Model

As introduced in Section 2.1.3, the information package is the core concept of the OAIS information model. Within the OAIS model, three types of information package are identified: the Submission Information Package (SIP), which is sent from the information Producer to the archive; the Archive Information Package (AIP), which is the information package actually stored by the archive; and the Dissemination Information Package (DIP), which is the information package transferred from the archive in response to a request by a consumer [CCSDS, 2002].

A detailed view of an AIP is depicted in Figure 38. An AIP is composed of four types of information objects: Content Information, Preservation Description Information, Packaging Information and Descriptive Information. Content Information is the primary information of interest (the Data Object and its associated Representation Information). The Representation Information is the combination of the Structure Information and the Semantic Information. Preservation Description Information (PDI) contains information necessary to adequately preserve its associated Content Information. Typically, PDI is divided into four sections: provenance information, context information, reference information, and fixity information. Packaging Information binds the components of the information package into an identifiable entity [CCSDS, 2002].

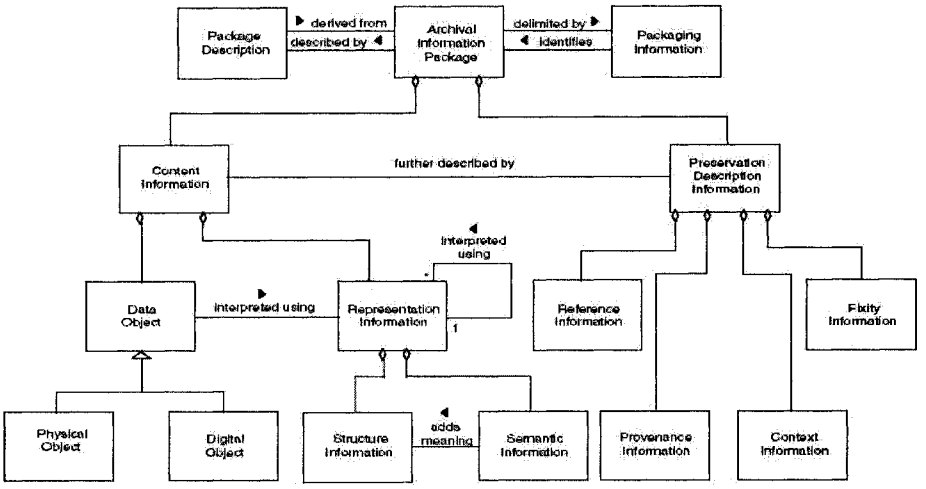


Figure 38 AIP Detailed View [CCSDS, 2002]

During the Ingest processing stage, a typical SIP submitted to the IDeAL Framework is the data in a data set and its related metadata about the data, such as data type, relationship, and fixity, amongst others. More metadata is added manually through the IDeAL Framework Portal. All metadata are stored in the Virtual Model. A digital object in the data set and its related metadata constitute an AIP. When requested, a DIP can be constructed based on the data in the data sets and the Virtual Models in the IDeAL Framework. Table 3 shows the mapping between the entities of the AIP (depicted in Figure 38) and the tables in the Virtual Model. The table indicates that the OAIS Information Model is successfully adopted into the IDeAL Framework, because almost all entities related to digital data are realized in the Virtual Model. The only exception, Packaging Information, can be obtained through the usage of METS, as stated in Section 5.3.2.

Entity in AIP	Table in Virtual Model	Comment
Package Description	object	
Packaging Information		Packaging is achieved through the usage of METS as stated in Section 5.3.2.
Data Object	object	The actual bit stream of the data object is in the data sets of the IDeAL Framework.
Physical Object		Not exist in the IDeAL Framework.
Digital Object		The data in the data sets of the IDeAL Framework.
Representation Information	environment, hardware, software, dependency, format, formatRegistry, creatingApplication	
Structure Information	object	
Semantic Information	objectCharacteristics, format, md, mdWrap	
Reference Information	object	ArkID of table object is a persistent identifier across various preservation respositories.
Provenance Information	Event, eventOutcomeInformation, object_event, event_user, md, mdWrap	
Context Information	Object, relationship, object_relationship, event_relationship, object_user	
Fixity Information	objectCharacteristics, signatureInformation	

Table 3 Mapping between Elements of AIP and Tables in the Virtual Model

Conclusions of the Compliance with OAIS Information Model

Two facts indicate that the IDeAL Framework conforms to the OAIS Information Model. First, as shown in Table 3, most of the entities of the OAIS information model are in-

cluded in the Virtual Model of the IDeAL Framework. The only exception (Packaging Information) can be obtained through the use of METS. Second, PREMIS defines an implementable, core preservation metadata framework based on OAIS reference model. Table 4 in Section 5.3.1 depicts that the metadata elements of PREMIS are successfully adopted into the Virtual Model, as discussed next.

5.3. Usage of the PREMIS and the METS in the IDeAL Framework

Since the OAIS only provides a reference framework without implementation guidance, a metadata schema must be designed for it. Further, we would like to establish a long-term data preservation framework, which can interoperate with other digital repositories. Thus, some standards, such as PREMIS and METS, were brought into consideration when the Virtual Model in the IDeAL Framework was designed.

5.3.1 Usage of the PREMIS in the IDeAL Framework

As introduced in Section 2.1.3, PREMIS is a metadata framework to support the preservation of digital objects, which is claimed to contain a set of preservation metadata elements. These metadata elements construct an essential and minimal metadata set [OCLC, RLG, 2007]. In [Deborah Woodyard-Robinson Holdings, 2005], sixteen preservation repositories are surveyed regarding their implementation of PREMIS, including the APSR (Australian Partnership for Sustainable Repositories), FDA (Florida Digital Archive), KB (Koninklijke Bibliotheek), amongst others. Although the implementation in these reposi-

tories may differ in scale and data management practices, their success justifies the rationale of PREMIS as a workable framework. Thus, the ideas of the PREMIS are adopted into the Virtual Model of the IDeAL Framework.

PREMIS contains a data model of five types of entities involved in digital preservation activities. All of the five types of entities have their own properties, named Semantic Units. In the IDeAL Framework, the metadata of the Virtual Model are stored in databases and are grouped into database tables as depicted in Figure 13. Since we would like to measure the extent of the implementation of PREMIS in the IDeAL Framework, the mapping between the semantic units of the PREMIS data model and the tables of the Virtual Model is assessed in the following Table 4. Note that the Intellectual Entity is generally treated as an entire preservation repository and it does not have specific semantic units, it is excluded from the following mapping.

Entities in PREMIS	Semantic Unit in PREMIS	Table in the Virtual Model	Comment
Object Entity	objectIdentifier	object	
	preservationLevel	object	
	objectCategory	object	
	objectCharactertistics	object, objectCharacteristics, format, formatRegistry	The properties of the semantic unit objectCharacteristics are in four tables of the Virtual Model.
	creatingApplication	creatingApplication	
	originalName		It is not adopted in the Virtual Model.
	Storage	hardware	It is included into the hardware table.
	signatureInformation	signatureInformation	
	Environment	environment, software, hardware, dependency	The properties of the semantic unit environment are in four tables of the Virtual Model.
	Relationship	relationship, object_relationship, event_relationship	The properties of the semantic unit relationship are in three tables of the Virtual Model.
Event Entity	eventIdentifier	event	
	eventType	event	
	eventDateTime	event	
	eventDetails	event	
	eventOutcomeInformation	eventOutcomeInformation	
	linkingAgentIdentifier	event_user	
	linkingObjectIdentifier	object_event	
Agent Entity	agentIdentifier	vmuser	
	agentName	vmuser	
	agentType	vmuser	
Rights Entity	all semantic units of rights entity	md, mdWrap, mdSchema	

Table 4 Mapping between the Semantic Units and Tables in the Virtual Model

Table 4 shows that the metadata elements of PREMIS are successfully adopted into the Virtual Model. Except for one Semantic Unit ('originalName'), all Semantic Units of PREMIS are mapping to tables in the Virtual Model. The 'originalName' of the Object

Entity is not included in any table of the Virtual Model, since a persistent identifier, 'ARK ID' in table 'object', is used to identify each object in the IDeAL Framework. Therefore, it is not feasible to give original name to each record in a database. Some Semantic Units are combined into one table, due to efficiency reasons. For example, table object contains properties of 'objectIdentifier', 'preservationLevel', and 'objectCategory'. However, some properties, belonging to one Semantic Unit in PREMIS, are distributed into several tables in the Virtual Model. This is due to our design requirement that unnecessary duplication of information is minimized. For example, properties of 'environment' are divided into table environment, table software, table hardware, and table dependency. Note that, since one software or hardware component may belong to multiple environments, this design minimizes storage overhead.

5.3.2 Usage of the METS in the IDeAL Framework

One of the main challenges facing all digital repositories is the provision of seamless access to the assets within the repository. Access is partially dependent on the provision of different levels of metadata to describe the assets. The METS system would appear to be a good overall metadata solution as it fulfils all the criteria within the Open Archives Initiative's Protocol for Metadata Harvesting (OAI-PMH) framework [DLF, 2007]. Given that the OAIS Reference Model allows the conceptual mapping between heterogeneous systems, METS is one method of implementing this concept [Beedham et al., 2004]. Moreover, as introduced in Section 2.1.3, METS is a flexible and software independent platform, which can be used for the interoperability between digital repositories by providing a framework for integrating various types of metadata. The METS standard pro-

vides a method to combine a digital object and its diverse metadata as a whole, which can be shared, exchanged, and searched. Except our essential requirement that the metadata schema is compliant with the OAIS, two crucial requirements are considered when designing the metadata schema: designing a dissemination template of the DIPs (data and its metadata) in the Framework and preparing for the potential of the interoperability with other digital repositories. Therefore, the METS approach is adopted into the Virtual Model of the IDeAL Framework. Currently, no standard DIP template is defined in the IDeAL Framework, but the architecture of the IDeAL Framework supports various possible designs of DIPs. The following is an example XML of these designs.

```
<mets OBJID="ARKA#123" LABEL="caesar_thumbnails_a" TYPE="Attribute">

      <metsHdr>
            <agent ROLE="CREATOR">
                  <name>Tom Lorie</name>
            </agent>
      </metsHdr>

      <amdSec>
      <techMD ID="premis1">
            <mdWrap MDTYPE="PREMIS">
                  <xmlData>
                        <PREMIS metadata/>
                  </xmlData>
            </mdWrap>
      </techMD>
      <techMD ID="image_tech1">
            <mdWrap MDTYPE="NISOMIX">
                  <xmlData>
                        <NISO Metadata for Images in XML />
                  </xmlData>
            </mdWrap>
      </techMD>
      </amdSec>
</mets>
```

In this example, the data in the element OBLID, LABEL, TYPE, and ROLE can be obtained directly from the Virtual Model. PREMIS related metadata in the Virtual Model can be stored in the techMD section for PREMIS. The techMD section for NISOMIX is an example for involving potential metadata that is not visibly included in the Virtual Model. Actually, the Virtual Model is designed with an interface for those metadata that are not involved but useful in some situations. The table md and mdWrap in the Virtual Model constitute the interface. For example, the NISO Metadata for Images in XML (NISO MIX) is only useful for images, so there are no tables in the Virtual Model especially designed for this kind of metadata. Thus, the NISO MIX can be stored in table md and mdWrap. When constructing a DIP, the data in the md and mdWrap can be extracted to fill in the techMD section for NISOMIX as shown in the above XML.

5.4. Description of the Test Data Sets

To test the functionalities of the IDeAL Framework, two data sets were used, which are named as CAESAR1 and CAESAR2. Their content is similar and they can be considered as two versions of the CAESAR^{TM} database, which is a repository of anthropometric data. 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) [Viktor, Paquet, & Guo, 2006]. Anthropometry is used in many application areas, such as the design of clothes, and the design of seats in airplanes and buses. The CAESAR^{TM} Project is an international anthropometric survey that was carried out in the United States, Italy, the Netherlands and Canada. This survey included many individuals in each country. For

each individual, various highly accurate anthropometric measurements were performed and questions of demographic nature were collected. The anthropometric measurements included forty-nine details which have been recorded by domain experts, using standard anthropometric practices [Viktor et al., 2006]. These include the stature, weight, thigh circumference, feet length, amongst others. The demographic data contains the family income, number of children, age, amongst others. We chose them because their characteristics and relationships matched the test data requirements of the experiments.

During the lifecycle of the IDeAL Framework, many types of data may be stored into the framework. Data sets with multiple data types are required to evaluate the capability of the Framework to manipulate and retrieve various data. The data types in CAESAR^{TM} contain 2D images, 3D images, STRING, SHORT, INTEGER, DECIMAL, DATE, and TIME. Most data in CAESAR1 and CAESAR2 are same, but there is some minor difference. Thus, they are used as two versions of one data set. Figure 39 shows the database schema of CAESAR1.

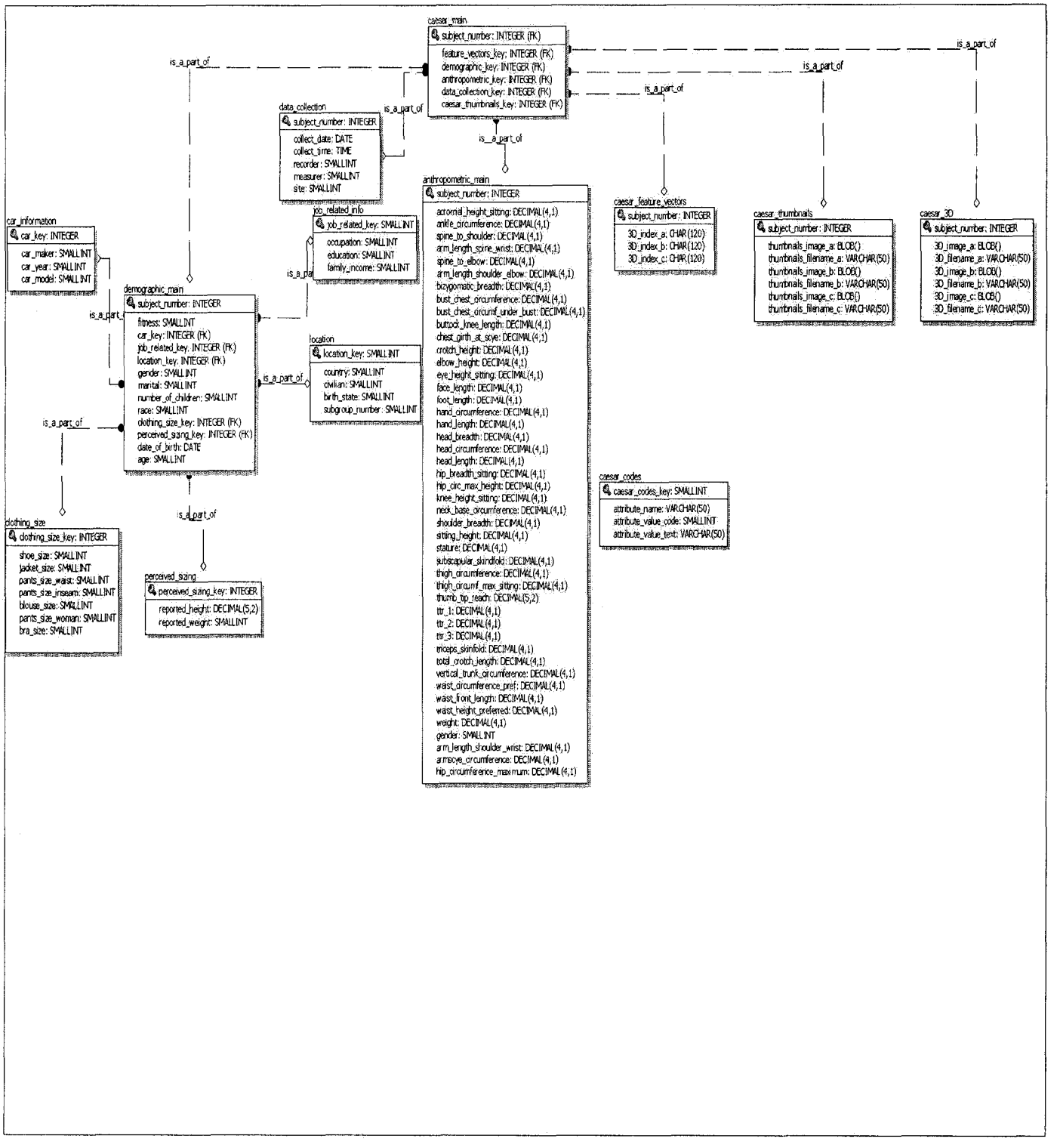


Figure 39 CAESAR1 Database Schema

The data set CAESAR1 contains 13 tables. Table anthropometric_main contains the major anthropometric data. Table caesar_thumbnails includes the 2D views of the human body scans. Table caesar_3D and caesar_feature_vectors houses 3D human body scans. Table demographic_main, car_information, location, job_related_info, clothing_size, and perceived_sizing involve the demographic data, such as family income, number of children, age, amongst others. Table caesar_codes is used as a reference table.

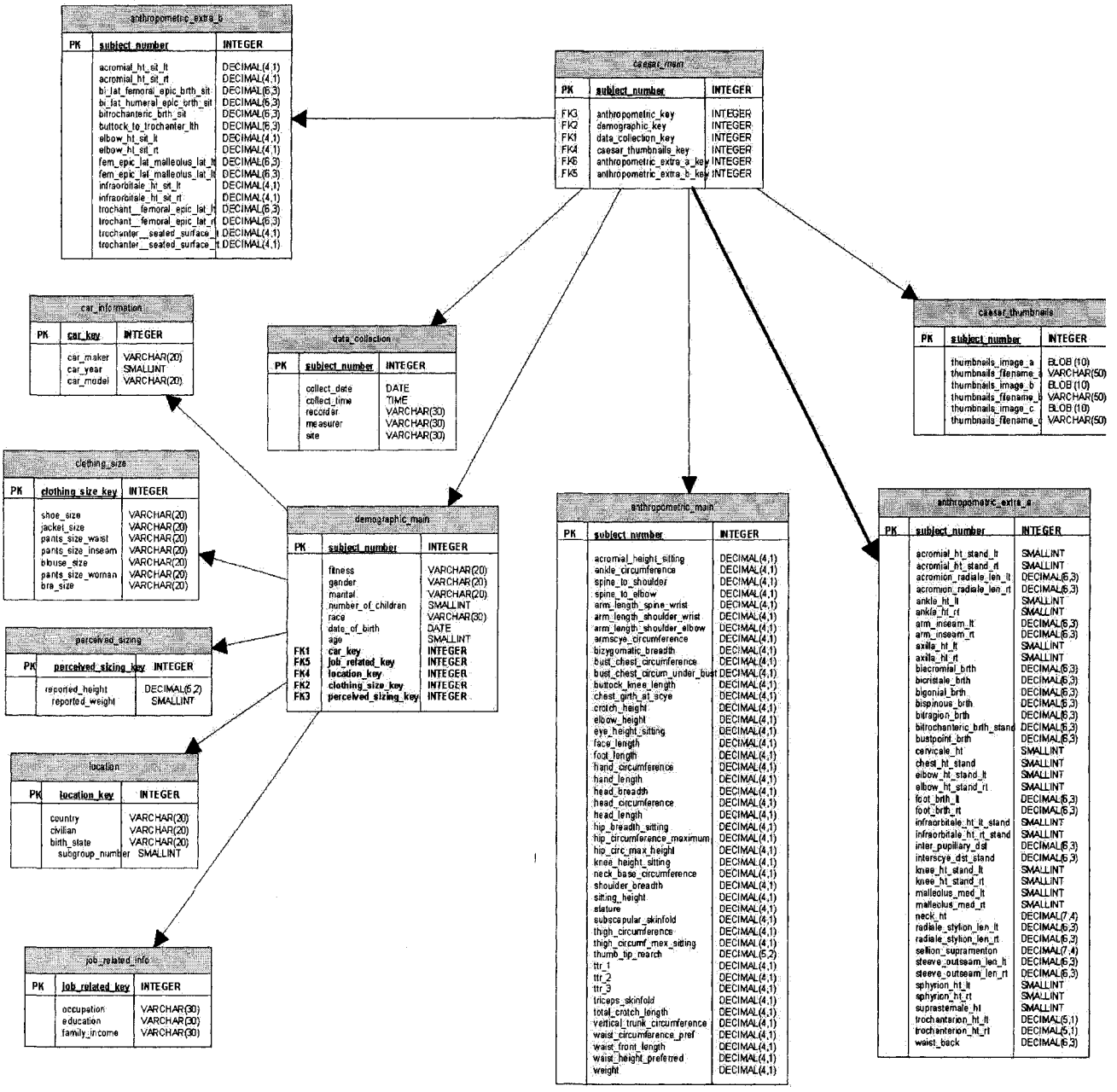


Figure 40 CAESAR2 Database Schema

Figure 40 presents the database schema of CAESAR2. The data set CAESAR2 contains 12 tables. Table anthropometric_main, anthropometric_extra_a, and anthropomet-

ric_extra_b contain the major anthropometric data. Table caesar_thumbnails includes the 2D views of the human body scans. Table demographic_main, car_information, location, job_related_info, clothing_size, and perceived_sizing involve the demographic data, such as family income, number of children, age, amongst others.

The major differences between CAESAR1 and CAESAR2 include four major points. First, CAESAR1 contains table CAESAR_3D and table CAESAR_FEATURE_VECTORS, which are for 3D data; while, CAESAR2 does not involves 3D data. Second, CAESAR2 contains two tables which are not in CAESAR1, ANTHROPOMETRIC_EXTRA_A and ANTHROPOMETRIC_EXTRA_B. Third, the content of table CAESAR_THUMBNAILS are different in CAESAR1 and CAESAR2, although the content represent the same information of the data. Finally, some column names are dissimilar in CAESAR1 and CAESAR2, even though the content in those columns are the same.

After loading CAESAR1 and CAESAR2 into the IDeAL Framework, the related metadata needed to be input into the Virtual Model. As described in Chapter 3, five specializations of IP are used in the IDeAL Framework: Information Collection, Information Package, Information Unit, Column, and Attribute. In our experiment, one Information Collection, CAESAR, was created. CAESAR contained two Information Packages, named as CAESAR1 and CAESAR2. The metadata of CAESAR1 and CAESAR2 resided in different Virtual Models.

In the IDeAL Framework, all of the relationships among specializations were first separated into two types: STRUCTURAL and DERIVATION. Then, the relationships were further divided into subtypes. In our experiment, the relationships between the specializations inside CAESAR1 and CAESAR2 included REFERENCE TABLE, FOREIGN KEY, ATTRIBUTE REFERENCE, and SIMILAR.

To clearly illustrate the metadata of the experimental data sets in the Virtual Models, we use an XML file as depicted in Appendix I. Figure 41 shows the corresponding XML schema. The schema clarifies the five data specializations in the Virtual Model and their relationships. As introduced in Section 3.3.1, the five data categories are Information Collection, Information Package, Information Unit, Column, and Attribute. Moreover, the relationships among the five specializations are included in the schema too.

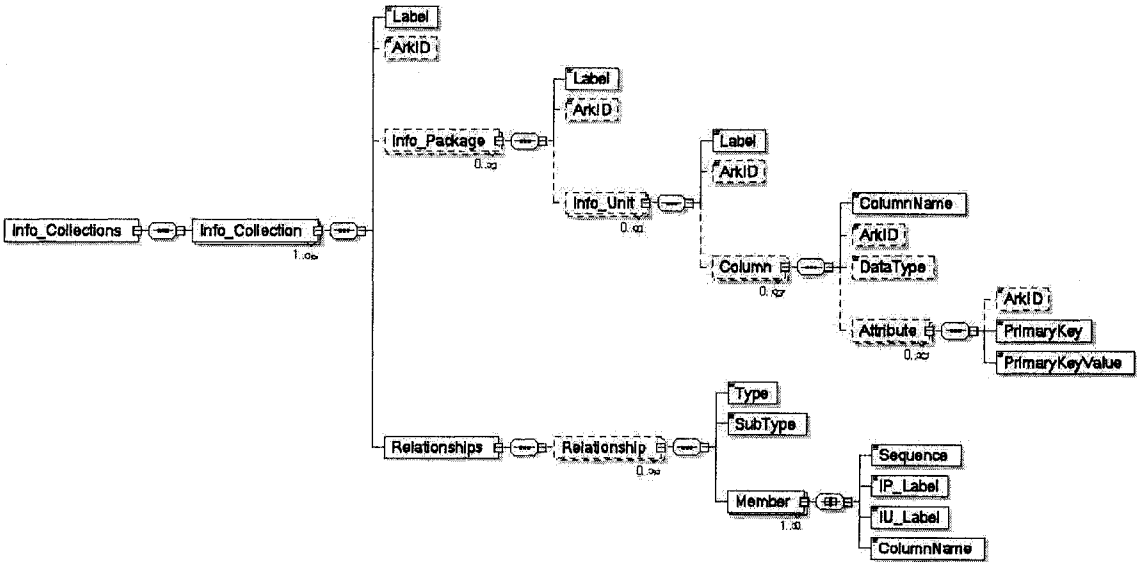


Figure 41 Test Data XML Schema

In the Appendix I, not only the specializations except for the attributes are listed, but also all necessary relationships are recorded. However, only an excerpt of the whole XML is cited, due to the considerable length of the entire XML document. The relationships are crucial for the data integration function. Currently, four relationships were used for the experiment. The following are detailed explanations of these relationships:

1. REFERENCE TABLE is a table into which an enumerated set of possible values of a certain field data type is divested. This relationship is depicted as below:

```
<Relationship>
    <Type>STRUCTURAL</Type>
    <SubType>REFERENTIAL TABLE</SubType>
    <Member>
        <Sequence>0</Sequence>
        <IP_Label>CAESAR1</IP_Label>
        <IU_Label>CAESAR_CODES</IU_Label>
        <ColumnName/>
    </Member>
</Relationship>
```

2. FOREIGN KEY is a referential constraint between two tables, which identifies a column or a set of columns in one (referencing) table that refers to a column or set of columns in another (referenced) table.

```
<Relationship>
    <Type>STRUCTURAL</Type>
    <SubType>FOREIGN KEY</SubType>
    <Member>
        <Sequence>0</Sequence>
        <IP_Label>CAESAR1</IP_Label>
        <IU_Label>CAESAR_MAIN</IU_Label>
        <ColumnName>
        ANTHROPOMETRIC_KEY</ColumnName>
    </Member>
    <Member>
        <Sequence>1</Sequence>
        <IP_Label>CAESAR1</IP_Label>
```

```
                <IU_Label>
                    ANTHROPOMETRIC_MAIN
                </IU_Label>
                <ColumnName>
                    SUBJECT_NUMBER
                </ColumnName>
            </Member>
        </Relationship>
```

3. ATTRIBUTE REFERENCE is the relationship for the columns that need to refer to the columns in the reference table.

```
<Relationship>
    <Type>STRUCTURAL</Type>
    <SubType>ATTRIBUTE REFERENCE</SubType>
    <Member>
        <Sequence>0</Sequence>
        <IP_Label>CAESAR1</IP_Label>
        <IU_Label>LOCATION</IU_Label>
        <ColumnName>COUNTRY</ColumnName>
    </Member>
    <Member>
        <Sequence>1</Sequence>
        <IP_Label>CAESAR1</IP_Label>
        <IU_Label>CAESAR_CODES</IU_Label>
        <ColumnName>ATTRIBUTE_NAME</ColumnName>
    </Member>
    <Member>
        <Sequence>2</Sequence>
        <IP_Label>CAESAR1</IP_Label>
        <IU_Label>CAESAR_CODES</IU_Label>
        <ColumnName>ATTRIBUTE_VALUE_CODE</ColumnName>
    </Member>
    <Member>
        <Sequence>3</Sequence>
        <IP_Label>CAESAR1</IP_Label>
        <IU_Label>CAESAR_CODES</IU_Label>
        <ColumnName>ATTRIBUTE_VALUE_TEXT</ColumnName>
    </Member>
</Relationship>
```

4. SIMILAR is the relationship between the similar columns in two related Information Packages of one Information Collection. This relationship is created for data integration.

```
<Relationship>
    <Type>STRUCTURAL</Type>
    <SubType>SIMILAR</SubType>
    <Member>
        <Sequence>0</Sequence>
        <IP_Label>CAESAR1</IP_Label>
        <IU_Label>DATA_COLLECTION</IU_Label>
        <ColumnName>COLLECT_TIME</ColumnName>
    </Member>
    <Member>
        <Sequence>1</Sequence>
        <IP_Label>CAESAR2</IP_Label>
        <IU_Label>DATA_COLLECTION</IU_Label>
        <ColumnName>COLLECT_TIME</ColumnName>
    </Member>
</Relationship>
```

5.5. Evaluation of the General Metrics for Trusted Digital Repositories

One of the central challenges to long term preservation in a digital repository is the ability to guarantee the interpretability of digital objects for users over a very long time, which includes an assurance of integrity, authenticity, and the necessary functionality of the repository. General metrics for trusted digital repositories are given in [Nestor, 2006]. 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 digital data, we mainly assess three metrics, namely integrity, authenticity, and the necessary functionality of digital repositories. In

[Nestor, 2006], the processing stages correspond in the OAIS reference model to the processes of submission (ingest), storage (archival storage), and usage (access). The evaluation of the general metrics may happen in any or all of these processing stages.

5.5.1 Evaluation of Integrity of Digital Objects

The IDeAL Framework should be able to ensure the integrity of the digital objects during all processing stages. Integrity refers to the completeness of the digital objects and to the exclusion of unintended modifications. Inappropriate modifications may be caused by human error, technical imperfections or damage to the technical infrastructure. We used the assurance of the integrity of a 2D image to evaluate the integrity of digital objects in the IDeAL Framework. Totally there were six steps in this evaluation.

The first step was to calculate the message digest of a 2D image (51.jpg) as shown in Figure 42. "md5deep" was a cross-platform set of programs to compute MD5, SHA-1, SHA-256 Tiger, or Whirlpool message digests on an arbitrary number of files.

Figure 42 Message Digest of 1.jpg

The second step was to logon into the IDeAL Framework with proper authorization. There were three types of user roles, each of which has specific access rights. As shown in Figure 43, the logon page was to ensure that no unauthorized user can obtain rights over digital objects and metadata.

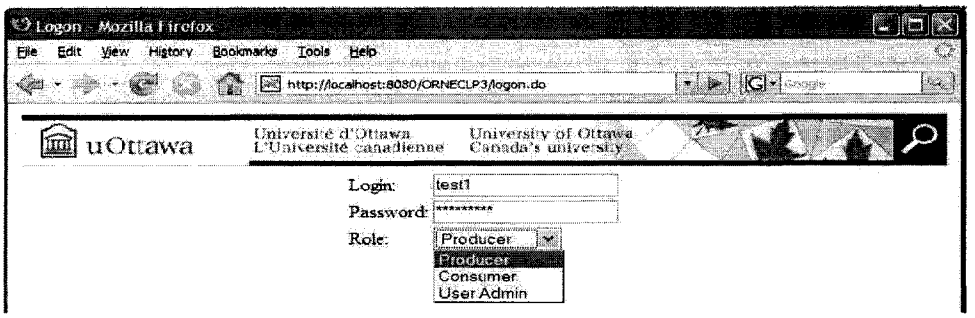


Figure 43 Logon Page

The third step was to manually input the message digest metadata as shown in Figure 44.

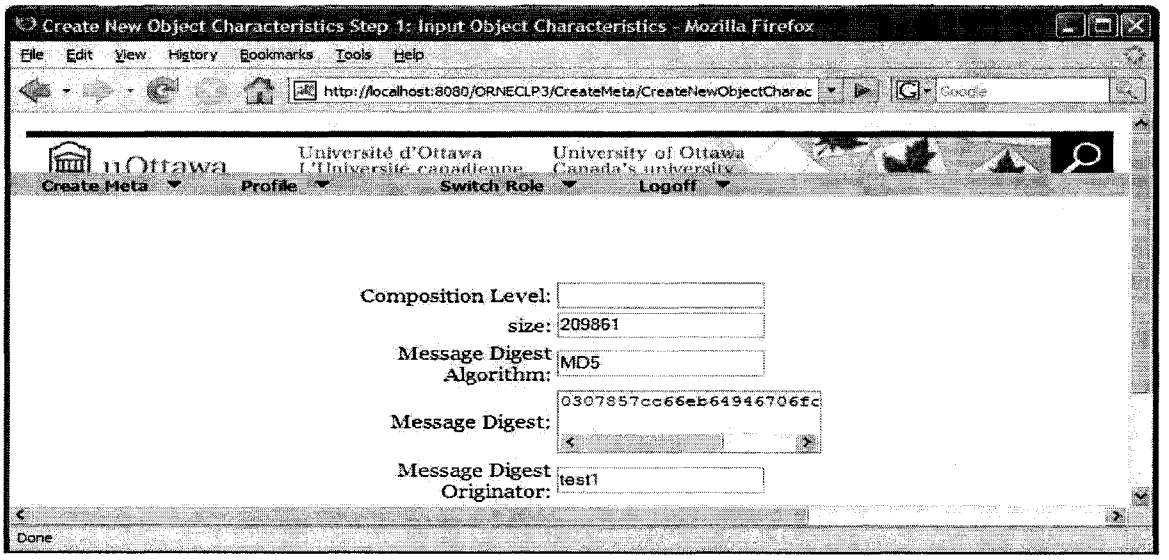


Figure 44 Input Message Digest

The fourth step was to retrieve the 2D image through the IDeAL Framework Portal after the image was loaded into a data set and its metadata is input into the Virtual Model. The query result is shown as Figure 45.

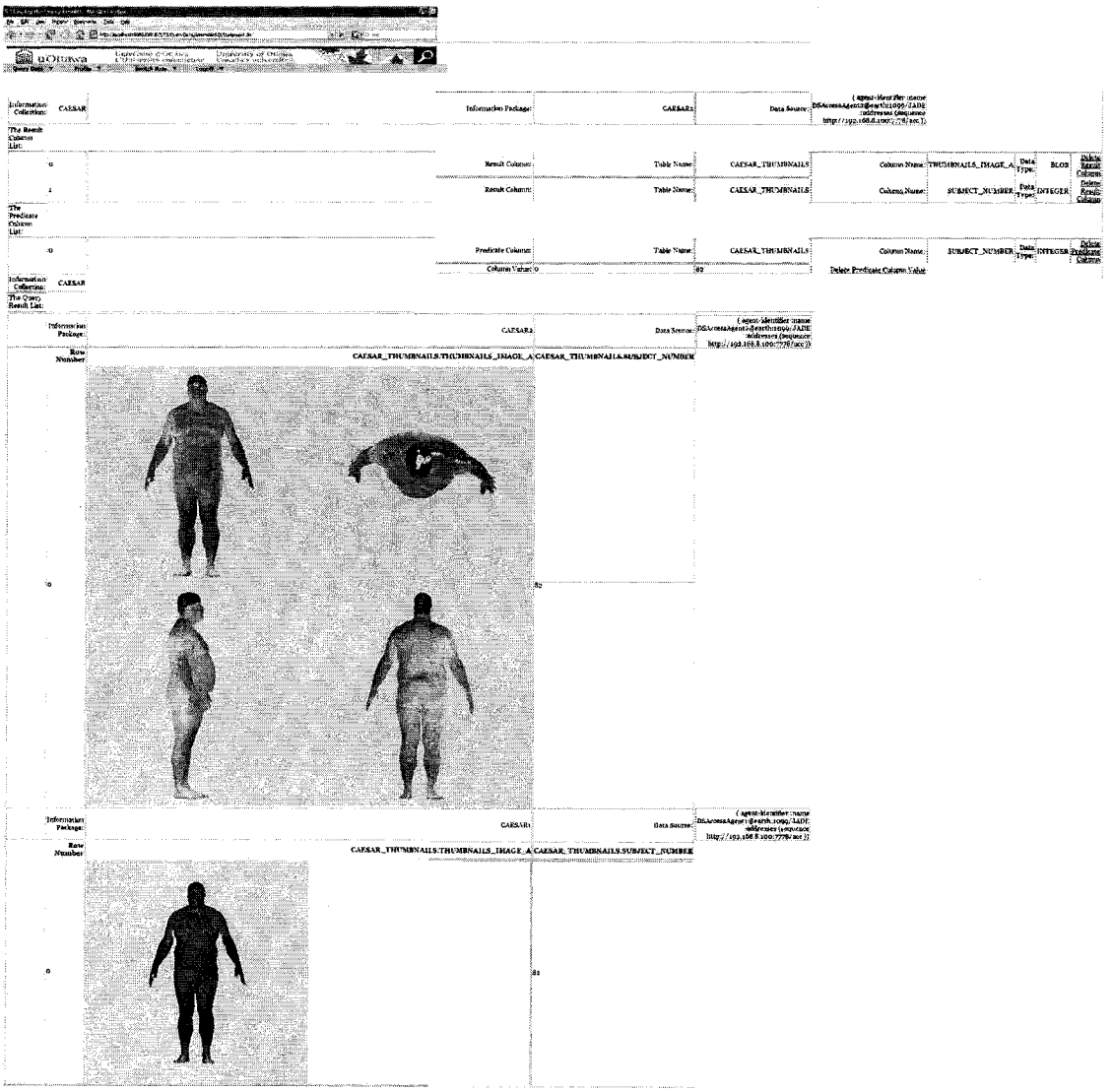


Figure 45 Query Result of a 2D Image

The fifth step was to save the result image to file "showImage51.jpg" and to calculate the message digest of "showImage.jpg". The experimental result is shown in Figure 46.

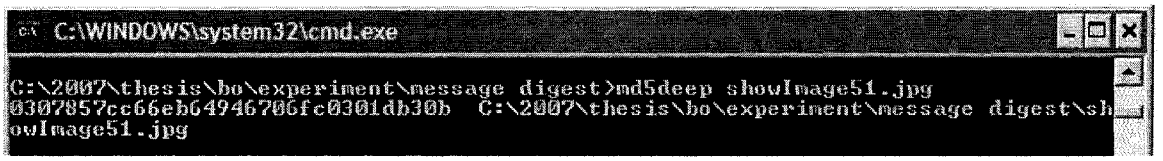


Figure 46 Message Digest of showImae.jpg

The sixth step was to retrieve the message digest of the 2D image as shown in Figure 47.

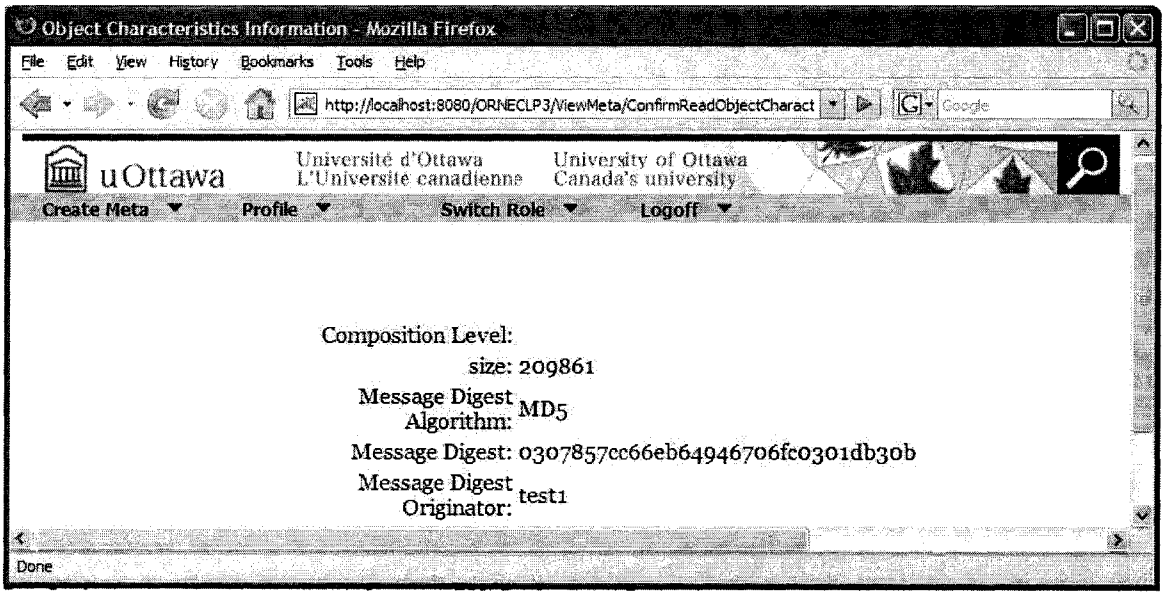


Figure 47 Get the Message Digest

The equality of the message digest obtained from the fifth step and the sixth step indicates that the IDeAL Framework can assure the integrity of digital objects.

5.5.2 Evaluation of Authenticity of Digital Objects

The IDeAL Framework should be able to ensure the authenticity of the digital objects during all stages of processing. For a definition of authenticity, we consider Rothernberg's authenticity principle which indicates that for a digital document to be interpreted authentically in the future, it should "exhibit as much as possible of its original behaviour, functionality, look and feel, as well as its content" [Rothenberg, 1999]. About how to ensure data authenticity, Gladney's viewpoint relates to trusting the content of a preserved digital document and he suggests signing a digital object to inform the users that the content can be trusted [Gladney, 2004]. Therefore, Public Key Infrastructure is adopted into the IDeAL Framework and digital signature is used for authenticating data. The authentication mechanism contains six steps as depicted in Figure 48. Since the size of data may be very large, normally the message digest of the data is calculated first. Then the message digest is signed with the Producer's private key to produce a digital signature. The combination of the message digest and its signature can be verified by the Producer's public key to make sure if the Producer is authenticated. This digital signature mechanism was carried out in the following example to verify whether user "test1" was the authenticated Producer of a 2D image.

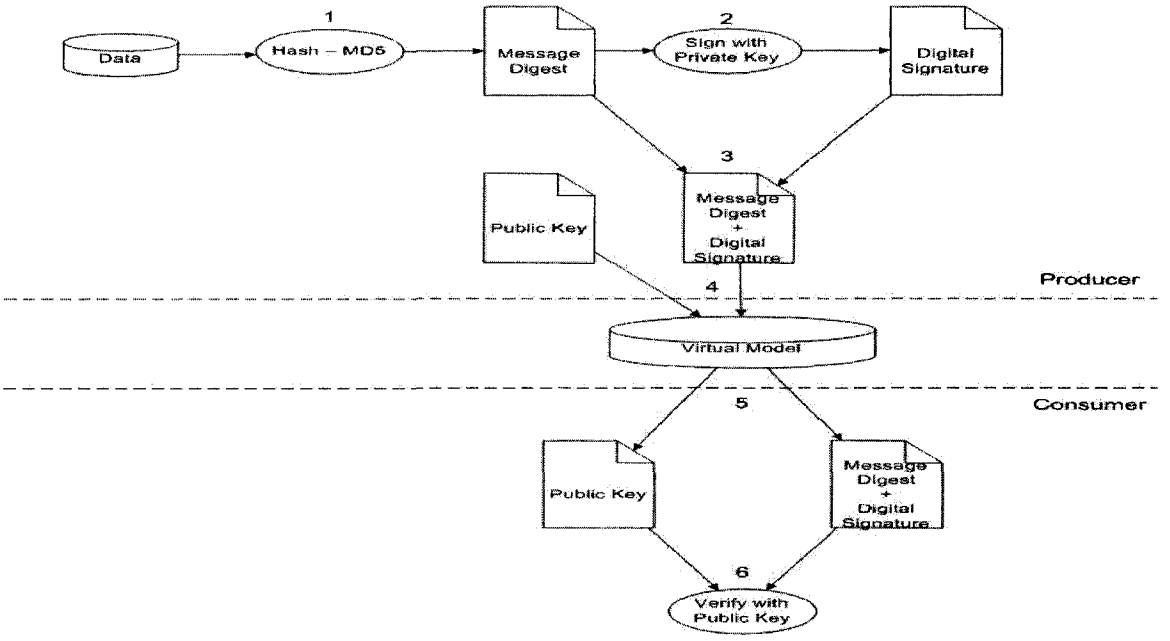


Figure 48 Digital Signature Mechanism [Gladney, 2004]

To assess this functionality, a security standard was needed to provide cryptographic privacy and authentication. Within tens of such standards, OpenPGP was chosen. OpenPGP is a widely used email encryption standard in the world. It is defined by the OpenPGP Working Group of the Internet Engineering Task Force (IETF) Proposed Standard RFC 2440. Moreover, the OpenPGP protocol defines standard formats for encrypted messages, signatures, and certificates for exchanging public keys. GnuPG, the software used in the experiment, can achieve complete implementation of the OpenPGP standard as defined by RFC 2440. GnuPG can encrypt and sign your data and communication, accompanying with a versatile key management system [Koch, 2007].

Before this evaluation, we assumed that the private key and public key of user "test1" had been created successfully. The first step was to calculate the message digest of a 2D im-

age (51.jpg), which is the same as the first step of the evaluation of integrity of digital objects in Section 5.5.1. The message digest was stored in a text file "caesar2_thumbnails_image_a_51.txt". The second step was to sign the file "caesar2_thumbnails_image_a_51.txt" with the private key of user "test1" as depicted in Figure 49.

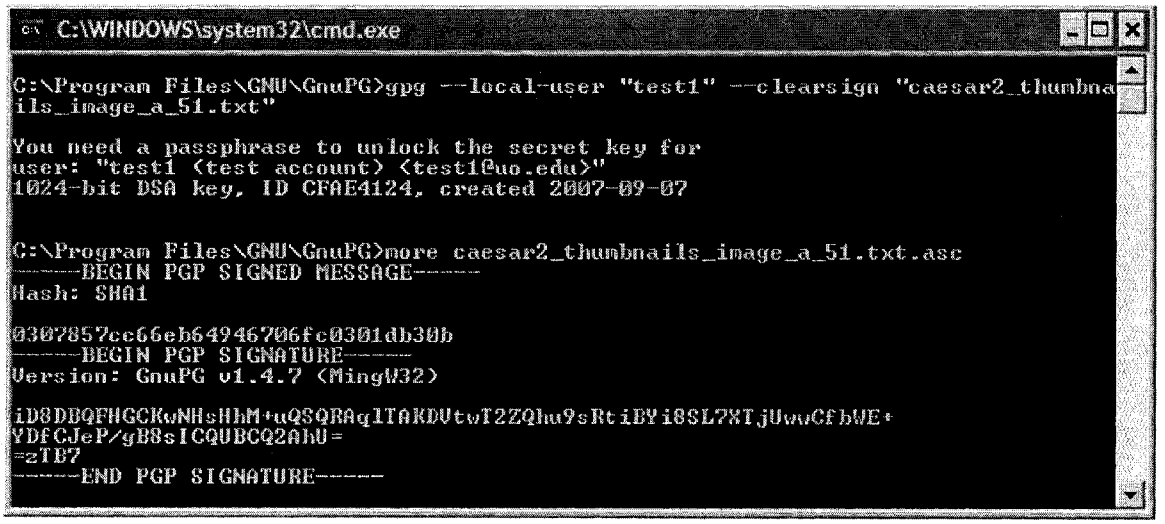


Figure 49 Sign with test1's Private Key

The third step was to combine the content of the "caesar2_thumbnails_image_a_51.txt" and its digital signature together. The fourth step was to input the public key and the digital signature into the Virtual Model as in Figure 50.

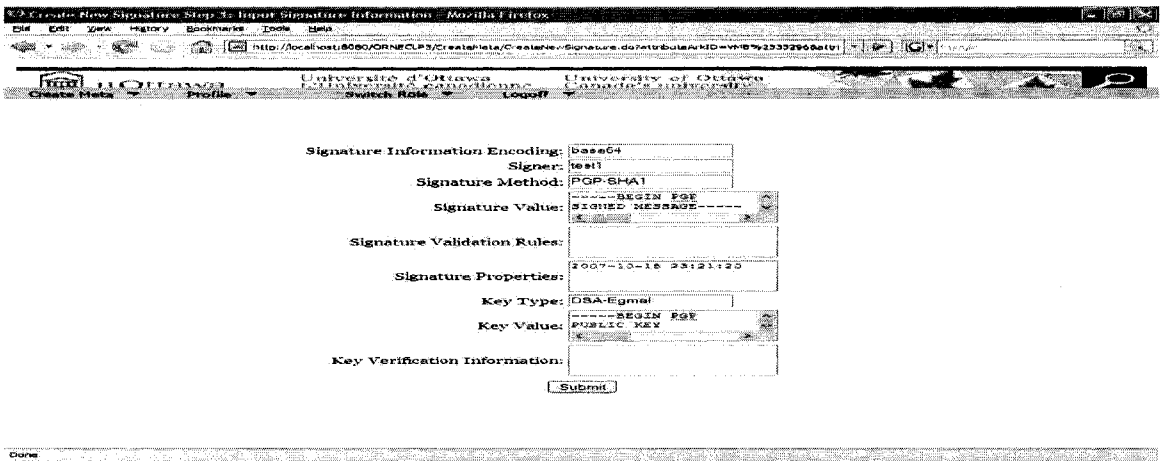


Figure 50 Input Public Key and Digital Signature

The step five was to retrieve the public key and the digital signature as in Figure 51.

Figure 51 Read Public Key and Digital Signature

The sixth step included two sub steps. Firstly, the message digest in the digital digest was needed to compare the digital digest obtained in the fifth step and sixth step of the evaluation of integrity of digital objects. If all of the three message digests were equal, the digital signature could be verified with the public key as in Figure 52. If not, the 2D image data was unauthentic or non-integrated.

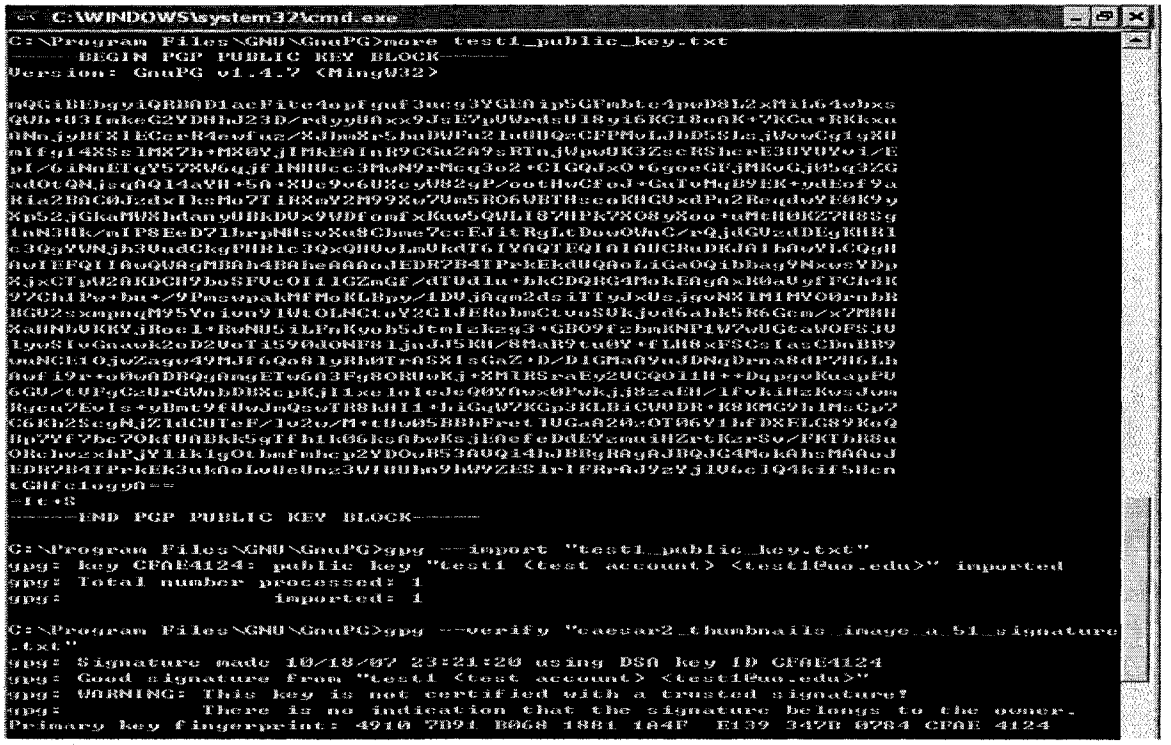


Figure 52 Verify Digital Signature

The results in Figure 52 indicate that the digital signature is a good signature from the user test1, which means that the IDeAL Framework can assure the authenticity of those digital objects.

5.5.3 Evaluation of the Necessary Functionality of the Digital Repository

As recommended in [Nestor, 2006], four aspects are examined to assess the necessary functionality of the digital repository. They are the identification of the digital objects and their relationships, formal description of digital objects content and structure, interpretability of digital objects, and documentations of all changes to digital objects.

Identification of the Digital Objects and Their Relationships

Identification of the digital objects and their relationships is essential for administration of the objects. A kind of standardised persistent identifier (ARK ID) is used to ensure the future interoperability with other digital repositories. A persistent identifier tracks a specific object regardless of its physical location or current ownership. It is similar in function to a Social Insurance Number which is assigned to an individual and does not change when that person's address changes. Likewise, in the long-term data preservation environment, digital objects must at least have one identifier, unique, persistent, and independent of specific digital repositories. The Archival Resource Key (ARK) identifier is a naming scheme of persistent identifier, emphasizing on persistent access to digital objects; so it is adopted into the IDeAL Framework. Figure 53 shows one object metadata in the IDeAL Framework, including its identifier (objectID), ARK ID (ArkID), its relationships, its attributes, amongst others. This figure indicates the IDeAL Framework can identify the digital objects and their relationships, since the 'Object ID', 'Object ARK ID', 'The Object Relationship List', and 'The Attribute List' in the following figure depict such information.

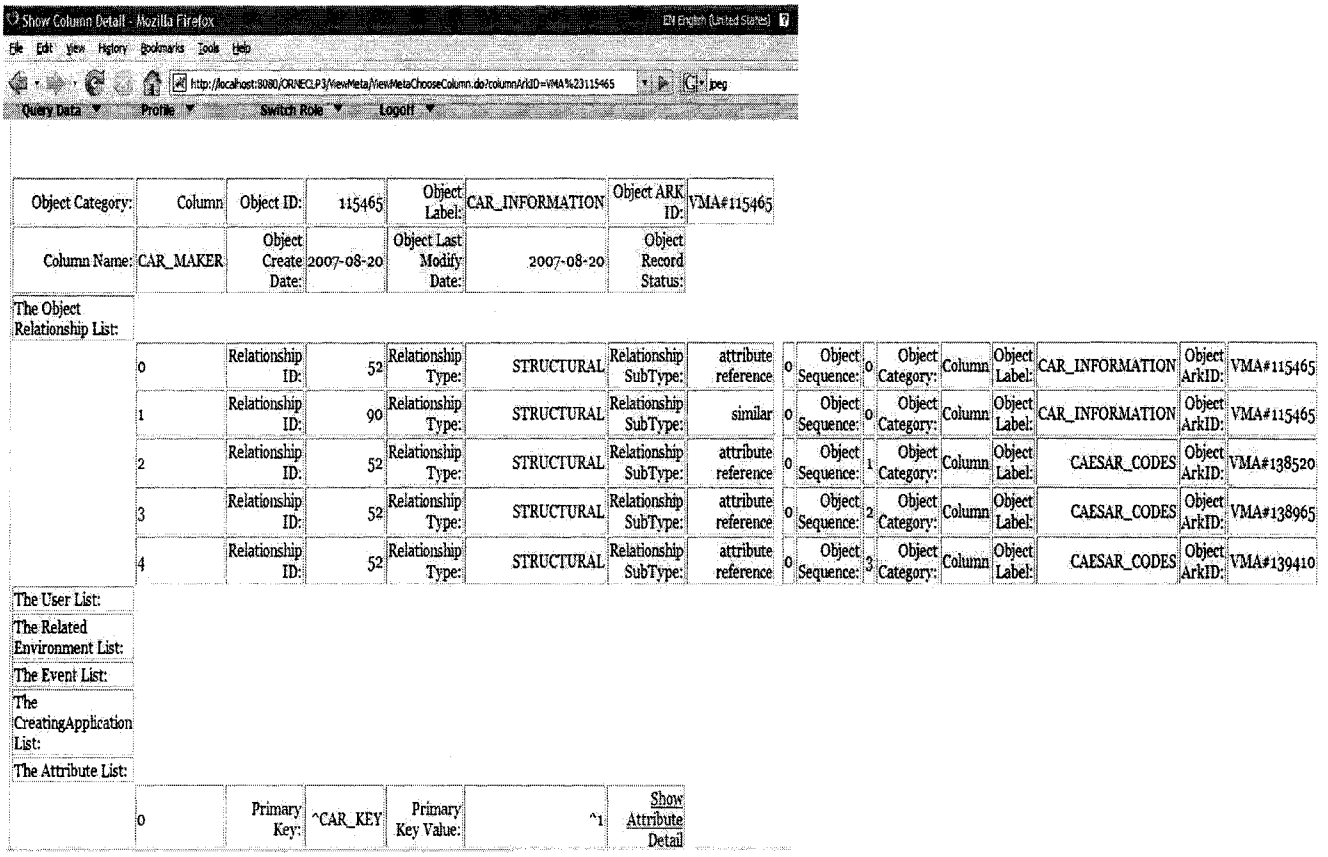


Figure 53 Metadata of a Column Object

Formal Description of Digital Objects Content and Structure

In addition to the identification of digital objects, their structures are also important. As described in Section 3.3.1, the data in the digital repositories of the IDeAL Framework are divided into five categories, which are Data Collection (Information Collection), Data

Set (Information Package), Table (Information Unit), Column and Attribute. The following five figures show description of these five kinds of data and the structure among them.

Figure 54 shows Information Collection 'CAESAR', which contains two Information Packages: 'CAESAR1' and 'CAESAR2'. The following web page also includes the features of Information Collection 'CAESAR', such as the 'Object Label', 'Object Create Date', amongst others.

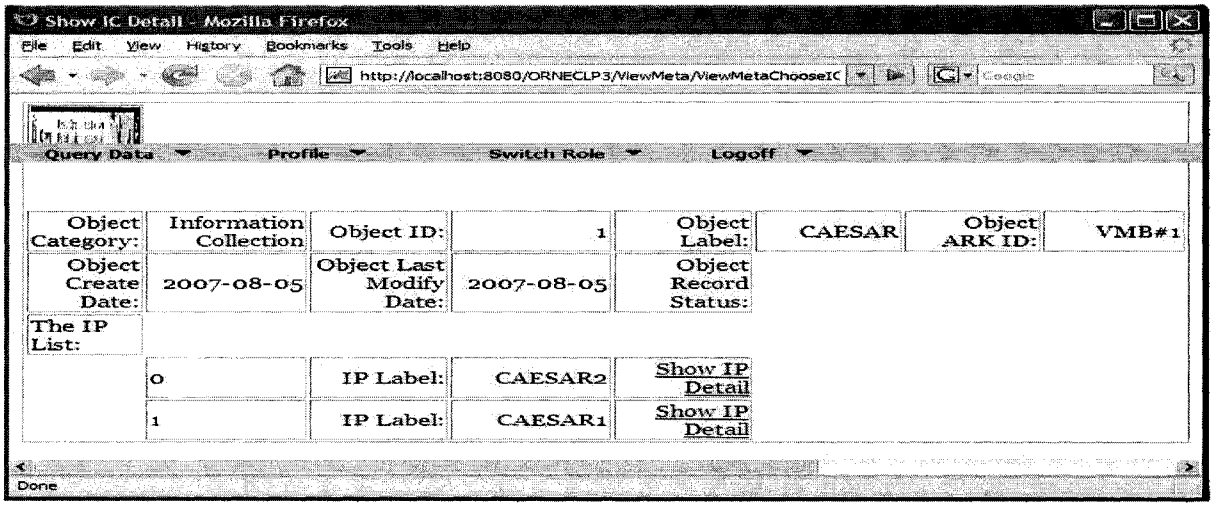


Figure 54 Information Collection

Figure 55 presents the details of Information Package 'CAESAR2'. This Information Package includes twelve Information Units (Tables). Moreover, the following web page contains the features of the Information Package 'CAESAR2', such as 'Object Label', 'Object Create Date', 'The Related Environment List', amongst others.

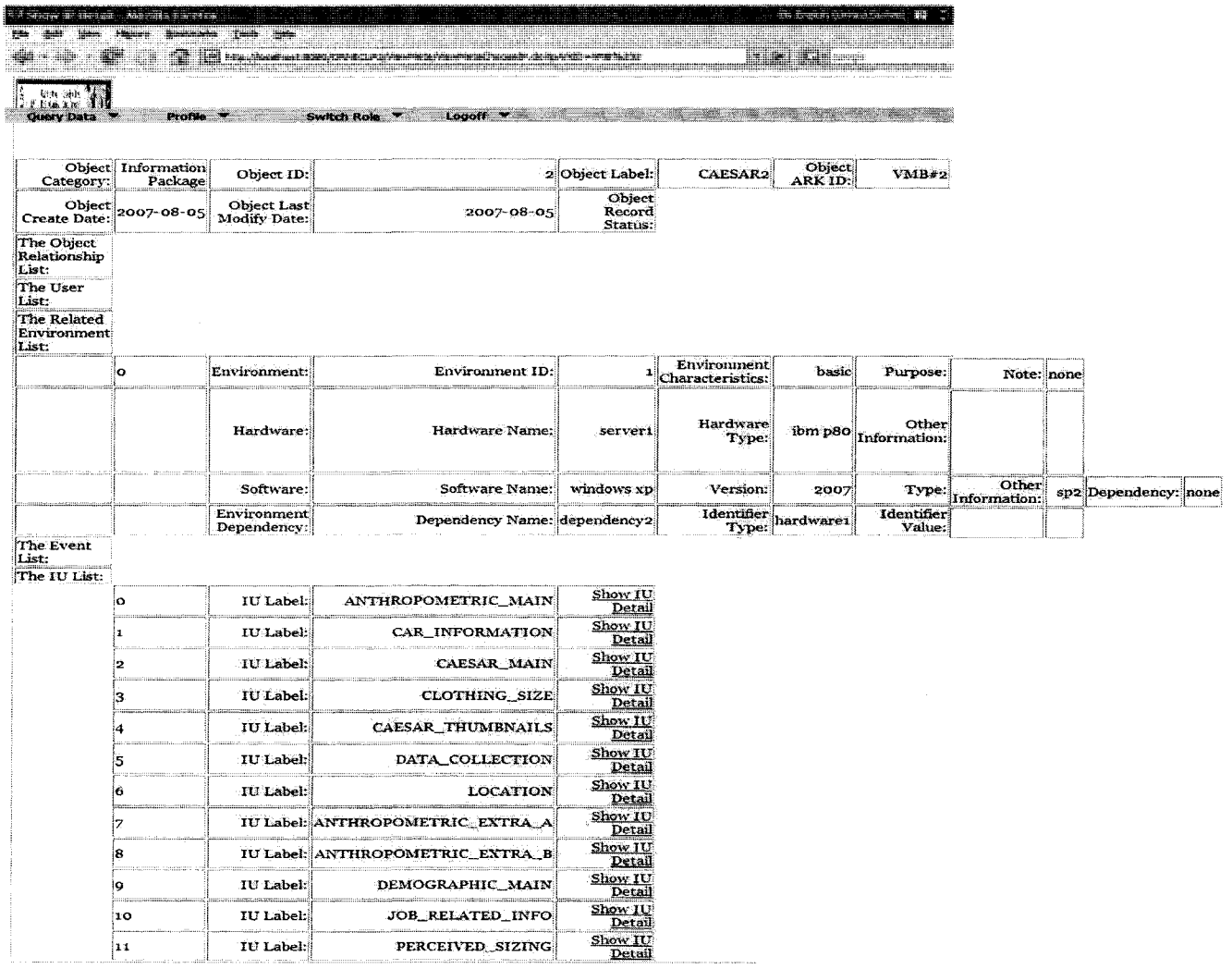


Figure 55 Information Package

Figure 56 presents the details of Information Unit 'CAESAR_THUMBNAILS'. This Information Unit includes several columns. Further, the following web page contains the features of the Information Unit 'CAESAR_THUMBNAILS', such as 'Object Label', 'Object Create Date', 'The Event List', amongst others.

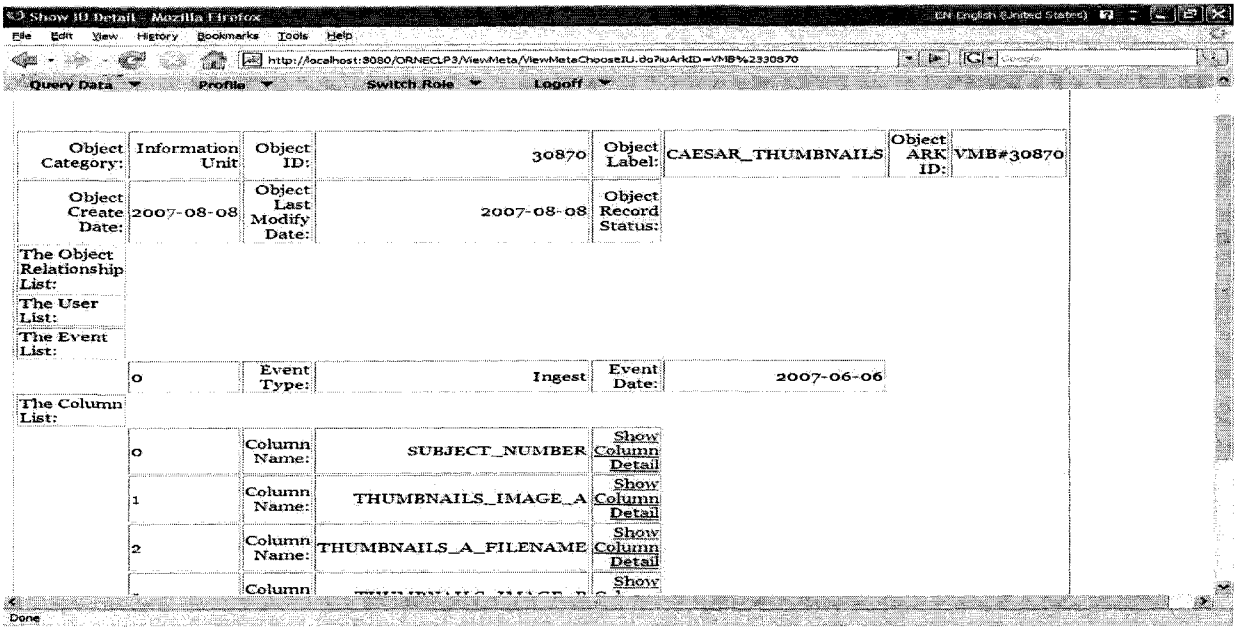


Figure 56 Information Unit

Figure 57 presents the details of column 'THUMBNAILS_IMAGE_A'. This column includes several attributes. Moreover, the following web page involves the features of the Column 'THUMBNAILS_IMAGE_A', such as 'Object Label', 'Object Create Date', 'The Object Relationship List', 'The Related Environment List' amongst others.

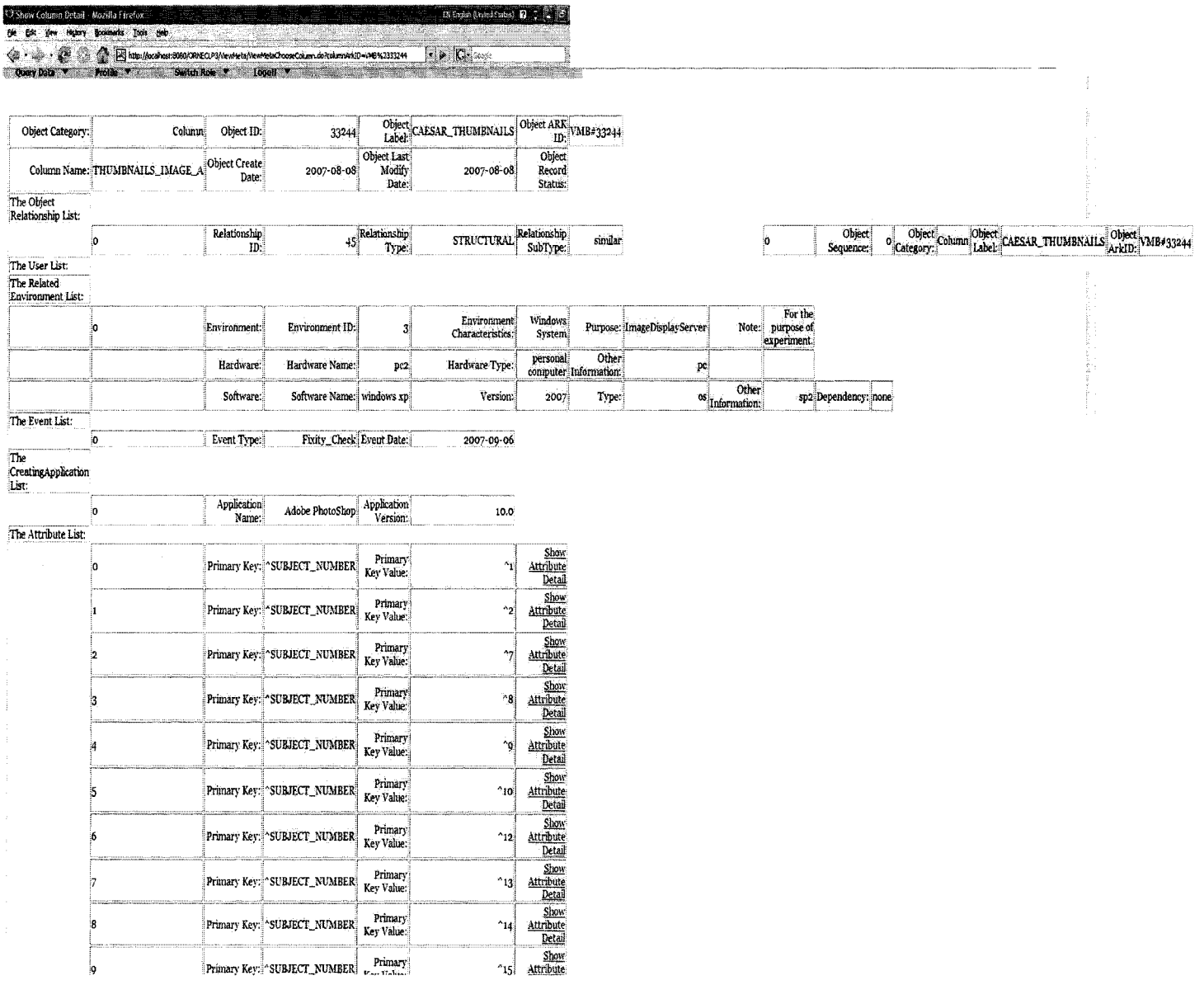
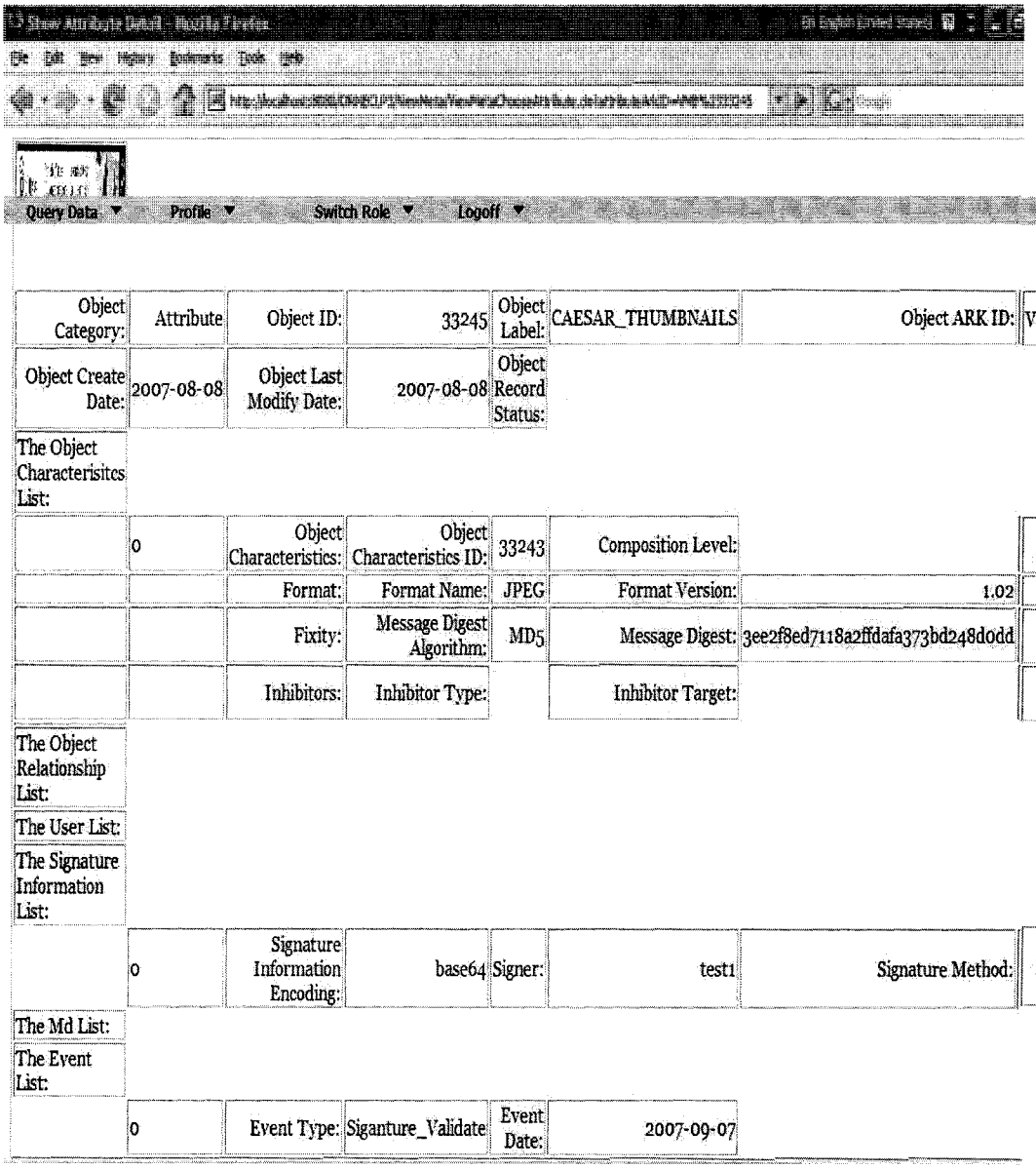


Figure 57 Column

Figure 58 shows the details of an attribute, such as 'Object Category', 'Object Create Date', 'The Event List', 'The Signature Information List' amongst others. Further, the web page contains some object characteristics, including object format, fixity information, amongst others.



Query Data ▼ Profile ▼ Switch Role ▼ Logoff ▼

Object Category:	Attribute	Object ID:	33245	Object Label:	CAESAR_THUMBNAILS	Object ARK ID:	VMB#33245		
Object Create Date:	2007-08-08	Object Last Modify Date:	2007-08-08	Object Record Status:					
The Object Characterisitcs List:									
	0	Object Characteristics:	Object Characteristics ID:	33243	Composition Level:		Significant Properties:	null	
		Format:	Format Name:	JPEG	Format Version:	1.02			
		Fixity:	Message Digest Algorithm:	MD5	Message Digest:	3ee2f8ed7118a2ffdafa373bd248d0dd			
		Inhibitors:	Inhibitor Type:		Inhibitor Target:		Inhibitor Key:		
The Object Relationship List:									
The User List:									
The Signature Information List:									
	0	Signature Information Encoding:	base64	Signer:	test1	Signature Method:	PGP-SHA1	Key Type:	DSA-Egmal
The Md List:									
The Event List:									
	0	Event Type:	Siganture_Validate	Event Date:	2007-09-07				

Figure 58 Attribute

Interpretability of Digital Objects

Since the preservation strategy of the IDeAL Framework is emulation, the digital object itself, its original rendering software, its related DBMS, its related operating system and software environment required by that software are all saved. If the metadata about the original computing environment and format are reserved correctly, the original interpreting environment can be established and the digital object can be interpreted successfully. The following experiments show how to get the required metadata for interpreting a 2D image. The computing environment of the Information Package, the computing environment of the Column, and the format of the 2D data are retrieved. In the Virtual Model, Information Packages map to databases. Therefore, the computing environment of Information Package is that of the database. In the following figure, the web page shows the hardware and the software information. Based on the information in the web page, we can obtain the relevant information that the Information Package CAESAR2 resides in the software (windows xp sp2 Version 2007) on top of the hardware (ibm p80).

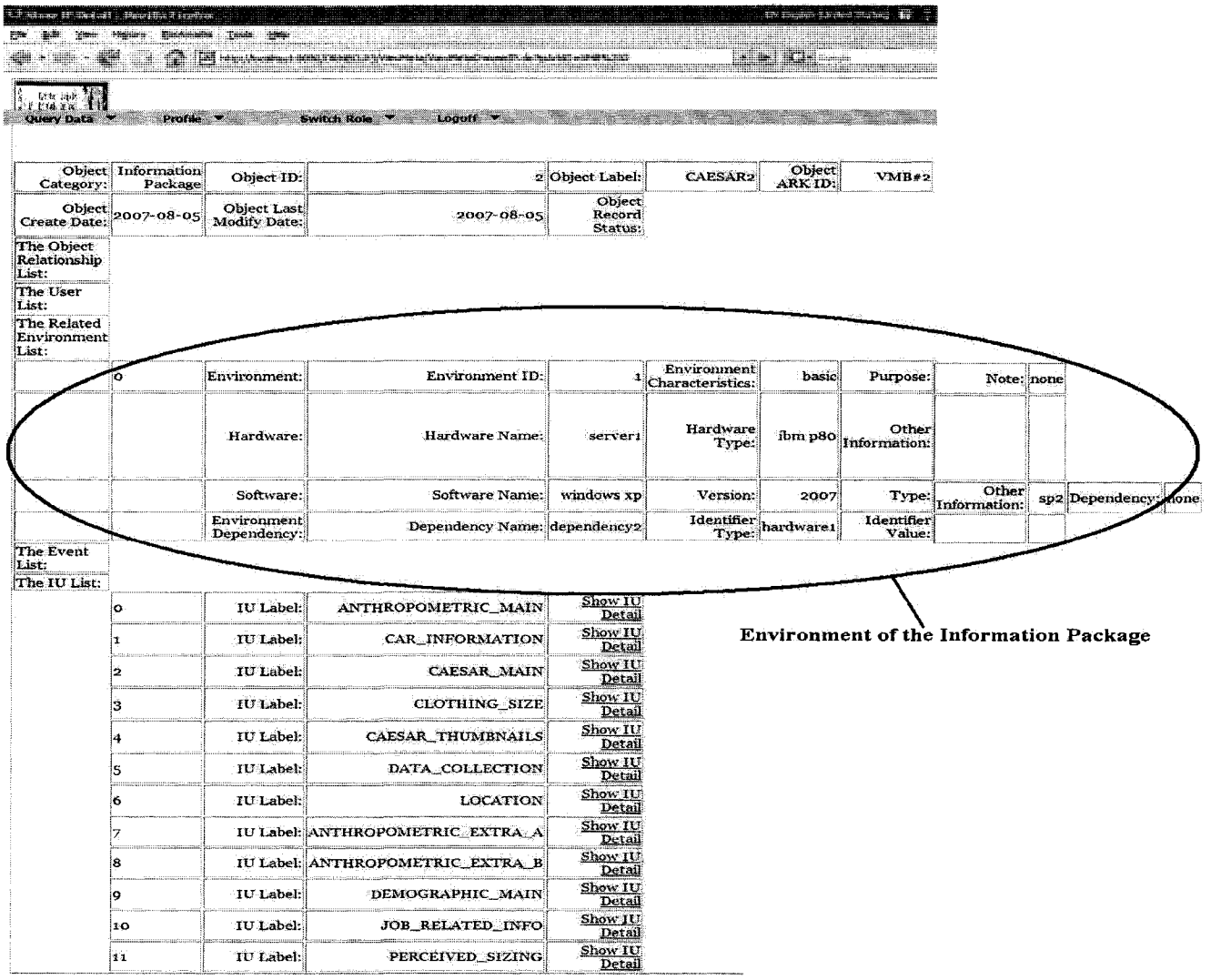


Figure 59 Environment of the Information Package

The computing environment of the 2D image column THUMBNAILS_IMAGE_A can be obtained from the web page in Figure 60. Based on the information in the web page, we can know that the 2D image can be viewed with software Microsoft Paint Version 5.1 on top of a personal computer.

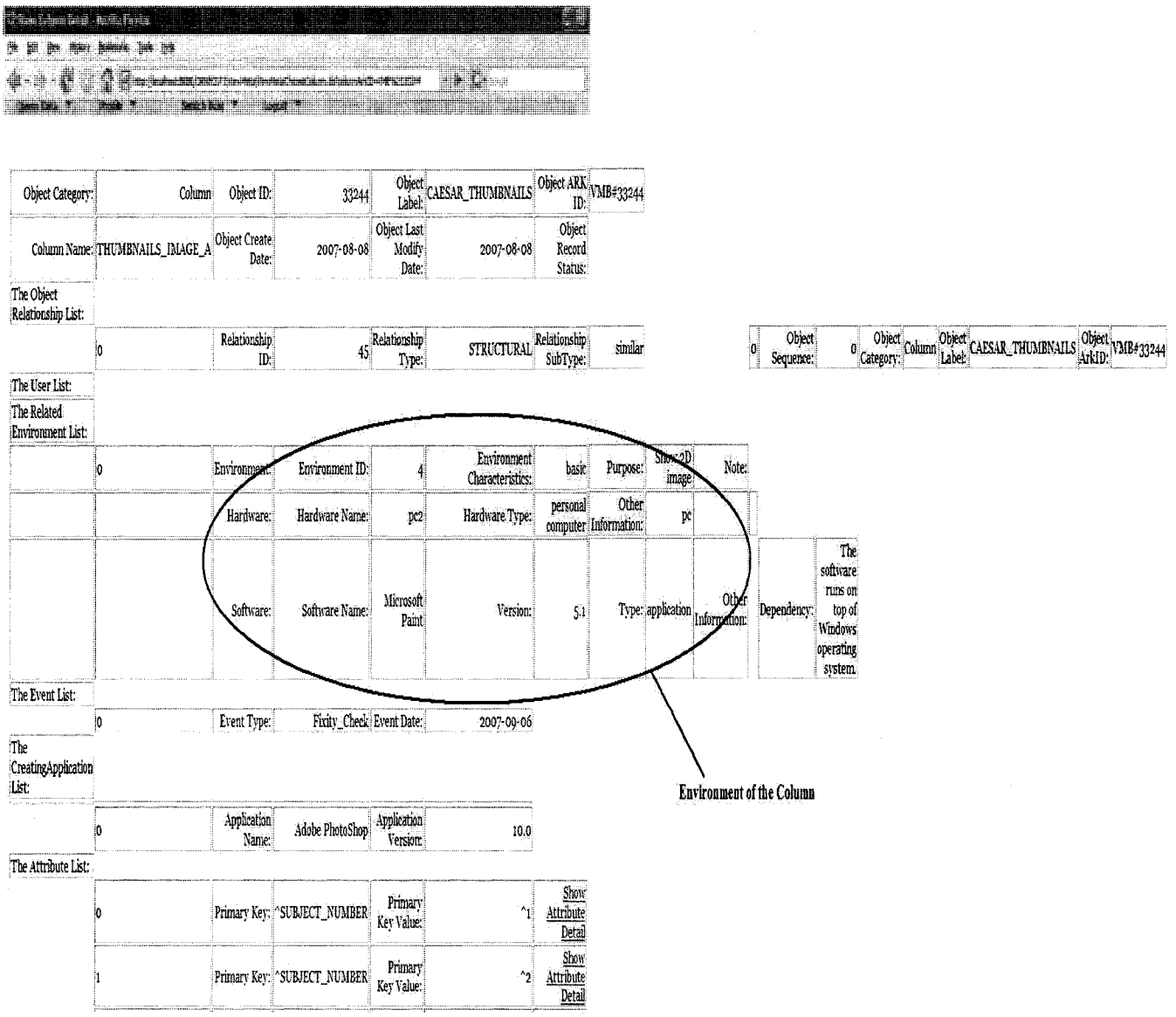
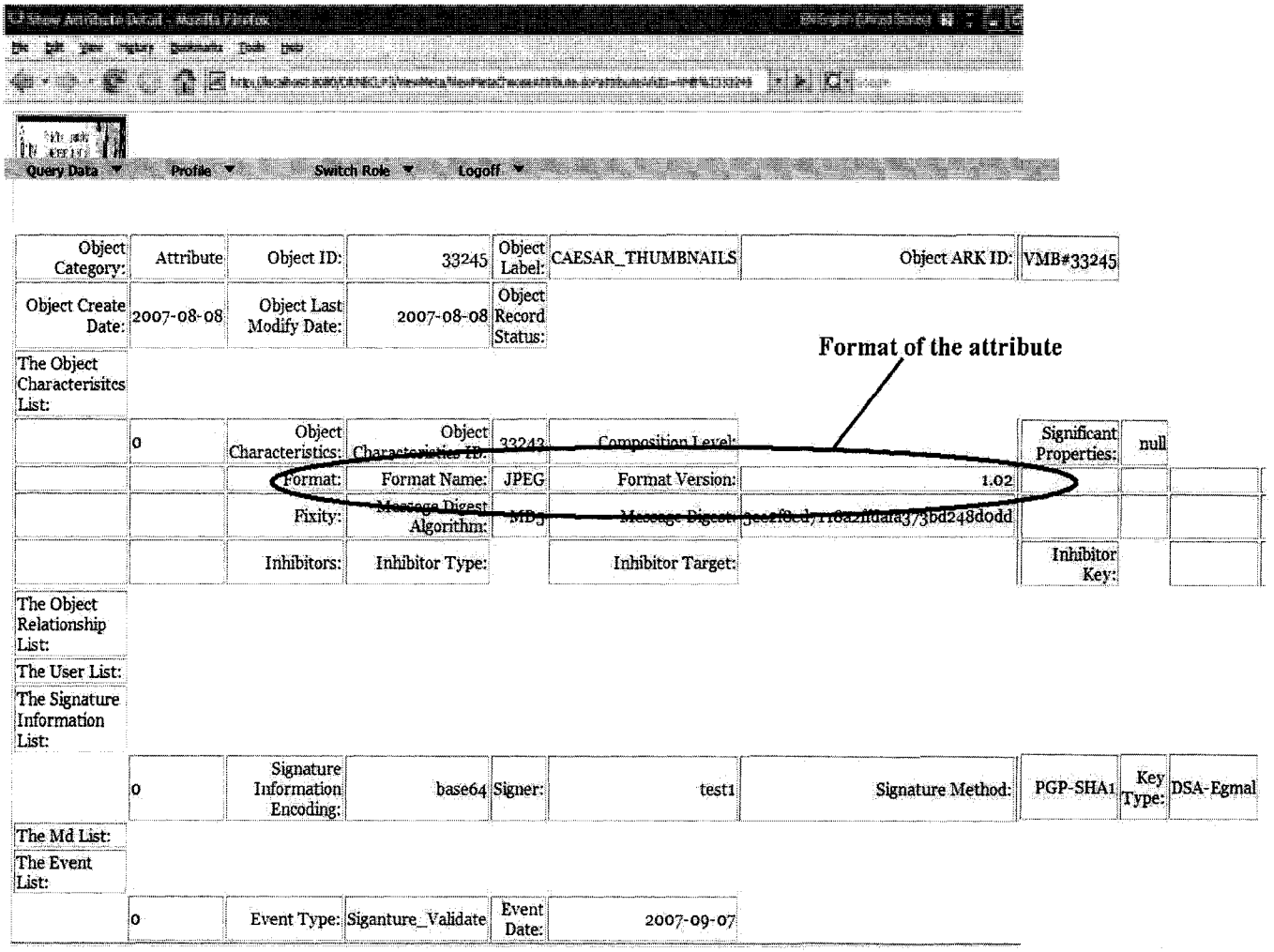


Figure 60 Environment of the Column

Furthermore, the format information of the 2D image can be obtained from the web page in Figure 61, which is JPEG Version1.02 in this example. Not only the information can be used to help the end user understand the 2D image, but also it can be used to aid the end user in choosing software tools to interpret the 2D image.



Object Category:	Attribute	Object ID:	33245	Object Label:	CAESAR_THUMBNAILS	Object ARK ID:	VMB#33245		
Object Create Date:	2007-08-08	Object Last Modify Date:	2007-08-08	Object Record Status:					
The Object Characterisites List:									
	0	Object Characteristics:	Object Characteristics ID:	33243	Composition Level:		Significant Properties:	null	
		Format:	Format Name:	JPEG	Format Version:	1.02			
		Fixity:	Message Digest Algorithm:	MD5	Message Digest:	3ee2f8cd7116a2ffdafa373bd248d0dd			
		Inhibitors:	Inhibitor Type:		Inhibitor Target:		Inhibitor Key:		
The Object Relationship List:									
The User List:									
The Signature Information List:									
	0	Signature Information Encoding:	base64	Signer:	test1	Signature Method:	PGP-SHA1	Key Type:	DSA-Egmal
The Md List:									
The Event List:									
	0	Event Type:	Siganture_Validate	Event Date:	2007-09-07				

Figure 61 Format of the Attribute

Documentations of All Changes to Digital Objects

The documentation of all changes to digital is necessary to ensure technical preservation of the digital objects, to assure the authenticity of the data and to trace the evolution of the digital repositories. In the IDeAL Framework, events are used to keep all the changes. The events can be many kinds, such as ingest, fixity check, digital signature validation, amongst others. These events can be recorded in the Virtual Model. Not only the event related metadata includes the characteristics of events, but also it involves the related users, related objects, and related relationships. Figure 62 shows a list of events in the IDeAL Framework, such as 'Ingest', 'Fixity_Check', and 'Signature_Validate'.

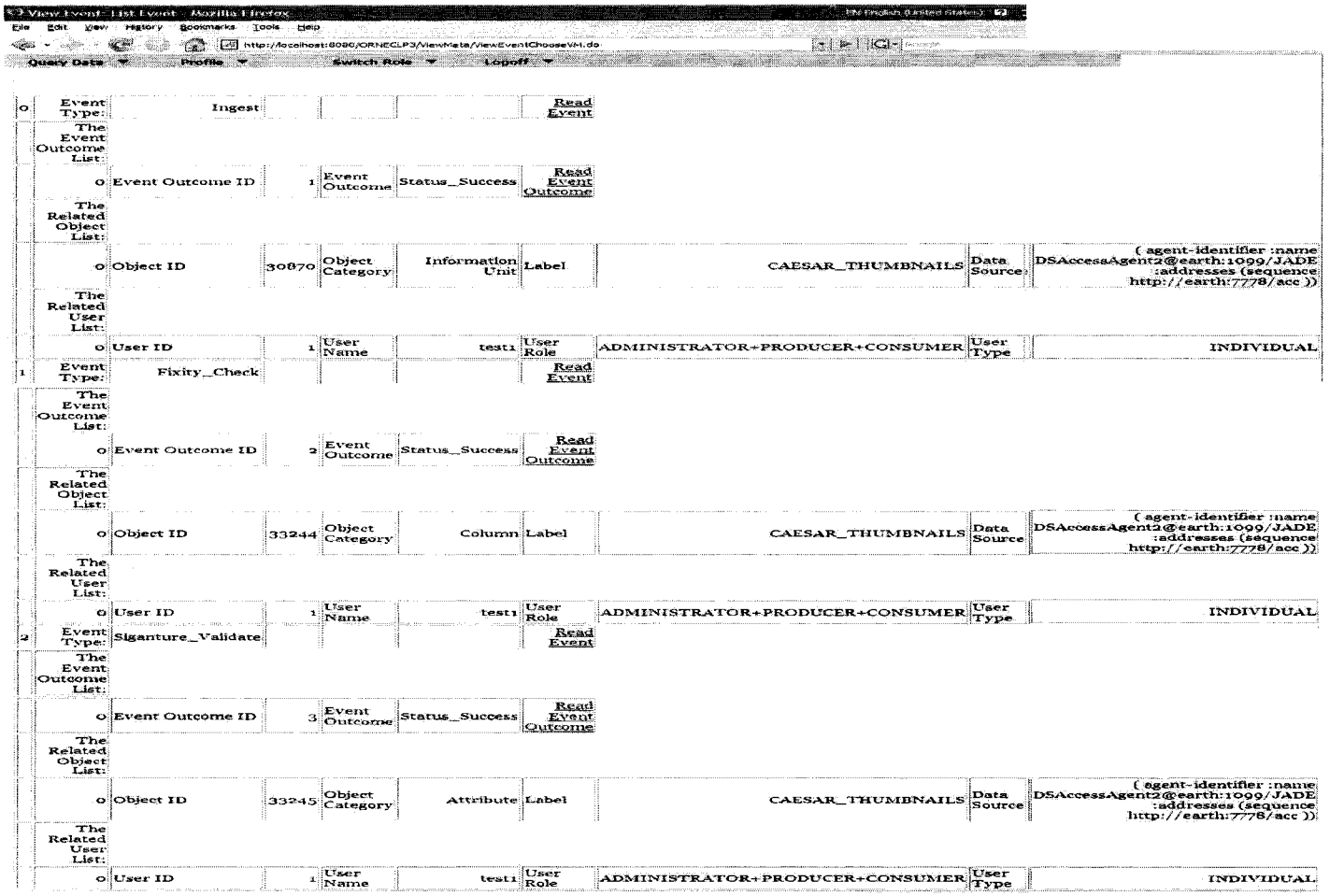


Figure 62 Read Events

5.6. Evaluation of the Function of the SRB Agent

Recall that the SRB (Storage Resource Broker) Agent acts as the bridge between the Business Logic Process System and the Multi-agent System. It presents a simple mechanism to access data distributed across multiple digital repositories. In Chapter 3, the SRB Agent Algorithm is described, and its implementation is detailed in Chapter 4. As indicated in Section 3.4.2, two kinds of SRB Agents in the Multi-agent System are similar. The experimental result of the Metadata Related SRB is shown in this thesis although both of the SRB Agents were tested.

5.6.1 Experimental Results of Metadata Related SRB Algorithm

The first step of the algorithm was to initiate a SRB Agent as shown in Figure 63.

```
enter VMBrokerGateway
************ init Broker AGENT    **************
Sep 12, 2007 9:28:09 AM jade.core.Runtime beginContainer
INFO: ----------------------------------
    This is JADE 3.5 - revision 5988 of 2007/06/21 11:02:30
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.tilab.com/
----------------------------------------
Sep 12, 2007 9:28:09 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Sep 12, 2007 9:28:09 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Sep 12, 2007 9:28:09 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Sep 12, 2007 9:28:09 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Sep 12, 2007 9:28:09 AM jade.core.messaging.MessagingService clearCachedSlice
INFO: Clearing cache
Sep 12, 2007 9:28:09 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Container-134@earth is ready.
--------------------------------------------
```

Figure 63 SRB Agent Initialize

The second step was that the SRB Agent searched for suitable VM Access Agent from the DF Agent if needed. Figure 64 shows that the SRB Agent sent the search request to the DF Agent and the DF Agent responded with suitable Access Agents in its reply message.

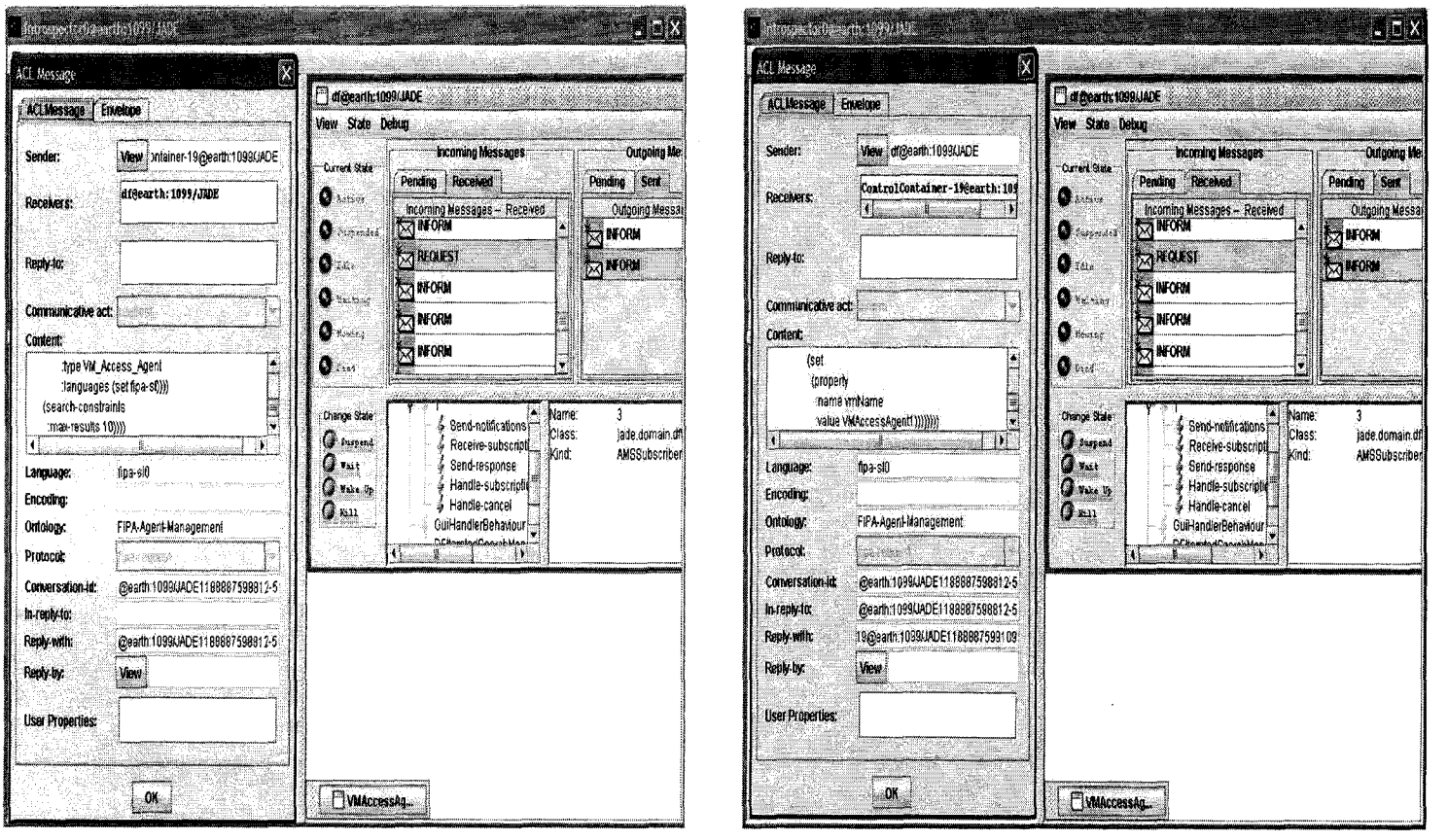
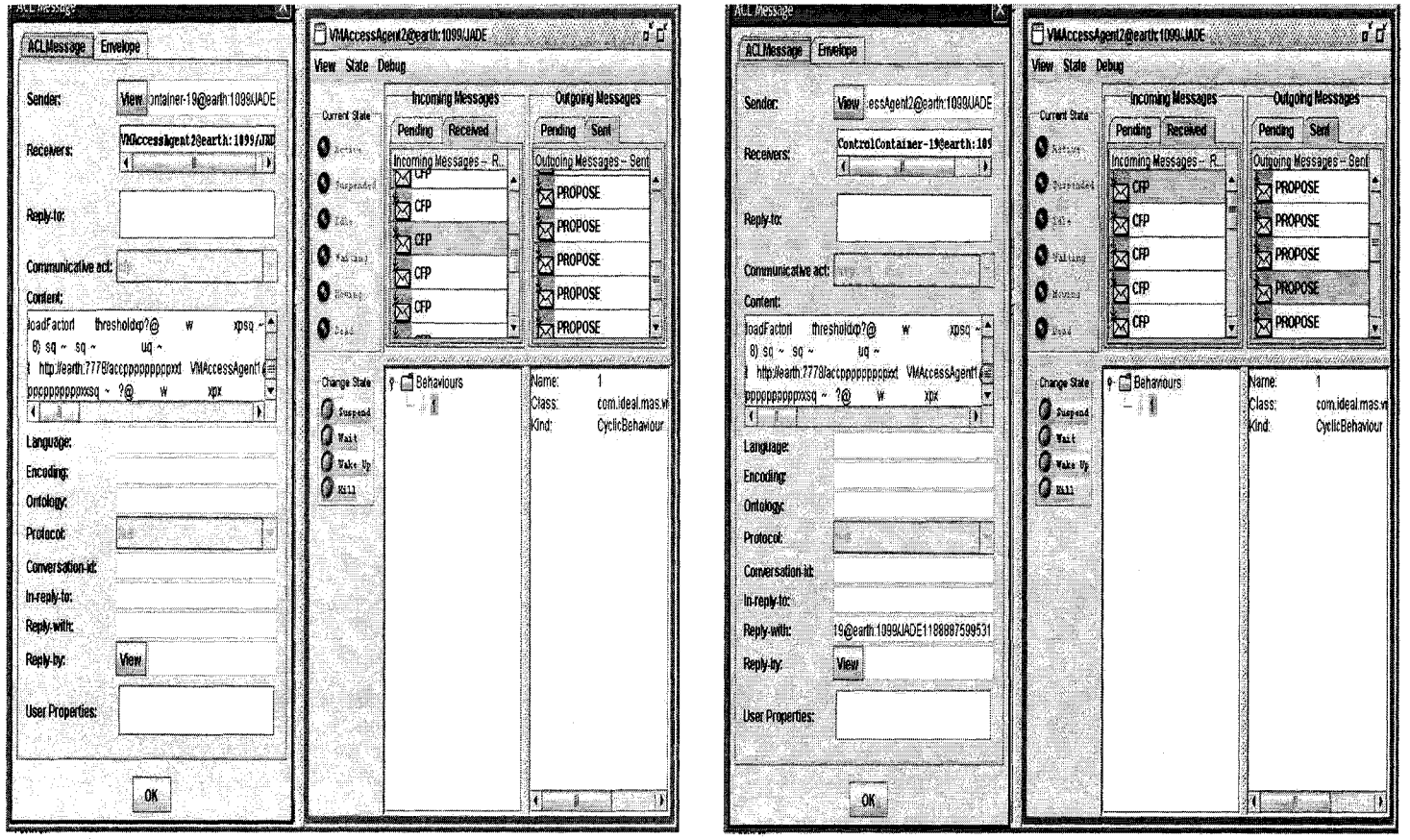


Figure 64 SRB Agent Interacted with DF Agent

The third step was that the SRB Agent sent the database request to the identified Access Agent and the identified VM Access Agent responded with results in its reply message. Figure 65 shows the message exchange between the SRB Agent and the VM Access Agent.



SRB Agent Sends Request ACLMessage to Access Agent

Access Agent Send Response ACLMessage to SRB Agent

Figure 65 SRB Agent Interacted with Access Agent

Further, the following Figure 66 gives out an outline of the message exchanges among the Multi-agent System.

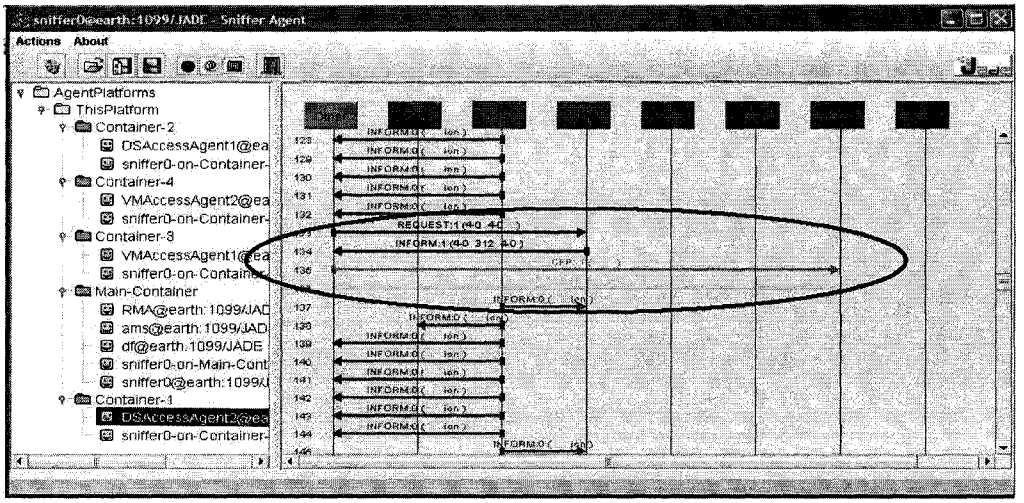


Figure 66 Message Exchanges in the MAS

The fifth step was that the broker agent returns the result to the Business Logic Process System. The result was further forwarded to the IDeAL Framework Portal and shown as Figure 67.

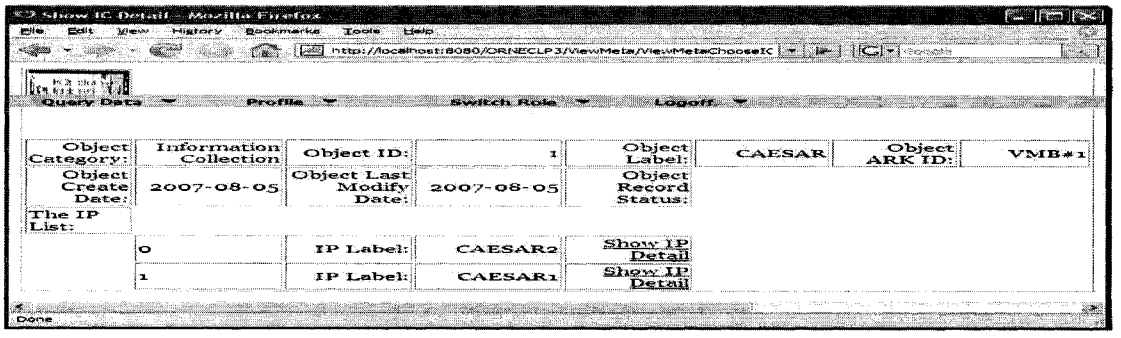


Figure 67 Result Displayed by the IDeAL Framework Portal

The above experimental results indicate that the SRB Agent can function properly and correctly.

5.7. Evaluation of the Scalability of the MAS

MAS provide an ideal mechanism for implementing heterogeneous and complex distributed systems. The agent technology is well suited for applications that are based on communication of loosely-coupled systems. In the current configuration, except the JADE system agents, the MAS have two VM Access Agents, two Data Access Agents, and various Broker Agents.

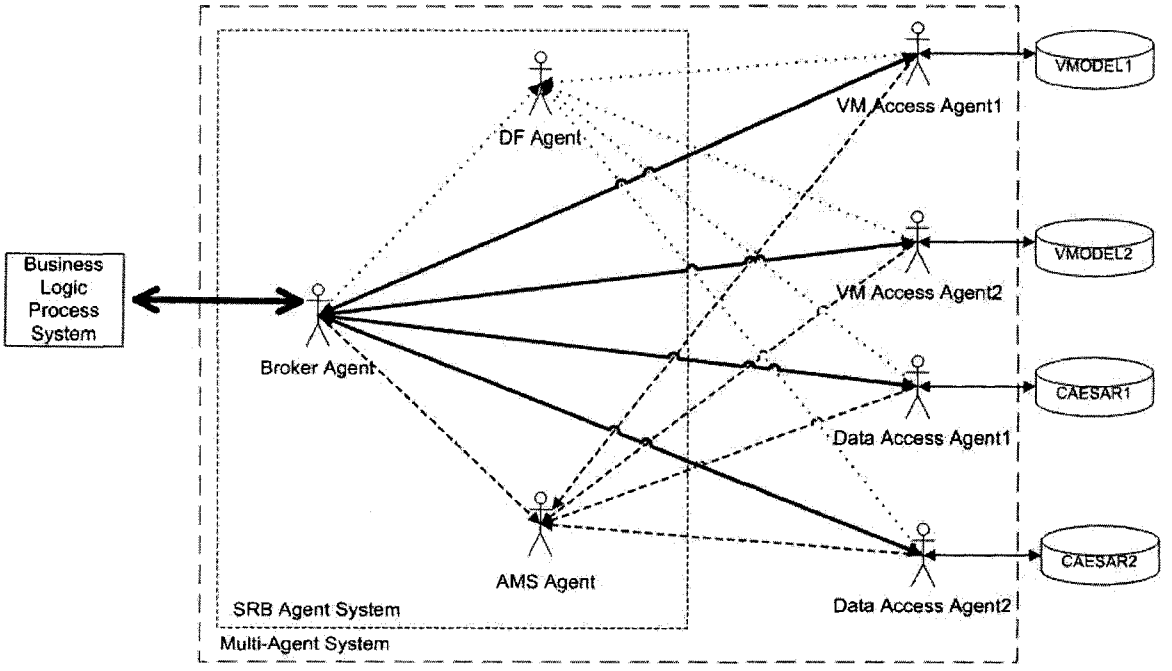


Figure 68 The MAS Implementation

As depicted in Figure 68, the experimental environment contains two data sets: CAESAR1 and CAESAR2, which reside in two different databases. The metadata of CAESAR1 is stored in database VMODEL1 and the metadata of CAESAR2 is in VMODEL2. All the four databases have their corresponding static access agents.

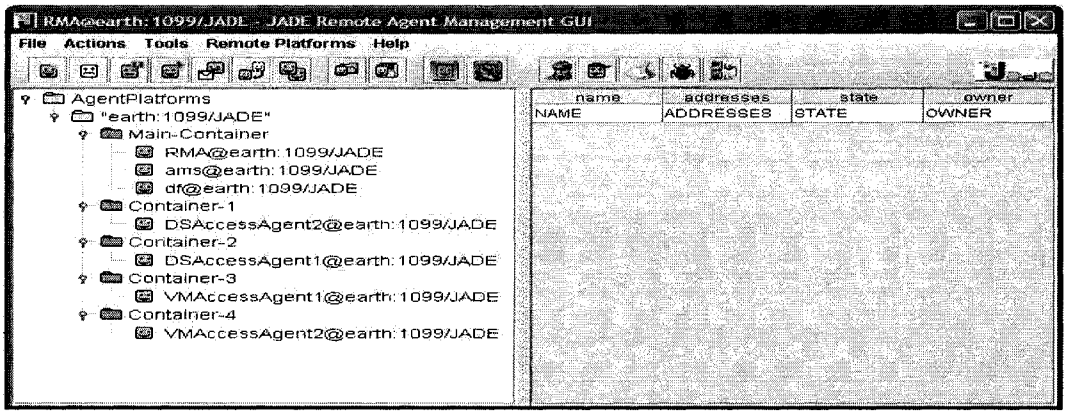
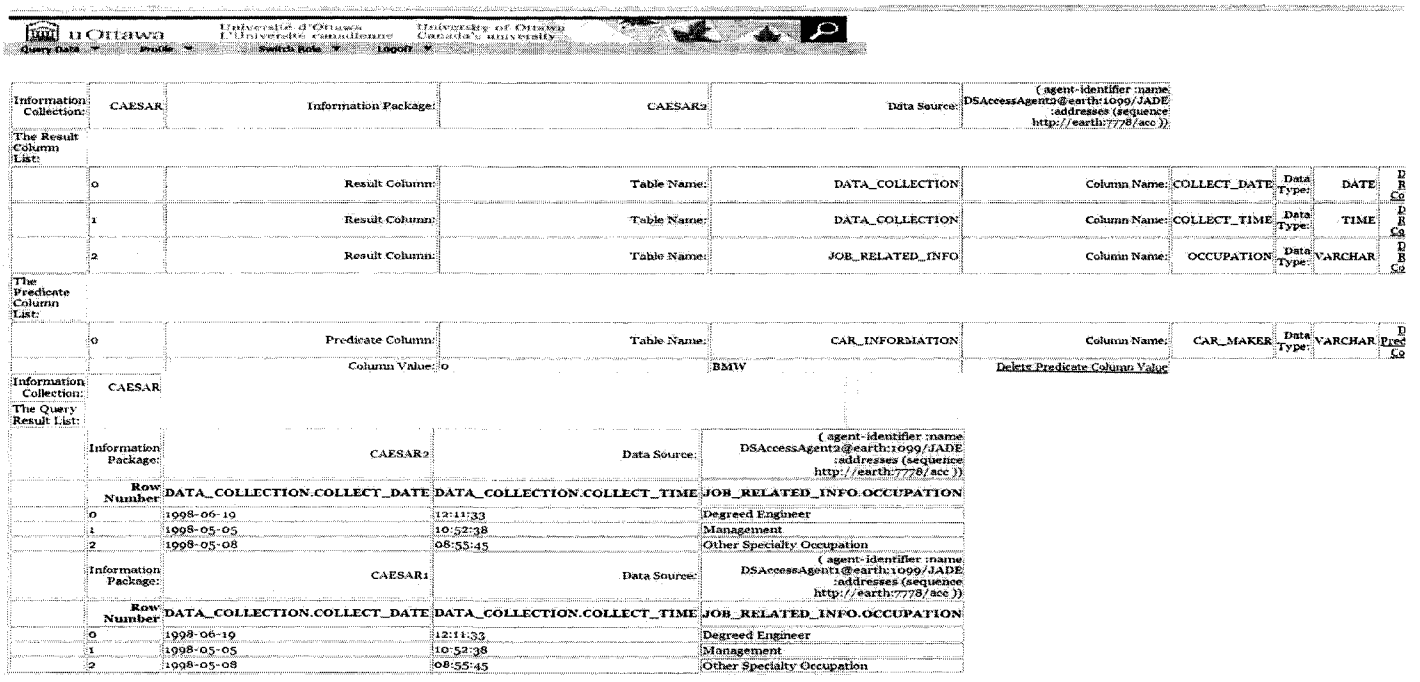


Figure 69 Agents Status in RMA

Figure 69 shows the status of the JADE system agents and the four static access agents. There is no status of the Broker Agent, because the status of the Broker Agent is shown dynamically. The success of the evaluation of the general metrics indicates that the MAS can handle the requests from the Logic Process System efficiently and effectively. The scalability and performance of the JADE framework have been discussed in [Chmiel et al., 2005; Cortese, & Quarta, 2002; Rahmi, Bjursell, Paprzycki, Cobb, & Ali, 2006; and Chmiel, Tomiak, Gawinecki, & Karczmarek, 2004]. Its scalability was acknowledged. The Chmiel [2005] concluded that the JADE framework could contain thousands of agents and tens of thousands of ACL messages if given enough hardware. Further, the architectture of the IdeAL Framework determines that handling the access to one data set only needs one Data Access Agent, and that the access to a Virtual Model needs one VM Access Agent; therefore, this suggests that the MAS can support the IdeAL Framework to possibly involve thousands of data sets.

5.8. Evaluation of Data Integration

The data integration function of the IdeAL Framework combines data from multiple data sets into a coherent data representation. Currently, the framework shows the integrated data from different data sets instead of combining them as a whole. Thus, the issues such as the redundancy of data, the detection and resolution of data value conflicts are not included in this implementation of the IdeAL Framework. The experimental results include three aspects pertaining to data integration: (1) present various data types already in the IdeAL Framework (2D, date, time, integer, decimal, and varchar); (2) show the difference of two data sets to indicate the existence of data evolution; and (3) depict the situation when one user-defined column only exists in part of all the similar data sets. Figure 70 shows data in both CAESAR1 and CAESAR2 with same query criteria and the data types are date, time, and varchar. The query results are same from CAESAR1 and CAESAR2.

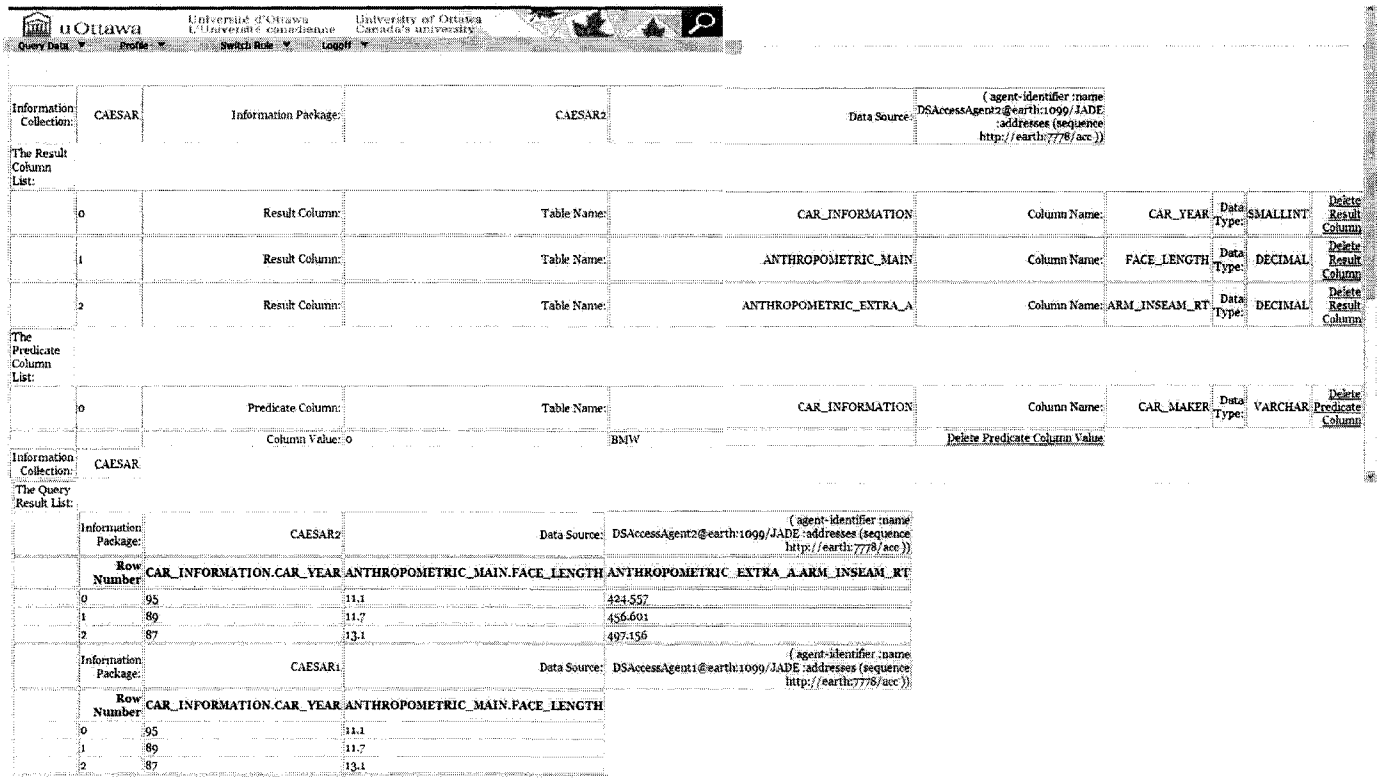


Information Collection:	CAESAR	Information Package:	CAESAR2	Data Source:	(agent-identifier :name DSAccessAgent2@earth:1099/JADE :addresses (sequence http://earth:7778/acc))					
The Result Column List:										
	0	Result Column:		Table Name:	DATA_COLLECTION	Column Name:	COLLECT_DATE	Data Type:	DATE	D R Co
	1	Result Column:		Table Name:	DATA_COLLECTION	Column Name:	COLLECT_TIME	Data Type:	TIME	D R Co
	2	Result Column:		Table Name:	JOB_RELATED_INFO	Column Name:	OCCUPATION	Data Type:	VARCHAR	D R Co
The Predicate Column List:										
	0	Predicate Column:		Table Name:	CAR_INFORMATION	Column Name:	CAR_MAKER	Data Type:	VARCHAR	D Pred Co
		Column Value:	0		BMW	Delete Predicate Column Value				

Information Collection:	CAESAR			
The Query Result List:				
	Information Package:	CAESAR2	Data Source:	(agent-identifier :name DSAccessAgent2@earth:1099/JADE :addresses (sequence http://earth:7778/acc))
	Row Number	DATA_COLLECTION.COLLECT_DATE	DATA_COLLECTION.COLLECT_TIME	JOB_RELATED_INFO.OCCUPATION
	0	1998-06-19	12:11:33	Degreed Engineer
	1	1998-05-05	10:52:38	Management
	2	1998-05-08	08:55:45	Other Specialty Occupation
	Information Package:	CAESAR1	Data Source:	(agent-identifier :name DSAccessAgent1@earth:1099/JADE :addresses (sequence http://earth:7778/acc))
	Row Number	DATA_COLLECTION.COLLECT_DATE	DATA_COLLECTION.COLLECT_TIME	JOB_RELATED_INFO.OCCUPATION
	0	1998-06-19	12:11:33	Degreed Engineer
	1	1998-05-05	10:52:38	Management
	2	1998-05-08	08:55:45	Other Specialty Occupation

Figure 70 Data of Type Date, Time and Varchar

Figure 71 shows data in both CAESAR1 and CAESAR2 with the same query situation. The data types are integer, and decimal. Parts of query results are same from CAESAR1 and CAESAR2. Since there is no column 'ARM_INSEAM_RT' or its similar columns in CAESAR1, no result is shown for this column in CAESAR1.



uOttawa Université d'Ottawa / L'Université canadienne / University of Ottawa / Canada's university

Query Data | Profile | Switch Role | Logoff

Information Collection:	CAESAR	Information Package:	CAESAR2	Data Source:	(agent-identifier :name DSAccessAgent2@earth:1099/JADE :addresses (sequence http://earth:7778/acc))					
The Result Column List:										
	0	Result Column:	Table Name:	CAR_INFORMATION	Column Name:	CAR_YEAR	Data Type:	SMALLINT	Delete Result Column	
	1	Result Column:	Table Name:	ANTHROPOMETRIC_MAIN	Column Name:	FACE_LENGTH	Data Type:	DECIMAL	Delete Result Column	
	2	Result Column:	Table Name:	ANTHROPOMETRIC_EXTRA_A	Column Name:	ARM_INSEAM_RT	Data Type:	DECIMAL	Delete Result Column	
The Predicate Column List:										
	0	Predicate Column:	Table Name:	CAR_INFORMATION	Column Name:	CAR_MAKER	Data Type:	VARCHAR	Delete Predicate Column	
		Column Value: 0		BMW	Delete Predicate Column Value					

Information Collection:	CAESAR				
The Query Result List:					
	Information Package:	CAESAR2	Data Source:	(agent-identifier :name DSAccessAgent2@earth:1099/JADE :addresses (sequence http://earth:7778/acc))	
	Row Number	CAR_INFORMATION.CAR_YEAR	ANTHROPOMETRIC_MAIN.FACE_LENGTH	ANTHROPOMETRIC_EXTRA_A.ARM_INSEAM_RT	
	0	95	11.1	424.557	
	1	89	11.7	456.601	
	2	87	13.1	497.156	
	Information Package:	CAESAR1	Data Source:	(agent-identifier :name DSAccessAgent1@earth:1099/JADE :addresses (sequence http://earth:7778/acc))	
	Row Number	CAR_INFORMATION.CAR_YEAR	ANTHROPOMETRIC_MAIN.FACE_LENGTH		
	0	95	11.1		
	1	89	11.7		
	2	87	13.1		

Figure 71 Data of Type Integer and Decimal

Figure 72 presents 2D data in both CAESAR1 and CAESAR2 with the same query criteria. Although the column 'THUMBNAILS_IMAGE_A' in CAESAR1 and the column 'THUMBNAILS_IMAGE_A' in CAESAR2 were similar columns and they were shown based on the same query situation, apparently their content were different. However, they were not totally unlike. We could recognize that the 2D images from CAESAR1 and CAESAR2 were of the same person. Thus, our system was able to retrieve the image of the same person from two different data sources. This example also illustrates the data evolution within our IdeAL Framework.

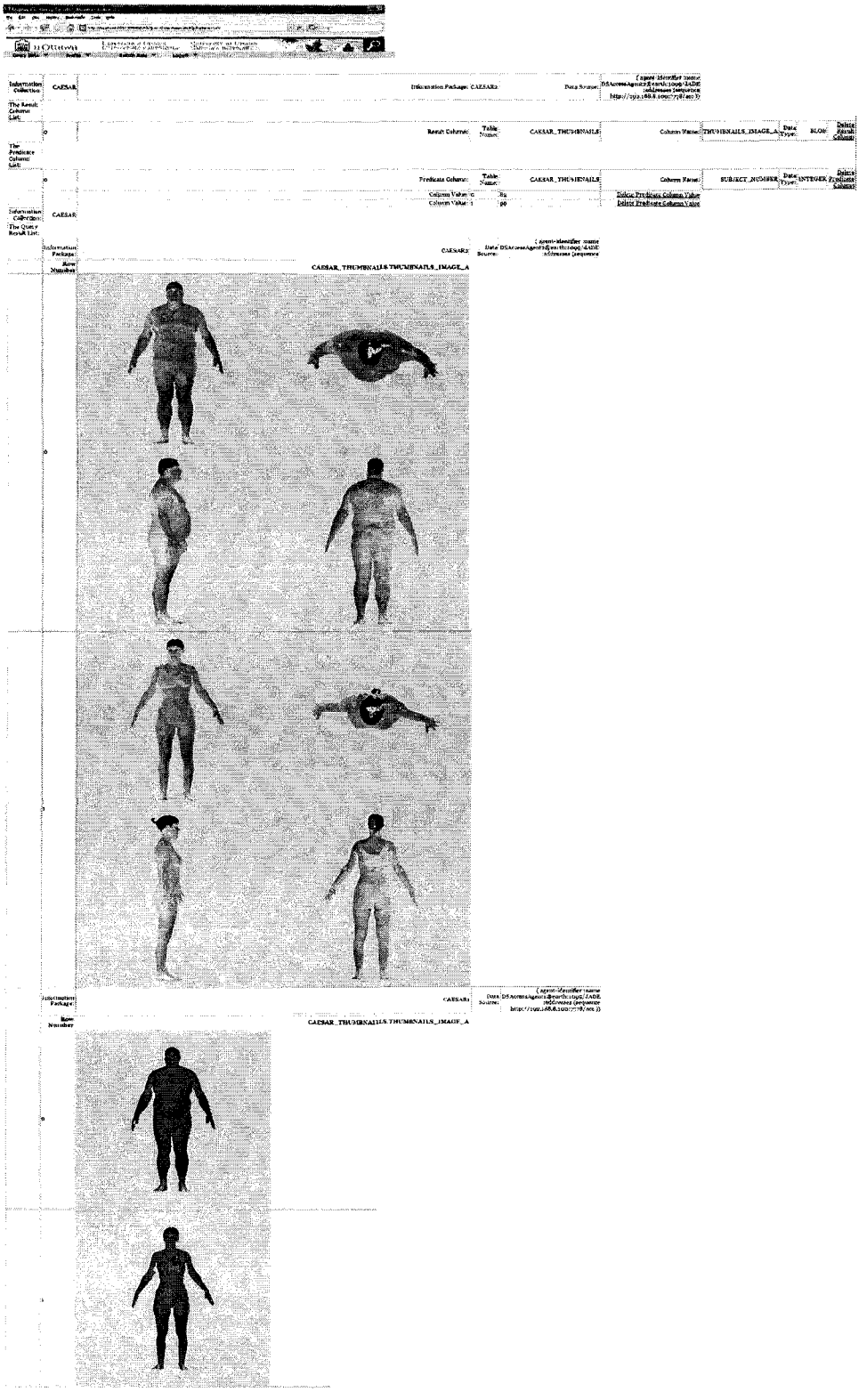


Figure 72 Data of Type 2D Image

5.9. Result Analysis

As stated at the beginning of this chapter, initially the compliance of the IdeAL Framework against some standards was evaluated.

Firstly, the IdeAL Framework can be claimed to be OAIS-compliant. The OAIS reference model provides a common set of concepts, responsibilities, information models, and processes. The IdeAL Framework uses the terms and concepts defined in the OAIS standard. The technical part of the OAIS mandatory responsibilities are accomplished in the IdeAL Framework. The definition of polices and procedures is left to the future implementation. Moreover, the IdeAL Framework fully conforms to the OAIS Information Model. Table 3 shows that the Virtual Model contains most of the entities of the OAIS information, and the only exception (Packaging Information) can be obtained through the use of METS in the IdeAL Framework. Further, the current IdeAL Framework concentrates on the technology-oriented parts of the OAIS functional model. Within all six functional entities, the core functions as a long-term data preservation system are implemented. The Preservation Planning function entity and the Administration function entity are barely realized, because they are mainly concerned about polices, procedures, agreements, and responsibilities.

Secondly, the design of the metadata (Virtual Model) adopts PREMIS and METS, because the OAIS only provides a reference framework without implementation guidance. Further, the adoption provides the IdeAL Framework the ability to interoperate with other digital repositories. As introduced in Section 2.1.3, PREMIS is a metadata framework

that claims to contain essential preservation metadata elements. 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. Moreover, as introduced in Section 2.1.3, METS is used for the interoperability between digital repositories by providing a framework for integrating various types of metadata. METS is employed for designing dissemination templates of the DIPs (data and its metadata) in the IdeAL Framework and for preparing for the potential of the interoperability with other digital preservation repositories. An example is presented to illustrate how METS is used to construct a DIP based on the metadata in the Virtual Model.

After introducing the test data sets (CAESAR^{TM}), the IdeAL Framework was evaluated against the general metrics for trusted digital repositories. The experimental results indicate that the IdeAL Framework can successfully accomplish the technical parts of the general metrics for trusted digital repository evaluation proposed in [Nestor, 2006]. These metrics include integrity, authenticity, and the necessary functionality of digital repositories. Integrity refers to the completeness of the digital objects and to the exclusion of unintended modifications. 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 data in the IdeAL Framework and it compares with the one in the Virtual Model. If they are same, the integrity is guaranteed. Digital data can be interpreted authentically if it can show its original behaviour, functionality, look, and feel [Rothenberg, 1999]. The Public Key Infrastructure is used to ensure the

data authenticity. The original data, its message digest, the public key of the data creator, and the digital signature are stored in the IdeAL Framework. The digital signature can be verified with the public key to authenticate the data in the IdeAL Framework as the experiments in Section 5.5.2. Moreover, the experiments show that the IdeAL Framework can fulfil the essential functions of digital repositories, including the identification of the digital objects, formal description of digital objects content and structure, interpretability of digital objects, and documentation of all changes to digital objects.

Finally, the achievement of other objectives of the IdeAL Framework was appraised. The MAS (Multi-Agent System) provides an ideal mechanism for the scalability and evolution of digital data. The SRB (Storage Resource Broker) Agent is used as a single interface to the non-agent environment to provide transparent database access to the Business Logic Process System. The experimental example in Section 5.6 shows that the SRB Agent functions properly and correctly. The scalability of the MAS is discussed based on the success implementation of the MAS and the capability of agent development framework JADE. The conclusion is that the MAS can potentially support the IdeAL Framework to involve thousands of data sets. Further, the data integration functionality is evaluated successfully from three aspects: (1) presenting the data types already in the IdeAL Framework, such as 2D, date, and integer, (2) showing the data evolution of two data sets, and (3) illustrating the situation when one user-defined column only exists in part of all the similar data sets.

Chapter 6. Conclusions

The IdeAL Framework presented in this thesis provides an effective and efficient environment for preserving data in multiple databases for a very long time. This is successfully demonstrated through the implementation and evaluation of the framework.

6.1. Summary of Contributions

This thesis has four contributions. Firstly, The IdeAL Framework is proposed to address the problem of long-term data preservation in multiple databases. The framework can be used to archive, maintain, and retrieve the data in persistent databases over a long period of several decades. As the cornerstone of digital preservation, a specific metadata schema, the Virtual Model, is created based on the ideas from OAIS, PREMIS, and METS. Thus, we provide a novel set of guidelines or directives to be used when evaluating long-term data preservation environments. Secondly, a multi-agent system is used in the IdeAL Framework to cope with the scalability and evolution of the data. Further, the SRB (Storage Resource Broker) Agent in the multi-agent system is used as a mechanism for accessing the data smoothly and transparently. Thirdly, an implementation is developed to assess the IdeAL Framework, and as the base for further research. The implementation contains a web-based portal, and essential functions of digital repositories, which include archiving, retrieving and analyzing data in these repositories. Finally, a novel evaluation method of combining theoretical proof and empirical confirmation is employed to evalu-

ate the IdeAL Framework. This method can provide some ideas for the establishment of the standards for assessing the trustworthiness and effectiveness of long-term data preservation systems.

6.2. Future work

The effectiveness of the IdeAL Framework has been illustrated and a workable implementation has been provided as the basis for further extension and improvement. Future research should address several issues to extend the capability of the IdeAL Framework. Various kinds of emulators can be integrated into the implementation of the framework. Additional research is needed into the automatic generation of metadata, through self-description of digital objects or the provision of archiving mechanisms in authoring tools. Moreover, except the general metadata definition in the Virtual Model, specific metadata for each individual kind of data can be further investigated, such as the NISO MIX (NISO Metadata for Images in XML) for images. Such new metadata can be easily integrated into the Virtual Model.

Although METS is adopted for encapsulating information packages (AIP, DIP, and SIP) and the metadata in the Virtual Model are well designed for constituting information packages, no standard templates of information packages are proposed at the current stage. However, the information packages are one of the key aspects for interoperability between the IdeAL Framework and other digital repositories. Some research efforts should be put into this issue because of the importance of the interoperations in the future.

Furthermore, the interoperation and the exchange between our framework and other digital repositories can be an interesting research area.

From the implementation aspect, only IBM DB2 databases are considered at current stage. However, during the long life cycle of an IdeAL Framework, data sets may be from other kinds of DBMS, such as ORACLE, Sybase, and MS SQL Server. Therefore, specific access agents for each DBMS are needed and the evaluation of the IdeAL Framework with heterogeneous DBMS is required.

Further investigation is needed on globally accepted evaluation criteria and methods for long-term preservation of digital data. The methodologies in this thesis can provide some suggestions for this task, such as combining theoretical proof and empirical confirmation, assessing the metadata against the OAIS standard, and testing the functions against the general metrics of trusted digital repositories.

Currently, the research has focused on the technology aspect of long-term data preservation. While complete policies and procedures are also crucial for long-term data preservation repositories and these are required for the IdeAL Framework to be a full-fledged system. Further, data mining functions can be added to the framework for making use of the value of the information already preserved therein. Moreover, more data integration applications and technologies can be incorporated into the IdeAL Framework.

References

[Alur, Crup, & Malks, 2003] Alur, D., Crupl, J., & Malks. D. (2003). *Core J2EE Patterns: Best Practices and Strategies*. (2nd Edition). : Prentice Hall / Sun Microsystems Press.

[Baklarz, & Wong, 2002] Baklarz, G., & Wong, B. (2002) *DB2 Universal Database V8 for Linux, UNIX, and Windows Database Administration Certification Guide (5th Edition)*: IBM Press.

[Beedham, Missen, & Palmer, 2004] Beedham, H., Missen, J., & Palmer, M. (2004). *ASSESSMENT OF UKDA AND TNA COMPLIANCE WITH OAIS AND METS STANDARDS*: Systems Committee, UK Data Archive, National Archives.

[Bellifemine, & Caire, 2007] Bellifemine, F., & Caire, G., (2007). *JADE Programmer's Guide* : TILAB. Retrieved October 8, 2007 from http://jade.tilab.com/doc/programmersguide.pdf.

[Bergsten, 2003] Bergsten, H. (2003) *JavaServer Pages, Third Edition*: O'Reilly.

[Bernon, Cossentino, Gleizes, & Turci, 2004] Bernon, C., Cossentino, M., Gleizes, M.P., & Turci, P. (2004) *A study of some multi-agent meta-models*. Lecture Notes in Computer Science. Volume 3382, pp. 62-77.

[Bodoff, Green, Haase, Jendrock, Pawlan, & Stearns, 2002] Bodoff, S., Green, D., Haase, K., Jendrock, E., Pawlan, M., Pawlan, M., & Stearns. B. (2002) *J2EE™ Tutorial*: Prentice Hall.

[Burrafato, & Cossentino, 2002] Burrafato, P., & Cossentino, M., (2002) *Designing a multi-agent solution for a bookstore with the PASSI methodology*: In Procs. Agent-Oriented Information Systems, pp. 102-118.

[CCSDS, 2002] Consultative Committee for Space Data Systems (CCSDS). (2002). *Reference Model for an Open Archival Information System (OAIS)*. Retrieved October 8, 2007 from http://public.ccsds.org/publications/archive/650x0b1.pdf.

[Chmiel, Tomiak, Gawinecki, & Karczmarek, 2004] Chmiel, K., Tomiak, D., Gawinecki, M., & Karczmarek, P. (2004). *Testing the efficiency of JADE agent platform*. Proceedings - ISPDC 2004: Third International Symposium on Parallel and Distributed Computing/HeteroPar '04: Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Hete. pp. 49-56.

[Chmiel, & Gawinecki, 2005] Chmiel, K., & Gawinecki, M. (2005) *Efficiency of JADE Agent Platform*: Scientific Programming. Volume 13, Issue 2, pp. 159-172.

[Cornell University Library, 2003] Cornell University Library. (2003). *Digital Preservation Management: Implementing Short-term Strategies for Long-term Problems*: Retrieved October 8, 2007 from http://www.library.cornell.edu/iris/dpworkshop/.

[Cortese, & Quarta, 2002] Cortese, E., Quarta, F. (2002) *Scalability and Performance of JADE Message Transport System.* Proceedings of AAMAS Workshop on AgentCities, Bologna. Retrieved October 8, 2007 from http://sharon.cselt.it/projects/jade/papers/Final-ScalPerfMessJADE.pdf.

[Cossentino, & Potts, 2002] Cossentino, M., & Potts, C. (2002) *PASSI: a Process for Specifying and Implementing Multi-Agent Systems Using UML*. Retrieved October 8, 2007 from http://citeseer.ist.psu.edu/cache/papers/cs/32683/http:zSzzSzwww.cc.gatech.eduzSzclasseszSzAY2002zSzcs6300_fallzSzICSE.pdf/passi-a-process-for.pdf

[Day, 2006] Day, M. (2006). *The long-term preservation of Web content,* Web Archiving (Berlin: Springer-Verlag), pp. 177-199.

[Deborah Woodyard-Robinson Holdings, 2005] Deborah Woodyard-Robinson Holdings Ltd. (2005) *Implementing the PREMIS data dictionary: a survey of approaches*: Retrieved October 8, 2007 from http://www.loc.gov/standards/premis/implementation-report-woodyard.pdf.

[DLF, 2007] Digital Library Federation. (2007) *Metadata Encoding and Transmission Standard: Premier and Reference Manual*. Retrieved October 8, 2007 from http://www.loc.gov/standards/mets/METS%20Documentation%20final%20070930%20msw.pdf.

[Doyle, Viktor, & Paquet, 2007] Doyle, J., Viktor, H.L., & Paquet, E. (2007) .*Long Term Digital Preservation – An End User's Perspective*. the 2nd IEEE International Conference on Digital Information Management (ICDIM2007), Lyon: France, October 28-31, 2007.

[Eckel, 2004] Eckel, B. (2004) "Thinking in JAVA" : Prentice Hall.

[Enders, Kehoe, & Smith, 2006] Enders, M., Kehoe, W., & Smith, A. (2006). *Bringing Many Tools Together: Building a System of Co-operating OAIS's in the MathArc Project.* Retrieved October 8, 2007 from http://hdl.handle.net/1813/3687

[FIPA, 2002] The Foundation for Intelligent Physical Agents. (2002) *The Foundation for Intelligent Physical Agents Specifications*: Retrieved October 8, 2007 from http://www.fipa.org/repository/standardspecs.html.

[Gewirtz, & Gano, 2006] Gewirtz, D., & Gano, G. (2006). *Towards a Preservation Content Model for Numeric Data Collections: PREMIS and FEDORA*. Retrieved October 8, 2007 from http://hdl.handle.net/1813/3694.

[Gladney, 2004] Gladney, H.M. (2004). *Trustworthy 100-Year Digital Objects: Evidence after Every Witness is Dead*: In CAM Transactions on Information Systems, Vol. 22, No. 3, pp. 406-436.

[Hedstrom, 2001] Hedstrom, M. (2001) *Digital Preservation: Problems and Prospects*: Digital Libraries, Volume 20, page 3-15.

[Hedstrom, Brandt, & Campbell, 2002] Hedstrom, M., Brandt, L., & Campbell, L. (2002). *It's about Time: Research Challenges in Digital Archiving and Long-term Preservation.* In Library of Congress (October 2002), Preserving Our Digital Heritage: Plan for the National Digital Information Infrastructure and Preservation Program, Washington, D.C.: Library of Congress: 205–220.

[Hedstrom, & Ross, 2003] Hedstrom, M., & Ross, S. (2003) *Invest to Save: Report and Recommendations of the NSF-DELOS Working Group on Digital Archiving and Preservation.* Retrieved October 8, 2007 from http://delos-noe.iei.pi.cnr.it/activities/ internationalforum/Joint-WGs/digitalarchiving/Digitalarchiving.pdf.

[Heminger, & Robertson, 1998] Heminger, A.R., & Robertson, S.B. (1998). *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, Volume 2, page 158-167.

[Hodge, 2002] Hodge, G.M., (2002). *Digital preservation: Overview of current developments.* Information Services and Use. Volume 22. Issue 2,3. pp. 73-82.

[Hodge, & Frangakis, 2004] Hodge, G., Frangakis, E. (2004) "Digital Preservation and Permanent Access to Scientific Information: The State of the Practice": Retrieved October 8, 2007 from http://www.cendi.gov/publications/04-3dig_preserv.html.

[Huhns, & Singh, 1998] Huhns, M.N., & Singh, M.P. (1998) *Readings in Agents.* Morgan Kaufmann Publishers.

[Husted, & Dumoulin, 2003] Husted, T.N., & Dumoulin, C. (2003). *Struts in Action: Building Web Applications with the Leading Java Framework.* NetLibrary, Incorporated.

[ISO, 2003] International Organization for Standardization. (2003). *Space data and information transfer systems - Open archival information system - Reference model*, Retrieved October 8, 2007 from http://www.iso.org/iso/iso_catalogue/ catalogue_tc/ catalogue_detail.htm?csnumber=24683

[Jennings, 2001] Jennings, N.R.. (2001). *An agent-based approach for building complex software systems.* Communications of the ACM. Volume 44. Issue:4. pp. 35-41.

[Koch, 2007] Koch, W. (2007). *The GNU Privacy Guard Manual.* Free Software Foundation.

[Koehler, 2004] Koehler, W. (2004) *A longitudinal study of Web pages continued: a consideration of document persistence.* Information Research, Volume 9, No. 2. Retrieved October 8, 2007 from http://informationr.net/ir/9-2/paper174.html.

[Lavoie, 2000] Lavoie, B. (2000). *Meeting the challenges of digital preservation: The OAIS reference model*: OCLC Newsletter 243 (January/February), pp. 26-30.

[Lavoie, & Gartner, 2005] Lavoie, B., & Gartner, R. (2005). *Preservation Metadata.* DPC Technology Watch Report (No. 05-01). Retrieved October 8, 2007 from http://www.dpconline.org/docs/reports/dpctw05-01.pdf.

[Lawrence, Pennock, Flake, Krovetz, Coetzee, Glover, Nielsen, Kruger, & Giles, 2001] Lawrence, S., Pennock, D.M., Flake, G.W., Krovetz, R., Coetzee, F.M., Glover, E., Niel-

sen, F.A., Kruger, A., & Giles, C.L. (2001). *Persistence of Web references in Scientific Research*: Computer, Volume 34, Issue 2, pp. 26-31.

[Lee, Clifton, & Langley, 2006] Lee, B., Clifton, G., & Langley, S. (2006). *Preservation Metadata: Adapting or Adopting PREMIS for APSR*. Retrieved October 8, 2007 from http://hdl.handle.net/1813/3693

[Lenzerini, 2002] Lenzerini, M. (2002) *Data Integration: A Theoretical Perspective*. In Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 233-246.

[Levy, 2001] Levy, A.Y. (2001). *Answering Queries Using Views: A Survey*. VLDB Journal 10 (4), pp. 270-294.

[Lorie, 2004] Lorie, R. (2004). *Long Term Preservation of Digital Cultural Heritage*. In Proceedings P4 Panel Long Term Preservation of Digital Content, 19th May ACM W3C – WWW2004 New York.

[Luck, ASHRI, & D'INVERNO, 2004] Luck, M., ASHRI, R., & D'INVERNO, M. (2004). *Agent-Based Software Development*. Artech House.

[Luck, McBurney, Shehory, & Willmott, 2005] Luck, M., McBurney, P., Shehory, O., & Willmott, S. (2005). "Agent Technology: Computing as Interaction". AgentLink.

[Maes, 1991] Maes, P. (1991) *The Agent Network Architecture*. SIGART Bulletin, 2(4). pp.115-120.

[Moczar, 2004] Moczar, L. (2004). *Tomcat 5 Unleashed*. Sams.

[Moore, 1998] Moore, G. (1998) *Cramming more components onto integrated circuits*. Proceedings of IEEE. Volume 86. Issue 1. pp. 82-85

[Nahm, & Ishikawa, 2005] Nahm, Y.E., & Ishikawa, H. (2005). *A hybrid multi-agent system architecture for enterprise integration using computer networks*. Robotics and computer-integrated manufacturing. Volume:21. Issue:3 pp. 217-234.

[Nestor, 2006] Nestor Working Group on Trusted Repositories Certification. (2006). *Catalogue of Criteria for Trusted Digital Repositories*. Version 1. Frankfurt, Germany: Retrieved October 8, 2007 from http://edoc.hu-berlin.de/series/nestor-materialien/8en/PDF/8en.pdf.

[Neuroth, & Strathmann, 2005] Neuroth, H., & Strathmann, S. (2005). *Developing a National Preservation Policy: Experiences in Germany*. Retrieved October 8, 2007 from http://rdd.sub.uni-goettingen.de/conferences/ipres05/programme.

[OCLC, RLG, 2007] A working group sponsored by OCLC and RLG. PREMIS Data Dictionary Version 1.0. Retrieved October 8, 2007 from http://www.oclc.org/research/projects/pmwg/premis-dd.pdf.

[Rahmi, Bjursell, Paprzycki, Cobb, & Ali, 2006] Rahmi, S., Bjursell, J., Paprzycki, M., Cobb, M., & Ali, D. (2006). *Performance evaluation of SDIAGENT, a multi-agent system for distributed fuzzy geospatial data conflation.* Information Sciences. Volume 176. Issue 9. pp. 1175-1189.

[Rothenberg, 1999] Rothenberg, J. (1999). *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation*, Council on Library and Information Resources, Washington, USA.

[Singh, Stearns, & Johnson, 2002] Singh, I., Stearns, B., & Johnson, M. (2002). *Designing Enterprise Applications with the Java™ 2, Enterprise Edition*. Prentice Hall.

[Stone, Veloso, 2000] Stone, P., Veloso, M. (2000). *Multiagent Systems: A Survey from a Machine Learning Perspective*. Autonomous Robots. Vol. 8 No.3. pp. 345-383.

[Sycara, 1998] Sycara, K. (1998). *A Roadmap of Agent Research and Development*. Autonomous Agents and Multi-Agent Systems 1 (1), pp. 7-38.

[Ullman, 1997] Ullman, J.D. (1997). *Information integration using logical views*. In Proceedings of the 6th International Conference on Database Theory (ICDT'97). Volume 1186 of Lecture Notes in Computer Science, pp. 19-40. Springer.

[Verdegem, 2003] Verdegem, R. (2003) *Database Preservation Issues*. Retrieved October 8, 2007 from www.digitaleduurzaamheid.nl/bibliotheek/docs/longterm_preservation_of_databases.pdf.

[Vidal, Buhler, & Huhns, 2001] Vidal, J.M., Buhler, P.A., & Huhns, M.N. (2001). *Inside an agent*. IEEE internet computing. Volume:5. Issue:1. pp. 82-86.

[Viktor, & Paquet, 2005] Viktor, H.L., & Paquet, E. (2005) *Long-term Preservation of 3-D Cultural Heritage Data Related To Architectural Sites*, In the Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005, Mestre-Venice, Italy, pp. 22-24.

[Viktor, Paquet, & Guo, 2006] Viktor, H.L., Paquet, E, & Guo, H. (2006) *Measuring to Fit: Virtual Tailoring through Cluster Analysis and Classification*. Lecture Notes in Computer Science. Volume 4213 LNAI. pp. 395-406.

[Vrba, 2003] Vrba, P. (2003). *JAVA-based agent platform evaluation*. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science). Volume 2744. pp. 47-58.

[Waters, & Garett, 1996] Waters, D., & Garett, J. (1996). *Preserving Digital Information*. Commission on Preservation and Access. Retrieved October 8, 2007 from http://www.oclc.org/programs/ourwork/past/digpresstudy/final-report.pdf

[Waugh, Wilkonson, & Hills, 2000] Waugh, A., Wilkonson, R., & Hills, B. (2000). *Preserving Digital Information Forever,* In 5th ACM Conference on Digital Libraries, page 175-184, San Antonio, Texas.

[Wheatley, 2001] Wheatley, P. (2001). *Migration – a CAMiLEON Discussion Paper*. In Ariagne, No. 49. Retrieved October 8, 2007 from http://www.ariadne.ac.uk/issue29/camileon/.

[White, 2005] White, C. (2005). *Data Integration: Using ETL, EAI, and EII Tools to Create an Integrated Enterprise*. Retrieved October 8, 2007 from http://whitepapers.techrepublic.com.com/thankyou.aspx?&docid=278917&view=278917&load=1

Appendix I: Depiction of Test Data

The following XML file illustrates an excerpt of the metadata of the experimental data sets in the Virtual Models.

```
<Info_Collection>
   <Label>CAESAR</Label>
      <Info_Package>
         <Label>CAESAR1</Label>
         <Info_Unit>
            <Label>CAESAR_MAIN</Label>
            <Column>
               <ColumnName>SUBJECT_NUMBER</ColumnName>
            </Column>
            <Column>
               <ColumnName>ANTHROPOMETRIC_KEY</ColumnName>
            </Column>
            <Column>
               <ColumnName>DEMOGRAPHIC_KEY</ColumnName>
            </Column>
            <Column>
               <ColumnName>DATA_COLLECTION_KEY</ColumnName>
            </Column>
            <Column>
               <ColumnName>CAESAR_THUMBNAILS_KEY</ColumnName>
            </Column>
            <Column>
               <ColumnName>FEATURE_VECTORS_KEY</ColumnName>
            </Column>
         </Info_Unit>
         <Info_Unit>
            <Label>DATA_COLLECTION</Label>
            <Column>
               <ColumnName>SUBJECT_NUMBER</ColumnName>
            </Column>
            <Column>
               <ColumnName>COLLECT_DATE</ColumnName>
            </Column>
            <Column>
               <ColumnName>COLLECT_TIME</ColumnName>
            </Column>
            <Column>
               <ColumnName>RECORDER</ColumnName>
            </Column>
            <Column>
               <ColumnName>MEASURER</ColumnName>
            </Column>
            <Column>
```

```
      <ColumnName>SITE</ColumnName>
   </Column>
</Info_Unit>

         •

         •

         •

   <Info_Unit>
      <Label>CAESAR_FEATURE_VECTORS</Label>
      <Column>
         <ColumnName>SUBJECT_NUMBER</ColumnName>
      </Column>
      <Column>
         <ColumnName>3D_INDEX_A</ColumnName>
      </Column>
      <Column>
         <ColumnName>3D_INDEX_B</ColumnName>
      </Column>
      <Column>
         <ColumnName>3D_INDEX_C</ColumnName>
      </Column>
   </Info_Unit>
</Info_Package>
<Info_Package>
   <Label>CAESAR2</Label>
   <Info_Unit>
      <Label>CAESAR_MAIN</Label>
      <Column>
         <ColumnName>SUBJECT_NUMBER</ColumnName>
      </Column>
      <Column>
         <ColumnName>ANTHROPOMETRIC_KEY</ColumnName>
      </Column>
      <Column>
         <ColumnName>DEMOGRAPHIC_KEY</ColumnName>
      </Column>
      <Column>
         <ColumnName>DATA_COLLECTION_KEY</ColumnName>
      </Column>
      <Column>
         <ColumnName>CAESAR_THUMBNAILS_KEY</ColumnName>
      </Column>
      <Column>
         <ColumnName>ANTHROPOMETRIC_EXTRA_A_KEY</ColumnName>
      </Column>
      <Column>
         <ColumnName>ANTHROPOMETRIC_EXTRA_B_KEY</ColumnName>
      </Column>
   </Info_Unit>
   <Info_Unit>
      <Label>DATA_COLLECTION</Label>
```

```
      <Column>
         <ColumnName>SUBJECT_NUMBER</ColumnName>
      </Column>
      <Column>
         <ColumnName>COLLECT_DATE</ColumnName>
      </Column>
      <Column>
         <ColumnName>COLLECT_TIME</ColumnName>
      </Column>
      <Column>
         <ColumnName>RECORDER</ColumnName>
      </Column>
      <Column>
         <ColumnName>MEASURER</ColumnName>
      </Column>
      <Column>
         <ColumnName>SITE</ColumnName>
      </Column>
   </Info_Unit>
         •

         •

         •

   <Info_Unit>
      <Label>CLOTHING_SIZE</Label>
      <Column>
         <ColumnName>CLOTHING_SIZE_KEY</ColumnName>
      </Column>
      <Column>
         <ColumnName>SHOE_SIZE</ColumnName>
      </Column>
      <Column>
         <ColumnName>JACKET_SIZE</ColumnName>
      </Column>
      <Column>
         <ColumnName>PANTS_SIZE_WAIST</ColumnName>
      </Column>
      <Column>
         <ColumnName>PANTS_SIZE_INSEAM</ColumnName>
      </Column>
      <Column>
         <ColumnName>PANTS_SIZE_WOMAN</ColumnName>
      </Column>
      <Column>
         <ColumnName>BLOUSE_SIZE</ColumnName>
      </Column>
      <Column>
         <ColumnName>BRA_SIZE</ColumnName>
      </Column>
   </Info_Unit>
</Info_Package>
<Relationships>
   <Relationship>
```

```
    <Type>STRUCTURAL</Type>
       <SubType>REFERENCE TABLE</SubType>
       <Member>
          <Sequence>0</Sequence>
          <IP_Label>CAESAR1</IP_Label>
          <IU_Label>CAESAR_CODES</IU_Label>
          <ColumnName/>
       </Member>
</Relationship>
<Relationship>
    <Type>STRUCTURAL</Type>
    <SubType>FOREIGN KEY</SubType>
    <Member>
       <Sequence>0</Sequence>
       <IP_Label>CAESAR1</IP_Label>
       <IU_Label>CAESAR_MAIN</IU_Label>
       <ColumnName>ANTHROPOMETRIC_KEY</ColumnName>
    </Member>
    <Member>
       <Sequence>1</Sequence>
       <IP_Label>CAESAR1</IP_Label>
       <IU_Label>ANTHROPOMETRIC_MAIN</IU_Label>
       <ColumnName>SUBJECT_NUMBER</ColumnName>
    </Member>
</Relationship>
       •

       •

       •

<Relationship>
    <Type>STRUCTURAL</Type>
    <SubType>SIMILAR</SubType>
    <Member>
       <Sequence>0</Sequence>
       <IP_Label>CAESAR1</IP_Label>
       <IU_Label>ANTHROPOMETRIC_MAIN</IU_Label>
       <ColumnName>WAIST_HEIGTH_PREFERRED</ColumnName>
    </Member>
    <Member>
       <Sequence>1</Sequence>
       <IP_Label>CAESAR2</IP_Label>
       <IU_Label>ANTHROPOMETRIC_MAIN</IU_Label>
       <ColumnName>WAIST_HEIGHT_PREFERRED</ColumnName>
    </Member>
</Relationship>
<Relationship>
    <Type>STRUCTURAL</Type>
    <SubType>SIMILAR</SubType>
    <Member>
       <Sequence>0</Sequence>
       <IP_Label>CAESAR1</IP_Label>
       <IU_Label>ANTHROPOMETRIC_MAIN</IU_Label>
       <ColumnName>WEIGHT</ColumnName>
```

```
                </Member>
                <Member>
                    <Sequence>1</Sequence>
                    <IP_Label>CAESAR2</IP_Label>
                    <IU_Label>ANTHROPOMETRIC_MAIN</IU_Label>
                    <ColumnName>WEIGHT</ColumnName>
                </Member>
            </Relationship>
        </Relationships>
    </Info_Collection>
</Info_Collections>
```