A Semi-Automated Digital Preservation System based on Semantic Web Services

Jane Hunter
DSTC PTY LTD
Brisbane, Australia
jane@dstc.edu.au

Sharmin Choudhury
DSTC PTY LTD
Brisbane, Australia
sharminc@dstc.edu.au

ABSTRACT
This paper describes a Web-services-based system which we have developed to enable organizations to semi-automatically preserve their digital collections by dynamically discovering and invoking the most appropriate preservation service, as it is required. By periodically comparing preservation metadata for digital objects in a collection with a software version registry, potential object obsolescence can be detected and a notification message sent to the relevant agent. By making preservation software modules available as Web services and describing them semantically using a machine-processable ontology (OWL-S), the most appropriate preservation service(s) for each object can then be automatically discovered, composed and invoked by software agents (with optional human input at critical decision-making steps). We believe that this approach represents a significant advance towards providing a viable, cost-effective solution to the long term preservation of large-scale collections of digital objects.

Categories and Subject Descriptors
H.3.7 [**Digital Libraries**]: Collection, Standards.
I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems.
H.3.5 [**Online Information Services**]: Web-based services.

General Terms
Design, Management, Standardization

Keywords
Digital Preservation, Semantic Web services

1. INTRODUCTION
Addressing the preservation and long-term access issues for digital resources is one of the key challenges facing informational organizations such as libraries, archives, cultural institutions, scientific organizations and government agencies today. Digital objects require constant and expensive maintenance because they depend on hardware, software, data, models and standards which are upgraded or replaced every few years. Accelerating rates of data collection and content creation and the increasing complexity of digital resources means that many organizations can no longer keep pace with the preservation needs of all of the data entrusted to them.

A number of major initiatives have been established to tackle the problem of preservation of digital content. The US Congress recently appropriated $100 million to the Library of Congress to establish a National Digital Information Infrastructure and Preservation Program (NDIIPP) [1]. Other initiatives such as the CEDARS project [2], CAMiLEON [3], the National Library of Australia's PANDORA project [4,5], Networked European Deposits Library (NEDLIB) [6] and the OCLC/RLG Working Group on Preservation Metadata [7] have all been investigating strategies for the preservation of digital content. These initiatives have primarily been focusing on three main strategies: emulation, migration or some amalgam of these which relies on the encapsulation of the digital object with detailed preservation metadata. In addition most preservation projects have focused on preserving digital objects of a particular media type e.g., web sites (HTML), electronic journals, electronic books, digitally recorded sound, digital moving images, or multimedia objects.

It is widely recognized that there is no single best solution to digital preservation. The most appropriate strategy depends on the particular requirements of the custodial organization, the producers and users of its collection and the nature of the objects in the collection.

In this paper we describe a flexible, dynamic, semi-automated approach which leverages existing work on preservation metadata and preservation software tools (e,g., emulation and migration) by integrating them and making them available through a Web services architecture.

The system which we have developed comprises metadata capture tools which enable the recording of preservation metadata (based on the METS schema) [8] for digital objects in a collection. Each object's formatting metadata is periodically compared with a software registry which stores the latest available versions of the authoring and viewing software required to ensure the object's accessibility. When there is incompatibility between an object's format and the latest available version, a notification is sent to the relevant agent (human or software). If preservation action is requested, then the most appropriate preservation service for the object is dynamically discovered and invoked. This is implemented by making preservation software modules available as Web services and describing them semantically using a machine-processable ontology (OWL-S) [9]. Software agents use the object's preservation metadata to find or compose the most appropriate preservation services for that object. Collection managers then have the option to choose and invoke particular preservation

JCDL'04, June 7–11, 2004, Tucson, Arizona, USA.

services discovered and recommended by the software agents. Figure 5 illustrates the overall system architecture.

We believe that the semi-automated, Web services approach which we describe in this paper is an optimum architecture to provide a viable, cost-effective solution to the long term preservation of large scale collections of complex digital objects.

The remainder of the paper is structured as follows. The next section describes the background to this work, related projects and how our work differs from and builds upon these earlier approaches. Section 3 describes an example scenario. Section 4 describes our extensions to OWL-S to describe preservation services. Section 5 describes the system architecture. Section 6 provides implementation details. Section 7 concludes with an evaluation of the system to date and a discussion of problem issues and future work.

2. BACKGROUND AND OBJECTIVE

2.1 The Digital Preservation Problem

The major threats to a digital object's long term accessibility are:

- Media obsolescence
- Technology obsolescence – both hardware and software

The problem of media obsolescence has diminished in recent years due to improvements in magnetic and optical media for digital storage. This has reduced the frequency at which digital objects must "refreshed" or copied to new storage media.

Hardware and software obsolescence however, is still a major problem which occurs due to incompatibilities between the hardware and software used to create the original digital objects and the current versions being used to retrieve, render and interpret the objects. In the next section, we describe the current technical strategies being applied to digital preservation.

2.2 Current Technical Strategies

The two main technical approaches to digital preservation have been *migration* and *emulation*.

Migration involves converting a document from its original format into successive subsequent formats as each previous format becomes obsolete. The disadvantages of migration are that it is highly labor-intensive, and in some situations it may either corrupt or alter the appearance, structure, meaning or behaviour of a digital resource. The advantage is that in the majority of cases it extends the longevity of a digital resource, at least until the new format becomes obsolete.

Emulation involves saving not just the data but also the program that was used to create/manipulate the information in the first place. Jeff Rothenberg [10,11], the principle advocate of the emulation method, suggests that the only way to decode the bit stream would be to run the old program, which would require the use of an emulator. Several different methods for defining how an emulator can be specified have been suggested, but the feasibility of these methods has not yet been fully demonstrated. The emulation approach suffers from two major drawbacks:

- Saving the original program is justifiable for re-enacting the behaviour of a program, but it is overkill for data archiving. In order to archive a collection of pictures, it should not be necessary to save the full system that enabled the original user to create, modify, and enhance pictures when only the final result is of interest.
- The original program generally shows the data (and more often, results derived from the data) in one particular output form. The program does not make the data accessible; it is therefore impossible to export the original data from the old system to a new one.

Lorie [12,13] suggests another approach which relies only partially on emulation by differentiating between data archiving, which does not require full emulation, and program archiving, which does. Lorie's approach relies on a Universal Virtual Computer (UVC). For data archiving the UVC can extract the data from the bit stream and return it to the caller in an understandable way, so that it may be exported to a new system. To archive a program's behaviour, emulation cannot be avoided and the UVC will emulate the current computer when interpreted on a future one.

Recent research [2,3] recommends an amalgam of these strategies which relies on the preservation of both the original bytestream or digital object, as well as detailed metadata which will enable it to be interpreted in the future. Preservation metadata provides sufficient technical information about the resources to support either migration or emulation. It facilitates long-term access to the digital resources by providing a complete description of the technical environment needed to view the work, the applications and version numbers needed, decompression schemes, other files that need to be linked to it etc.

A number of initiatives have been focusing on the use of metadata to support the digital preservation process: Reference Model for an Open Archival Information System (OAIS) [14], the CURL Exemplars in Digital Archives project (CEDARS) [2], the National Library of Australia's (NLA) PANDORA project [4], the Networked European Deposit Library (NEDLIB) [6] and the Online Computer Library Centre/Research Libraries Group (OCLC/RLG) Working Group on Preservation Metadata [7]. Several research projects have recently begun developing tools and workflows to streamline the capture of preservation metadata [15,16].

A growing number of projects [3,15] are using the Library of Congress's Metadata Encoding and Transmission Standard (METS) [8] schema. METS provides a flexible mechanism for encoding descriptive, administrative, and structural metadata for a digital library object, and for expressing the complex links between these various forms of metadata. Extensions to METS have been developed to support the preservation of audiovisual content [17].

Two other projects, the Typed-Object Model (TOM) [18] and the Format REgistry Demonstration (FRED) [19], are designed to offer disturbed service for preservation of digital objects. However it is unclear how either of these two projects will interface with current initiatives and existing systems.

Clearly there is no single one-size-fits-all solution to digital preservation. There are a large number of radically different preservation strategies available. Our proposal is to build a system which provides access to the wide range of preservation services available but also provides decision-support tools or a

recommender service which can assist the librarian, archivist or collections manager to select the best single service or combination of services for a particular digital object or a particular set of circumstances.

2.3 Semantic Web Services

Web services are enabling networked computer programs to process and consume information. Based on the following open standards, Web services provide a standardized way of enabling Web-based application-to-application interoperability:

- XML (Extensible Markup Language) – structures the application-to-application messages;
- SOAP (Simple Object Access Protocol) – provides the message format for communicating and invoking Web services;
- Web services Description Language (WSDL) – describes how to access Web services;
- Universal Description, Discovery and Integration (UDDI) - provides a registry that clients can use to discover available services.

More recently the Semantic Web services initiative has developed OWL-S/DAML-S, an OWL ontology which enables Web services to be described semantically and their descriptions to be processed and understood by software agents. A number of projects are using OWL-S/DAML-S to describe their domain-specific services and enable software agents to automatically discover, compose, invoke and monitor the most appropriate Web services [20,21]. As far as we are aware, no one is currently applying or extending OWL-S/DAML-S to generate semantic descriptions of digital preservation services so that they can be discovered, invoked and composed by software agents in order to automate the preservation tasks of large archival organizations.

2.4 Objective

Our hypothesis is that the modular, distributed nature of the Semantic Web services architecture makes it perfectly suited to the dynamic, large-scale, heterogeneous nature of the digital preservation problem. Our key objective is to test this hypothesis by developing and evaluating a semi-automated preservation system based on the Semantic Web services architecture which provides access to a suite of independent preservation service components which can be discovered, linked, and used in arbitrary combinations to fulfill the specific preservation tasks and requirements of different archival organizations.

As Hedstrom observed in [22], previous preservation work has focused on conceptual frameworks (OAIS), specific standards for data formats, particular technical strategies and the development of a few persistent repositories. But there are very few working models of well-integrated and cost-effective digital preservation programs or frameworks, such as we have developed and describe here.

3. MOTIVATIONAL SCENARIO

It's the year 2004 and Nancy Pearl, librarian at the *Library of Congress's American Folklife Centre* has just received an email notifying her that a set of valuable digital photographs, stored in TIFF version 5.0 format within the Centre's digital collection, are in danger of becoming obsolete because the latest version of the display software used by the library to render the images (ImageViewer) no longer supports TIFF 5.0. The email message includes a list of the endangered photographs. The message also recommends that the TIFF preservation image format being used by the *Folklife Centre* be replaced by JPEG 2000 which has now become the defacto image preservation standard, recommended by the RLG. Nancy Pearl must now find a TIFF to JPEG2000 conversion service that meets all her service quality parameters. She uses the PANIC system to specify the parameters she requires in the conversion service. For example, she specifies that she requires a service that converts from TIFF 5.0 to JPEG-2000. She prefers a distributed converter since she does not want to download a converter to her local machine, which is old, slow and has limited memory. She also specifies that the distributed converter must be highly reliable, high-speed and not result in any loss in image quality. Her request is handled by a Discovery Agent which searches a Web Service registry for the appropriate service description. The Discovery Agent cannot find any exact matches to Nancy's request but it can find one near match and two conversion services which can be chained to approximate the required service:

1. A service which converts from TIFF 5.0 to JPEG-2000 but is slow - developed by *Tirion Technology*, a software development company in Bath, UK;
2. One service which converts from TIFF 5.0 to TIFF 6.0 (developed by the *National Library of Australia*) and another which converts from TIFF 6.0 to JPEG-2000 (developed by the JPEG2000 Working Group) – both are lossless, reliable and high speed.

The Discovery Agent presents these alternatives to Nancy, ranked according to how well they match her request. Nancy is able to choose her preferred option and the selected Provider Agent then executes this service and returns the JPEG2000 images. After migration is complete , the associated provenance and events metadata (which records a history of preservation actions associated with each digital object in the collection) is also automatically updated.

The next two sections of this paper describe the architecture, components and implementation details of the PANIC system that we have developed to enable the scenario outlined above to become a reality.

4. OWL-S ONTOLOGIES FOR PRESERVATION SERVICES

OWL-S is an OWL-based Web service ontology that has been developed by the DAML Services arm of the DARPA Agent Markup Language program [9] (Earlier releases of this ontology were under the name of DAML-S and were based on DAML+OIL instead of OWL.) The purpose of OWL-S is to provide computer-interpretable descriptions of services so that they can be located, selected, employed, composed and monitored automatically over the Internet. Multiple web services can be matched and chained - interoperating to perform complex tasks and transactions for users dynamically and on-demand.

Figure 1 shows the structure of the OWL-S ontology. There are three main subontologies to the top-level Service ontology:

1. ServiceProfile – provides a description of what the service does, enabling advertising and discovery
2. ServiceModel – provides a detailed description of a service's operation or how it works
3. ServiceGrounding – provides details of how to interoperate with or access a service using messages.

The advantage of OWL-S is that it is very general and can be adapted into describing any Web service. However, the generality of OWL-S is also a disadvantage because it can be TOO general. As such, have extended the OWL-S classes to create more preservation specific class that are to be used to describe preservation Web services. In the remainder of this section, we will describe how OWL-S has been extended through the addition of sub-classes and new properties to enable the semantic description of preservation services.

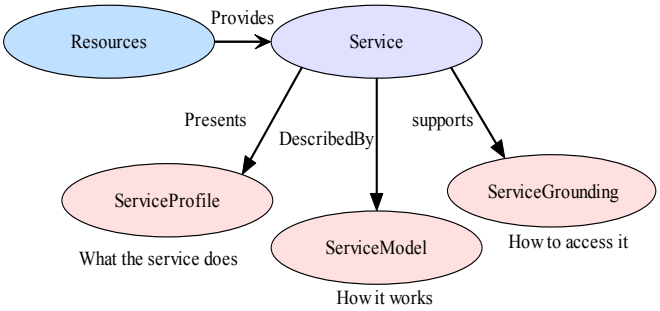

Figure 1: Top level View of the Service ontology [9]

4.1 *Service* Ontology

Service is the upper level ontology which provides the generic *Service* class. Figure 2 illustrates how we have extended Service by defining a *PreservationService* subclass of Service. *PreservationService* has two subclasses – *emulation* and *migration*. These new types of service are defined in PreservationService ontology, which extends Service ontology provided by the DAML-Service Coalition [9].

4.2 *ServiceProfile* ontology

Within the ServiceProfile ontology, the *Profile* class provides three types of information:

1. Service name, description and contact (person or organization);
2. Functional description in terms of inputs, outputs, pre-conditions and effects;
3. An extensible set of properties used to describe features of the service e.g., service category, quality rating, etc.

We have extended the ServiceProfile ontology to create PreservationServiceProfile. Figure 2 illustrates how the Service and ServiceProfile ontologies have been extended to support the description of preservation services.

An existing service that uses ServiceProfiles for matching service requesters and service providers is the *Semantic Matchmaker* [23]. The *Semantic Matchmaker* stores the ServiceProfiles of service providers in its repository. Queries from the service requesters, are submitted as ServiceProfile documents and compared against stored ServiceProfiles. The *Semantic Matchmaker* then returns the ServiceProfiles that best match the query.

We use the *Semantic Matchmaker* service as the Discovery Agent in the PANIC system. When Nancy Pearl specifies the parameters she requires in the preservation web service, a query is created and submitted to the *Semantic Matchmaker*. The *Semantic Matchmaker* then finds the best matches to Nancy's request and sends the results back to her.

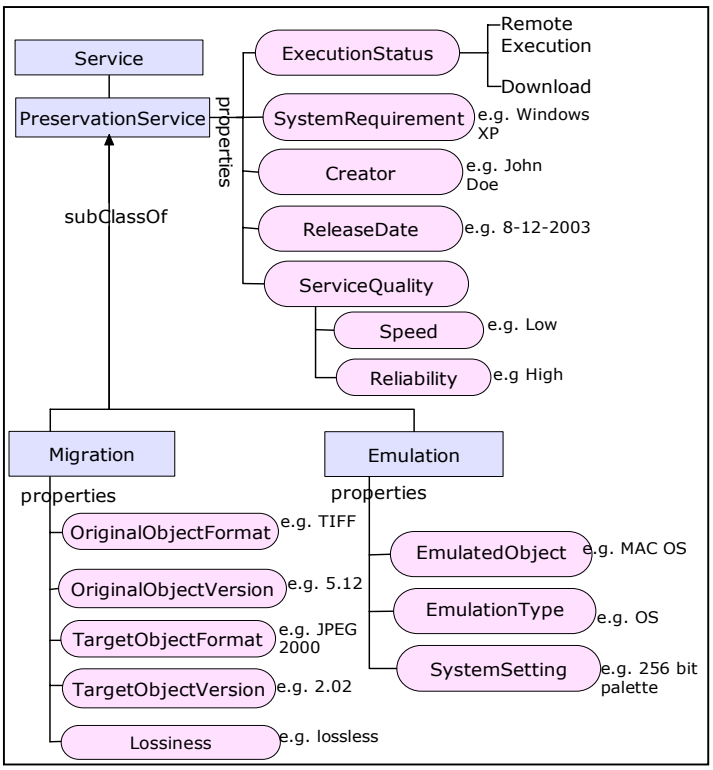

Figure 2: Owl-S Service and ServiceProfile extensions

4.3 *ServiceModel* ontology

The purpose of the ServiceModel subontology is to provide details of how a service operates. OWL-S views a service as a process and defines the *ProcessModel* subclass of *ServiceModel*. A process is defined as producing a data transformation from a set of inputs to a set of outputs. It also produces a transition in the world from one state to another, described by a set of preconditions and effects. Hence a process can have any number of inputs - representing the information that is, under some conditions, required for the execution of the process. A process can also have any number of outputs - the information that the process provides, conditionally, after its execution. Finally, the process can have any number of effects. Outputs and effects can have conditions associated with them.

Three types of process are defined:

- AtomicProcess - a service which executes in a single step, has no subprocesses and is directly invocable;
- SimpleProcess – not invocable and not associated with a grounding but they are conceived as having single-step execution;
- CompositeProcess – decomposable into sub-processes and specified using control constructs such as

"sequence", "split", "choice", "if-then-else", "iterate", "repeat-until".

Figure 3 illustrates the extensions we made to ServiceModel ontology for preservation services and which were defined in the PreservationServiceModel ontology.

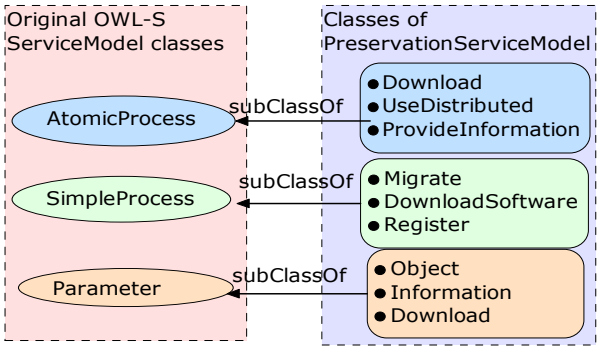

Figure 3: ServiceModel to PreservationServiceModel extensions

We created a class called "Migrate" that is a subclass of "SimpleProcess". This allows, *Tirion Technology* for example, to advertise a migration service that converts TIFF images to JPEG-2000. The PreservationServiceModel instance provided by *Tirion Technology* (in Extract 1) describes the process "tiff_to_jpeg2000" that is an instance of the class "migrate". This description specifies that *Tirion*'s "tiff_to_jpeg2000" process takes a "TIFF" object as input and outputs a "JPEG2000" object.

```
<owl:class rdf:ID="TIFF">
    <rdfs:subClassOf rdf:resource="preProcess:Object"/>
</owl:class>
<owl:class rdf:ID="JPEG2000">
    <rdfs:subClassOf rdf:resource="preProcess:Object"/>
</owl:class>
<preProcess:migrate rdf:ID="tiff_to_jpeg2000">
    <process:hasInput>
        <process:Input rdf:resource="TIFF">
…
     </process:hasInput>
…
     <process:hasOutput>
        <process:Output rdf:recource="JPEG2000">
            <process:parameterType
rdf:resource="http://www.w3.org/2001/XMLSchema#hexBinary" />
        </process:Output>
    </process:hasOutput>
</preProcess:migrate>
```

Extract 1: Tirion's PreservationServiceModel

4.4 *ServiceGrounding* ontology

Within OWL-S the ServiceGrounding ontology specifies how to access a service. It provides details such as protocol and message formats, serialization, transport, and addressing. A grounding can be thought of as a mapping from an abstract to a concrete specification of those service description elements that are required for interacting with the service - in particular the inputs and outputs of atomic processes. Because the ServiceGrounding deals with the technical implementation description it does not require specific extensions for preservation Web services.

The ServiceGrounding ontology utilizes the W3C's Web service Description Language (WSDL), an established industry standard for concrete message specification, for semantically grounding a Web service. The interlinking between the OWL-S ServiceGrounding and WSDL is shown in Figure 4. WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information [24]. OWL-S/WSDL grounding involves a complementary use of the two languages because the two languages do not cover the same conceptual space [9]. This complementary use is best highlighted with an example. Extract 2 shows Tirion Technology's ServiceGrounding document and Extract 3 shows Tirion Technology's corresponding WSDL document. These examples illustrate how the ServiceGrounding acts as a link between WSDL and the semantic descriptions in the OWL-S subontologies.

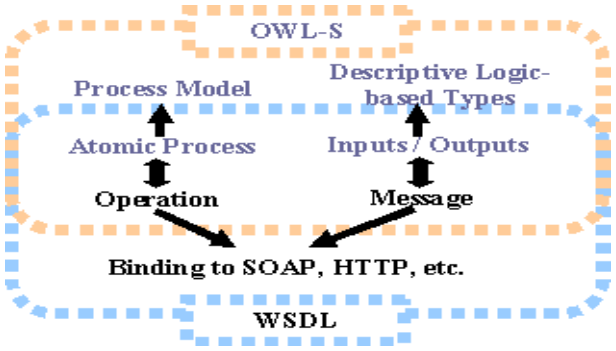

Figure 4: Mapping between OWL-S and WSDL [9]

```
<grounding:WsdlAtomicProcessGrounding rdf:ID="
tiff_to_jpeg2000">
    <grounding:wsdlOperation rdf:resource="#convertImage"/>
    <grounding:wsdlInputs rdf:parseType="Collection">
        <grounding:WsdlInputMessageMap>
        </grounding:WsdlInputMessageMap>
    </grounding:wsdlInputs>
…
    <grounding:wsdlOutputs rdf:parseType="Collection">
    </grounding:wsdlOutputs>
</grounding:WsdlAtomicProcessGrounding>
```

Extract 2: Tirion's ServiceGrounding document

```
<message name="originalImage"/>
<message name="convertedImage"/>
<portType name="tiff_to_jpeg2000">
    <operation name="convertImage">….
</portType>
```

Extract 3: Tirion's WSDL document

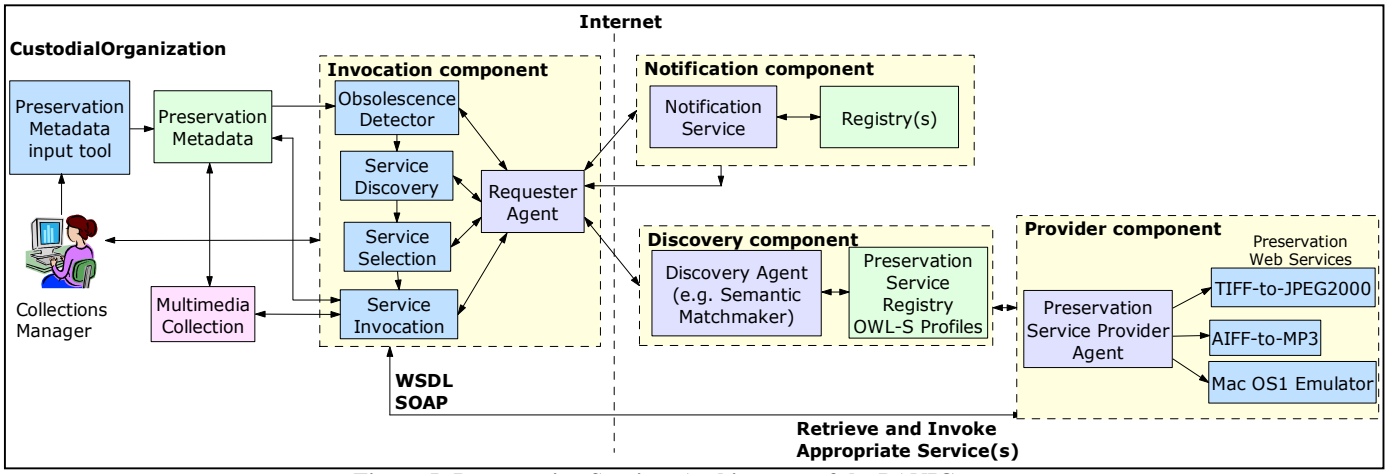

Figure 5: Preservation Services Architecture of the PANIC system

5. SYSTEM ARCHITECTURE

Figure 5 provides an architectural overview of the system which we have developed.

This section describes the four main system components:

1. Invocation component
2. Notification component
3. Discovery component
4. Provider component

5.1 Invocation component

The Invocation component is the set of software modules located on the custodial organization's server which control the transfer of information and the invocation of preservation services. This component also provides the user interface for the collections manager who is making decisions and setting parameters associated with preservation actions. There are five sub-components:

- Obsolescence detector – this module periodically compares the preservation/formatting (METS-based) metadata for each object in the collection (which includes current authoring and viewing software) with information retrieved from the Software Version Registry via the Notification agent. The registry stores information about the latest versions of authoring and viewing software required to access objects in the collection;

  If there is an incompatibility between an object's current preservation/formatting metadata and the latest version recorded in the registry, then a message is sent to the relevant person or software agent, notifying them of a potential accessibility problem.

- Service Discovery – this module provides an optional user interface through which the collections manager can interactively specify the properties of the preservation service they are looking for. The Requester agent then sends a message to the Discovery agent, with a request to search for a service which matches the given description.

- Service Selection – this module presents the services retrieved by the Discovery agent to the user for selection. The retrieved options are ranked according to how well they match the users' request. Alternatively, this module can be configured so that a service is automatically selected and invoked, based on predefined criteria.

- Service Invocation – Upon selection of a service, this module invokes the specified service – either migrating the specified objects or downloading a particular emulator. This module also updates the relevant preservation metadata where necessary;

- Requester Agent – this program acts as the mediator between the organization's collections manager(s) and the preservation services and software agents on the Web. It controls and choreographs the delivery of information, messages and services associated with the system.

5.2 Notification component

The Notification service provides a search and retrieval interface to the information in the Software Version registry which is used to determine potential obsolescence. The registry records information about the latest (authoring and display) software versions, including the digital formats which the latest version of the software supports. More specifically, for each software product, the registry stores: Title, Description, Company, CurrentVersion, ReleaseDate, DeveloperPage, License, Requirements, DownloadSite, DownloadSize, Rating, EaseOfUse, FormatsSupported, Features, Stability, Price. Currently we are using MySQL as the database for this Registry.

The concept of a software registry is not new. VersionTracker [25] is a software application that discovers updates, upgrades and patches for software applications on the users' workstation. To complement the software the makers of VersionTracker also maintain a website with a human searchable registry that enables users to determine whether they should update, upgrade or patch their existing applications. Also, attempts to make a universal formats registry is being made through initiatives such as FRED [19].

An additional Standards Registry that tracks the current recommended preservation standards and the authority making the recommendation is also under development e.g., "the

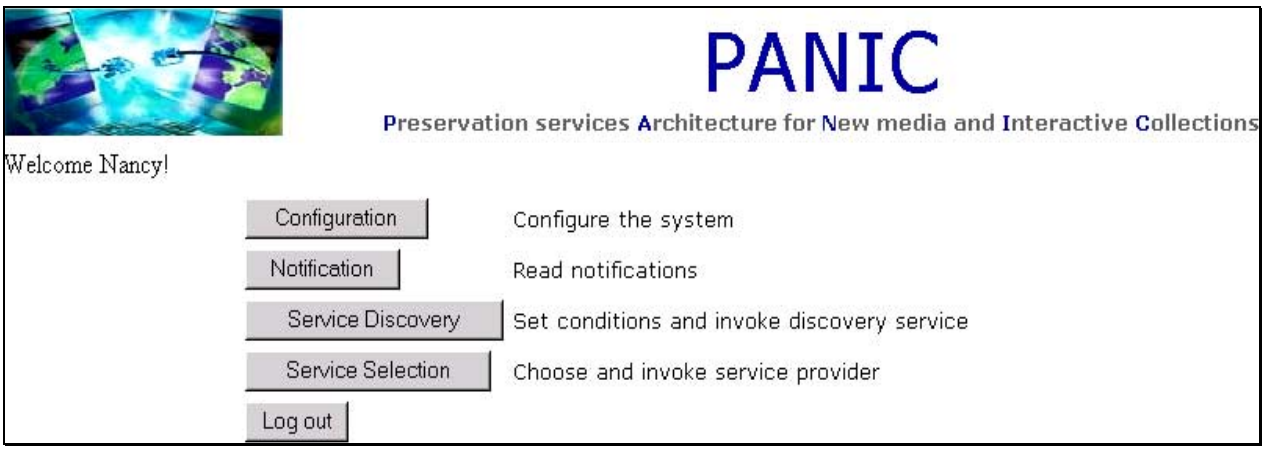

Figure 6: Main Screen

Research Libraries Group (RLG) recommends JPEG-2000 as the preferred preservation format for images". Such information is complementary to the Software Version Registry – increasing the chance of detecting potential obsolescence before it occurs.

The Notification service queries the registries which have been specified by the collections manager and returns the information to the Requester Agent which interprets it and passes it back to the Obsolescence Detection software. Our architecture has the potential to interface with any registry, including FRED [19], provided that the said registry has a Web service agent.

In the example in section 3, querying the Software Registry reveals that the latest version of ImageViewer, the image viewer used by the LoC *Folklife Centre*, no longer supports TIFF images but does support JPEG-2000. Moreover from querying the Standards Registry, it is discovered that the *RLG* is recommending that TIFF files be migrated JPEG2000. Based on this information, the Obsolescence Detector determines the list of potentially endangered (TIFF) objects and sends the list by e-mail to Nancy Pearl, informing her of the problem.

5.3 Discovery component

Nancy Pearl decides to look for suitable preservation services to solve this potential problem. Using the PANIC system, she specifies that she requires a service that converts from TIFF 4.0 to JPEG-2000. She prefers a distributed converter since she does not want to download a converter to her local machine, which is old, slow and has limited memory. She also specifies that the distributed converter must be highly reliable, high-speed and not result in any loss in image quality.

Her request is forwarded by the Requester Agent to a Discovery Agent which searches a Web Service registry for a service description matching Nancy's specification. In the past UDDI registries [26] were used to advertise available Web services but dynamic discovery was difficult due to the lack of rich semantics. Because we are using OWL-S, we can use an existing service such the *Semantic Matchmaker* which performs more precise and dynamic discovery of appropriate services.

In our example, the *Semantic Matchmaker* determines that there are two matches to the service request: one simple process and one composite process. It sends the matching service descriptions (the WSDL, ServiceGrounding, PreservationServiceProfile and PreservationServiceProcess documents) back to the *Requester Agent*. These are then used to generate an email and present the results of the search to Nancy Pearl.

5.4 Provider Component

Given the results of the search and the recommendations of the Discovery Agent (based on pre-defined constraints), the collections manager can either allow the system to automatically invoke the best matching service or interactively choose a particular preservation action and invoke it manually. The collections manager may need to set certain runtime parameters prior to service execution e.g., where to save the output files, whether to update preservation metadata, where to email the logfile.

The Requester Agent then sends the inputs (TIFF files) to the Provider Agent which executes the service and returns the outputs (JPEG-2000 files) to the Requester Agent. The Requester Agent saves the output files locally to the specified location and updates the preservation action metadata – recording what files were converted, when, authorized by whom, and the service that was used. Finally an email is sent to Nancy Pearl, notifying her that the migration of TIFF files has been completed.

6. SYSTEM IMPLEMENTATION

Based on the architecture described in section 5, we have implemented a prototype PANIC system using Java Web services and Java-Server Pages (JSP) for the Graphical User Interface (GUI) development.

Figure 6 shows the main menu of the PANIC system. There are four steps in using the preservation system:

- Configuration
- Notification
- Service Discovery
- Service Selection and Invocation

6.1 Configuration

The configuration interface allows collections managers to set default parameters for the PANIC system. For example, the system manager can define the interval at which the Software

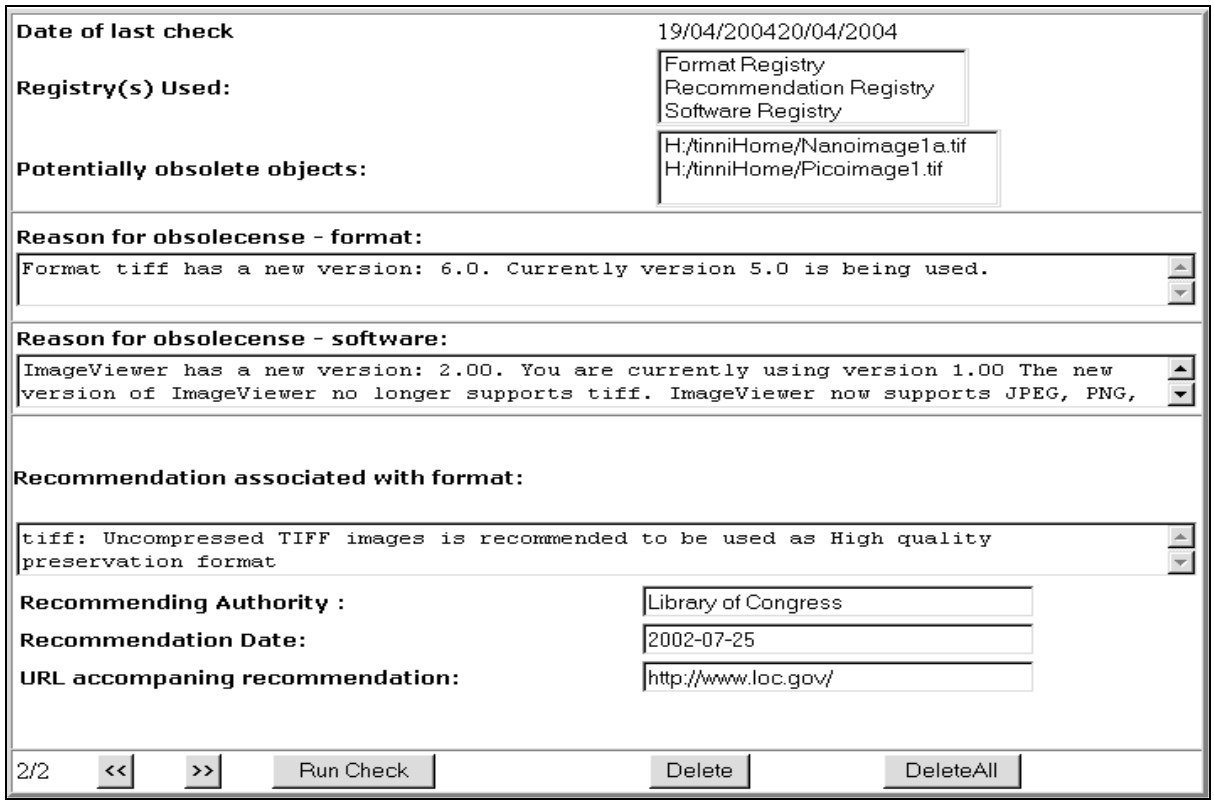

Figure 7: Notification Screen

Version Registry is compared with the preservation metadata of the objects in the collection. The method of notification (email addresses) can also be defined through the configuration interface. Either automatic invocation of preservation services or manual selection and invocation can be specified.

6.2 Notification of Potential Obsolescence

The Notification interface (Figure 7) enables the user to specify which registries will be queried to determine potential digital object obsolescence or inaccessibility. We are currently using a MySQL database to store the Software Version information (described in Section 5.2). The actual information being retrieved is specified through queries that can be pre-defined and built-in to the system or interactively constructed and applied by the user. The list of objects at potential risk and the reason for this, are emailed to the collections manager or accessible through the PANIC Notification interface.

6.3 Service Discovery

The Service Discovery interface (Figure 8) allows the collections manager to specify required service parameters and process inputs and outputs. This information is used to generate a *PreservationServiceProfile* document which is submitted to the *Semantic Matchmaker*. The services which match the request, are then either emailed to the collections manager or accessible via the PANIC interface (Figure 9).

6.4 Service Selection and Invocation

The user can then choose between the suggested services and define runtime parameters such as where the output JPEG-2000 images should be saved (Figure 9).

7. EVALUATION AND CONCLUSIONS

7.1 System Evaluation

To date we have only tested the system on a small (approx. 100) collection of images of different formats - but it has proved highly effective in semi-automatically migrating these images. Based on a comparison of the images' (METS-based) preservation metadata with registries that track the latest software versions and recommended preservation formats, the system can discover and execute the most appropriate migration services. We need to carry out further testing and evaluation of the system using: larger collections; objects of other formats (text, Word, web pages, video, audio, multimedia etc) and a wider variety of preservation Web services. We also need to install and evaluate the system within a real archive or library to determine if there are additional problems or constraints which we need to consider or additional requirements which the system should but does not currently support.

To date we have only invoked atomic and simple processes or services - further work is required to enable the automatic composition of more complex, composite or chained preservation services to match a collections' manager specifications and perform more complex preservation tasks.

7.2 Conclusions

In this paper we have described a prototype digital preservation system which we have developed based on preservation metadata and Semantic Web Services. We believe that the semi-automated, distributed Web services approach which we describe in this paper is the optimum architecture to provide a

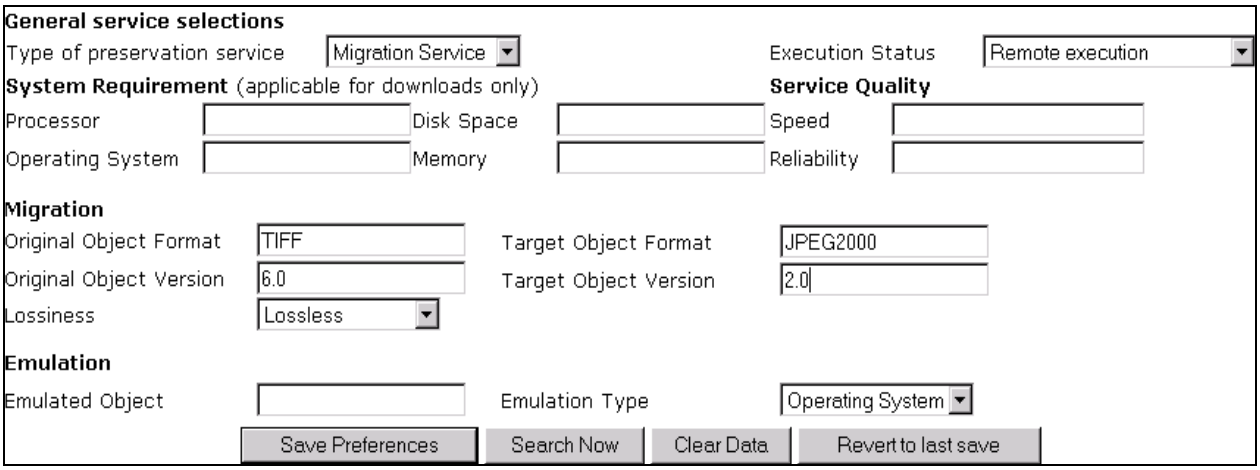

Figure 8: Service Discovery Screen

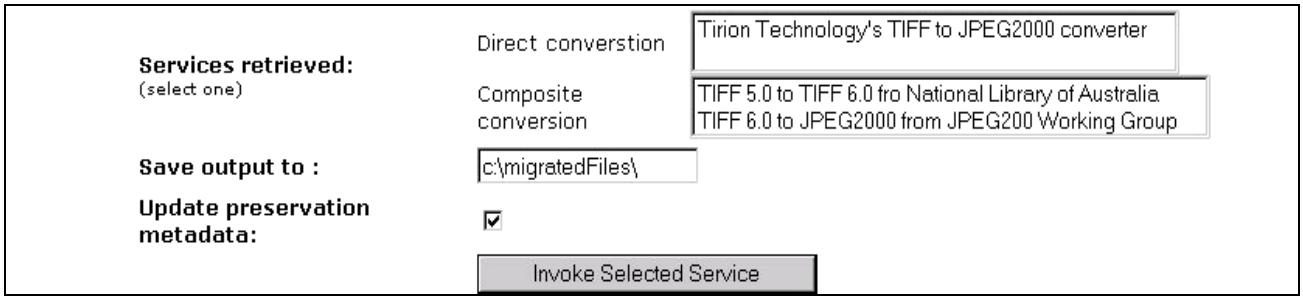

Figure 9: Service Selection and Invocation Screen

viable, cost-effective solution to the long term preservation of large scale collections of complex digital objects. By enabling the automatic detection of potentially obsolescent digital objects and the subsequent discovery and execution of the most appropriate preservation service – the system can potentially save organizations vast amounts of time and effort, as well as prevent the loss of valuable digital assets.

The distributed nature of the proposed Web services architecture offers many advantages. It leverages existing work on preservation metadata and preservation software tools (e.g., emulation and migration services) by integrating them and making them available through a single interface. It enables institutions to coordinate and share their digital preservation activities whilst retaining the flexibility to meet local requirements. The proposed system is scalable and extensible. It has the potential to provide preservation services for all media types and genres. As advances are made in preservation software, they can automatically be incorporated and adopted. Because the system is based on standards including: METS, XML, SOAP, WSDL, UDDI, OWL, OWL-S, interoperability between services and information is optimized. The design offers maximum flexibility - as an organizations preservation needs change, the system can grow and change accordingly. As new preservation services, tools, standards and recommendations evolve, they can automatically be incorporated into the system by adding their semantic descriptions to the relevant registries. As well as providing unified access to the wide range of preservation services available, the system also provides decision-support and recommender services to assist the librarian, archivist or collections manager to select the best single service or combination of services for a particular digital object or a particular set of circumstances. The user interface allows easy customization of the system and human intervention where required – offering the best combination of human and software agents.

To conclude, we believe that the PANIC system which we describe in this paper represents a significant advance towards the development of an integrated and cost-effective digital preservation system for libraries and archival organizations who are currently losing the digital preservation battle.

8. ACKNOWLEDGMENTS

The work described in this paper has been funded by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Education, Science and Training).

9. REFERENCES

[1] National Digital Information Infrastructure and Preservation Program, http://www.digitalpreservation.gov/ndiipp/

[2] CEDARS, CURL Exemplars in Digital Archives http://www.leeds.ac.uk/cedars/

[3] CAMiLEON http://www.si.umich.edu/CAMILEON/

[4] PANDORA http://pandora.nla.gov.au/

[5] National Library of Australia, Digital Archiving and Preservation at the National Library http://www.nla.gov.au/initiatives/digarch.html

[6] Networked European Deposits Library (NEDLIB) http://www.kb.nl/coop/nedlib/

[7] OCLC/RLG Working Group on Preservation Metadata http://www.oclc.org/research/pmwg/

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

[9] The DAML Services Coalition, OWL-S: Semantic Markup for Web services, DAML, 27th of December 2003 http://www.daml.org/services/daml-s/0.9/daml-s.html

[10] Jeff Rothenberg: "Ensuring the Longevity of Digital Documents". Scientific American, 272(1), January 1995.

[11] Jeff Rothenberg, "An Experiment in Using Emulation to Preserve Digital Publications", RAND-Europe, published by The Koninklijke Bibliotheek, Den Haag, Zuid-Holland, Netherlands, April 2000 http://www.kb.nl/coop/nedlib/results/emulationpreservationreport.pdf

[12] Raymond Lorie, A Project on Preservation of Digital Data, 15th June 2001 http://www.rlg.org/preserv/diginews/diginews5-3.html#feature2

[13] Raymond Lorie, The UVC: A Method for Preserving Digital Documents - proof of concept, 2002 http://www.kb.nl/kb/ict/dea/ltp/reports/4-uvc.pdf

[14] OAIS Resources http://www.rlg.org/longterm/oais.html

[15] J. Hunter, S. Choudhury, "Implementing Preservation Strategies for Complex Multimedia Objects", The Seventh European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2003, Trondheim, Sør-Trøndelag, Norway, 17th – 22nd August 2003 http://metadata.net/panic/Papers/ECDL2003_paper.pdf

[16] Eva Müller, Stefan Andersson, Uwe Klosa and Peter Hansson, "Metadata Workflow Based on Reuse of Original Data, 6th International Symposium On Electronic Theses and Dissertations", ETD 2003, Berlin, Germany, 21st to 24th May 2003 http://publications.uu.se/etd2003/papers/MetadataWorkflow.pdf

[17] AV Prototype Project Working Documents, "Extension Schemas for the Metadata Encoding and Transmission Standard", Revised February 2003. http://lcweb.loc.gov/rr/mopic/avprot/metsmenu2.html

[18] Jeannette Wing and John Ockerbloom, "Respectful Type Converters for Mutable Types", IEEE Transactions on Software Engineering (TSE), Volume 26 Number 7, p 579-593, July 2000. http://www-2.cs.cmu.edu/afs/cs/project/calder/papers/focbs/paper.ps

[19] Stephen Abrams, "Global Digital Format Registry (GDFR): Data Model v.3", Revised December 2003. http://tom.library.upenn.edu/fred/docs/Data%20Model%20v3.pdf

[20] Dan Wu, Bijan Parsia, Evren Sirin, James Hendler and Dana Nau, "Automating DAML-S Web Services Composition Using SHOP2", 2nd International Semantic Web Conference, ISWC 2003, Sanibel Island, Florida, USA, 20th – 23rd October 2003 http://www.mindswap.org/papers/ISWC03-SHOP2.pdf

[21] Tse-Ming Tsai, Han-Kuan Yu, Hsin-Te Shih, Ping-Yao Liao, Ren-Dar Yan, Seng-cho T. Chou, "Ontology-Mediated Integration of Intranet Web Services", Computer No 10, Volume 36, October 2003 http://computer.org

[22] Margaret Hedstrom, "Digital Preservation: Problems and Prospects", Journal of Digital Libraries No 20, March 2001 http://www.dl.ulis.ac.jp/Dljournal/No_20/1-hedstrom/1-hedstrom.html

[23] Massimo Paolucci, Katia Sycara, Takuya Nishimura, Naveen Srinivasan, "Using DAML-S for P2P Discovery", International Conference on Web Services, ISWS 2003, Las Vegas, Nevada, USA, 23rd to 26th June 2003 http://www-2.cs.cmu.edu/~softagents/papers/p2p_icws.pdf

[24] W3C. "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language", W3C Working Draft, 10th November 2003 http://www.w3.org/TR/wsdl20/

[25] Tech Tracker, VersionTracker, 2004 http://www.versiontracker.com/vtpro/

[26] UDDI, UDDI Technical white paper, September 6 2000 http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf

[27] Intelligent Software Agents group, "Semantic Matchmaker", http://www-2.cs.cmu.edu/~softagents/a-match/index.html