MIAKT: Combining Grid and Web Services for Collaborative Medical Decision Making

Nigel Shadbolt*, Paul Lewis*, Srinandan Dasmahapatra*, David Dupplaw*,
Bo Hu*, Hugh Lewis+
* IAM Group, School of Electronics and Computer Science,
+ School of Engineering Sciences
University of Southampton, Southampton, UK.
{nrs,phl,sd,dpd,bh,hgl}@ecs.soton.ac.uk

Abstract

Providing semantic web technologies in a medical domain has its obvious advantages. Having distributed services using shared domain vocabularies provides a great impetus for the integration of disparate hospital information systems, as well as the possibility of providing more accurate diagnoses and a well organised knowledge base for sharing, tutoring and researching. Using such disparate systems requires careful consideration both technically, medically and ethically. This paper describes the way the system we have developed offers evidence of the promise of such enhancements in the specific area of screening for breast cancer.

1 Introduction

Web technologies have become ubiquitous in the workplace and GRID services are becoming a reality for computationally intensive applications [14]. Increasingly in the medical domain distributed information sources are being accessed by both medical practitioners and patients alike. Given the promise of the Semantic Web with machine processable documents and the possibility of knowledge-based and grid services over the internet, one might ask the question whether providing for these new affordances in a medical domain can improve the experience and the results thus obtained. Machine understandable documents are provided by the development of relevant ontologies (shared vocabularies for a domain). If users commit to a particular descriptive regimentation of a domain, the data is not only available to machines for unambiguous interpretation, but widely understood by users of the data. In the following sections we briefly describe the domain in which we have undertaken our work, before describing the technologies we use in our architecture.

1.1 Breast Cancer Screening

Breast cancer screening is now mandatory for women over the age of 50. This process consists of the capturing of an X-ray mammogram. Any areas considered abnormal on the mammogram are assessed by means of pathology tests (biopsies). Data from the radiologist, responsible for the mammogram, the histopathologist, responsible for the biopsies, and the clinician, with knowledge of the history of the patient, are brought together to make a consultative appraisal of each particular case. This process is known as the Triple Assessment Procedure and we have undertaken the MIAKT (Medical Imaging with Advanced Knowledge Technologies) project to support this collaborative meeting and the knowledge that goes with it, using the Semantic Web technologies.

The MIAKT system [13] provides knowledge management for the data that the screening process generates, as well as providing a means for medical staff to investigate, annotate, and analyse the data using web and GRID services. Use of ontologies to store knowledge facilitates a mechanism for providing ancillary diagnosis to the consultation. The application software allows viewing and annotation of various types of images, from x-ray mammograms to 3-dimensional MRI scans, provides for searching of patient data, and supports invocation of services on the web for image analysis and data analysis.

2 Ontologies

The organization of the components that make up the system is centered round the different domains of knowledge in use and the ways in which this knowledge is accessed by invoking the functionalities (image processing,

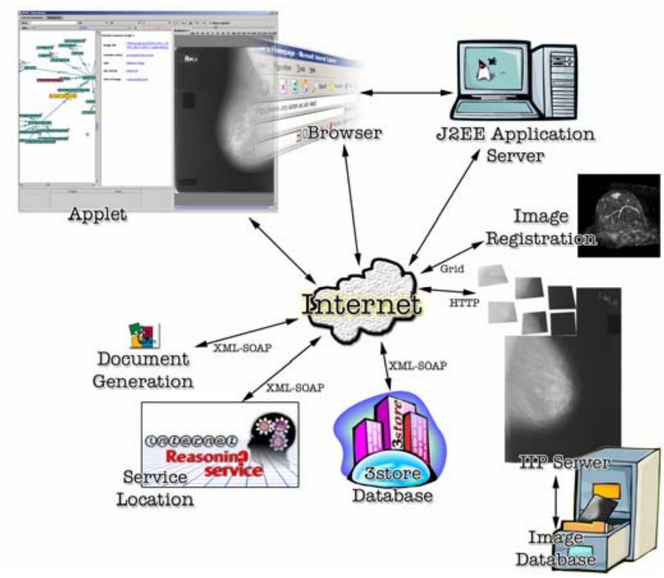

Figure 1 - The MIAKT Framework

annotation, retrieval, patient information records, etc.) that the system provides. These assemblies of domain knowledge as well as the principal concepts underlying the inputs and outputs of system methods are codified as ontologies. The ontologies control the terminological variability in expression of common concepts. We have created an ontology which codifies the relevant domain concepts for X-ray and MRI images, and another for histopathological concepts used in describing results of biopsies.

An essential part of ontology construction is the input from domain experts and users to define the shared vocabulary. This paper will not dwell on the substantive issues that surround the issue of ontology construction in a collaborative problem solving environment such as the one we encounter in Triple Assessment. However, there are real issues regarding the nature and extent of terminological and conceptual consensus between experts in any domain as complex as this.

The X-ray/MRI ontology developed in the project is called the breast cancer imaging ontology (BCIO) [16][17] and it is based on a standardized lexicon called BI-RADS (Breast Imaging Reporting and Data System) developed by the American College of Radiographers (ACR) [1]. Although there are no standards at the moment for MRI images or for histopathology, the BCIO has been built utilizing recommended guidelines by the ACR and the National Health Service (NHS) in the UK. The ontology is structured by simple subsumption relationships to aid in both reusability and reasoning. For example, the class Mammography is defined through a subsumption relationship with the class Medical Exam of which it is a more refined type. Other relationships between concepts are encoded within the ontology for more complex reasoning.

These ontologies we construct are compliant with the Web Ontology Language (OWL) standard [2]. They are used as the receptacles of available data and make the process of data instantiation easier through their limited nomenclature. In order that this arrangement be made scalable, the instances are stored as triples based on the Resource Description Framework (RDF) [10], in a database called the 3store [4], developed on the AKT project [3]. 3store is a set of core C libraries that use MySQL to store raw RDF data and it has been shown to scale well for querying [4]. Below is an example of the RDF that is generated for a patient and stored in the 3store. This particular snippet links a patient identified by the identifier 01261 to a triple assessment procedure, which is identified by the time-stamp of when it occurred. It also explicitly links the patient to the ontology class *Patient.*

```
<rdf:Description
   rdf:about='&n;01261_patient'>
   <rdf:type rdf:resource='&n;Patient'/>
   <n:has_age>51</n:has_age>
   <n:involved_in_ta
      rdf:resource='&n;ta-soton-
      1076933131487'/>
</rdf:Description>
```

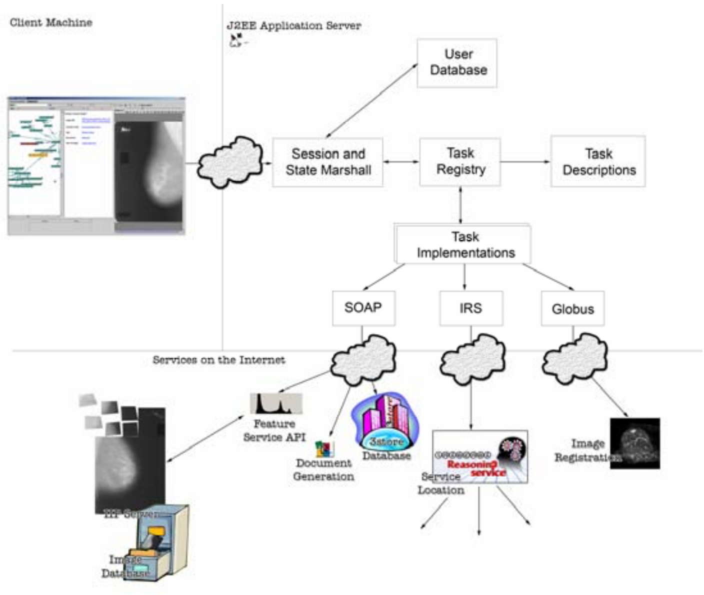

Figure 2 - The MIAKT Task Invocation Framework

As well as this data being stored in the 3store, the ontology definition on which it is based is also stored in the 3store. This allows the application to query the actual structure of the ontology using the servlet-based implementation of the querying engine provided by the 3store interface. The query engine provides a means for searching the RDF triples using a query language called the RDF query language (RDQL) [8]. This in turn provides a means for making inferences over the data that would otherwise be more complicated using a regular database management system. This ontological database technology is integrated into the MIAKT system using the Java Native Interface (JNI) with a web-service API. Providing domain data behind a web-service helps to facilitate the many potentially available applications to be accessed from one, or more, roving clients.

3 The Architecture

The MIAKT application is built around a novel distributed architecture (represented in Figure 1) which uses web-based services to provide discrete and disparate functionality to a generic client application. The architecture is deliberately abstracted from any particular application domain and its description, providing a powerful structure for rapidly prototyping new knowledge management applications in new domains. In this respect, MIAKT becomes a particular application of this architecture.

The architecture is divided by location into three distinct regions of execution (see layout of Figure 2) – the client application, the server (J2EE Enterprise Server) and remote web-services (over various protocols). The client and server deal with presentation and distribution of data while web-services are used for application functionality and also for domain-data storage. Providing distributed service and storage methods means that any roving client application can access the data from wherever they are instantiated.

Distributed web-service driven 3store databases provide the underlying data source for domain dependant instance information while the description of this data, and the functionality that is required to facilitate rational manipulation of this data by the application, is provided to the application by distinct ontological databases. This instance data can be used to provide contextual, user-driven delivery of the application functionality.

Multimedia data, that is always domain-specific, is gathered on remote servers which can be maintained under institutional control, while still being accessible to the framework via relevant internet delivery mechanisms. For example, large x-ray images are accessed remotely through servlets that run the Internet Imaging Protocol (IIP) which deliver image tiles on-demand to the clients that may be on low-bandwidth connections, and MRI-images are delivered to the client from a specialised MRI-

```
<miakt_fw:Instance-Handler rdf:about="&n;MIAKT-System-Preferences_00043"
    miakt_fw:has-name="PatientHandler">
   <miakt_fw:has-class> domain.miakt.PatientHandler </miakt_fw:has-class>
   <miakt_fw:has-description>
        Handler for instance information display of patient information.
   </miakt_fw:has-description>
   <miakt_fw:has-type>&domain;Patient</miakt_fw:has-type>
</miakt_fw:Instance-Handler>
```

Figure 3 - RDF From the *application ontology* that drives the framework and client.

image server that delivers MRI-image-slices on-demand.

Other services are presented to the client application according to the application-specific *application ontology*. The server instantiates the framework using this ontology and provides a simple and homogeneous API to the application's web-services which may be running over various protocols. The MIAKT application, as depicted in Figure 1, has a number of application web-services providing medical-based services for MIAKT. The RDF code snippet in Figure 3 shows part of the MIAKT application ontology that defines the special instance handler for patients (described later).

The client application is delivered to the user through a web-browser interface, using Java applet technology. The client application gives users access to the ontological database instances and provides a means for browsing and manipulating domain-based data. The client application has been designed in such a way that the interface is, for the most part, entirely generic while still providing a means for domain-dependant user interface widgets to be instantiated when necessary. Based on the instances of an *application ontology*, the client application instantiates potentially domain-dependant viewers when the user clicks on an instance that requires specialised viewing, or annotating. The client's architecture has the concept of an annotation, which is an item of interest specified within some domain data, such as a region of interest in an x-ray image. Through the viewers it provides annotations to be generated and for features to be automatically calculated based on the annotation. These features are then able to be stored as instances in the ontological database, based on the ontology structure.

The server-side architecture is depicted in Figure 2, along with an example of how a few of the services that are used for the MIAKT application are integrated. The architecture uses a task-level mapping to map task names to web-service methods. These web-services can be of any of the supported types. Currently the architecture supports XML-SOAP and has native support for the Internet Reasoning Service (IRS) [7]. We may ultimately extend this support to utilise a GRID service invocation framework, such as the Open Grid Services Infrastructure (OGSI) based Globus-3, but we currently support GRID services indirectly by use of web services.[1]

The idea in this task invocation sub-system is first and foremost to protect the client application from changes in the remote web-resources. Because the mapping is all ontology driven, changes of services that may cause client application failure can be corrected with remapping in the ontology. For example, the small snippet of code below is the Java code used to call a specific web-service through the sub-system that looks up the meaning of a medical term in the UMLS.

```
HashMap arguments = new HashMap();
arguments.put( "term", "carcinoma" );
stateMarshall.invokeTask( "lookup",
arguments );
```

This code invokes a task *lookup* on the state marshal. The task registry is queried for tasks matching this task name, and a list of task-implementations that provide functionality for this task are returned. Argument name mapping is applied at both task description and task implementation level, based on the application ontology, and, in the example above, the argument *term* is mapped to the web-service method parameter *theTerm*. Also, through the ontology, default values for arguments can be supplied, allowing common task invocations to be simply invoked. Default values are always able to be overridden. The task invocation system can also be used to provide some service resilience; if more than one task implementation is provided for a single task then calls can be re-routed to different services.

[1] For many users this may be the actual experience of GRID services.

Secondly, the sub-system provides a clean interface for the execution of disparate web-resources that are accessed through different interfaces to the client application. The flexibility provided by this system gives a good base for general application prototyping and deployment.

Automatic service discovery is supported through the use of remote service respositories and description mining. Available services are automatically mapped to tasks in the task invocation system, thereby publishing them for use to the client application. Services are automatically inserted into the generic client application by examining their input and output roles. This context-based application delivery facilitates rapid prototyping of new applications. All these services are remote to the client and server and provide the main functionality of any particular application. Although it is likely that many of these services would be application dependant, application independent services can also be delivered, where necessary, by appropriate application ontology settings.

Image processing and feature extraction algorithms are supported through a supplementary web-service API that provides generation, comparison and retrieval of feature vectors. These 'feature-services' would be delivered from resources with large enough computational and data storage capacity to generate and store the feature vectors resulting from analysis of, potentially large, images. These services retrieve the images from the relevant image servers themselves, and generate feature vectors which they retain in their own databases. This is a pragmatic solution to having a world of incompatible feature vectors and finding a module that can compare a random one of them; the feature vector is always stored with the module that generated it and understands how to interpret it. It also allows the feature module to have a database of feature vectors that is indexed in a way that is beneficial to its retrieval performance. Of course, these restrictions do not preclude the retrieval and re-use of a vector from one of these services.

For MIAKT, classification, image registration, image analysis, and natural language document generation have all been integrated using this framework, and we briefly describe these in more detail later.

4 Client Application

The client application is delivered to the user by means of a Java applet. The client application provides a knowledge management system that is general to any domain yet is able to be tailored to specific domains. To this end, the client application uses a generic architecture that is also driven from the application ontology, deriving from it the available media viewers, ontology instance viewers, and other components that are dynamically loaded into the generic knowledge management application. The client application utilises the authenticated servlet, which the server provides, to access the application's functionality. This allows the client application to be much smaller than if all the functionality were included within it, giving a faster start up time, as well as facilitating the distributed application.

New media viewers are dynamically loaded into the generic application client, instantiated based on the media-type and its delivery mechanism. The locations of viewers are sourced from the framework and the application ontology allowing the client application to be viewer agnostic until the user requests the viewing of a specific media item. The MIAKT client application provides media viewers for X-ray image viewing, sourcing the X-ray images over IIP from the relevant image server, MRI images, which are sourced from the relevant MRI image server, as well as general image, web-page and timeline viewers. The sourcing of the images for these viewers could easily be integrated in large image infrastructures, such as the e-Science e-Diamond system, or Mammogrid [19]. The media viewers are the producers of annotations within the system.

Ontology-instance browsing is provided by the client application using a generic instance viewer. This makes the client agnostic to the ontology-based application. However, should the viewing of some domain-instance require a special handler, the client provides the means for dynamically loading the relevant handler into the user interface at run-time. In the MIAKT application, for example, there is a special instance handler for the *Patient* domain-ontology concept, as shown in Figure 3. When the user attempts to view the information for a patient instance, this handler is automatically instantiated by the client's architecture instead of the generic instance viewer. This domain-specific handler automatically invokes a task to generate a natural language patient report (see below), and automatically dereferences domain-specific links within the instance to the medical images that were taken during the patient's exam. As described earlier, the client application automatically inserts available tasks into the instance handler's interface, based on the types of their input and output roles, thereby giving "free" application prototyping. In the MIAKT application tasks that find similarity between patient instances are provided for through this method.

The client supports the association of descriptive feature vectors from application-dependant media annotations with application-dependant semantic concepts by the use of a defined API between the client and the media viewers, and the framework and the ontology's data-sources. The regimented API allows the architecture to provide automatic activation of modules that perform media processing on the instantiation of a new media item, or the generation of a new annotation on some media. The vector generation is driven by a brokering object that distributes the annotation to feature modules, both remote and local, that are able to generate features for a given type of annotation. The framework's indifference to local and remote activation of media modules allows sites with large computational power, or storage capacity, to be used to generate media descriptive vectors from media which is remote to the module and the client.

The on-demand delivery of the application description to the generic client provides a means to customise the application to a given domain, while the distributed nature of the framework provides unlimited interoperability, giving access to any web and grid service based from any application domain.

5 MIAKT Application Services

Supporting medical decision making is achieved by using the general architecture to provide the knowledge management, while using a specific set of services that provide domain-based reasoning. Firstly, we have developed some simple classifiers (described in more detail in [9]) to investigate the reasoning power of the ontology to predict the finding (malignant or benign) of a given mass. We took a large set of images from the University of South Florida's Digital Mammography Database [5] and used the BIRADS descriptors that have been generated for those images as a training-set for classifying test cases as malignant or benign. We normalise the data from the training-set metadata by assigning binary values for each of the metadata features like Shape, Margin, Architectural Distortion, and so on. These binary values represent whether or not such a feature has been recorded in the case notes accompanying each image. The classifiers are built with the statistics accumulated from the co-occurrence of the features available. The naïve Bayesian classifier maximizes the conditional probability of a label given the observed features *P(label | features)*. We also compared a simple linear classifier (with a binary output variable obtained by taking a logistic function on a linear combination of input data with trainable coefficients) to a MLP with different numbers of hidden variables – 2, 4, 8, 16. All three classifiers give correct classifications in around 75% of the cases. It is clear that the classification accuracy is limited by the data source, and we are investigating more fully-annotated patient cases for achieving better classifications. We also believe that the use of image feature vectors, directly extracted from the annotation boundaries, could be used to increase the accuracy.

Currently, the ontology view is based on a TouchGraph [12] viewer, which mirrors the internal structure of the ontology. We have experimented with instance (as opposed to class or concept) browsing using a viewer based on a formal concept analysis (FCA) [11] of the annotations of the patient cases from [5]. This lattice-based view [9] is constructed by identifying various intersections of predicative sets (each set containing elements possessing one common feature value) where the partial ordering relation is derived from set inclusion. This partial order is used to stitch a lattice together. The number of attributes increases down the page and the corresponding number of elements in the intersection of these predicative sets reduces. The viewer allows browsing of the patient cases based on how similar those cases are to one another.

To provide the radiologist with more detailed information, regions of interest drawn on images can be refined to the fit the object of interest within the region. This refinement is completed prior to calculating useful statistics of the object of interest. Based on these statistics, automatic semantic labelling of the object can be suggested; for example, whether the highlighted mass is round, speculated, or irregular. Web-services developed on the

MIAKT project provide these refinements and statistics for both X-ray images and MRI images. Also, to aid in comparison of MRI images taken at different times, or over a period of time, image registration morphs a target image to match a source image, allowing subtraction of images and change identification. These services execute on the GRID, which allows the highly computational registration calculation to run over many machines [14].

Instances in the ontology represent the patient's examination history including the outcome of those examinations. Medical staff produce reports to document this information, however, using natural language generation, based on the breast cancer ontology as a lexicon [15], reports of the patient's history can be automatically generated [6], saving time. This is currently presented with the patient's instance information in the MIAKT client application, by calling the webservice using RDF data and receiving report text in return. The quote below gives an example of the report.

"The 68 years old patient is involved in a triple assessment procedure. The triple assessment procedure contains a mammography exam. The mammography exam is carried out on the patient on 22 9 1995. The mammography exam produced a right CC image. The right CC image contains an abnormality… The abnormality has a mass, a probably malignant assessment, a microlobulated margin, and a round shape."

The output from the natural language generator is entirely based on the data stored within the ontology. This means that the more information that is stored about a patient in the ontology, the more detailed the report is able to be [18]. Services such as this are of real interest to clinicians who are required to produce summary reports on various aspects of a patient's management.

The Internet Reasoning Service (IRS) [7] provides a means for ontology-based web-service selection using reasoning, by describing web-services semantically. The IRS is built upon the Unified Problem Solving Method Development Language (UPML) framework which distinguishes between *domain models*, *task models*, *problem solving methods*, and *bridges*. Domain models are effectively the domain ontology, while the task models provide a generic description of tasks to be solved. Problem solving methods provide implementation-independent descriptions of tasks, while the bridges map between the various components. It takes a task-centric view, where the client asks for a task to be achieved, and the IRS broker calls the appropriate problem solving method. Through this architecture, the MIAKT application has a set of services that are provided based on their input and output roles. These provide methods such as finding similar patients by their findings, or by their morphology.

6 Summary and Future Work

In this paper we have described the architecture, and the application developed for the MIAKT project, to provide support for medical decision making during the multi-disciplinary meetings that take place during breast-cancer screening, diagnosis and treatment. We described the generic framework that provides knowledge management in any domain using services on the web for both domain and non-domain specific functionality. We have briefly described the services we have developed specifically for the MIAKT application, providing medical-based services for classification and for x-ray and MRI images.

Our interactions with medical staff from the Royal South Hants Hospital and Guys Hospital have shown that there is real and substantial interest in the project. Continued input from them to provide a better evaluation of the techniques involved. We envisage that the client's interface, which has initially been designed as a test-bed for integration, may require redesigning. Ideally this would be a process-model based interface that provides the user with the relevant interface to the relevant services at the right time. We are also designing interfaces that match more closely with the medical staff's current paper-based methods of storage, providing greater familiarity to the users and aiding ease of use.

The MIAKT project has a webpage at http://www.aktors.org/miakt that contains further information about the project.

7 Acknowledgements

The authors are grateful to their collaborators on the project – Kalina Bontcheva, Michael Brady, Liliana Cabral, Fabio Ciravegna, John Domingue, David Hawkes, Enrico Motta, Maud

Poissonnier, Christine Tanner, Jonathan Whiteley, Yorick Wilks and Yalin Zheng – for many valuable discussions. This project (MIAKT) is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number GR/R85150/01. MIAKT is part of the e-Science initiative and is a collaboration between two IRCs supported by the EPSRC – AKT (GR/N15764/01) and MIAS (GR/N14248/01).

8 References

[1] American College of Radiology. *Breast Imaging Reporting and Data System: BIRADS*. Available at http://www.acr.org/departments/stand_accred/birads/.

[2] D. McGuinnes, and F. van Harmelen. *Ontology Web Language (OWL) Overview*. Available at http://www.w3.org/TR/owl-features/. August 2003.

[3] *The AKT Project*. Available at http://www.aktors.org/.

[4] S. Harris, and N. Gibbins. *3store: Efficient Bulk RDF Storage*. In Proceedings 1st Intl. Workshop on Practical and Scalable Semantic Web Systems, Sanibel Island, Florida, USA, 2003.

[5] M. Heath, K.W. Bowyer, D. Kopans et al. *Current Status of the Digital Database for Screening Mammography*, pages 457-460 in Digital Mammography, Kluwer Academic Publishers, 1998

[6] K. Bontcheva, and Y. Wilks. *Automatic Report Generation from Ontologies: the MIAKT approach*. In Proceedings Nineth Intl. Conference on Applications of Natural Language to Information Systems (NLDB'2004), Manchester, UK, 2004.

[7] E. Motta, E., J. Domingue, L. Cabral, and M. Gaspari. *IRS-II: A Framework and Infrastructure for Semantic Web Services*. 2nd Intl. Semantic Web Conference (ISWC2003), Florida, USA, 20th-23rd October 2003

[8] W3C. *RDQL Specification*. Available at http://www.w3.org/Submission/RDQL/. January 2003.

[9] S. Dasmahapatra, D. Dupplaw, B. Hu, H. Lewis, P. Lewis, M. Poissonier, N. Shadbolt. *Ontology-based Decision Support for Multidisciplinary Management of Breast Cancer*. Intl. Workshop on Digital Mammography, North Carolina, 18th-21st June 2004.

[10] O. Lassila and R.R. Swick. 1999. *Resource Description Framework (RDF) Model and Syntax Specification*. W3C. Available at http://www.w3c.org/.

[11] B. Ganter, and R. Wille. *Formal Concept Analysis*. Springer-Verlag, Berlin, 1999.

[12] TouchGraph LLC. *TouchGraph*. Available at http://www.touchgraph.com/.

[13] S. Dasmahapatra, D. Dupplaw, B. Hu, H. Lewis, P. Lewis and N. Shadbolt, *Facilitating Multi-Disciplinary Knowledge-Based Support for Breast Cancer Screening*. Intl. Journal of Healthcare Technology and Management, 2004.

[14] Y. Zheng, C. Tanner, D.L.G. Hill, D.J. Hawkes, M.J. White, M. Khazen, and M.O. Leach. *Alignment of Dynamic Contrast Enhanced MR Volumes of the Breast for a Multicenter Trial: An Exemplar Grid Application*. In Medical Imaging 2004, San Diego, CA, USA. SPIE, February 2004.

[15] K. Bontcheva. *Open-source Tools for Creation, Maintenance, and Storage of Lexical Resources for Language Generation from Ontologies*. Fourth Intl. Conference on Language Resources and Evaluation (LREC'2004). Lisbon, Portugal. 2004.

[16] B. Hu, S. Dasmahapatra, P. Lewis, and N. Shadbolt. *Ontology-based Medical Image Annotation with Description Logics*. In Proceedings of the 15th IEEE Intl. Conference on Tools with Artificial Intelligence. IEEE Computer Society Press, Sacramento, CA, USA, 2003.

[17] B. Hu, S. Dasmahapatra, and N. Shadbolt. *From Lexicon To Mammographic Ontology: Experiences and Lessons*. In D. Calvanese, G. De Giacomo, and E. Franconi, editors, Proceedings of the Intl. Workshop on Description Logics (DL'2003), pages 229-233, 2003.

[18] K. Bontcheva. *Reuse and Problems in the Evaluation of NLG systems*. In Proceedings of EACL’03 Workshop on Evaluation Initiatives, April 2003.

[19] A. Solomonides, R. McClatchey, M. Odeh, M. Brady, M. Mulet-Parada, D. Schottlander and S.R Amendolia. *MammoGrid and eDiamond: Grids Applications in Mammogram Analysis* Proceedings of the IADIS Intl. Conference: e-Society 2003. Lisbon, Portugal. June 2003, pp 1032-1033. ISBN 972-98947-0-1 IADIS Press, Lisbon.