The Knowledge Engineering Review, Vol. 26:3, 329–351.
doi:10.1017/S0269888911000142

A Web-accessible distributed data warehouse for brain tumour diagnosis

FRANCESC ESTANYOL[1], XAVIER RAFAEL[1], ROMAN ROSET[1,*], MIGUEL LURGI[1], MARIOLA MIER[2,†] and MAGI LLUCH-ARIET[1]

[1]*MicroArt, Parc Cientific de Barcelona, Baldiri Reixac 4-6, 08028, Barcelona, Catalonia, Spain;*
e-mail: festanyol@microart.eu, xrafael@microart.eu, roman.roset@gmail.com, mlurgi@microart.eu, mlluch@microart.eu;
[2]*Pharma Quality Europe, Via Brunetto degli Innocenti, 2-50063 Figline Valdarno (FI), Italy;*
e-mail: m.mier@pqe.eu

Abstract

Currently, biological databases (DBs) are a common tool to complement the research of a wide range of biomedical disciplines, but there are only a few specialized medical DBs for human brain tumour magnetic resonance spectroscopy (MRS) data; they typically store a limited range of biological data (i.e. clinical information, magnetic resonance imaging and MRS data) and are not offered as open-source Structured Query Language relational DB schemas. We present a novel approach to biological DBs: a distributed Web-accessible DB for storing and managing clinical and biomedical data related to brain tumours from different clinical centres. This tool is designed for multi-platform systems with dissimilar DB management systems. Being the main data repository of the HealthAgents (HA) project, it uses multi-agent technology and allows the centres to share data and obtain diagnosis classifications from other centres distributed around the world in a reliable way.

The HA project aims to create an agent-based distributed decision support system (DSS) to assist doctors to provide a brain tumour diagnosis and prognosis. The HA DB enables the DSS to totally integrate with its Graphical User Interface to perform classifications with the stored data and visualize the results using the HA distributed agents framework. This new feature converts the system presented in the first application in the world to combine a storage and management tool for brain tumour data and a complete Web-based DSS to obtain automatic diagnosis.

1 Introduction

The introduction of information technologies into the biological and biomedical fields has generated a huge amount of data related to their different specialities, pushing forward the development of databases (DBs) specialized in biomedical data (Stumpflen *et al.*, 2007). Having the data well organized and structured is the first step to extracting relevant biological knowledge (Radhouene *et al.*, 2009) and it opens new lines of investigation to foster research within the scientific community. This community believes that future advances in the biological sciences will depend both on the creation of new knowledge and on the effective management of proliferating information (Blaschke *et al.*, 2002).

Brain tumours are not an exception, owing to the full range of experimental and clinical data related to their diagnosis and prognosis (Mamrak & Yates, 2006) in which a typical clinical centre uses different data to diagnose and make the prognosis for each particular tumour. There is no

* Roman Roset is now a MicroArt associate researcher.
† This work was fully developed while working at MicroArt.

single standard structure to map all the possible needs that may arise when studying different kinds of tumours. There is therefore a real need for tools to organize and manage specialized data in order to be able to extract knowledge in a smooth and reliable way.

This paper documents the HealthAgents (HA) DB and its novel approach to storing and managing brain tumour-related data. The HA project aims to create an agent-based distributed decision support system (DSS), and its main objective is to help doctors make decisions regarding brain tumour diagnosis and prognosis using intelligent agents (Arús *et al.*, 2006). In addition, we should mention that we are providing an application with the ability to help with the follow-up of patients by storing complete information about them over time and by integrating a DSS to obtain a diagnosis based on their acquired data.

The remainder of this paper is organized as follows. First, we outline the rationale behind our approach in Section 2. Second, we describe the methodology that we have followed to design and develop our Web-accessible DB in Section 3. Next, we delineate the actual development of the HA DB in Section 4, including the integration of the DB with the rest of the system. The results and their relevance are described in Section 6. Finally, in Section 7 we conclude by summarizing our work.

2 Rationale and proposed solution

DSSs have a long history in the field of oncology. Almost from the birth of the so-called 'expert systems', there have been programmes dedicated to assisting practitioners to diagnose, treat, or monitor cancers (Shortliffe *et al.*, 1981). However, although there have been significant advances resulting in several prototypes and experiments (Golodetz *et al.*, 2007; Jarman *et al.*, 2008), none have ever been able to support the daily practice of neuro-oncology.

However, the underlying technologies for DSSs have evolved immensely since the early expert systems, and currently their application in cancer diagnosis is increasing. Most of these systems are concerned with breast (Ruggiero *et al.*, 1998; Van Diest *et al.*, 2007), colon (Exarchos *et al.*, 2007), or renal (Chiang *et al.*, 2005) cancers, and there is a lack of effective systems to deal with brain cancers.

In this paper, we will focus on these types of DBs and their relationship to DSSs. Some initiatives provide clear examples of the need for creating DBs that can store and manage all the information related to brain tumour diagnosis.

- INTERPRET: INTERPRET (International Network for Pattern Recognition of Tumours using magnetic resonance (MR); Julià-Sapé *et al.*, 2006; Tate *et al.*, 2006) intends to improve the non-invasive characterization of brain tumours with 1H-MRS (proton magnetic resonance spectroscopy). This project provides an Internet-accessible DB containing validated *in vivo* MR spectra and clinical data of patients with brain tumour. Its model follows a hierarchical DB using plain-text files. A centralized DSS is available to facilitate the clinical use of MRS in brain tumour diagnosis, which uses a classification based on histopathological diagnosis.
- eTumour (Manjón-Herrera *et al.*, 2004; Martínez-Bisbal *et al.*, 2004): eTumour is a DSS for brain tumour diagnosis and prognosis, and incorporates *in vivo* and *ex vivo* genomic and metabolomic data. It uses an Oracle DB and provides manual mechanisms to query data and create new brain tumour classifiers.

Similar to HA, both of these projects have the same final clinical goal: to build a DSS for the early diagnosis of brain tumours using non-invasive techniques (MRS). Both of them aim to create a DB to store all the data required to build the classifiers, but they do not offer a DB with features to perform the monitoring of patients. In fact, none of them have the patient as the basic unit or entity, but rather organize the information in *cases*. A case is a set of experiments carried out on a patient at a given moment. This approach, which is a perfectly valid one for building classifiers, is unfortunately of no help whatsoever when trying to follow a patient's progress, and consequently these DBs cannot be used as a part of the DSS or as a clinical management tool. Both projects have produced a centralized DSS, independent from the DB, which uses local classifiers trained with data coming from a centralized DB.

In contrast, HA follows a different approach to achieve the same clinical goal. It has created a distributed DSS in which the data and classifiers are located in different nodes. As the HA DB installations can be a node of the HA network, it is necessary to integrate the entire agent framework to perform the communications, the data sharing, and the classifications. These characteristics imply a set of particular DB features, as they have to be deployed over a wide range of different platforms and provide facilities to keep the local systems right across the network of centres permanently updated. This innovative distributed approach and the different DB schema to provide a clinical tool to perform the monitoring of the patients are the main differences from previous initiatives and are the focus of this paper.

2.1 Proposed solution

This section describes the main characteristics of the HA DB, focusing on five main aspects: (1) its innovative DB structure that converts it into a clinical management tool while offering resources to store data and produce classifiers; (2) the use of a brain tumour domain ontology to permit the interoperability between different distributed DBs and other systems; (3) the use of multi-agent technology to perform all the distributed tasks: communication between nodes and data transferring and collaboration; (4) the integration of the DSS and its classifiers to perform automatic diagnosis; and (5) a set of extra tools that only the HA DB offers and which represents an added value to the DB itself.

2.1.1 Monitoring tool

The monitoring information is based on clinical disease histories (CDHs), medical controls (MCs), and medical experiments. This approach to storing and managing the data, and, consequently, the DB structure itself is an improvement on the previous schemas that only allow us to store the situation of a patient in a particular moment, excluding the possibility of performing data traceability and monitoring of the disease by only focusing on storing brain tumour data to create classifiers.

All data are stored anonymously and securely through the HA network of datamarts in order to create a distributed data warehouse. This data warehouse contains a collection of clinical data that have been properly anonymized, and all the other relevant information is acquired and stored at the various participating European clinical centres. This incipient network grants *bona fide* access to any qualified organization in return for its contribution of clinical data to the DSS. Personal patient information is kept confidential.

With this approach, the HA DB includes features to manage data, not only related to the classifier training, but also all the data and information related to studies and diagnosis, converting the application into a useful Web clinical management and monitoring tool for patients with brain tumour. All the details of the data and the DB structure are described in Section 3.1.1.

2.1.2 Data interoperability

The HA DB structure is mapped to the HA ontology of brain tumours (Croitoru *et al.*, 2007) and it enables an improved machine interpretability of Web content when compared to XML, RDF, and RDF schema. The ontology follows rigorous security protocols for clinical data exchange (Xiao *et al.*, 2008), permitting users to preserve their local centre policies for sharing information while allowing them to benefit from the use of a distributed data warehouse without losing the capacity to design local classifiers targeting specific patient populations.

The Semantic Web accessibility of the stored data takes advantage of some previous initiatives using agent technology and ontologies in the biomedical field (Merelli *et al.*, 2007; Turck *et al.*, 2007) and improves interoperability with other DBs and systems. It allows communication between the hospitals from the HA network and data to be accessed by agents.

With the use of the ontology, it is possible to perform queries to the DB manually and also automatically using specified agents. For example, to create new classifiers, a data collector agent will collect all the cases from the distributed DBs as soon as new data become available, boosting classifier production and reducing effort on the part of the manual.

The minimum number of cases necessary to produce a reliable classifier oscillates between 30 and 50 per tumour type, and there are over 130 brain tumour types (Manjón-Herrera *et al.*, 2004). Furthermore, clinical centres with a high capacity of data acquisition are able to collect between 50 and 80 cases per year. This implies that not only does the complexity of the production of classifiers increase dramatically, but it also opens up the possibility of creating multi-centric classifiers (Tate *et al.*, 2006). Acquiring data from different geographical areas adds reliability to the classifiers and improves their performance and, in consequence, the final DSS offers more accurate classifications to doctors during the diagnosis.

The use of an ontology to access the DB is indeed one of the major contributions of HA, as it introduces a novel approach of data sharing and distributed classifiers that determine the final HA DB design. The use of the HA ontology to access the data permits centres with their own DB schema to join the HA network after a DB mapping to the HA ontology.

2.1.3 Multi-agent technology

The use of agent technology offers an increasingly popular paradigm for the design and development of distributed systems, in which contributors need to communicate and discuss the information they exchange. Simlilarly, HA is particularly useful because it permits us to solve all the problems derived from the communication between nodes, data transfer, collaboration between centres, or the access to remote data and classifiers. This also expands the possibilities of HA from our initial approach based on a centralized DB (González-Vélez *et al.*, 2009).

The integration of the distributed agent framework and the DSS into the HA DB Graphical User Interface (GUI) converted the HA DB from an application to manage and retrieve brain tumour data to a complete system that can manage the brain tumour data but also to perform classifications to obtain diagnosis using the classifiers around the HA network.

It is possible to classify the stored data directly just by clicking a button and visualizing the results using the integrated DSS GUI. With this approach, the users from clinical centres do not need to use an extra application to perform the classification, as a unique system permits them to store, manage, monitor, and classify their patients' brain tumour data. The integration of the HA DSS system into the HA DB GUI is explained in further detail in Section 5.

2.1.4 Classifiers of the decision support system

The integrated HA DSS offers a wide range of classifiers to classify new cases of brain tumour. Particular types of tumours are described as having particular characteristics. A classifier can be created to present these characteristics, and when an unknown type of tumour case is presented to the classifier it can identify whether the unknown case is similar to the ones the programme 'knows'.

Having built the classifier, an estimate of its performance is made. This estimation is sometimes known in the literature as guessed performance and is generally carried out by using re-sampling techniques. This is a typical approach with cross-validation. As more data are acquired and stored in the distributed data warehouse, more classifiers can be built and trained. In that sense, the distributed multi-centre approach of the HA data warehouse enhances the data acquisition process and permits the building of more and more reliable classifiers. Currently, the system offers more than 15 classifiers. All these classifiers have been built using MRS data that have been shown as useful data for classifying brain tumours (García-Gómez *et al.*, 2008). The integration of the HA DSS into the DB GUI, its use, and how to obtain classification are described in more detail in Section 5.

2.1.5 Extras

The HA DB has a large set of integrated subsystems that makes it very interesting for clinical centres, such as a complete users' administration system implemented following strict security restrictions, an information report generator, a system installation verification tool, a software validation tool, a trouble ticketing facility for reporting potential problems, and an audit and back-up system to avoid data disasters. Section 3.2 describes each of these subsystems in more detail.

3 Methodology

The HA DB is a three-tier application that has a Web-accessible interface developed using Java Server Faces (JSF) and Ajax (Asynchronous JavaScript And XML) technologies. Altogether, these two technologies enhance the experience of the user in terms of human–computer interaction (Kiris *et al.*, 2007) giving a user-centred GUI. The three-tier architecture provides increased performance, flexibility, maintainability, reusability, and scalability, while hiding any unnecessary complexities of distributed processing from the user. These characteristics have made the three-tier architecture a popular choice for the Web. The three tiers are: *Web browser*, *Java middleware*, and *database server*, as illustrated in Figure 1.

- The *first tier* uses a Web browser to take advantage of the installed base of this universal client. An HTML (Hyper Text Markup Language) form is used for user input and the results of the DB queries are returned as an HTML page. Using HTML for user input and displaying the data lowers the requirement of the client's browser version, and using JSF technology the final step of rendering the HTML code is delegated to its rendered kit, avoiding problems of different browser characteristics and making development easier.
- The *second tier* is implemented via a Web server running all the Java code and the HTTP (Hypertext Transfer Protocol) requests on the server side. The Java Beans are able to access the DB and return the results to the first tier for its presentation. The Web server chosen for maintaining the HA Web system was Jakarta Tomcat.
- The *third tier* is the back-end DB server; the database management system (DBMS) chosen was MySQL.

The three-tier architecture allows us to establish the Model View Controller (MVC) software engineering pattern (Kwon *et al.*, 2004). The MVC is particularly appropriate for the design of DB Web applications because it isolates business logic, where the DB queries are generated, from the user interface, resulting in an application in which changing the presentation or visualization layer does not affect the underlying business rules and vice versa. While maintaining the consistency of the same presentation layer, it provides the possibility of using a different DBMS provided that changes are made to the logic layer to adapt the Structured Query Language (SQL) syntax for each SQL or ontology query language implementation (Singh *et al.*, 2002).

This architecture permits the modification of the business logic layer using a set of calls to the HA ontology and the subsequent retrieval of data from the ontology, without directly recourse to SQL queries or changes to the presentation layer. This is a key point for the final integration of HA network nodes, because after the DB-ontology mapping process all the HA nodes will use the HA ontology to communicate with each other. The fact that changing an approach based on SQL queries to a new architecture based on queries to the ontology only requires a single change of the business layer is of crucial importance.

3.1 Data entry: database and GUI

In the context of HA, we have envisioned two different scenarios depending on the clinical centre:

1. Centres with electronic data: HA allows the use of different DB schemas in each local DB, mapping a local DB schema to the HA ontology (Roset *et al.*, 2008). In other words, to join the

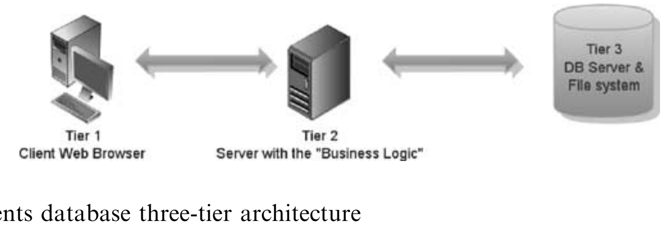

Figure 1 HealthAgents database three-tier architecture

HA network it is mandatory to perform a mapping process between the particular DB schema and the HA ontology. To facilitate this complex process, the HA project offers a software tool, called 'Map And Drop,' which graphically associates each DB table to its equivalent ontology concept. It is a manual process and requires the participation of experts in the field to determine the correspondence between the table fields and the ontology concept attributes.
2. Centres with paper-based data: HA offers the standard and open DB presented in this paper fulfilling all the HA requirements and already mapped into the HA ontology.

The HA DB is fully integrated within the rest of the HA system and, once all the data types have been mapped into the ontology, it is not necessary to perform any further mapping or conversion.

The data entry is more than a simple front-end for the data repository; it is a complete data management system that includes features to organize all the patients' data from a hospital, monitor their treatment, visualize spectra, display MRI images, generate activity reports, extract information to build classifiers, and use the integrated DSS to obtain an automatic diagnosis. It also allows new partners and contributing clinical centres to be added and offers a complete security system based on a range of user rights on a per centre basis. It is possible, for example, to assign edition privileges to one user over the data from a specific centre, or only download privileges over the data from another centre.

To ensure that no data are lost, either during transfers or because of any possible errors, a complete audit system is available throughout the HA data entry. It records all the actions performed, and if any of these actions produce some changes in the content of the DB the previous value is also stored. With this mechanism, it is possible to restore data from any given point in history, from a simple register to the whole DB.

3.1.1 Data and database

Here, we describe the data that can be stored in the HA DB, how it is stored therein, and the way it is presented through the GUI. It is important to emphasize that all sensitive data are fully anonymized before storing them in the DB. The HA system provides a set of specific client-side applications to anonymize all necessary data types. The anonymization tools are explained in more detail in Section 3.2.1.

HA encompasses two main groups of data—clinical and management data. The former represents patients' data coming from the clinical centres (shown in Figure 2) and the latter data to manage the DB. Obviously, the former is the most important, but without the latter the overall system cannot work properly.

1. Clinical data
   a Patient data: In this section, all the information related to the patient, for example, *age*, *origin*, *date of birth*, *date of death*, *gender*, or the *centre* to which they belong, is stored. In addition, the information about their clinical history and all the MCs made during the clinical studies that the patient has been the subject of are stored in this section.
   b CDH: When a patient comes to a hospital with some symptoms, a clinical disease history file is opened and a set of MCs are made to study the disease. Finally, the diagnosis is given and the CDH is closed. Each patient may have several clinical histories, as many as the number of diseases he/she has had. Each of these may have a different number of MCs associated with it. A CDH has a *start date*, an *end date*, and also fields to specify the reasons for the *opening* and *closure*. It is an entity to map the treatment of a disease, from its beginning to its end, and is used to group all the performed MCs until the final diagnosis.
   c MC: An MC is used to map every medical intervention with regard to a patient and is the main entity of the HA DB. These interventions can be MR experiments (single voxel (SV), multi-voxel (MV), MRI, and reference images), or biopsies (RNA micro arrays, HRMAS (High Resolution Magic Angle Spinning), and anatomopathologist images). An MC can be one or a set of these interventions. Each MC has a *start date* and an *end date* to specify the period when these experiments are performed and contains information about the initial and

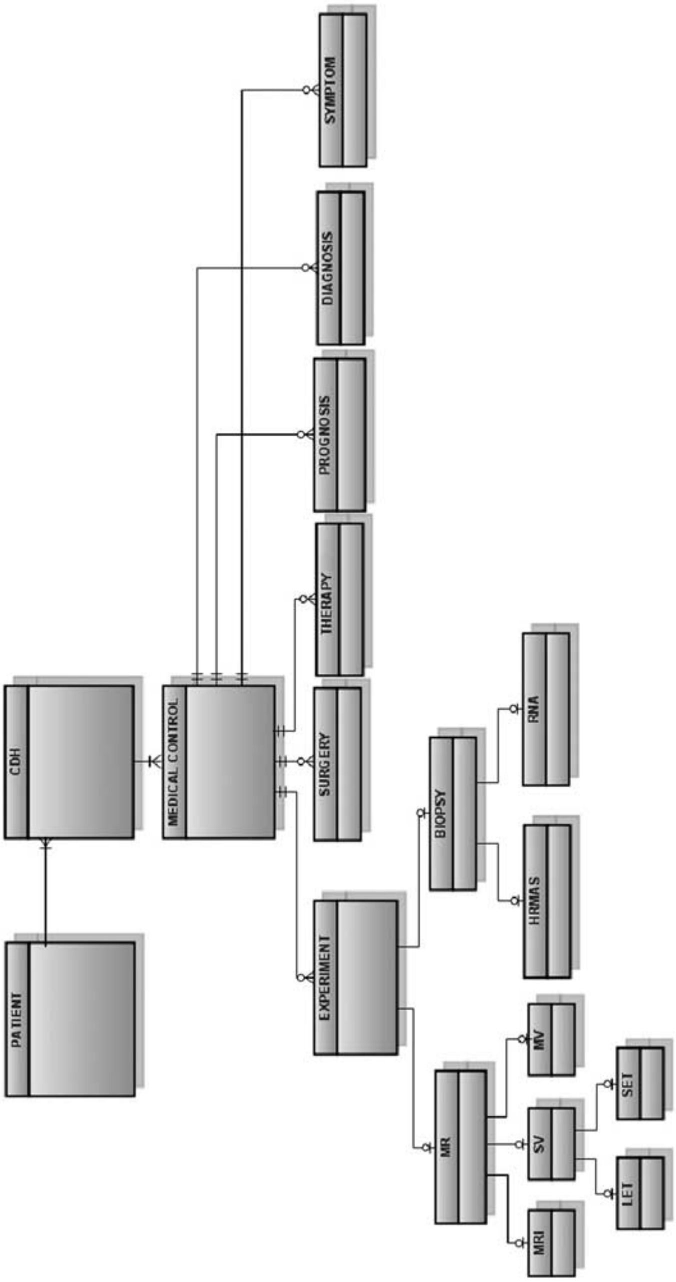

Figure 2 Schema of HealthAgents clinical data

final *observations* and other important information about the *status* of the patient during the whole period.

In addition, each MC can also register the symptoms, diagnosis, surgery (if any), prognosis, and therapies. It is therefore possible to display the status of a given patient at a particular moment.

The structured information in CDH and MC and the possibility of creating one or more experiment in each MC offers users the ability to organize their data according to their personal preferences. It is possible, for example, to create an MC for every experiment or to group a set of related experiments in a determined period of time.

For the design and development of the GUI, the recommendations from experts in the field of human–computer interaction, as well as the opinions from end-users were considered. With their opinions and feedback, it has been possible to design an application that permits us to easily display patient CDH and related MC. Users are able to follow on a patient's progress, global disease history, diagnoses, symptoms, and response to applied therapies.

2. Management data: These data contain tables to store the system users, partners, and centres. They also describe how to support extra functions such as data entry deployment and communication mechanisms for end-users, system administrators, and developers.
   a Users, centres, and partners: There is a specific set of tables to manage the users, centres, and partners of HA nodes. Partners are institutions that belong to the HA network and may have several associated clinical centres. The clinical centres can be any clinical site able to acquire data related to the HA system, and finally each user belongs to a clinical centre.

     The HA DB implements a complete security system to guarantee that the data are only available for the users (or agents) with enough rights to access it. The system offers the possibility of specifying a set of roles. Each role defines a profile that can be assigned to the users in the DB. This feature enables the possibility of having different types of users such as administrators, data loaders, clinicians, radiologists, and software validation users.

     In addition, for each user, and independently of their role, the system allows us to specify the types of privileges each user has for data coming from each specific centre. It is possible to specify four types of privileges for each centre; these are: creation, editing, visualization, and downloading.

     Apart from these data rights assigned to users, there is a system that informs the rest of the HA network whether the data from a given patient is public. If patient data are public, they can be used for training classifiers and be available for users in other centres; henceforth, the role and user right system is used to determine the level of privacy of the data, otherwise the patient data are private and are only accessible by users in the originating centre or node.
   b Metadata
     - News: Informs users of the latest updates or modifications to the HA system. The system administrators and the developers can add, edit, or delete these updates, and there is a set of parameters specific to each update; that is, availability of a new release, the section that it affects, the kind of announcement, etc. The update screen is the first page that the user sees when he/she logs into the system.
     - Master tables: Contain values that are used by other tables in the system, for example, MRS scanner manufacturers and their versions, the WHO (Smirniotopoulos & Meltzer, 2001) classification for brain tumours, countries, and tumour locations. Having these values in the DB, instead of in files or directly in the code, facilitates its maintenance and makes it easy to add, edit, or delete elements without changing the application implementation.

3.2 Data entry subsystems

3.2.1 Anonymization tools

One of the major concerns regarding HA is the ethical usage of data. This system is highly demanding in terms of security, confidentiality, and patient record linkage. Compliance to all

Table 1 Magnetic resonance scanners compatible with the HealthAgents anonymizer tool

Manufacturer	Version
Phillips	Gyroscan
Siemens	NUMARIS 3 (RDA, IMA) and NUMARIS 4 (RDA, IMA, DCM)
Siemens	MAGNETOM VISION VB33A/D/E/G
General Electric	SAGE
General Electric	Raw LX 5.x, 8.x, 9.x, 11.x, 12.x and 14.x PROBE

ethical regulations for data acquisition and management requires that all data included in every HA data repository outside the originating centre be dissociated, ensuring confidentiality and protection of the data.

Dissociation is a process by which personal, sensitive data are removed from the original source data files. Once this process is performed, data can be transferred without affecting the patient's right to privacy. In HA, the dissociation process is carried out by the anonymization tools. These remove all the data that could be used to trace the patient and subsequently identify them. With this approach, only totally anonymized data can be accessed outside the centre and the original patient information cannot be retrieved using only the linked data.

The process takes place locally at the originating centre and before any data transfer to the DB, as the anonymization tools are client-side applications. More specifically, these tools use Java Network Launch Protocol (JNLP) technology. The availability of anonymization software modules ensures that data will be anonymous before starting any data transfer.

The anonymization process removes the *patient name* and the *scanner patient ID* from all the raw data files coming from the manufacturers listed in Table 1. The HA DB application does not store any kind of personal data such as the address or postal code, and only date of birth, gender, and country are stored because these are important parameters for classification purposes. An internal HA identifier is created when a new patient is entered into the DB (e.g. 'ha0003'). The task of correlating these identifiers with the original patient register files is delegated to the hospitals and has to be performed outside the HA system. An example could be a simple list with the relationship between HA patients' IDs and real names. With this approach, it is impossible to correlate the data stored in the HA DB system to the patients' identity.

Depending on the manufacturer, the raw data can be composed of a different number of files. Normally, for example in the case of Phillips, there are separate files for the MR experiment information: *examination name*, *date*, *position*, *orientation,* and the binary data of the spectra. There are other manufacturers, such as General Electric (Fairfield, Connecticut (United States)), which provide all this information in the same file, mixing the experiment information (in a header) with the spectra (in binary format after the header section). The first step to anonymize the files then is to detect the manufacturer type. After that, the anonymizer performs the operations, taking into account the file specifications and removing the patient's personal data. The anonymization process is a complex task, and the details of the whole process are beyond the scope of this paper, which focuses on the DB system.

3.2.2 Automatic parameter extraction tools

HA has a set of automatic parameter extraction tools to read information from the raw data files of the different experiments and to fill this information automatically into the DB fields. This process avoids manual mistakes when the user is filling in data and reduces the time taken to insert a new case. Currently, HA has automatic parameter reading for SV, MV, HRMAS, and RNA data. It is still not possible to read all the data for all the files from the existing manufacturers and versions, but this forms part of the ongoing work that depends on manufacturer specifications and the availability of accurate documentation on the different formats. Fields such as the *date of measurement*, *manufacturer*, *version*, *sequence*, *magnetic field*, *echo time*, *repetition time*, *size*, *number of spectral points,* or *spectra width* are examples of automatic read parameters.

This process takes place at the same time as the anonymization; therefore, it also uses JNLP, which is a client-side application, but it is fully integrated within the rest of the HA data entry. As it is a task related to the anonymization process, it is possible to extract the parameters from the same manufacturers described in Table 1.

3.2.3 Magnetic resonance spectroscopy processing tools

HA uses tools to transform the original MR SV raw data into a readable spectral format that the classifiers use as a part of their input to determine the tumour type of a patient. This tool, the Data Manipulation Software (DMS; Tate *et al*., 2006), was developed during the INTERPRET project and has been integrated into the HA data entry.

Currently, we are exploring other alternatives, that is, the use of the JMRUI (Java Magnetic Resonance User Interface) tool (Naressi *et al*., 2001) as a middle layer inside the MRS conversion pipeline process. For this reason, the DMS was updated to permit the reading of files pre-processed using the JMRUI tool. This allows us to focus development purely on the last part of the MRS conversion pipeline, avoiding the tedious task of format conversion from different raw data versions, and therefore we only need to work on the transformation of the JMRUI output to a standard MRS format used to build classifiers.

Therefore, the HA system now allows the direct upload of raw data, processed with the DMS for all the compatible formats, and JMRUI data. In both cases, the data are converted into the same MRS format, a list of 512 floating points of the spectrum that the system is able to display in the GUI. A simple linear interpolation is used to produce the same spectral resolution for the Phillips and Siemens spectra, as both formats have a different spectral width and number of points. Intensity values between 4.2 and 5.1 ppm in the normalized spectra are set to zero in order to remove the effects of remaining unfiltered water signal (Tate *et al*., 2006). After the uploading process, the HA DB anonymously stores the original raw data (or JMRUI) and the files from the processing process (e.g. spectrum file).

The resulting spectrum, combined with the contextual clinical data such as age, gender, and tumour location, is the most important parameter to classify a case and build classifiers for brain tumours.

3.2.4 Magnetic Resonance Image conversion modules

HA allows the uploading of MRI images. Usually, these images come in Digital Imaging and Communications in Medicine (DICOM) format, but this is non-Web compatible and also changes according to each version and manufacturer of MR scanners. HA has two options to visualize the MRI images, the first one is an adapted ImageJ[1] Applet application that permits the loading of a remote ZIP file containing anonymized DICOM images. With this adapted tool, it is possible to use all the ImageJ operations and features, increasing the possibilities for the final HA DB user.

The other option is to convert the DICOM images to a Web-compatible format. For this purpose, the free software image converters ImageMagick[2] and the DCMTK–DICOM[3] are used. The original DICOM images are converted to the PNG (Portable Network Graphics) format on the server side, and can then be visualized on the Web browser. It is important to remark here that the original images in the DICOM format are also stored within the system, fully anonymized, and totally available for the users because these kinds of files contain more information than just the image.

3.2.5 Other subsystems

There are complementary tools that facilitate the enhancement of the system and also assist the daily work of system administrators and system developers.

Some of these subsystems benefit from the distributed HA approach; and even though the subsystems are presented in the local DB GUI installation as an additional section, internally they

[1] http://rsb.info.nih.gov/ij/

[2] http://www.imagemagick.org/

[3] http://dicom.offis.de/dcmtk.php.en

work using a remote central DB. They use this central DB, accessible to all HA DB installations, because its scope is not local but global, via the HA system. Without a centralized approach, the global information would have to be replicated on each node, making it difficult to manage and duplicate data across the network. By centralizing common services (i.e. the trouble ticketing, the software validation, and the system installation verification tools), the distributed nodes can access the same resources from everywhere, and management becomes easy because the system administrators only have to use one system. Nonetheless, the backup mechanisms and the audit system have local scope, and all their necessary tables are contained in every local HA schema.

- Trouble ticketing: A ticketing system to report and fix application bugs. It improves the communication between end-users and developers and boosts the improvement of the final HA data entry version. It is necessary that all HA network users can see all the reported bugs from other nodes because the aim of the trouble ticketing tool is to provide a mechanism to fix bugs in the HA system, not only bugs in local installations. From the developers' perspective, it would be impossible to manage bugs for each node separately with a non-centralized approach, and probably users from different nodes would find and report the same bugs, adding redundancy to the system and duplicating the developers' work. With the centralized approach, all the HA Data entry installations have the tool fully integrated in its GUI. All the actions regarding the trouble ticketing section are performed to the centralized DB, instead of the local DB. This process is totally transparent to the final user, and it does not require any extra effort from him.
- Software validation: Tool for system quality control management. It is used to verify that the system does what is intended to, fulfilling the initial requirements and ensuring that the users are able to work with it easily. This tool is quite versatile, as it allows the use of different methodologies to validate the software such as Technology Acceptance Model (TAM) (Lederer *et al.*, 2000) or an internal one proposed by expert partners of the HA consortium. This subsystem is fully integrated within the Data Entry and allows the validation experts to introduce all the items to be checked (i.e. upload and process a SV case, perform a query or login). Later on, users are able to check whether the specified items work properly, allowing the possibility of providing feedback about the user's experience during the item validation. This information is stored and the experts can evaluate the results using a complete set of reports with statistics and graphical charts. The idea of this tool is to provide a homogeneous environment to validate the HA system using online tools, avoiding the use of paper forms, e-mails, and other less optimal techniques, and ensuring that the system fits with initial user requirements. This distributed approach indicates that every HA user is able to validate the same items and that the validation process can be carried out at the same time for all the nodes, instead of having an individual validation for each node. From the user's point of view, this centralized approach is transparent and it does not differentiate whether he/she is working with the local DB or with the centralized one.
- System installation verification tool: Application that verifies that all the deployed software that the HA data entry requires (Web application, DB, agent framework, DMS and MRI processing modules) is correct. In addition, it compares the current release with the latest release in the server to provide information as to whether the local system needs an upgrade, as well as providing updates with an easy and faster mechanism. All the nodes perform these operations using a unique central point that is only administrated by the HA developers. With this approach, when a new version is available for the nodes, the HA manager updates the central DB with the proper information, and all the HA installations can see immediately that a new version is available and download the resources to update its system.
- Back-up mechanisms: There are tools to back-up the data directly from the data entry GUI. This option is only available for the nodes administrator users and the back-up files (in ZIP format) are stored in a location previously specified. It is a local service.
- Audit system: System that tracks and analyzes any DB activity including DB access and usage, data creation, change, or deletion. It reports every change to the DB content and stores all the

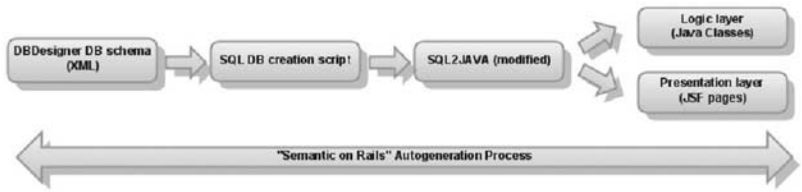

Figure 3 Autogeneration process pipeline

actions performed by a user during his/her visit to the Website. It permits the recovery of any previous data if a disaster occurs. This system works locally.

4 Development

4.1 Autogeneration process

The DBDesigner[4] tool was used to design the HA DB schema, and to export it as an SQL script for the DB creation. DBDesigner is a visual DB design system that integrates DB design, modelling, creation, and maintenance into a single, seamless environment and is optimized for the open-source MySQL DB. It is an open-source program, and this fact allows us to modify it and combine it with SQL2JAVA[5] to complete the pipeline from the design of the DB schema in the DBDesigner until the creation of the Java classes and JSF pages with the modified SQL2JAVA. In the HA system, SQL2JAVA is used to adopt the MVC pattern and to use JSF and AJAX technologies in the presentation layer.

Modifying this program, and combining it with the already modified DBDesigner, we created a new program: the *Semantic On Rails*. This application, developed using Java technologies, permits the creation of a complete Web application to manage a DB, taking the output of the DBDesigner as the input of the modified SQL2JAVA. It includes all the possibilities offered by DBDesigner and SQL2JAVA with the added advantage of allowing the creation of user-customized templates to generate the Web front-end based on the DB tables. We adopted a template design based on JSF, because it was the adopted technology for the project, but it is possible to use any other Web technology such as Java Server Pages, Active Server Pages, or Hypertext Preprocessor.

SQL2JAVA offers a powerful DB layer API (application programming interface) to access a given relational DB through a JDBC (Java database connectivity)-compliant driver. Therefore, using Semantic On Rails, it is possible to develop a Web application to manage an Oracle, SQL server, or MySQL DB just by editing a properties file. (Figure 3)

A unique generator process was used to simultaneously generate a set of JSF pages (presentation layer) to manage the java classes (business layer) from the graphical user interface. With this approach, from the original DB schema, we obtained a set of java classes to manage it and a set of JSF pages to manage the interaction between the user and the business layer.

4.2 Database layer

The global HA DB is split into several sections in which each corresponds to one domain and contains all its related tables. There are sections for patient management, clinical trial, and MC management for each of the medical experiments that can be carried out. These isolated sections meet the requirements of each different centre and constitute the data warehouses of the system to be managed through the HA data entry interface.

The derived logic and presentation layers to manage the DB also follow the same DB schema section structure, generating different packages to manage themselves.

[4] http://www.fabforce.net/dbdesigner4/

[5] http://sql2java.sourceforge.net/

4.3 Logic layer

As a result of the autogeneration process described in Section 4.1, we obtained all the necessary Java classes to manage the DB from a DB schema designed with the DBDesigner tool. This approach guarantees the following points:

- Configuration: Automatic mapping of the DB fields.
- Database: Provides support for most versions of MySQL and other DBMS.
- Identity: Retrieves autogenerated keys.
- Transaction: Convenient transaction handling. Uses data source or simple Driver getConnection.
- Performance: Performs update/save only when needed. Uses prepared statements so that it is not necessary to re-parse the queries every time.
- Configuration**:** Includes a listener for life-cycle events that permit operators to add the code before and after inserting, updating, and deleting DB operations. This feature allows us to implement the HA audit system.
- Manipulation: Requires minimal knowledge of SQL.
- Relationship: manages one-to-one, many-to-one, and many-to-many relationships.
- Extension: both the generator and the generated code that is template based are easily extendible.

Some of these characteristics are key points of the HA data entry system. For example, the option to extend the initial capacities of SQL2JAVA allowed us to create JSF and Java templates and simultaneously generate the business and presentation layers. Each table has a set of automatically generated classes and a set of implementations of them. These two sets of classes allow a group of classes in which the developers can include their code without facing the problem of overriding existing code when a change to the DB requires a new autogeneration process. With this approach, the classes that the developer must edit are all included in a subpackage called *IMPL*. All the classes of this package have a suffix *Impl* (e.g. *PatientTblImpl*). Another good characteristic of this approach is that it separates the autogenerated code from the developer code, offering the option of splitting the code into different kinds of classes depending on its purposes.

4.4 Presentation layer

JSF (Holmes & Schalk, 2006) was chosen as the most suitable technology for the presentation layer because it guarantees the MVC design pattern and allowed us to develop a system separating control logic (the system knows what, but not how), business logic (knows how), and presentation logic (knows how to interact with the users). It also facilitates the reuse of components, gives some guidelines, and provides inner functionalities such as libraries, sub-applications, and configuration files among others.

The HA DB GUI has different modules. All the common parts for several pages are externalized separately and are then included in the desired pages. With this approach, if it is necessary to customize certain components of a page all the application pages in which they are included change automatically. This feature makes the application easy to maintain and reduces programming effort. It is a system that also allows changes to the structure of the whole application only by changing the position of the components included within the page code. For example, if we change the position of the 'include element', which contains the main menu, this could appear horizontally on the top of the page, vertically on the left, and so on. Examples of this modularity are the menu, header, footer, the 'includes' with table lists, search options, or the forms to add data. All the HA JSF pages are simply the result of a rendered set of includes.

A tool to autogenerate the logic layer classes and the JSF pages was developed. The resulting autogenerated JSF pages have the following structure: for each DB table, a set of six JSF pages are generated automatically. These pages will not build the GUI for themselves, but are the basis of it. Some of the six pages are sections to be included in the main screens and have the following purposes:

1. xxxADD: Adds elements to the table XXX, where XXX represents a table name in the DB. It contains an 'include' of xxxEditPG file and two buttons—one to save the data and another to

cancel the operation and to go back to the previous screen. When the user clicks the save button, a method of its associated Java Bean that calls SQL *insert* is fired.
2. xxxEDIT: Edits previous elements of the table XXX. It contains an 'include' of xxxEditPG file and two buttons: one to save the new data and another to cancel the operation and to go back to the previous screen. When the user clicks the save button, a method of its associated Java Bean that does an SQL *update* is fired.
3. xxxEditPG: Included in xxxADD and/or xxxEDIT, and contains the form used to specify the values of the XXX table.
4. xxxList: Shows a list of elements in the table XXX. It contains 'includes' of xxxFilteredList and xxxFilterAndActions files to construct the final page.
5. xxxFilteredList: Features a data table JSF component that shows a sortable list of registers from the xxx table.
6. xxxFilterAndActions: Contains common operations performed by a list of elements such as add and filter by some text.

4.4.1 External files

All the resources that the HA data entry requires to work properly are externalized. Examples of these resources are the Java script files, the CSS files, the bundle files with the different language character literals, or the properties files with system configuration parameters. Having all the resources outside improves the non-dependency of the code and makes it easy to change only the desired files. The CSS and the bundle files are clear examples that HA has implemented a CSS file style that defines the styles and colours for all the components that appear in the application: tables, images, input text, texts, lists, and so on. If a centre requires a customization of the user interface, it is possible to make the modification just by changing the CSS (Cascading Style Sheets) file because of which the whole application will change, offering a completely different look and feel.

The bundle files are used to store all the application character literals and strings. By using them, the GUI presentation layer has no literal specified in the code. All of them are references to properties of the bundles. When these properties are modified, the values displayed change. The bundle files are also used to translate the application into several languages. Currently, HA is fully translated into English, Spanish, and Catalan, but could easily be extended to other languages by adding further bundle files. Each file contains the same properties, but with different values depending on the language.

5 Integration within the rest of the HealthAgents system

The main objectives of the HA project were: (1) to provide a DB that can store brain tumour data in order to create classifiers and with features to organize and manage patient information; and (2) to provide a DSS to classify and perform the diagnosis of brain tumour cases to help the clinicians in the brain tumour diagnosis process.

Initially, we considered developing two different systems for each of these objectives: the HA DB and the HA DSS, but when the HA DB development was in its final phase it was decided to integrate the agent platform and the HA DSS into the HA DB GUI. All the systems were implemented using Java technologies: the agent framework over a JADE platform and the DSS as a J2EE application and consequently their integration into the HA DB GUI was feasible.

The HA agent framework was packed in a set of JAR (Java ARchive) libraries. By adding these libraries to the HA GUI application, it was possible to use all the framework features, such as creating agents, requesting classifiers, or obtaining diagnosis results from any agent across the distributed network. With the inclusion of the HA DSS as a Java Applet in the HA DB GUI, it was possible to perform classifications and obtain diagnosis results within the same application. Both objectives were achieved in a single system, avoiding the need to use two different systems, thus reducing the time needed to obtain a diagnosis and converting HA DB GUI into the first application in the world to offer a Web- accessible brain tumour DSS to the scientific community.

To link the HA DB GUI to the rest of the HA system, the *agentGUI* was developed. This agent is responsible for performing all the actions requested from the GUI that require interaction with the rest of the HA system; it is the common interface between the data entry and the framework. The access to the DB is performed using the *DBAgent*, included in the HA framework that provides ontological access to the content of the DB using a mapping file. It is responsible for performing the translation between the ontology language and the SQL language and facilitates interoperability within the HA system because the information of any DB can be accessed using the HA ontology. It also avoids the need for any additional knowledge about the specific data structure of the different DBs.

From the HA DB GUI, it is possible to run the HA DSS in two different ways. The first one involves simply selecting an already processed file from the local disk and execute the DSS. The second one involves classifying the stored data by clicking some of the direct accesses through the User Interface.

This system allows users to identify a classification of a particular case by creating a question-to-solve. The question is created by identifying the constraints that would limit the possible tumour types that the patient could have. As a minimum, the question consists of a possible tumour type, but could also include contextual patient data such as gender, age, or geographical area. Once the data to classify are selected, the system searches for classifiers able to answer this question. These are presented to the user. Once the candidate classifiers are selected, the HA system executes every candidate. Candidate results are ranked. The ranking is based on the similarity of the patient data to the data the classifiers were trained on, the performance evaluation it achieved when created, or the usefulness and the accuracy reported by users so far during their use of the classifier as a classification tool. When the classification is performed, a list with all the results from the classifiers is shown. For each classifier result, it is possible to see the spectra from the classified case in comparison to the means of the classified tumour type result. A chart with the accuracy probabilities, a representation of the classifier performance and other information to help the clinicians to make a decision about the diagnosis is also presented in the same results GUI. Figure 4 shows the HA DSS results GUI integrated within the HA DB.

6 Results

The facilities that the HA DB offers that were considered in the initial requirements are to create and list patients, to organize their information with CDHs and MCs, to store all the clinical and management data that has been explained in the previous sections, and to use the necessary conversion and anonymization tools to guarantee the security and privacy of the data. In addition to this, at the end of the project, all the complementary subsystems described in Section 3.2.5 were added. The integration of these tools represents an added value to the DB and differentiates it from other existing clinical DBs that only focus on data management.

The possibility of using the DB as a central and independent node at the same time as using it as a distributed node of the HA network represents a clear advance from previous approaches, but the most important feature with regard to similar brain tumour DBs is the possibility of performing classifications and obtaining automatic diagnosis using the integrated HA DSS.

A detailed explanation of the HA system: the agent framework, the ontology, the DSS, or the techniques and algorithms to create and train the classifiers beyond the scope of this paper and are covered in (González-Vélez *et al.*, 2009).

This paper focuses on the DB, and this section describes the results of this system and the possibilities that it offers to the scientific community.

Figures 5 and 6 show some of the features in the current GUI.

6.1 Relevance of the results for the scientific community

The possibility of having a free tool to organize and manage their clinical information is of great interest to clinical centres. There are a lot of centres using old mechanisms to store patients'

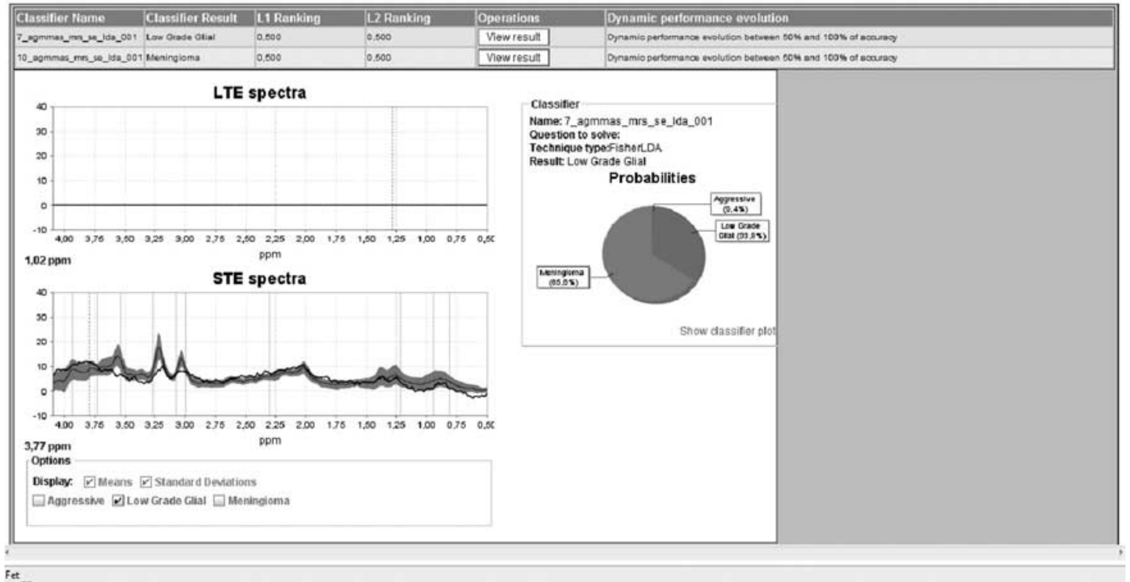

Figure 4 The HealthAgents decision support system integrated within the HA DB GUI

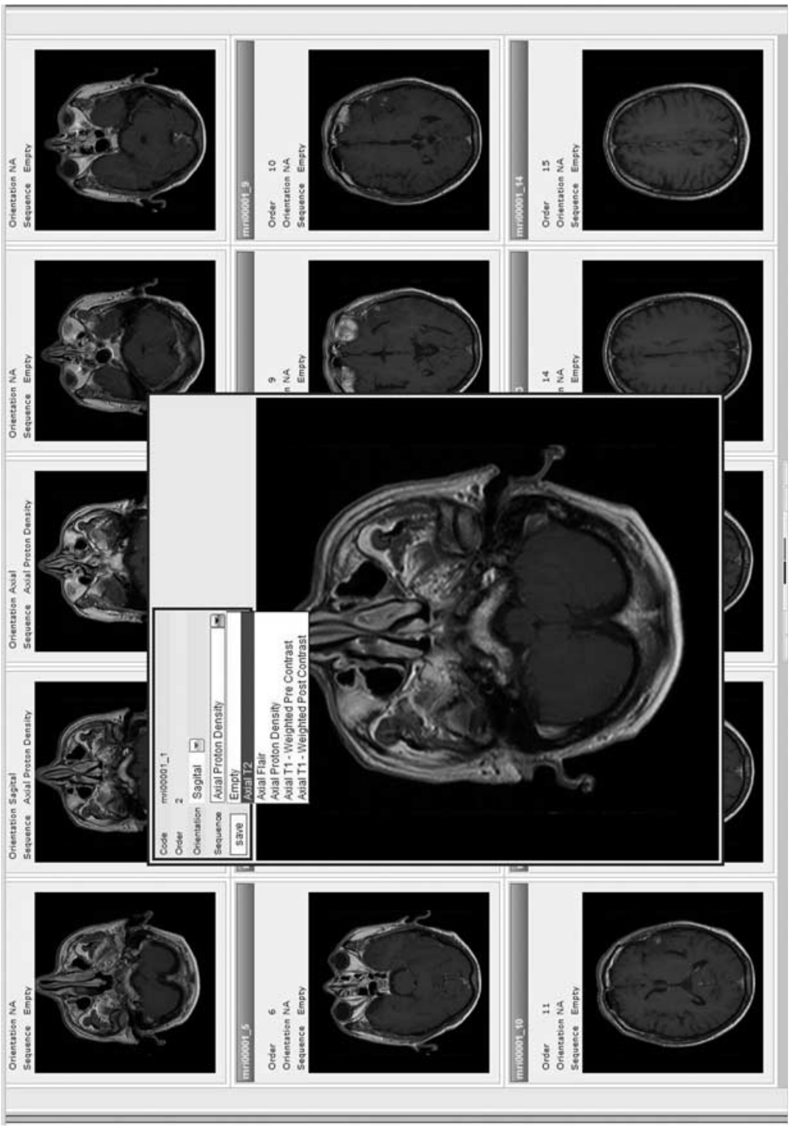

Figure 5 HealthAgents magnetic resonance imaging section

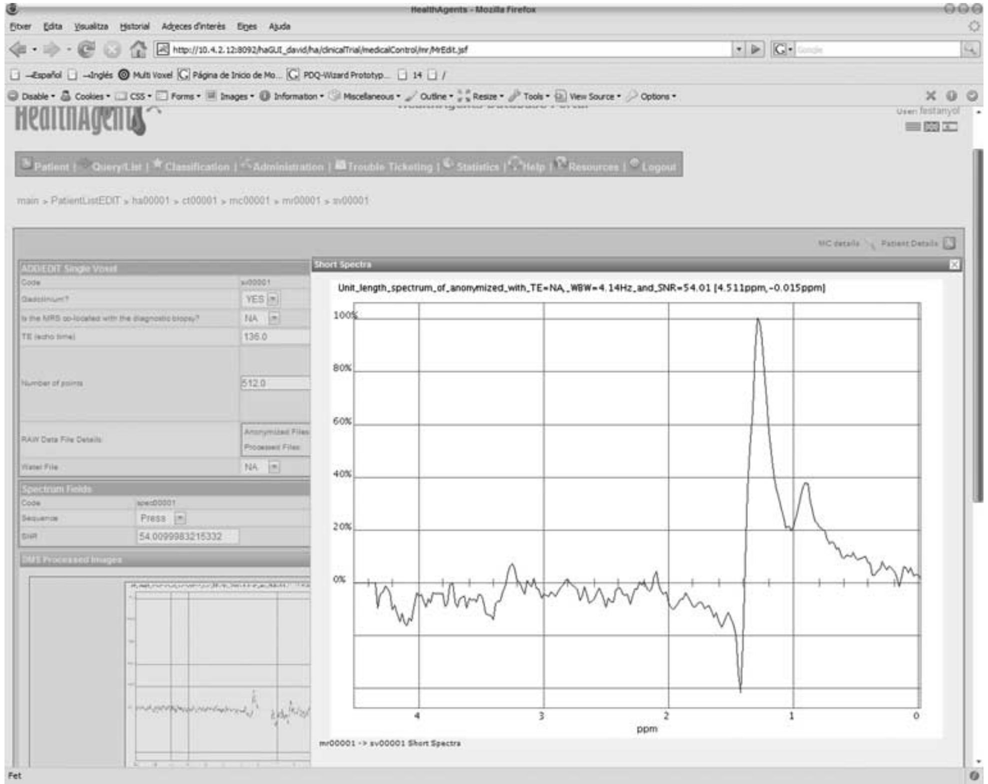

Figure 6 HealthAgents magnetic resonance spectroscopy single-voxel (SV) section (SV spectrum is zoomed out)

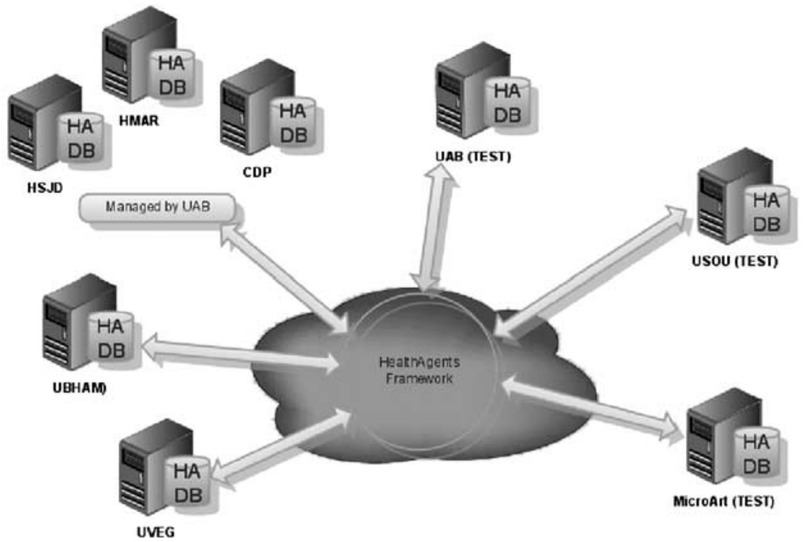

Figure 7 Nodes with the HealthAgents database deployed

information such as paper-based ones. Some of them are using electronic formats, such as Microsoft Excel files or small DBs implemented with Microsoft access or custom DBs developed by non-expert users. Only a small group of hospitals are using robust and Web-accessible DBs to store information related to patients with brain tumour (Alger *et al.*, 2002; Galperin, 2008).

However, the major interest of the centres is to access the HA DSS and obtain an easy tool to help the clinicians make their diagnosis. Most other DBs do not contain all the fields and features of HA (as mentioned previously though, HA is compatible with any other DB after DB-ontology mapping). From our experience during the project, clinical centres consider our DB as a real asset in organizing and managing their clinical data and also appreciate the benefits of joining the HA network. (Figure 7)

The compatibility of the HA DB with other subsystems, its distribution under free software licences, the possibility of customizing the DB and the GUI code to specific needs, and the possibility of accessing an international network of centres with accurate classifiers make the HA DB a good option for those centres with brain tumour patients.

At present, the HA data entry has already been deployed in the following nodes: Universitat Autònoma de Barcelona and its associated hospitals, which are Hospital St. Joan de Déu, Hospital Clínic Corporació Sanitària, Centre Diagnòstic Pedralbes, Institut de Diagnosi per la Imatge de l'Hospital Vall Hebron, Hospital de Bellvitge, and the Hospital del Mar. The University of Birmingham is currently acquiring child data from 22 hospitals of the Children's Cancer and Leukemia Group. This node is currently storing all their data acquired within the HA DB, but as they have their own DB (National Health Service, UK) they are mapping it to the ontology to be able to use the HA system with their own storage tools. In addition, there is also a node in Universititat de Valencia, which is acquiring MRS data from the following hospitals in the Valencia area: La Ribera-Alzira, Instituto Valenciano de Oncologa, and Grupo Hospitalario Quiron, and finally two test nodes: MicroArt and the University of Southampton. Finally, we would like to mention the Jiangsu Province Hospital, China, which is installing the HA DB and will join the existing network soon.

In spite of the HA and the HA DB still being under development, some data were acquired to create and train new classifiers, validate the system, and test their integration within the HA framework.

Once the HA DSS was integrated into the HA DB GUI, a prospective study to demonstrate that the system performs the requirements of the HA project and to verify that the system provides added value to the clinicians' decision-making process was performed.

The study involved 26 respondents, of whom 12 were female and the remainder were male. Their average age was 39 years and the average amount of clinical experience was 13 years.

All were involved, as a major part of their practice, with caring for patients with brain tumours, and the group included radiologists, oncologists,, and histopathologists. An evaluation of the results, based on TAM questionnaires, showed that the clinicians responded positively and could see the benefits of the HA system in supporting the diagnosis and prognosis of brain tumours within their working environment.

However, the relevance for the scientific community goes further than its medical advantages. Facilitating the integration of new applications was a key issue to guarantee the project's success. In that sense, all the technologies and architecture adopted facilitate the integration of complementary tools. During the project, different systems have been integrated into the HA DB to test and verify that the system offers enough facilities to easily integrate subsystems that will improve its original functions.

The INTERPRET DSS was the first third-party application integrated into the HA data entry. It was decided to integrate the DSS to prove the utility of the DB in storing data to build classifiers and to classify patients' brain tumour data with an external and already validated DSS.

Other functions that have been added to the system and that have benefitted from its technological approach are the collaborative agents (Rafael-Palou & Rovatsos, 2009) or the Evidence-based Search Service. The first, included in the HA framework, provides an autonomous and collaborative approach in the classification learning process that attempts to provide new classifiers with improved classification performance. The second is a system developed by the School of Informatics of Edinburgh University, which, when given search words, consults scientific literature DBs (e.g. PubMed or Google Scholar) to retrieve related articles.

The possibility of integrating third-party systems in the HA DB GUI would expand its initial capabilities and permits the scientific and research communities test, in a real and deployed application, the latest advances and theories from different scientific fields.

7 Conclusion

Although the HA project is still under development, most of the initial objectives have been achieved. There is a DB able to store all the required information to build classifiers and improve the diagnosis of brain tumours using a distributed DSS. This DB has been installed in several clinical sites successfully and they are currently finishing the validation and testing phases.

The HA DB goes a step beyond previous and similar DBs. The GUI uses cutting-edge technologies such as JSF and AJAX to provide a more user-friendly experience than the previous ones and offers a wide range of services that the others do not have. It is more than a simple data repository; it is a complete brain tumour data management system that permits its users to develop tasks related to the DSS. The integration of the other subsystems is another added benefit that makes HA data entry the most innovative DB for storing brain tumour data.

Another important feature is its DBMS. Previous initiatives were developed as a hierarchical flat file system DB, with all the inconveniences that this causes. Others used commercial DBMS, with the derived high costs for participants. Some of these features were not critical in centralizsed DBs. In such cases, costs were shared by all participating hospitals, but in the case of HA, where each hospital may have its own DB, these solutions are unfeasible. In that sense, having an open and free DB that is continually improving and which easily integrates with DSSs and extra tools makes the HA DB an excellent option for those centres interested in the HA network (or in only having a tool to store their information, as it is possible to install it without the DSS and the HA framework).

As the HA DB could be installed in any centre around the world, the internationalization feature that HA provides is a key aspect for its success. It is the only DB for brain tumour data that offers three languages from its installation. Adding a new language is an easy and fast process; it only requires editing a new bundle file with the keywords and translate the literals to the desired language. This task can be carried out by someone at the node, and it does not require any intervention on the part of the HA DB development team.

It is important to remark that there is a lot of future work to do on the HA DB. The development of these kinds of systems is an ongoing task and new features may be added as extra tools. The idea is to offer a tool that will satisfy the preferences of clinical centres. With this in mind, new customization and adaptive features will be developed in the near future. For the scope of the project, the initial objectives have been more than achieved, and the current system is able to be installed in clinical centres around the world.

Acknowledgements

Many people have participated in the research and development of different aspects of the work described in this paper. In particular, the authors thank Dr Bo Hu and Dr Madalina Croitoru for the development of the HADOM ontology; Dr Liang Xiao for the security model in the HA DSS; Dr David Dupplaw for the HA Framework—without whose key contribution, nothing else in the project would have been possible. The authors also thank Juan Miguel García-Gómez and Javier Vicente from ITACA, who are in charge of the development of the Classifier Agents; Carlos Sáez for the DSS GUI development; and Dr Margarida Julià-Sapé and Dr Ana Paula Candiota for their key collaboration on the DB schema design from a radiological and clinical point of view. Raw data conversion modules were kindly provided by the FAST consortium in the framework of the scientific cooperation between both projects; we particularly thank Prof. Danielle Graveron and Dan Stefan for their support on this issue. DMS routines were kindly provided by the University of Sussex, and further adaptations of DMS and also of jMRUI output to DMS format are credited to Guillem Mercadal and Iván Olier. Finally, we would also thank Prof. Paul Lewis for reviewing this article and his key contribution to the project's success by way of global scientific coordination; and to Chris Roland for his exhaustive English revision.

This paper reflects only the authors' views. This has been carried out within the EU FP6 Project HealthAgents: Agent-Based Distributed Decision Support System for Brain Tumour Diagnosis and Prognosis (IST-2004-027214).

References

Alger, J. R., Frew, A. J., Cloughesy, T. F., del Vecchio, W., Villablanca, J. P. & Curran, J. G. 2002. Novel methodology for the archiving and interactive reading of clinical magnetic resonance spectroscopic imaging. *Magnetic Resonance in Medicine* **48**(3), 411–418.

Arús, C., Celda, B., Dasmahapatra, S., Dupplaw, D., González-Vélez, H., van Huffel, S., Lewis, P., Lluch i Ariet, M., Mier, M., Peet, A. & Robles, M. 2006. On the design of a web-based decision support system for brain tumour diagnosis using distributed agents. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 208–211.

Blaschke, C., Hirschman, L. & Valencia, A. 2002. Information extraction in molecular biology. *Briefings in Bioinformatics* **3**(2), 154–165.

Chiang, I.-J., Shieh, M.-J., Yung-Jen Hsu, J. & Wong, J.-M. 2005. Building a medical decision support system for colon polyp screening by using fuzzy classification trees. *Applied Intelligence* **22**(1), 61–75.

Croitoru, M., Hu, B., Dasmahapatra, S., Lewis, P., Dupplaw, D., Gibb, A., Julià-Sapé, M., Vicente, J., Saez, C., García-Gómez, J., Roset, R., Estanyol, F., Rafael, X. & Mier, M. 2007. Conceptual graphs based information retrieval in HealthAgents. In *CBMS '07: 20th IEEE International Symposium on Computer-Based Medical Systems*, IEEE Computer Society, 618–623.

Exarchos, T., Giannakeas, N., Goletsis, Y., Papaloukas, C. & Fotiadis, D. 2007. A framework for cancer decision support based on profiling by integrating clinical and genomic data: Application to colon cancer. In *8th International Workkshop on Mathematical Methods in Scattering Theory and Biomedical Engineering*, 261–268.

Galperin, M. Y. 2008. The molecular biology database collection: 2008 update. *Nucleic Acids Research* **36**, D2–D4.

García-Gómez, J., Luts, J., Julià-Sapé, M., Krooshof, P., Tortajada, S., Vicente Robledo, J., Melssen, W., Fuster-García, E., Olier, I., Postma, G., Monleón, D., Moreno-Torres, A., Pujol, J., Candiota, A., Martínez-Bisbal, M., Suykens, J., Buydens, L., Celda, B., Van Huffel, S., Arús, C. & Robles, M. 2008. Multiproject-multicenter evaluation of automatic brain tumor classification by magnetic resonance spectroscopy. *Magnetic Resonance Materials in Physics, Biology and Medicine* **22**(1), 5–18.

Golodetz, S., Voiculescu, I. & Cameron, S. 2007. A proposed decision-support system for (renal) cancer imaging. In: *FBIT '07: 2007 Frontiers in the Convergence of Bioscience and Information Technologies*, 361–366.

González-Vélez, H., Mier, M., Julià-Sapé, M., Arvanitis, T. N., García-Gómez, J. M., Robles, M., Lewis, P. H., Dasmahapatra, S., Dupplaw, D., Peet, A., Arús, C., Celda, B., Van Huffel, S. & Lluch-Ariet, M. 2009. HealthAgents: distributed multi-agent brain tumor diagnosis and prognosis. *Applied Intelligence* **30**(3), 191–202.

Holmes, J. & Schalk, C. 2006. *JavaServer Faces: The Complete Reference*. McGraw-Hill Osborne Media.

Jarman, I. H., Etchells, T. A., Martín, J. D. & Lisboa, P. J. G. 2008. An integrated framework for risk profiling of breast cancer patients following surgery. *Artificial Intelligence in Medicine* **42**(3), 165–188.

Julià-Sapé, M., Acosta D., Mier M., Arús C., Watson D. & The INTERPRET consortium 2006. A multi-centre, web-accessible and quality control-checked database of in vivo MR spectra of brain tumour patients. *Magnetic Resonance Materials in Physics, Biology and Medicine* **19**(1), 22–33.

Kiris, E., Abrams, H. & Longoria, R. 2007. User-centered design: Component-based web technology. In *HCI (11),* Lecture Notes in Computer Science **4560**, 114–122. Springer.

Kwon, K., Kim, N., Cheon, S. & Choi, H. 2004. A framework for developing distributed application with web-tier object modeling. *MWSCAS '04: 47th IEEE International Midwest Symposium on Circuits and Systems* **3**, 251–254.

Lederer, A. L., Maupin, D. J., Sena, M. P. & Zhuang, Y. 2000. The technology acceptance model and the world wide web. *Decision Support Systems* **29**(3), 269–282.

Mamrak, S. A. & Yates, A. J. 2006. An information system to support collaborative brain-tumor research. *Brain Tumor Pathology* **14**(3), 131–137.

Manjón-Herrera, J. V., Martínez-Bisbal, M. C., Celda, B., Martí-Bonmatí, L. & Robles, M. 2004. SIView 2.0: A new MR spectroscopy imaging tool. *European Radiology Supplements* **14**(2), 300.

Martínez-Bisbal, M. C., Martí-Bonmatí, L., Piquer, J., Revert, A., Ferrer, P., Llácer, J. L., Piotto, M., Assemat, O. & Celda, B. 2004. 1H and 13C HR-MAS spectroscopy of intact biopsy samples ex vivo and in vivo. *NMR in Biomedicine* **17**(4), 191–205.

Merelli, E., Armano, G., Cannata, N., Corradini, F., d'Inverno, M., Doms, A., Lord, P., Martin, A., Milanesi, L., Möller, S., Schroeder, M. & Luck, M. 2007. Agents in bioinformatics, computational and systems biology. *Briefings in Bioinformatics* **8**(1), 45–59.

Naressi, A., Couturier, C., Devos, J. M., Janssen, M., Mangeat, C., de Beer, R. & Graveron-Demilly, D. 2001. Java-based graphical user interface for the MRUI quantitation package. *Magnetic Resonance Materials in Physics, Biology and Medicine* **12**(2), 141–152.

Radhouene, A. M., Sophie, S., Anne, F., Frederic, P., Olivier, P., Aron, M.-B. & Dawn, T. J. 2009. Knowledge-based expert systems and a proof-of-concept case study for multiple sequence alignment construction and analysis. *Briefings in Bioinformatics* **10**(1), 11–23.

Rafael-Palou, X. & Rovatsos, M. 2009. Collaborative agent-based learning with limited data exchange. In *AAMAS 2009: 8th International Conference on Autonomous Agents and Multiagent Systems*, 1191–1192.

Roset, R., Lurgi, M., Croitoru, M., Hu, B., Lluch-Ariet, M. & Lewis, P. 2008. A visual mapping tool for database interoperability: the HealthAgents case. In *CS-TIW 2008: 3rd Conceptual Structures Tool Interoperability Workshop*, Springer-Verlag, 1–10.

Ruggiero, C., Giacomini, M., Bonadonnac, G., Valagussa, P. & Demicheli, R. 1998. The database design for a decision support system for breast cancer chemotherapy. In *20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* **3**, 1230–1233.

Shortliffe, E. H., Scott, A. C., Bischoff, M. B., Campbell, A. B., Melle, W. V. & Jacobs, C. D. 1981. Oncocin: an expert system for oncology protocol management. In *IJCAI: 7th International Joint Conference on Artificial Intelligence*, William Kaufmann, 876–881.

Singh, I., Stearns, B. & Johnson, M. 2002. *Designing Enterprise Applications with the J2EE Platform*. Addison-Wesley Longman.

Smirniotopoulos, J. G. & Meltzer, C. C. 2001. The new WHO 2000 classification of brain tumors: Imaging correlations. URL `http://rad.usuhs.mil/`.

Stumpflen, V., Gregory, R. & Nenova, K. 2007. From biological data to biological knowledge. In *TMRA'06: 2nd International Conference on Topic Maps Research and Applications*, Lecture Notes on Artificial Intelligence **4438**, 62–66. Springer-Verlag.

Tate, A. R., Underwood, J., Acosta, D. M., Julià-Sapé, M., Majós, C, Moreno-Torres, A., Howe, F. A., van der Graaf, M., Lefournier, V., Murphy, M. M., Loosemore, A., Ladroue, C., Wesseling, P., Bosson, J. L., nas, M. E. C., Simonetti, A. W., Gajewicz, W., Calvar, J., Capdevila, A., Wilkins, P. R., Bell, B. A., Rémy, C., Heerschap, A., Watson, D., Griffiths, J. R. & Arús, C. 2006. Development of a decision support system for diagnosis and grading of brain tumours using in vivo magnetic resonance single voxel spectra. *NMR in Biomedicine* **19**(4), 411–434.

Turck, F. D., Decruyenaere, J., Thysebaert, P., Hoecke, S. V., Volckaert, B., Danneels, C., Colpaert, K. & Moor, G. D. 2007. Design of a flexible platform for execution of medical decision support agents in the intensive care unit. *Computers in Biology and Medicine* **37**(1), 97–112.

Van Diest, P., Belien, J., Zanstra, P., Wilhelm, W. & Baak, J. 2007. Integrated decision support system/image archive for histological typing of breast cancer using a relation oriented inference system. *Histopathology* **25**(3), 253–259.

Xiao, L., Lewis, P. & Gibb, A. 2008. Developing a security protocol for a distributed decision support system in a healthcare environment. In *ICSE'08: 30th International Conference on Software Engineering*, ACM, 673–682.