The Knowledge Engineering Review, Vol. 26:3, 261–282. © Cambridge University Press, 2011
doi:10.1017/S0269888911000117

The design and implementation of a novel security model for HealthAgents

LIANG XIAO[1,*], SRINANDAN DASMAHAPATRA[1], PAUL LEWIS[1], BO HU[1,†], ANDREW PEET[2], ALEX GIBB[2], DAVID DUPPLAW[1], MADALINA CROITORU[3], FRANCESC ESTANYOL[4], JUAN MARTÍNEZ-MIRANDA[4], HORACIO GONZÁLEZ-VÉLEZ[5] and MAGÍLLUCH I ARIET[4]

[1]*Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK;*
e-mail: lx@ecs.soton.ac.uk, bh@ecs.soton.ac.uk, dpd@ecs.soton.ac.uk, sd@ecs.soton.ac.uk, phl@ecs.soton.ac.uk;
[2]*School of Cancer Science, College of Medical and Dental Sciences, University of Birmingham, Birmingham BIJ 2TT, UK;*
e-mail: acpeet@doctors.org.uk, a.j.gibb@bham.ac.uk;
[3]*LIRMM, 161 rue ADA, F34392 Montpellier Cedex 5, Montpellier, France;*
e-mail: croitoru@lirmm.fr;
[4]*MicroArt, Parc Cientific de Barcelona, Baldiri Reixac 4-6, 08028 Barcelona, Spain;*
e-mail: rroset@microart.eu, mlurgi@microart.eu, jmartinez@microart.eu;
[5]*School of Computing and IDEAS Research Institute, Robert Gordon University, St Andrew Street, Aberdeen AB25 1HG, UK;*
e-mail: h.gonzalez-velez@rgu.ac.uk

Abstract

In this paper, we analyze the special security requirements for software support in health care and the HealthAgents system in particular. Our security solution consists of a link-anonymized data scheme, a secure data transportation service, a secure data sharing and collection service, and a more advanced access control mechanism. The novel security service architecture, as part of the integrated system architecture, provides a secure health-care infrastructure for HealthAgents and can be easily adapted for other health-care applications.

1 Introduction

In a distributed collaborative health-care environment, multiple clinical organizations from geographically different sites, each having its own users, resources, and access policies may be involved in the delivery of health-care services. Access provided to clinicians who use such systems from their own sites must be provided securely.

We take the view that security concerns, particularly in a distributed environment, must be integrated into the design stage of a system, or else the integrity and usability of the system may be critically compromised. We note that there are several aspects to the security challenges that need to be overcome in a distributed health-care system. First, in the scenario exemplified by our particular 'HealthAgents' system, no global user repository will be available for distributed authorization. Clinical centres may join or leave independently. The management and administration of resource access will have to be decentralized in the network, in which each site maintains its own users and resources to be accessed. Second, although access control becomes complicated

* This author is now at the Royal College of Surgeons, Ireland. e-mail: liangxiao@rcsi.ie
† This author is now at SAP Research. e-mail: bo01.hu@sap.com

in a distributed environment, we shall bear in mind that a significant improvement in clinical decision making will be predicated by enabling many hospitals and users to join the federated system and share their knowledge. Third, in such an open environment, health-care records containing sensitive private information must never be disclosed, even to collaborative centres and friendly clinicians, except for health-care purposes, and under the conditions agreed upon. Lastly, and this brings a different degree of complication, our consideration of access control shall not be restricted to users who want access, nor to sites that want to be accessed. Instead, more refined access conditions such as access user type or group, individual case-based access, or even run-time access context will be taken into account. In addition, all of the above security controls must be checked, not only against human users initiating interactions with the system, but also software agents that perform the tasks in an autonomous way. This is an added complexity induced by the nature of agent-based systems, particularly those in which the source code is made open.

The paper is structured as follows. The remainder of this section analyzes security needs in health-care systems in general. Section 2 describes the particular health-care system, HealthAgents, as our running example. Section 3 provides a literature review and analyzes the weaknesses of the existing security approaches. In Section 4 an overview of our model of security layers is presented with justification for its design decisions. Implementation of various security services of the model are discussed in the sections that follow. Section 5 distinguishes the different resources, the attributes of which determine whether or not they are sharable in the network. The use of a link-anonymized data scheme for protecting patient privacy in the architecture is depicted, conforming to ethical regulations. Section 6 describes a secure data transportation service. Section 7 discusses a secure data sharing and collection system. Section 8 illustrates a more advanced security model including role-based access control (RBAC) and other powerful features and Section 9 concludes the paper.

1.1 Security and health care

We begin by drawing distinctions between the types of threats health-care systems face and the likelihood of their occurrence. Although eavesdropping or hacking is a major concern to computer network security, it is so expensive that dedicated and capable intruders may consider using a more convenient way. It is more likely that improper design or use of the system may lead to privacy being compromised and the leaking of confidential information. This is often caused by pre-assumed design decisions regarding how the system will work and may conflict with the manner in which end users work with the system in practice. It is the interaction between humans and computers that has been accredited as the root cause of the security problem, and actually hackers pay more attention to the human link in the security chain than do security designers (Smith, 2003). If security analysis is always restricted to computers but not to the human processes and users, problems will continuously occur. For example, password sharing is typical among general practitioners (GPs) because doctors do not want to wait for the system to switch between accounts. If inappropriate privileges are bestowed on unwanted users, their behaviour is not traceable by the system. This implies that, apart from the necessary education on security for end users, a system should be well designed not only to protect the communicating sites and end users, but also to carefully authenticate and authorize users who have institutionally approved rights to have access to specific information, without exposing the additional information under protection. This security need has currently not been well addressed in health-care information systems (Zhang *et al.*, 2002). In the following, we outline the challenges and common security requirements of health-care systems, in which privacy and confidentiality must be maintained in an open and distributed access environment.

1.2 The distributed health-care environment

Aggregating dispersed data into large databases is expensive and practically infeasible, as geographically different health-care centres have to have control over their data sets and at the same

time maintain a globally consistent data schema. A more important reason to oppose data consolidation involves health-care data confidentiality. In the United Kingdom, for instance, the National Health Service (NHS) attempted to build a unified electronic patient record system to enable easier central administration and better information availability by giving access to the extended NHS community. This has been opposed (Anderson, 2001) because such a system, which collects data from existing GP systems but is out of their control, is in conflict with the ethical principle that no patient should be identifiable other than to their GPs unless consent is given (GMSC/RCGP, 1988). Results from a survey indicate that most patients are unwilling to share their information with the NHS (Hawker, 1995). Another objection arises from the overwhelming workload that such a centralized system could possibly put upon a security officer responsible for managing the data sharing (Zhang *et al.*, 2002).

A distributed health-care service infrastructure, however, promises the ability to cope with the administrative burden and the continuous maintenance needs arising from fully functional and networked clinical centres, each of which has its own users, data, access policies, and which assumes that cross-centre access is the norm. A distributed environment and its associated dynamics bring other concerns to the information-sharing health-care network, such as preserving the patient's privacy.

2 The HealthAgents project and the agent architecture

The HealthAgents project (González-Vélez *et al.*, 2009) is creating a multiagent distributed Decision Support System (d-DSS) to help determine the diagnosis and prognosis of brain tumours using non-invasive techniques. Brain tumours are an important cause of morbidity and mortality (Bray *et al.*, 2002), and there is a need to improve their classification and management. Novel medical imaging techniques such as magnetic resonance spectroscopy (MRS) provide information on the spectral (metabolite) content of neural tissue, and laboratory techniques such as gene expression arrays provide additional correlated information to histopathological studies of surgically extracted tissue. These techniques promise to deliver these advances but suffer from a paucity of extensive case studies to enable reliable interpretation, which has hindered their incorporation into routine clinical practice. The new techniques provide an excellent test bed for the development of a computer-aided decision support system. Furthermore, the rarity of many brain tumour types requires that information must be sought from many hospitals to increase the evidence base upon which signals from MRS can be correlated with gene expression and histopathological validation. The use of a distributed system for data collection and management is therefore a necessity.

The HealthAgents system uses a set of distributed nodes that either store patient case data, build classifiers trained on case data that are capable of classifying tumour types, or use the results of classification algorithms to aid clinical procedures for the diagnosis and prognosis of brain tumours. The MRS data used by the system are built up using anonymous information from child and adult cases. Classifier agents encapsulate pattern recognition modules and are created at specialist nodes that receive requests from the clinicians to generate classifiers for particular tumours. Clinicians will use classifiers to assist in the differential diagnosis of patients for particular tumours. The HealthAgents system consists of a variety of agents each charged with a different task. In the current state of deployment, the main sites are located at the University of Birmingham in the United Kingdom with 20 different contributing centres, and in Spain at the Universitat Autónoma de Barcelona with six centres, as well as at the Universitat de Valencia with four centres.

Figure 1 shows a schema of the distributed architecture of the HealthAgents d-DSS. Each clinical node, as part of the inter-networked system, can represent either a user where requests for classification of a given case are initiated, or a producer where classifiers are created or retrained based on pattern recognition techniques, or both. In any case, they all contribute their data for the training of classifiers. New classifiers may be produced or existing ones improved when new cases become available, because of the growth of data in existing centres or the participation of

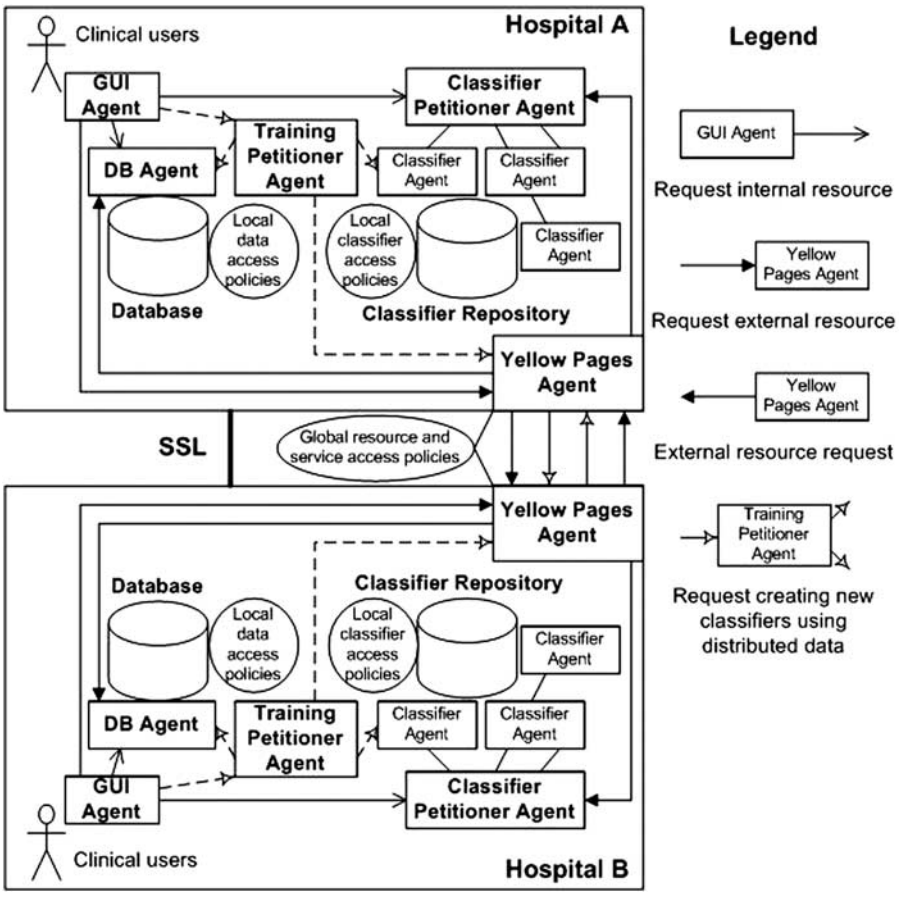

Figure 1 The distributed architecture of the HealthAgents system and its resource access flow control

a new centre. When a clinical user requests the classification of a case that resides internally, its associated GUI (Graphical User Interface) Agent will retrieve the patient data from the local hospital database via a Database Agent, and local data access policies will be applicable. Alternatively, if the case under classification resides externally, then the GUI Agent will contact the local Yellow Pages Agent to find an appropriate Database Agent elsewhere on the network from which patient data are retrieved, external data access policies being applicable. One Yellow Pages Agent resides in each hospital's local node. They synchronize with each other and together maintain a directory of available nodes and agents, as well as the classifiers for the entire HealthAgents network. Global resource and service access policies will apply when (1) cross-centre resource access is requested by an agent and (2) global services such as the query service provided by the Yellow Pages Agent are requested.

Once a case has been loaded into the GUI application, it may be classified. The local Yellow Pages Agent has registered within it classifiers that can discriminate among tumour classes, including descriptions of their capabilities, reputation, and the training data with which they have been produced. The clinician may ask the Yellow Pages to search for the existing classifiers which can solve various questions. These questions are related to the patient's condition, including, for example, whether the tumour state is aggressive or non-aggressive, or the specific type of cancer. The Yellow Pages Agent looks up its local registry, contacts external Yellow Pages Agents, and compiles a list of appropriate classifiers. This list is returned to the clinician, and the clinician can now send the list of selected classifiers that can solve questions, accompanied by the patient data for these classifiers, to the Classifier Petitioner Agent. The Classifier Petitioner Agent will invoke and provide patient data to each Classifier Agent associated with the classifiers in the list. Internal or external classifier access policies will apply, depending upon the location of the classifiers.

Although this may involve remote classifier access, which gives the system a sense of full distribution, in practice, once a classifier is produced a copy might be obtained by every node in the network for local classifier running and better performance.

After the execution of classifiers, classification results are collected by the Classifier Petitioner Agent from multiple classifiers and ranked using performance statistics, and finally sent back to the clinician. The clinician can now do the diagnosis, supported by the answers and recommendations provided by the system. Eventually, when the diagnosis is completed, the clinician evaluates the classification results produced by the selected classifiers, and their reputation is updated. The above scenario assumes that classifiers exist to solve the questions. If no such classifier exists, a clinician requests the Training Petitioner Agent to create one using data from distributed sites and registers the new classifier in the Yellow Pages Agent for later use.

3 Literature review of existing security approaches for health-care information systems

Agent technology is promising in both the building of health-care information systems and ensuring their security. On the one hand, agents have the capability to represent the different services required by the system, providing the framework and functionality to ensure the distribution of data, and offering intelligent answers to the demands of the users. On the other hand, their abstraction of different processes in which resources are accessed can be under security control if appropriate measures are imposed upon them. Several approaches that use agents in health-care domains for providing security have been investigated.

The concept of heuristic security agents has been introduced in a scheme (Keese & Motzo, 2005) in which all calls to files, networks, library modules, and components, as well as other resources, are intercepted. They are checked in a 'sandpit' against behavioural rules before an 'allow' or 'deny' decision is made, preventing the entire classes of attacks to health-care information systems.

Security concerns have also been raised regarding the private patient information sharing among interconnected hospitals. Secure access of distributed electronic health-care records (EHR) has been considered in Gritzalis and Lambrinoudakis (2004). A scheme is proposed that uses a security agent per site, which authenticates users and controls the access to the local resources by observing user roles. The dedication of an agent for the full security control of each site suffices for the protection of a simple resource type of patient records from a single point of access. However, this approach will expose its insufficiency in three ways: (1) when multiple resource types are available, each corresponding to a responsible party in an individual's site, (2) when some common services are shared amongst multiple sites, and (3) when the differentiated access privileges of each user are necessary at the same time.

Another approach to the similar problem of exchanging private patient records among distributed hospitals introduces a four-tier architecture, a central access control (CAC) system, and multiple local access control (LAC) systems sitting between the client application and hospital information systems (Choe & Yoo, 2008). CAC and LAC are multi-agent systems (MAS) that use authentication agents, encryption agents, and access control agents. Multiple LACs enable hospital managers to maintain their distinct access control policies over patient records. The single CAC serves as a communication hub, establishing secure communication networks with each LAC so that data access requests can be forwarded amongst LACs and actual data can be passed among them in a secure manner. In this architecture, the security level is determined by the weakest LAC, and the central CAC may impose a performance bottleneck and a single point of failure to the entire system.

All the above methods introduce agents or MAS explicitly for the purpose of access control. Security was not considered as part of an integrated software design by software engineers in the first place. It has been shown in the Agent.Hospital framework (Kirn *et al.*, 2003) that it is feasible and beneficial to use MAS as well as ontology technology for modelling and integrating existing individualized health-care processes into distributed decision-making processes. This provides

improved assistance for enabling diagnosis and subsequent treatment plans for cancer patients. The addition of security-specific agents will impose extra design requirements for existing health-care system implementations, and will require a runtime communication overhead in addition to the load of maintaining the security components. Our hypothesis is that MAS will be most effective in securing a health-care information system if its participant agents serve core clinical functions with associated security measures or policies that serve as behavioural constraints to their normal function in the clinical setting. In doing so, functionality and security are integrated into a single architecture; however security policies can be separately maintained, thus improving the software design and the resultant application.

4 Design decisions for the HealthAgents security model and its service overview

In view of the various weaknesses of existing approaches, a new comprehensive security framework must be purpose-built for the HealthAgents system, which embraces the security characteristics and requirements from health-care information systems. A design decision of such a security framework needs the broad consideration of the security objectives or requirements of information systems in general, as well as the specific needs of the HealthAgents system. National and international standards and recommendations defining security requirements provide the guidance we need.

First of all, the Common Criteria (CC; CCRA, 2006), standardized as ISO/IEC 15408, is a framework in which security requirements can be specified by users. Security attributes can be implemented and claimed by vendors, and such features and claims can be evaluated by a testing group. CC's security functionality requirements include *communication/non-repudiation*, *cryptographic support*, *user data protection*, *identification and authentication*, and *privacy*.

According to the IEEE Std 730.1-1995 (IEEE, 1996), a software security plan must be in place to address the way in which software and data will be protected. It has been suggested that the plan should include the following:

1. how the data should be classified and communicated (e.g. 'no trespassing' messages);
2. how the users of the software access the application and how that access is to be controlled;
3. network design;
4. user identifications, passwords, security logging, and auditing.

Finally, the National Institute of Standards and Technology (NIST, 2006) requires that information systems be categorized by assigning impact values to the security objectives of *confidentiality*, *integrity*, and *availability*. From these security objectives, a number of security areas have been identified to meet the minimum security requirements, including *identification and authentication*, *access control*, *communications protection*, and *system and information integrity*. These security objectives are termed as CIA Triad (confidentiality, integrity and availability; Pfleeger & Pfleeger, 2002). One needs to identify methods that may result in the breaking of the CIA Triad: *confidentiality*, being concerned about unauthorized access to private information; *integrity*, being concerned about the creation, change, or deletion of data without authorization; and *availability*, being concerned about the loss of control over the functioning system and its security measures.

Obviously, such national and international standards agree on the security metrics to some extent and share some common characteristics. The existing HealthAgents architecture, as described in Section 2, which distributes data and services, already helps to maintain some of those objectives and metrics. First, the *integrity* among hospitals is sustainable as individual centres can retain the control over their local patient cases and the policies for sharing them, the responsibility for overall data protection being spread. In addition, the *availability* of the system is leveraged with some built-in fault tolerance in place. An example is an interconnected network: when one node is down, requests for a service (e.g. classification) can still be fulfilled because of multiple copies of classifiers being available across centres. Furthermore, the shift to classifier access from patient case access, which is now usually limited to the principal treating doctor and the classification software, should help to improve the *confidentiality* of individual patient privacy.

Yet, a complete secure framework demands identifying potential scenarios wherein the security requirements and CIA Triad could be broken in the context of all use cases. Below is a table of the complete cases we have identified in HealthAgents.

Planning security measures and making design decisions for adding security to the system involves the avoidance of potential security breaches and the placing of the relevant solutions outlined above. Overall, the design of such a framework takes into account the nature of the data to be exchanged, the channels by which they may be exchanged, the senders and receivers, and the more flexible policy options the data controller may hold towards sharing their data with data consumers. Clearly, as we go each step further, the access constraints become tighter and the system becomes more secure. According to the categories given by the international standards and our analysis of the HealthAgents security requirements, we design our security model in the following layers:

Security violation	Broken security requirement	CIA Triad	Consequence	Solution
Theft and disclosure of patient privacy information by a hacker due to insecure transportation network	Transportation and identification	Confidentiality	Patient privacy compromised	1. Remove the sharing of patient identifiable information if unnecessary 2. Add message encryption to the transportation layer of the HealthAgents, in addition to an authentication system for user identity recognition
Abuse of system services (Yellow Pages, Classifier Training, etc.) by hackers, making them unavailable or replace them with malicious alternatives	Transportation and identification	Availability and integrity	System services becoming unavailable or directing wrong diagnosis	2. Add message encryption to the transportation layer of the HealthAgents, in addition to an authentication system for user identity recognition
Malicious users may create low-quality classifiers	Identification	Integrity	Misleading decision support	2. Add message encryption to the transportation layer of the HealthAgents, in addition to an authentication system for user identity recognition
Accidentally, valid but inexperienced users may assign unreasonable reputation values to classifiers	Authorization	Integrity	Misleading diagnosis results because of incorrect alteration of classifier attributes	3. Set up mutual access agreements between partners with regard to the access of resources by valid users
Users from one hospital access data or execute classifiers from another hospital without the proper permission	Authorization and access control	Confidentiality and integrity	Patient privacy compromised and unwanted information or service disclosure	4. Allow resource owners to define comprehensive security policies to represent their sophisticated control policies for access to their resources across the network

1. The first consideration is distinguishing *the types of information that can possibly be made sharable* in the first place, according to ethical regulations and the resources present in the architecture with their global attributes and access context. Direct case-based sharing is subject to stricter regulations, whereas classifiers, which are trained upon cases but contain no private patient information, are the main resource shared for decision support. Cases are even distinguished between public and private ones to further constrain their access. It will be assumed, for the purpose of this paper, that the only data that are shared for classifier building cannot be used to reconstruct the patient's identity from it. Section 5 describes this layer in detail.

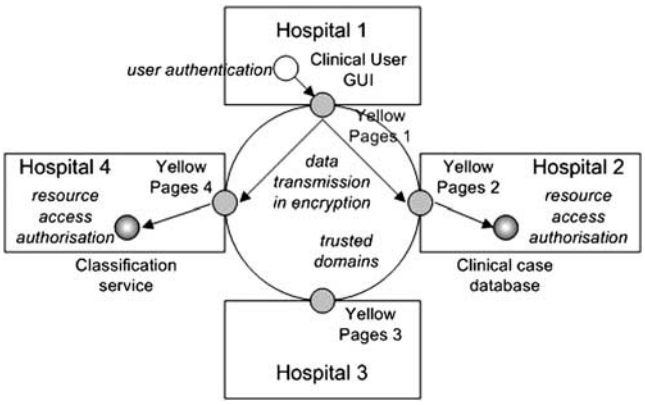

Figure 2 Overview of the security model layers using a cross-hospital resource access scenario

2. The next layer of security implementation comes in *the encryption and decryption at the transportation level*, which has a global effect on all messages passing through the network, the layer aiming at preventing invalid users from any access. All messages passing through the HealthAgents network must be encrypted. Secure transportation applies to the entire network. Section 6 discusses this perspective in more details.
3. Further, *mutual access agreements* must be set up between pairs of partners, the layer aiming at preventing valid users from unauthorized access. Individual case access or data collection must respect data owners' regulations against users from other HealthAgents sites. Basic yes/no permissions can be defined to restrict access across sites, based on user origin, request data residence, and case publicity. One user from site A may be allowed to access a data set marked as public located in site B only if the administrator at site B gives the user from site A such a permission (possibly), after a request has been made. Section 7 discusses this perspective in detail.
4. Although simple yes/no permissions can be defined for mutually agreed access arrangement, and stored in a global repository for independent but separate management, more comprehensive but generic permissions have to be defined. Such permissions need to be represented in an explicit form of *access control policies*, stored across data owners' sites and subject to their continuous review and configuration, where fully distributed control is required. Users may be grouped into certain types (e.g. according to their positions in clinical centres, clinical roles, and workgroup memberships) and access policies applied to them (e.g. perform certain operations in certain contexts). Section 8 discusses this perspective in more detail.

In Figure 2, an overview of the layered security model is illustrated, in which a clinician from hospital 1 retrieves and classifies a case from hospital 2 using a classifier from hospital 4. In this scenario, the case will be anonymized and made public before it is shared by hospital 2. The clinician must be authenticated in hospital 1 before access to the local network is granted, after which all messages passing through for the classification purpose in the interconnected HealthAgents network will be encrypted. It must be agreed, between hospital 1 and hospital 2, that the case under discussion can be shared and between hospital 1 and hospital 4, and that the classifier under discussion can be shared, prior to the case from hospital 2 being classified by the classifier from hospital 4 at hospital 1. More restricted security policies can be defined to govern the authorization of specific resource access by specific users.

We allow agents, as building blocks of the system, to function as system components that are capable of providing various data and services in the architecture, and at the same time these same agents control secure access to the resources. The information exchange will be encrypted and decrypted by agents automatically at the transportation layer, and agents respect the agreements and policies handed over to them by data controllers, when the data and services are being used. Therefore, the architecture reduces the extra complexity added to the system, which is present in

the existing approaches described in Section 3, where security engineering was not considered as part of the software requirements engineering. In the approach proposed, functional and non-functional requirements will be met together by the agents, and no extra agent layer working upon the existing system will be introduced to achieve security. In the next sections, we discuss the layers of the security model.

5 The use of a link-anonymized data scheme for information sharing

Before incorporation into clinical practice, new methods must be fully tested within a clinical trials setting, which is the context for the HealthAgents system. Such trials are subject not only to data protection laws, but also to regulations governing clinical trials, including ethical approval and informed consent of the participants. For multi-national projects, ethical approval is devolved to regional bodies without any coordinated or uniform decision making and so data gathered from different centres may be subject to different restrictions. Allowing for flexibility within the data security model is therefore essential.

Clinical trials are usually supported by a centralized database, the data from which personal information (e.g. name, address, date of birth) is removed, but to which a unique patient identifier is added, often termed *link-anonymized* data. This allows the patients to be reassured that their data will be afforded a high level of security, and allows regulatory bodies ease of access to inspect the processes in place. Such a scheme has the advantage of having a high chance of preserving patient anonymity while allowing data from the same patient to be added at a later date. This scheme also allows a specific patient's data to be located and removed from the project at any time they request, a condition usually imposed by ethics committees. Full patient records are maintained for clinical purposes within the treating hospital, and with the patient's permission may be used to generate and periodically update the clinical trial data.

For a distributed system, similarl robust arrangements must be designed to reassure ethics committees and patients that the data are secure. However, achieving this is a significant challenge and here we discuss a potential model. Each data collecting centre could have an associated link-anonymized database as approved by their appropriate ethics committee. Patient identifiers could then be maintained along with the clinical patient record in the treating hospital. These databases need to be the only databases kept within the system, giving a truly distributed data warehouse. The limited data required for analysis could then be subject to stringent anonymization processes and sent to a small number of specific sites for processing, for example, the production of tumour classifiers. In this way, the distributed nature of the system could be preserved while allowing appropriate regulatory access to data repositories. Additional security measures will need to be in place, which can allow each centre to potentially limit the type of data transmitted and the locations it is transmitted to.

Figure 3 shows such a data transmission model in a multilayered manner. Although complete patient records may only be accessed by hospitals and local nodes, link-anonymized records may be exchanged between a limited number of centres producing classifiers. Furthermore, only limited amounts of totally anonymized data may be accessed outside the closed project network. It is important to point out that there is no universal consensus on whether it is possible to rule out the identifying ability of individuals from the data used to classify tumours. Although this is not a concern within the project, our layered security arrangements are designed to systematically undo degrees of access that are switched on once decisions have been made at a regulatory level about the extent to which individuals can be identified using (aspects of), for example, MRS or gene regulatory data.

Apart from the link-anonymized data scheme, the system turns the majority of resource sharing for decision making from direct case-based comparison to classifier production and running, which offers a further level of protection to patient privacy. In the system, cases are gathered by particular sites in order to produce tumour classifiers. Cases are only known to the classifier producer software and it is the classifiers and not the cases that are used for decision making. If no

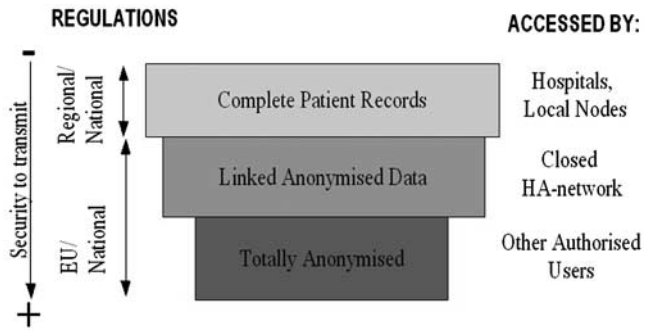

Figure 3 The link-anonymized data transmission model

such classifier is available, a new one may be produced using the available cases. Therefore, private patient data that are involved in the production of classifiers will normally not be available or accessible to clinical users. When new clinical centres join the existing collaborating centres, they can immediately start to use the classification services based on data from around the network, as well as providing new brain tumour cases from their local databases for the distributed data warehouse. New classifiers can then be produced, or existing ones improved, using the newly available data.

Although the use of classifiers is maximized for decision making in the HealthAgents system, direct access to patient records may be necessary in some situations and this requires some access principles. The age and the gender of patients, for example, can be associated with tumour types and so may be useful for diagnosis. Thus, a contract signed between two clinical centres working closely with each other may allow some cases to be transferred between the two, but not a third party. In addition, some classifiers may be trained internally for scientific experiments upon a specific set of data, and the creators may not wish them to be accessible to the general public owing to their applicability and reliability. These requirements demand the differentiating attributes of HealthAgents resources and their associated access principles.

An anonymized patient case is associated with a status. The status of the patient case can be changed to, for example, *validated*. That is to say that the patient diagnosis has been confirmed. The case can be *public*, being accessible by every HealthAgents node, or could remain *private*, being accessible only by its owner node or for producing classifiers. A selection of the validated cases labelled as public at each site can be shared altogether to produce *global* classifiers that are always public. A node can also request the creation of *local* classifiers that are trained uniquely with its own public and private data as defined by the requesting user. Apart from the global and the local classifiers, a node may want to develop specific classifiers that are trained with all the public cases available in the network in addition to its own private cases, being given a special weight to gain more accurate classification results for this particular site's cases. Again, they can be defined by the requesting users as either public or private. Once a classifier is produced, no matter how it is produced and with what data, all the cases sent from individual databases for training purposes will be discarded. The case data will only be temporarily stored in any other site apart from its origin.

The sharing of resources of different types and attributes for different purposes and circumstances makes the system conform to legal and ethical obligations. One of them, the UK Data Protection Act 1998, which came into force in 2000 and defines a list of principles, regulates the processing of data of individuals, including the obtaining, holding, use, or disclosure of such information:

1. Personal data shall be processed fairly and lawfully (and under certain conditions).
2. Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.
3. Personal data shall be adequate, relevant, and not excessive in relation to the purpose or purposes for which they are processed.
4. Personal data shall be accurate and, where necessary, maintained up to date.

5. Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.
6. Personal data shall be processed in accordance with the rights of data subjects under this Act.
7. Appropriate technical and organizational measures shall be taken against unauthorized or unlawful processing of personal data, and against accidental loss or destruction of, or damage to, personal data.
8. Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedom of data subjects in relation to the processing of personal data.

In the HealthAgents system, patient case records are only processed for either the diagnosis of that particular patient or for training classifiers, fairly and lawfully in compliance with *Principle 1*. Access to a case is strictly controlled by the node at which the case is stored inside the HealthAgents network, and reference to it can only be traced as metadata associated with classifiers trained on the data. Thus, cases will not be processed in any manner that conflicts with *Principle 2*. Clinical centres are responsible for their cases and, wherever possible, link-anonymized data are used for the preservation of patient privacy, in compliance with *Principle 3* and *Principle 4*. All cases used for the purpose of training classifiers will be discarded when classifiers are produced and will not be kept for longer than is necessary to comply with *Principle 5*. Patients retain the right to withdraw their cases and if requested they will be removed from the databases immediately (via the unique patient identifier being added to the link-anonymized data), as per *Principle 6*. Each clinical centre enforces the described case access principles, and therefore unauthorized or unlawful processing of personal data or damage to data will be avoided, in accordance with *Principle 7*. The HealthAgents project is building a network inside the EU boundary and may allow data transfer outside its network only if it is in a fully anonymized form and protected at an adequate level as agreed upon; this is in compliance with *Principle 8*.

6 Secure data transportation

The secure transportation service of the HealthAgents system provides a layer of protection for the agent messaging service in the framework. The platform-independent secure message transportation service has been developed on the Java Cryptography Architecture (JCA), which provides tools for implementing a standard interface for encrypting and decrypting. In the HealthAgents system, all interaction protocols requiring secure message passing will implicitly include the security procedure in the communication between HealthAgents pairs.

In HealthAgents, the transfer of large amounts of imaging and spectral data would make asymmetric encryption computationally expensive. Hence, a symmetric encryption scheme is adopted, Advanced Encryption Standard (AES), with asymmetric keys from the Rivest Shamir and Adleman (RSA) scheme used for the exchange of symmetric (AES) keys between communicating parties, as is standard, for example, the Secure Sockets Layer framework uses this approach. The agent architecture (described in a separate paper in this issue) is implemented in a layered manner, with the necessary cryptographic functionality built into every message exchanged between agents, without requiring it to be programmed into every conversation.

A generic scenario in which a sender agent, Agent 1, sends a message to a receiver agent, Agent 2, is shown in Figure 4. Such a procedure and JCA's API (Application programming interface) support can be split into steps as follows:

1. Both agents must, at the start-up stage, register their public keys to the system via the Yellow Pages Agent and retain the private keys. JCA's engine class *KeyPairGenerator* can be used for the generation of a pair of public and private keys.
2. Agent 1 can now retrieve Agent 2's public key, at runtime, from a key store maintained by the Yellow Pages Agent. After obtaining this public key, Agent 1 generates a secret key that will be used to encrypt the plain-text message to be secured. JCA's engine class *KeyGenerator* can be used to generate new secret keys, each only valid for a given conversation.

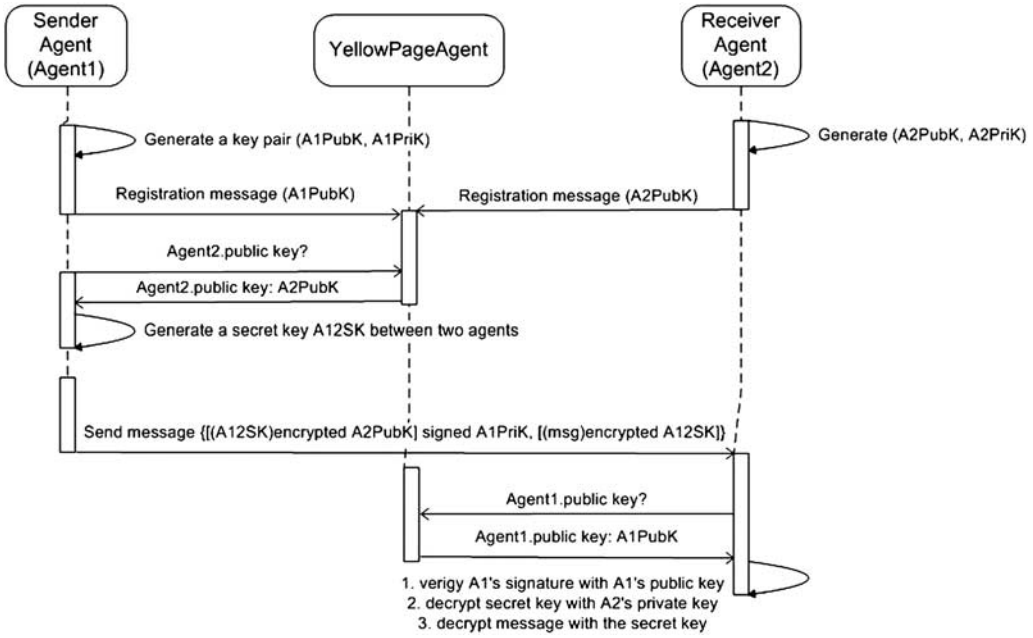

Figure 4 The sequence diagram of secure message transportation

3. The secret key must be shared between two agents. This can be achieved via Agent 1's encryption of the secret key using Agent 2's public key. JCA's engine class *Cipher*, once initialized with keys, can be used for encrypting and decrypting data. The symmetric algorithm of DES and asymmetric algorithm of RSA have been used in HealthAgents for the encryption and decryption of secret keys and messages, respectively, but can be easily switched to other algorithms.
4. These data with the secret key encrypted are signed by Agent 1's private key. JCA's engine class *Signature*, once initialized with keys, can be used to sign data and verify digital signatures. The eventual message for passing in the network will include the message encrypted by the secret key and the secret key encrypted by the public key of the receiver with the sender's signature attached.
5. Upon receiving the message sent from Agent 1, a reverse process will carry on for signature verification and message decryption. First, Agent 2 reads Agent 1's signed data and verifies its identity by retrieving the public key of Agent 1 from the common public key store. Further, the data regarding the key are decrypted by Agent 2 using its own private key and the secret key is revealed. Finally, the encrypted message will be decrypted using the secret key.

The secure message transportation service, called `JCACryptor` in the HealthAgents framework, is transparent among all communicating agents, and it has been configured in the system that

```
{message-cryptor = net.healthagents.agent.crypt.JCACryptor}
```

This configuration item will be dynamically read by the `MessagingService`, and its `encrypt()` (steps 1–4) and `decrypt()` (step 5 including the entire reverse decryption procedure) methods will be invoked by agents implicitly through message passing. Both methods are declared in the `Cryptor` interface according to which `JCACryptor` is implemented. The following depicts the procedure of using the `encrypt()` method of the service. It is performed in the framework rather than in the individual agent code:

```
Cryptor c = Configuration.getCryptor();
// Send a message
ACLMessage msg = new ACLMessage( ACLMessage.INFORM );
msg.setContent( c.encrypt( me.encode( m ), m.getMessageID()) );
```

The code above shows that a swappable cryptography implementation of the transportation service is retrieved according to the current configuration. This enables us to easily exchange the message encryption module based on providers, algorithms, or other variants. Next, a new message is constructed by the message sender and then the plain text content of the message is encrypted using the message cryptor before being sent. On the other end of the message exchange, the message receiver will use the same cryptor as is currently configured, to decrypt the received message and read the content. The actual cryptor construction and instantiation using the keys of the sender and the receiver are encapsulated in the specific implementation and are independent from the framework level code shown here. In turn, the framework level code is separated from the individual agent level code. This indicates that all agents joining the network for message exchange will have the encryption and decryption protocols applied to them transparently and automatically. Their implementation is subject to flexible replacement, if necessary.

7 Secure data sharing and collection

The secure message transportation service ensures that someone with a valid user account for system login can send or receive messages that are protected from eavesdropping. However, ensuring that the right users access the right resources relies on the fact that users will be properly authenticated; in a distributed network this brings special challenges. As authentication will be carried out at local partner sites before a valid user logs into the GUI and performs system operations, a concern is raised about how each partner site can recognize the validity of a user from another partner site, who requests access to their resources but the account of which resides outside their local databases. When a user logs in from the local site, the user should have the proper permissions, if assigned previously, to access the data distributed across the network without a second authentication by the data centre sites. This should work algorithmically according to the aggregated permissions, which are dynamically configurable and immediately accessible by the centres.

Although a global user account repository with a pool of all the periodically synchronized user accounts is not considered a proper solution, a repository including all users' cross-site permissions can address the issue. The context for setting up such a repository is that, while the majority of case-based access in the system remains at a local level, users may occasionally wish to legitimately access particular cases of special interest outside their centres, if mutual agreements have been reached among the sites. More often, data collectors must be authorized to collect data from various partner sites and produce classifiers. The registry of their permissions and the allowing of data centres to authorize the collection of their data to the data collectors instantly is therefore essential to a successful working system.

Such a global user permission repository must contain no redundant user information already included in local databases, but a registry of the very essential cross-site user access agreements. The simple access agreements residing in this global registry adopt the following scheme:

(HealthAgents unique global user ID, Data centre ID, Boolean values of permission (read/collect))

Several components are involved in the secure data sharing and collection architecture:

- A client node or a producer node is connected to the HealthAgents network and requests resources from the network. This could either be clinical users who want to view cases of interest, via the main system GUI, or data collectors who require a set of relevant cases for producing classifiers.
- A data centre node is connected to the HealthAgents network and provides its data. A DatabaseAgent is deployed at every data centre for answering user queries or returning data sets to producers.
- A DatabaseSecurityService is used by every DatabaseAgent for cross-site user access permission checking before the main functions are being carried out by the users. The module provides secure data access and collection service. It checks external user access, data collection, and

provides other facility functionalities such as filtering cases being collected according to their public/private attributes.

- A global user permission repository is deployed throughout the entire network, which stores all HealthAgents users' permissions of cross-site data access and data collection.
- A global user permission management GUI is deployed and is accessible by (a) all users or data collectors to request access permissions from data centres, as well as by (b) local administrators to view such requests and approve or revoke permissions as appropriate, reflecting the current access agreements.

These components work together in the following order:

1. Being authenticated locally, a clinical user logs into the main system GUI and wants to view a case of interest but is denied permission to do so. This could also happen when a data collector wants to collect relevant cases for producing classifiers.
2. The user is redirected to the global user repository management GUI, the identity of which is recognized by the already supplied local username and password together with the user's originating site, from which the global user identity is mapped. The user is presented with a complete list, showing all data centres currently involved in HealthAgents. The user can then make a request to whichever centre necessary.
3. Being informed by a system generated message of the request, a local administrator, responsible for managing the data centre to which the request has been made, logs into the same GUI and can view the current user access requests, and can approve them or revoke existing permissions.
4. Being informed by a system generated message of the approved permission, the user can resume the previous operations via the main system GUI and now the request will be re-evaluated by the secure data access and collection service, using the reconfigured global user permission repository. As the repository now has the appropriate permission settings for the user, such a request will be approved. The repository is always reconfigurable at runtime and the security service always checks the current security settings dynamically against the requests made at that particular time.
5. Eventually, the DatabaseAgent returns the requested case to the clinical user or the collection of cases to the classifier producer, the private ones being filtered out from the collection.

Figure 5 shows all the components of the architecture and the interaction protocols between them. Such a distributed permission management system has all the cross-site access permissions maintained separately by individual site administrators but stored in a central repository. It is anticipated that the most existing and future clinical participants of HealthAgents will agree upon a common database schema for user and clinical information storage and sharing. Nevertheless, users from new participant sites, without the implementation of such a schema, can still access the HealthAgents distributed databases via the maintenance of such a permissions repository, if appropriate access agreements have been reached. This provides a quick and easy network participation framework.

A central component in the architecture is the `DatabaseSecurityService`. It looks up the repository and checks permissions via one of the several provided methods, depending on the nature of access requested:

- The `checkLocalUser()` method provided by the service supports all access control models including in-site access; it ascertains whether a local user has permission to perform an action in the local database, prior to presenting cases or returning results to the user.
- The `checkExternalUser()` method takes as input (i) a global user ID that is mapped from the user's ID at the local node and the user's node ID, (ii) a data centre ID where data are requested, and (iii) an action that produces a yes/no output by determining whether the user has access permission to the database in that centre, before the actual presentation of cases to the user.
- The `checkCollectorUser()` method takes a data collector's global user ID, a data centre ID where data are to be collected, and evaluates whether the collector user has permission to collect data from the data centre, before the actual data collection.

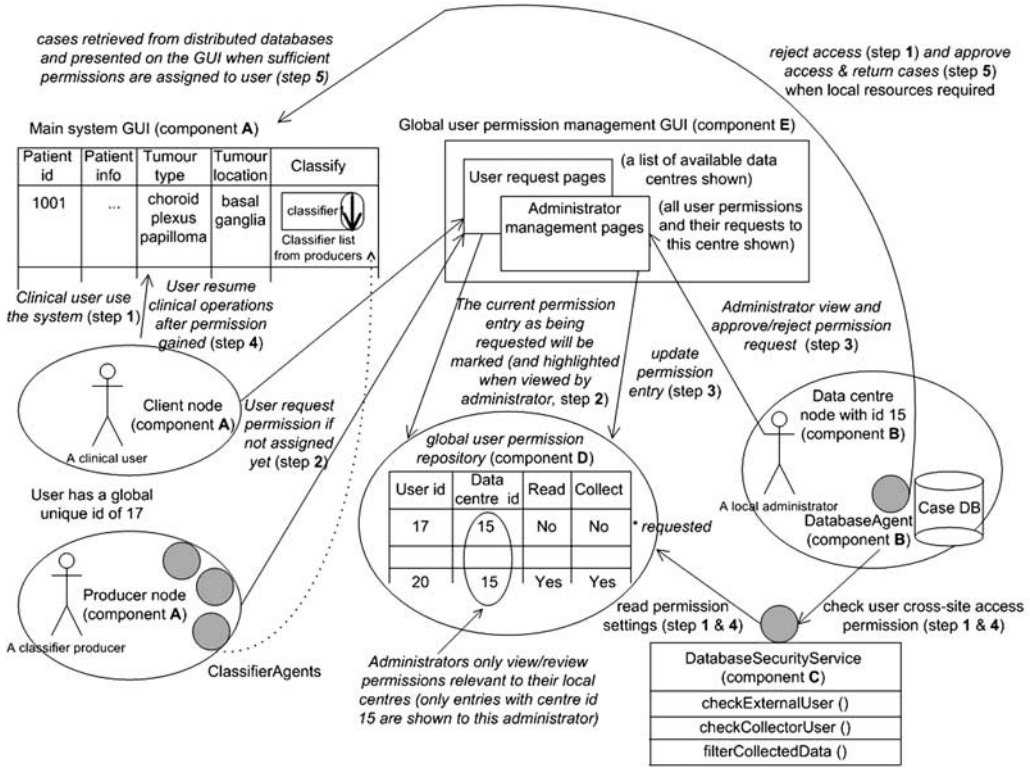

Figure 5 The secure data access and collection service architecture

- The `checkCollectorUser()` method takes a collector's Internet Protocol (IP) address and evaluates whether the collector is trusted. Only two sets of data collectors, physically running in Valencia and Leuven are currently involved in the project's network and allowed to collect data. Their IP addresses are pre-registered in the network and when the system is running, data collectors must demonstrate ownership of the pre-agreed IP before any data collection is permitted.
- The `method filterCollectedData()` filters the collection of relevant cases and only allows those open for building classifiers to be returned, given the data collection has already been approved. The data centre must give consent to cases being delivered outside of the centre for classifier building purposes, otherwise for some reasons, for example, if a patient is unhappy with the sharing of their confidential data, cases will be marked as private and filtered out from any data collection process. In the architecture, the security procedure must follow three basic steps for a producer to collect data: collector agent identification and trustworthy evaluation via `checkCollectorAgent ()`; data collector identification and permission checking via `checkCollectorUser ()`; private data set filtering via `filterCollectedData ()`.

8 A security model for more advanced controllability

The previous section describes a secure data sharing and collection service implemented in the system, where a user will either be able to access all (public) cases from a data centre site or none. In this scheme, no refined policy can be defined to allow more fine-grained access control, for example, based on access to individual cases.

The global user permission scheme based on the triple—user global ID, data centre ID, type of operation—is also a limitation. Introducing a case ID into the scheme will make the system cumbersome. Not only must a complicated synchronization mechanism be designed to ensure that the system always maintains the up-to-date cases and their IDs, the permission management of

many permission items will also bring heavy maintenance burdens to local administrators. Nevertheless, the availability of the existing repository puts the basic user permissions in place and building a more advanced policy-based access control system on top of them for more specific control becomes easier.

The need for a case-based access control rather than a site-based access control is justified by established clinical principles. It is not rational to allow a professional to have access to all patient records from a single hospital or even the entire network. Only relevant clinicians who have real-life relationships with patients in clinical centres should access their records. This is documented in the British Medical Association's security policy principles for clinical information systems (Anderson, 1996). A triangle relationship is described in Calam (n.d.): a patient is associated with a workgroup of which a user is a member, so that a user is permitted access via the workgroup to the patient (the patient may originally have been assigned to a clinical consultant or by referral from another, with transitive transfer of access rights). This implies that the organization of users in groups or roles is needed for access control. Indeed, maliciously or accidentally, users may create low-quality classifiers, or assign unmatched ranking values to classifiers. The abuse or misuse of classifiers could have a significant negative effect on routine clinical diagnosis. The definition of policies based on the types of users according to their trustworthiness, organization positions, or job functions, rather than individual identical users gives more configuration flexibility.

An established access control model that supports efficient management is the widely accepted US National Institute of Standards and Technology model of RBAC (Sandhu *et al.*, 1996). In RBAC, roles represent job functions in an organization. Roles bring together users and permissions in one representational scheme. Permissions that describe operations upon resources are associated with roles. Users are assigned to roles to gain permissions that allow them to perform particular job functions. For example, a clinician role can be created in a hospital, and permission giving access to patient data can be associated with this role. When a new clinician joins the hospital, he/she can be assigned the clinician role and so have the permission to access patient data. A major benefit of using this type of model is that the reconfiguration of user–role, role–-permission, and role–role relationships, directed by administrators, can reflect changing organizational policies. The maintenance of such a sub-system that is independent of the core application minimizes the impact of requirement changes on the overall system with regard to security.

However, the classic RBAC and its associated approaches have limitations. It is assumed that a large number of users can be grouped into several role groups requiring certain access levels in involved organizations. Given the requirement that information access or transfer may alter from patient to patient, RBAC will be confronted with difficulties. In a hospital, different users with the same role as a clinician may have different permissions to particular resources. For example, one clinician that created a patient case in a hospital might have more rights than other clinicians in the same hospital. Clinicians in one hospital could have more rights to data in that hospital than clinicians from another hospital. As permissions are not directly assignable to individual users, it is impossible to use RBAC to differentiate users with the same designated roles but with different capabilities in the system. Another insufficiency in the RBAC model is the lack of context of access provision in the way permissions are modelled. The context can constrain specific conditions that must be met before access is given. In the above example of clinicians accessing patient data, access permission is different depending on the different context (whether a clinician created the patient case). Finally, if there is no explicit concept of organizationally determined access rights and restrictions on defined groups, it is inconvenient to grant permissions to specific collections of users except by granting permissions on an individual basis.

The design of an advanced security model, taking the advantages of RBAC and avoiding its weaknesses, must also consider several unique security needs of HealthAgents and support a configurable but general purpose security system. Through the collaboration of multiple centres, which not only provide their cases but also require classifiers for their own use, the system should be able to respect the individual access control policies separately managed by each centre. In addition, there might be global constraints that are applicable to shared resources. All these

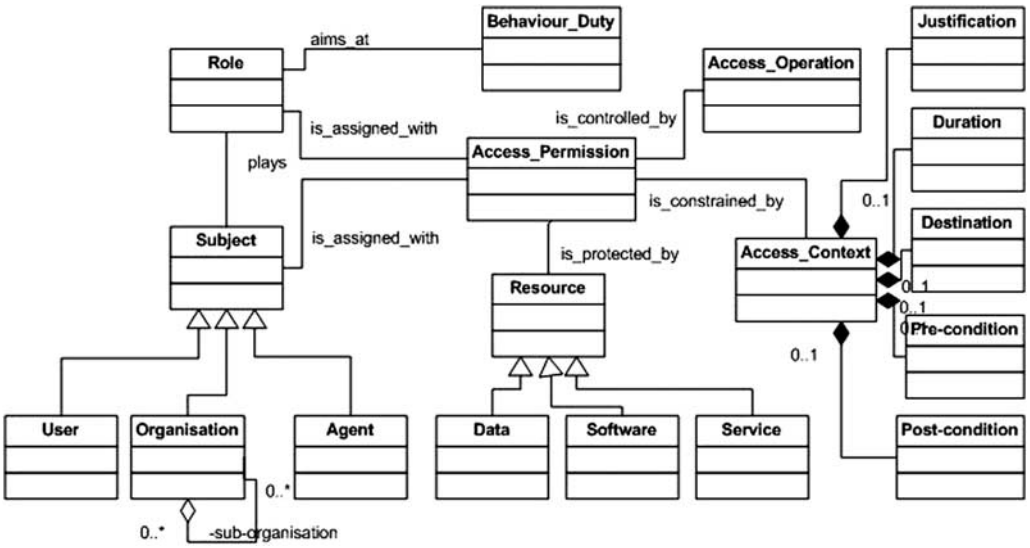

Figure 6 The security meta-model

policies and constraints could change continuously according to the system needs. For instance, a new junior clinician who recruits at a collaborative centre may not have the privileges required to create a new classifier or update the reputation of existing classifiers, as this could have a global impact on all diagnoses across centres. However, they may be allowed to perform such operations upon building up work experience. The system may have to assign to different users or even the same user at a different time or under a different context, appropriate access rights to system resources distributed among the centres.

8.1 Security meta-model

RBAC has been extended to avoid its weaknesses and to meet the characteristic requirements of HealthAgents.

The fundamental access permission policies take the form of the following 5-tuple: `{Role, Subject (Id, Organisation), Access Operation (Op), Access Context (Co), Resource (Id, Type)}`, and is shown in Figure 6.

Policy rules externalize security requirements and are structured in this form for later continuous management. The meta-model has been motivated by the particular requirements of the HealthAgents project, but it is generic so that other domains and applications may use it. In the HealthAgents d-DSS, a user who logs on to the system will be associated with an agent with ID and roles. Permissions are granted to agents through those directly associated (via the subject ID), roles they are assigned to (via subject–role relationships), or organizations they belong to (via clinical organization membership). Role definitions and user–role assignment are managed locally in individual hospitals. An administrative role can be assigned to a HealthAgents project manager to manage users and roles globally. On an individual basis, a clinician may have full access rights to his/her patient, whereas other clinicians may not. A clinician role hierarchy may also be defined (manager, principle clinician, senior clinician, junior clinician, apprentice, etc.) so that some clinicians have more access to operations (e.g. who can add new cases to the system) than others (who, for instance, can only run a classifier).

RBAC has been extended with permissions assignable to individuals as well as organizations. It might be necessary, for example, to define that senior clinicians can access all instances of a particular type of resource, the classifiers. More likely, individual entities of a resource type are deemed accessible by individual subjects. Permissions can be assigned upon a set (or type) of resources or for a group of subjects with exceptions. This can be configured by a positive permission

policy for the whole collection and a negative permission for individual exceptions. Although this kind of reasoning with exceptions is often difficult to integrate into declarative representations, we envisage a procedural implementation within which such permission constraints are processed as outlined below (see also Xiao *et al.*, 2008a).

RBAC has also been extended with the concept of context to provide additional flexibility. Access context might include a descriptive justification of the access operation, where/when the requested data goes, the duration of the use of the data, and the pre-condition and post-condition of the access operation. Agents have roles during their interaction, context varies and agents behave differently while evaluating certain instance values populated at runtime. A clinician may have special control over the data of a patient under the pre-condition (a type of context) that he/she is the principal doctor of the patient, and this special identification must be checked against before a special operation is carried out. Context can also be used to enable access normally not seen through rights delegation, for example, when two hospitals (or clinicians) reach some agreements. A hospital can then delegate the use of its private classifiers to another hospital or delegate the access of its patient data to some particular external clinicians or bodies for classification, given the appropriate ethical and patient permission has been obtained. Context specification is also useful to give special access to appointed individuals, even being outside the HealthAgents network and having no user account or role assignment. By supplying a justification of how the required data will be used and the destination of the data transmission, the access may be granted if such information is approved under appropriate contracts and with specified permissions.

The Role is an important concept in Agent-oriented Software Engineering (AOSE) and is tightly associated with agent behaviour. However, the role concept in the AOSE research community and that in the RBAC community are completely distinct and no research has ever been carried out to reconcile the two definitions of the concept for security control in MAS. In our security meta-model, agent behaviour is specified in roles that not only realize functional requirements but also enforce security policy requirements. RBAC has no concept of duty and AOSE has no permission constraint for agents. The complementary nature prompts us to define a role interaction model as one that integrates the concept in an agent paradigm and that in RBAC. In our meta-model, we stipulate the following:

> a role enacts its expected behaviour only if its permission constraint is satisfied.

8.2 HealthAgents interaction model and security policy specifications

The security model and the associated policy rule model avoid the weaknesses of RBAC and extend it towards a seamless integration with the role-playing pattern of AOSE. The security model sitting in MAS will not let agents fulfil regular functional requirements unless security requirements are met. A role plays its functional duty only if its social constraints are satisfied. This, therefore, achieves the separation of the functional and non-functional requirements for easier management and maintenance but at the same time the two parts are integrated in the running system with unified agent-playing behaviour according to the combined specification.

The model provides fine-grained access permission configuration based on individuals, roles, and organizations. A resource access request message can be traced to its origin and mapped to the roles that subject plays. Role-based policies are easier for management, but identity-based policies allow customization and exception handling. Policies can be defined in both forms. In HealthAgents, we have case records, classifiers, services (Yellow Pages, etc.), and their access must be protected by policies. Access operations should be distinguished for resources. One clinician may be able to execute a classifier but not update its reputation.

A context provides flexibility to the model, such as (1) allowing, in particular situations, certain specially delegated access in the name of a particular role; (2) providing justification of the special access; and (3) constraining the valid time period associated with the access. A comprehensive interaction model involving all of them is used to demonstrate the application of the above policy rule scheme to meet the requirements described in that section. The model includes most of the

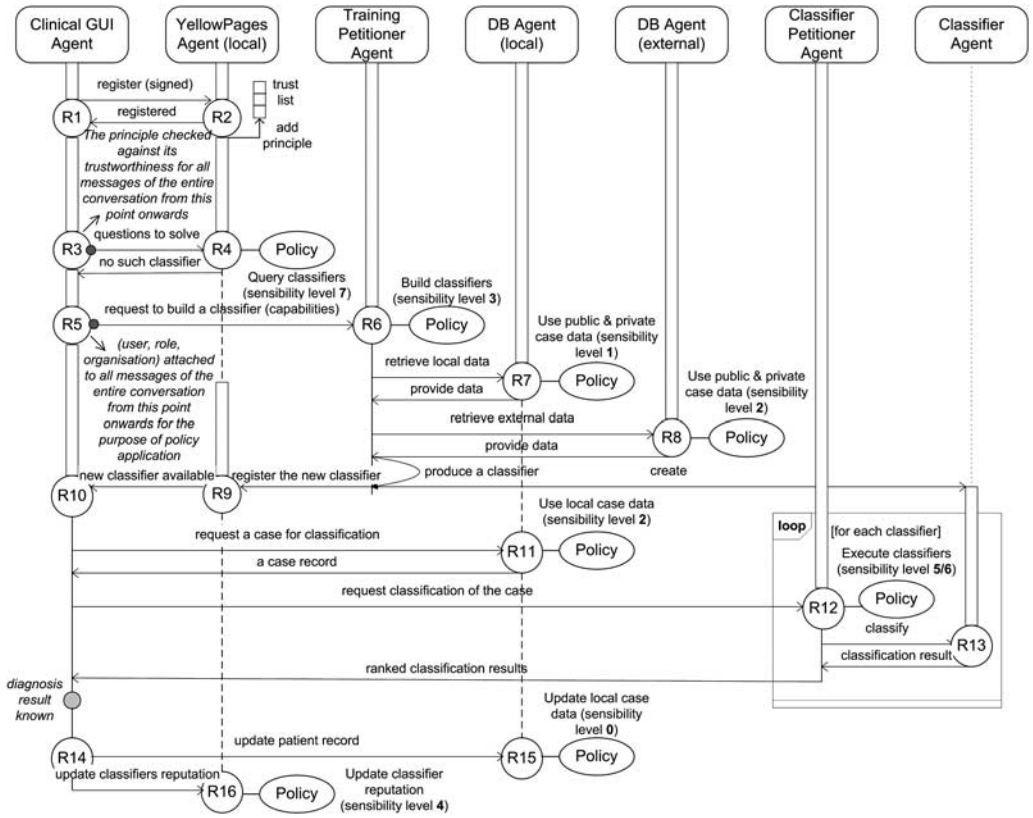

Figure 7 Agent interaction model with security policy set application in HealthAgents

HealthAgents business functions as well as resource access flows. Briefly, the scenario is that a new hospital joins the HealthAgents network with a new MAS setup in that site, new clinician users wish to perform classifications on cases from there, and they do so by creating new classifiers for the purpose. The role interaction model (referring to Figure 7) can be described as follows, referring to Figure 1 for the HealthAgents architecture and previous sections for supporting layers of secure communication and authentication:

- The new clinician is authenticated via the local GUI Agent and his/her principle is bound with the interface for the entire interactive session (R1).
- The GUI Agent registers this new node via the Yellow Pages Agent, which recognizes its identity (before this, the local hospital manager may have to acknowledge the participation of the new site to the HealthAgents network administrator through conventional phone calls, R1 and R2).
- The Yellow Pages Agent adds this new node to the trusted node list (R2).
- The GUI Agent at that node can start to communicate in the HealthAgents network and now it wants to perform a classification upon a local case (R3).
- The GUI Agent searches the Yellow Pages Agent for available classifiers by sending questions to solve as the first message it initializes for a new conversation (R3 and R4).
- The Yellow Pages Agent has its principal registered in the trusted list, therefore all ongoing communication in this conversation with all other agents will be allowed and all these messages will be signed and encrypted (R4).
- The Yellow Pages Agent checks this GUI Agent against the permission of using its Yellow Pages query service and will perform the query to its registered classifiers, but unfortunately no such classifier is available (R3 and R4).
- The GUI Agent requires the building of a new specific classifier using distributed data sets (R5 and R6).

- The Training Petitioner Agent applies a local policy repository and allows the request operation of building a new classifier (R6).
- Relevant public cases as well as local private cases from the request site will be sent to the building site for the production of the new classifier, and data access policy rules will be applied before the data are sent from each site (R6 and R7, R6 and R8).
- A new classifier is produced and registered to the Yellow Pages Agent, a copy becoming available to the original request site (R6 and R9).
- The clinician now wants to execute the new classifier on the case when being informed of the availability of the classifier (R9 and R10).
- The local policy rules on the use of the classifier and the particular case will be applied against this specific clinician and he/she will be allowed to perform the operation (R10 and R11, R10 and R12, R12 and R13).
- Decision-making support is received from the results of the classification and a diagnosis will be made later on (R12 and R14).
- When an actual diagnosis result is known, the clinician wants to update the classifier reputation, and the case he/she just diagnosed and the local policy rules on both operations will be applied against the clinician and he/she will be allowed to do so eventually (R14 and R15, R14 and R16).

The interaction model shown in Figure 7 captures the interactive behaviour of the involved agents each playing their respective roles, subject to the satisfaction of associated security policy constraints. We have demonstrated this integrated role function in a separate paper (Xiao *et al.*, 2008a), where workflows punctuated by checks for the satisfaction of security constraints have been designed in a process calculus-like language for agent interactions, the Lightweight Co-ordination Calculus (LCC; Robertson, 2004). The detailed interaction model in Figure 7 has been written in LCC and parsed to demonstrate the feasibility of integrating security into a workflow specification language for agent interactions. Not only is the functionality and security implementable in that framework, but also the execution of the process model leaves behind an audit trail of system behaviour. However, for its actual utility in software implementations, such as HealthAgents, the platform for such interactions would have to use the OpenKnowledge kernel[1], which supports LCC-compliant implementation of software agent choreography.

9 Conclusions

The security issues involved in health-care domains have been discussed in this paper. The practical solutions of these security issues have been addressed to the needs of the HealthAgents project. Our work includes the design and development of a security architecture in several levels. Various Software Engineering techniques and security protocols have been developed to provide a secure and maintainable health-care infrastructure.

A link-anonymized data scheme protects basic patient privacy. Sharing of classifiers instead of cases in major decision-support processes, and controlling the travelling of cases across sites by setting public or private attributes, further supports the maintenance of patient confidentiality. A secure transportation service protects data transmission. A secure data access and collection service controls cross-site case access. A RBAC mechanism enables fully customized resource access control.

Using a security policy rule scheme and applying it to the interaction model for the HealthAgents Multi-Agent System, security policies can be separately configured, but dynamically integrated, into the running agents of the distributed network. Security policies enable easy and separate maintenance tasks across centres as they can be independently defined and maintained in each individual site; however, their application is also under a unified access control scheme for resources with diverse types and locations. These make our security model adaptive. When a new hospital joins, new policy sets can be defined locally by the hospital managers. When its resources

[1] `http://www.openk.org`

are required from other sites, these policies will be applied by responsible manager agents residing in that site uniformly, conforming to the regulations set in that site. When its users require access to resources from other sites, the external policies will be applied in the same manner in which users and their assigned roles determine their access privileges. Once any policy rule is changed, the effect is immediate to all roles or individuals associated with the rule. Policies are automatically deployed and immediately available, requiring no coding, just the minimum administrative overhead.

The layered security model developed for HealthAgents has several novel features. First, the existing security solutions in agent-based systems are not adequate in Computer Science literature. Addressing the security issues in the agent-based clinical decision support systems is even rarer. The approach described in the paper provides practical implementation in this area in a rigorous manner. Second, a dual model of user identification and role management is embedded in a human–agent interaction environment. Agent architectures decompose functionality, focusing on methods. Composition occurs upon message passing or notification. The HealthAgents agents cooperate/collaborate with human actors who play multiple roles within the constraints of institutionally arranged access rights and privileges. Although securing messages relies on standardized protocols, simultaneous user identification and flexible role management and behavioural control, according to system level interaction and requirements delivery, is considered novel. Third, the approach allows the security requirements to be accommodated incrementally, in an adaptive way, in which security policies can be reconfigured and applied dynamically as constraints associated with functional interaction models. RBAC has been extended for this purpose. Lastly, the overall interaction model and the security policy model together contribute a model-driven architecture for the development of secure and adaptive health-care applications.

Because there is no single agent for securing the system, validation is performed on whether the messaging protocols are secure and standard. There are two aspects. One, in the transportation level, whether messaging is secure syntactically and second, in the policy control level, whether the correct policies are applied in given conditions semantically. The first is defined within the messaging and message handling layers of the agent architecture. Once the handling of private and public keys is done correctly, there can be no issues because of transitive fault (insecurity) propagation, no matter what the protocol is. This assumes a correct identity management system, specifically as it intersects with the role-based access abstractions. This leaves the validation of policy rule matching and application. A way of doing this is to encode all workflows in a rigorous language such as LCC, with constraints for security checks, and to run a model checker for temporal properties under security breaches locally. The description of the modelling at this level is discussed in Xiao *et al.* (2008a) and model checking will be part of our future work.

Elsewhere in the continuous work of the project, we have described the security policy rule scheme and their Software Engineering support (Xiao *et al.*, 2007), as well as their full integration with functional rule model into the established Adaptive Agent Model (Xiao & Greer, 2007, 2009); the ontology support to the security policy representation and reasoning for consistency checking (Croitoru *et al.*, 2008); the development of the overall model according to major Software Engineering principles to achieve software quality (Xiao *et al.*, 2008a); and the extension of the security model into a layered architecture to meet closely related clinical requirements (Xiao *et al.*, 2008b). We believe this work has established a comprehensive and practically useful security model and can provide a valuable reference for other distributed health-care systems.

Acknowledgements

This work is supported under the HealthAgents and OpenKnowledge STREP projects funded by the EU Framework 6 under grants IST-FP6-027214 and IST-FP6-027253.

References

Anderson, R. 1996. Clinical system security: interim guidelines. *British Medical Journal* **312**, 109–112.
Anderson, R. 2001. Undermining data privacy in health information. *British Medical Journal* **322**, 442–443.

Bray, F., Sankila, R., Ferlay, J. & Parkin, D. 2002. Estimates of cancer incidence and mortality in Europe in 1995. *European Journal of Cancer* **38**(1), 99–166.

Calam, D. (n.d.) Information governance—security, confidentiality and patient identifiable information. URL: http://etdevents.connectingforhealth.nhs.uk/eventmanager/uploads/ig.ppt.

CCRA. 2006. Common criteria for information technology security evaluation. URL: http://www.commoncriteriaportal.org/.

Choe, J. & Yoo, S. K. 2008. Web-based secure access from multiple patient repositories. *International Journal of Medical Informatics* **77**(4), 242–248.

Croitoru, M., Xiao, L., Dupplaw, D. & Lewis, P. 2008. Expressive security policy rules using layered conceptual graphs. *Knowledge-based System* **21**(3), 209–216.

GMSC/RCGP. 1988. GMSC and RCGP Guidelines for the Extraction and Use of Data from General Practitioner Computer Systems by Organisations External to the Practice. Technical report, GMSC/RCGP Joint Computer Group.

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., Huffel, S. & Lluch-Ariet, M. 2009. Healthagents: distributed multi-agent brain tumour diagnosis and prognosis. *Applied Intelligence* **30**(3), 191–202.

Gritzalis, D. & Lambrinoudakis, C. 2004. A security architecture for interconnecting health information systems. *International Journal of Medical Informatics* **73**(3), 305–309.

Hawker, A. 1995. Confidentiality of personal information: a patient survey. *Journal of Informatics in Primary Care* **March**, 16–19.

IEEE. 1996. *IEEE Guide for Software Quality Assurance Planning*.

Keese, J. & Motzo, L. 2005. Pro-active approach to malware for healthcare information and imaging systems. *International Congress Series* **1281**, 943–947.

Kirn, S., Heine, C., Herrler, R. & Krempels, K.-H. 2003. Agent.Hospital—agent-based open framework for clinical applications. *IEEE International Workshops on Enabling Technologies*, 36. doi/10.1109/ENABL.2003.1231379.

NIST. 2006. *Minimum Security Requirements for Federal Information and Information Systems*. Technical report, National Institute of Standards and Technology. http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf.

Pfleeger, C. & Pfleeger, S. 2002. *Security in Computing*, 3rd edn. Prentice Hall.

Robertson, D. 2004. A lightweight coordination calculus for agent systems. In *Declarative Agent Languages and Technologies* Lecture Notes in Computer Science **3476**, 183–197. Springer.

Sandhu, R. S., Coyne, E. J., Feinstein, H. L. & Youman, C. E. 1996. Role-based access control models. *IEEE Computer* **29**(2), 38–47.

Smith, S. W. 2003. Humans in the loop: human–computer interaction and security. *IEEE Security and Privacy* **1**(3), 75–79.

Xiao, L. & Greer, D. 2007. Towards agent-oriented model-driven architecture. *European Journal of Information Systems* **16**(4), 390–406.

Xiao, L. & Greer, D. 2009. Adaptive agent model: software adaptivity using an agent-oriented model-driven architecture. *Information and Software Technology* **51**(1), 109–137.

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

Xiao, L., Lewis, P. H. & Dasmahapatra, S. 2008b. Secure interaction models for the HealthAgents system. In *'SAFECOMP'*, Springer, 167–180.

Xiao, L., Peet, A., Lewis, P., Dashmapatra, S., Saez, C., Croitoru, M., Vicente, J., González-Vélez, H. & Lluch i Ariet, M. 2007. An adaptive security model for multi-agent systems and application to a clinical trials environment. In *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, **2**, 261–268.

Zhang, L., Ahn, G.-J. & Chu, B.-T. 2002. A role-based delegation framework for healthcare information systems. In *SACMAT '02: Proceedings of the 7th ACM Symposium on Access Control Models and Technologies*, ACM, New York, NY, USA, 125–134.