available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A knowledgeable security model for distributed health information systems

Liang Xiao[*], *Bo Hu, Madalina Croitoru, Paul Lewis, Srinandan Dasmahapatra*

University of Southampton, UK

ARTICLE INFO

Article history:
Received 6 October 2008
Received in revised form
1 May 2009
Accepted 1 August 2009

Keywords:
Decision support system (DSS)
Distributed clinical information system
Interaction model
Lightweight Coordination Calculus (LCC)
Multi-agent system (MAS)
Security model

ABSTRACT

Realising the vision of pervasive healthcare will generate new challenges to system security. Such challenges are fundamentally different from issues and problems that we face in centralised approaches as well as non-clinical scenarios. In this paper, we reflect upon our experiences in the HealthAgents project wherein a prototype system was developed and a novel approach employed that supports data transfer and decision making in human brain tumour diagnosis and treatment. While the decision making needs to rely on different clinical expertise, the HealthAgents system leveraged a domain ontology to align different sub-domain vocabularies and we have experimented with a process calculus to glue together distributed services. We examine the capability of the Lightweight Coordination Calculus (LCC), a process calculus based language, in meeting security challenges in pervasive settings, especially in the healthcare domain. The key difference in approach lies in making the representational abstraction reflect the relative autonomy of the various clinical specialisms involved in contributing to patient management. The scope within LCC of accommodating Boolean-valued constraints allows for flexible integration of heterogeneous sources in multiple formats, which are characteristic features of a pervasive healthcare environment.

1. Introduction and motivation

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it". This is Mark Weiser's vision (Weiser, 2001) of how technologies might eventually blend in with our surroundings. Projecting this vision on to healthcare gives a picture wherein "smart" software agents would act on behalf of human specialists in collecting/monitoring critical life support data, extracting information from the data, jigsawing information/data together, and eventually enabling decisions and actions to be taken on the outcome of such processes. One of the most far-reaching consequences of such a vision is the emergence of a different paradigm of patient care – pervasive healthcare. Currently, a person experiencing a perceptible ailment invokes the "patient-seeing-doctor" pattern, where a doctor is often an array of specialists. Instead, the new healthcare paradigm emphasises a degree of continuous medical surveillance, with key decisions for medical follow-ups requiring automated processing, and in a decentralised manner.

One of the fundamental questions concerning pervasive healthcare is how to ensure the data are delivered to the right person at the right moment. Thus far, knowledge in healthcare, to some extent, remains a "cottage industry" with largely tacit knowledge only explicit to isolated specialists,

[*] *Corresponding author*. Royal College of Surgeons in Ireland (RCSI), Homepage: http://hrb.rcsi.ie/liang/index.html.
E-mail addresses: liangxiao@rcsi.ie, lx@ecs.soton.ac.uk (L. Xiao), bh@ecs.soton.ac.uk (B. Hu), mc3@ecs.soton.ac.uk (M. Croitoru), phl@ecs.soton.ac.uk (P. Lewis), sd@ecs.soton.ac.uk (S. Dasmahapatra).

0167-4048/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2009.08.002

organisations and professional guilds. Although the necessity of collaboration has been recognised, there is little systematic knowledge sharing of clinical intervention outcomes. With the advance of modern transportation, communication, and tele-medicine, patients are no longer restricted by physical and geographical constraints. In the situation of comorbidity (e.g. heart disease, AIDS, cancer, diabetes, or mental health), it is not a surprise to find that a patient is examined in one hospital; his/her case is reviewed by clinicians from another hospital; and he/she is treated in a third hospital by yet another group of clinicians due to speciality and availability. Data about a particular patient might be held by different departments within one hospital, from different hospitals and/or even from hospitals located in different countries. Data requests might come from members of a dedicated team accessing from their office or home, members of auditing committees, interns for educational purposes, and patients themselves all with different access privileges and access capabilities. Differences in work idioms in different situations evidently have the potential to significantly impinge on the quality of services and data security. Apart from the wide spread in geographic regions and a diverse landscape of users, the heterogeneity of clinical data is also demonstrated in the different levels of granularity of domain knowledge, different nomenclatures used in sub clinical domains, different protocols followed, different levels of details passed on in the form of medical records, and different standards reinforced by industrial manufacturers. In such an environment, knowledge which is a prime capital can only be based upon distributed and heterogeneous data/information sources and needs to be processed automatically in streaming mode. Users, therefore, need to locate the correct data providers, retrieve the most appropriate parts of the exposed data and glue together all the bits and pieces of information to make sensible conclusions. In the meantime, one needs to observe the data integrity and obey the data privacy and ethical regulations enforced by organisational and national clinical guidelines and protocols.

The HealthAgents[1] prototype provides us with an ideal platform to investigate the impacts and implications of experimenting with semantic-rich data and knowledge management technologies in a decentralised/pervasive healthcare system. In practice, a centralised repository gains credit for its effectiveness, security, and manageability. This is true as long as patients do not go beyond the catchment area of a hospital. Centralised solutions become less attractive when one is injured while visiting another country; when one needs daily care while on holiday in a retreat cabin; and when specialists are summoned up from different areas in a teleconference to discuss a rare case. Such a list of counterexamples could continue whilst they all share the same characteristics: decentralisation. As illustrated in Fig. 1, in a fully pervasive environment, we observe the relative independence of each participating agent or intelligent device which fulfils its designated responsibilities, either automatically, semi-automatically or under the supervision of human experts.

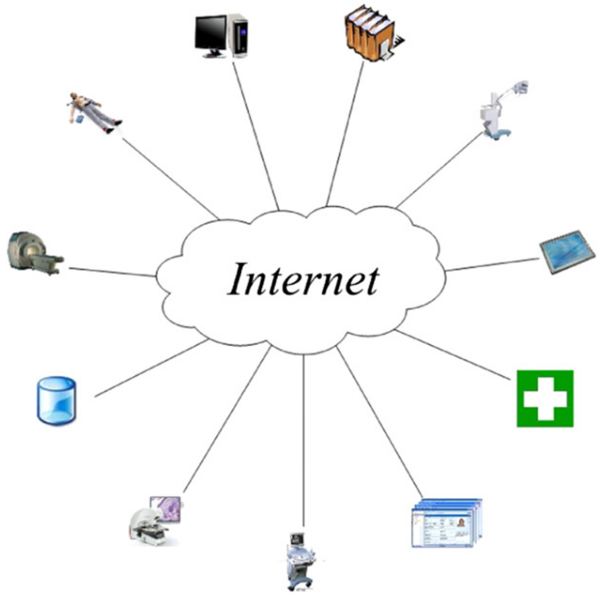


Fig. 1 – Pervasive healthcare architecture.

We shall lay out the various information fragments that are produced by these participating agents and build up a clinically appropriate and coherent representation of a patient based on potentially very different views. We tolerate the diversity and heterogeneity while systematically choreographing individual information resources so as to combine their knowledge of a particular patient or a particular disease. While interactions among individuals play an important role in engineering together distributed services underpinning the envisioned healthcare paradigm, security becomes a major concern when sharing, transferring, and modifying patient data/profiles. Prior to the discussion of the details of these interaction models and their scientific background, security requirements of clinical information systems are analysed for building up the proper models tailored for the need of secure interactions within Healthcare Information System (HIS).

2. Security requirements of healthcare information systems

We shall, in the beginning, draw distinctions between the types of threats imposed on healthcare systems and their likelihood. Though 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. Actually, 10% of GPs (general practitioners) in the UK have experienced their computers being physically stolen (Pitchford and Kay, 1995). More likely, improper use of the system may lead to privacy leaks, by careless (or malicious) users and when inappropriate privileges being given to them by the system. A well-designed system should not only protect the communication sites and end users, but also carefully authorise users with genuine needs to have access to selective sharing of information without exposing additional

[1] http://www.healthagents.cat.

information under protection. This particular security need has currently not been well addressed in healthcare information systems (Zhang et al., 2002). In this section, we outline the challenges and common security requirements of healthcare systems in a distributed environment, where preserving privacy and maintaining openness are crucial and information access decisions depend upon role and context.

2.1. *The distributed environment of healthcare information systems*

Aggregating dispersed data into large databases is expensive and practically unfeasible, since geographically different healthcare centres have to have control over their datasets and at the same time maintain a globally consistent data schema. A more important reason to oppose data consolidation is concerned with healthcare data confidentiality. In the UK, the National Health Service (NHS), driven by the motives of easier central administration and better information availability, attempted to build a unified electronic patient record system and give access to extended NHS community. This has been opposed (Anderson, 1996a, 2001) for the reason that such a system, collecting data from existing GP systems but out of their control, is in conflict with the ethical principle that no patient should be identifiable other than to the GP without patient consent (Joint Computer Group of the GMSC and RCGP, 1988) and the result from a survey that most patients are unwilling to share their information with NHS (Hawker, 1995). Another objection arises from the overwhelming workload such a centralised system could possibly put upon a security officer responsible for managing the data sharing (Zhang et al., 2002).

A distributed healthcare service infrastructure, however, implies the capability that is required 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, such as patient privacy preserving, to the information-sharing healthcare network.

2.2. *Preserving privacy and confidentiality in shared access*

The privacy of patient information is an important issue and failure to recognise this will lead to risk of patient safety, loss of public confidence in clinical organisations, and so on (Denley and Smith, 1999). A fundamental ethical principle stated by both the EU and the General Medical Council in the UK is that, patients must consent to data sharing. The British Medical Association advises that clinical professionals, who have access to patient confidential information in order to perform their duties, are responsible for the information they hold under ethical or professional obligations of confidentiality and shall not use or disclose such information for any purpose other than the clinical care of the patient to whom it relates. This means patients shall be assured that they can trust the access of their information, by a care team within their treating hospitals or experts involved from collaborative centres, if any, is safe and accords with their agreement. The moving from a traditional patient–doctor relationship towards a modern patient-healthcare service relationship implies trust in clinical systems must be maintained rather than reliance on doctor responsibilities. The absence of a mechanism or policy framework in the interest of information governance and confidentiality protection, hence, may damage the healthcare services aimed to be delivered, since private information of any individual patient may be made available by systems to people not directly related with the care of that patient. This will give opportunities to potential threats, possibly coming from inside workers, as well as outside hackers. Such threats include ungraceful private information disclosure and abuse or even more risky, incorrect clinical decisions made for vulnerable patients due to clinical data being wrongly altered, accidentally or deliberately. It is worth noting that threats from outside intruding into the network are much rarer than from inside. The security risks tend to increase dramatically, therefore, when an inter-connected clinical system network is in place which makes separately stored patient records and clinical information easily accessible and lets a wider range of people have access to them. Appropriate access control to patient records is the fundamental need for patient privacy and information security (Denley and Smith, 1999).

2.3. *Maintaining an open access*

Two aspects of openness must be maintained: 1) open for joining the system and not preventing any friendly but previously unknown clinical centre (bringing in its previously unrecognised users) from accessing information available across organisational boundaries; 2) open for information sharing to the network. Conducting healthcare research with more open use of information (identifiable data, etc.) under legitimate constraints and user acceptance, though not related with the clinical care directly, advances medical knowledge and promotes higher quality of healthcare service in the long run and is welcomed by the society. A clinical system can benefit most from clinical data as well as patient-specific data if such information can be machine-analysed and digested. The knowledge accumulated can be useful for later decision makings, particularly for rare but similar cases encountered in the future, confidential information contained in cases not being revealed.

2.4. *The different access needs to data subsets due to distinct job nature*

The need of distinguishing only the relevant data for sharing among clinical professionals rather than the whole records arises from preserving privacy while maintaining open access. Even if name, address and other privacy information is removed to produce a seemingly anonymised record, an NHS clinician can easily identify a patient by the NHS number and they must be able to do so to perform their jobs. Therefore, it is sensible to grant access permission to particular record parts on the basis of users' expertise. This expertise determines their actual needs of access, to the data parts they routinely work with and by doing so, healthcare roles are fulfilled. For example, pathology medical records or reports may be sent to a pathologist involved in a patient's care; prescription sent to a pharmacist; and sensitive parts not sent out at all. A specialist may have more control over their own partitions, e.g. write their reports or order certain tests, but limited permissions to other specialists' partitions or even not at all, e.g. to very sensitive medical test results.

2.5. *The access policies and principles pertinent to patients as individuals*

It is not rational to allow a professional to have access to all patient records, even if limited to the data subset fitting his/her expertise. Only relevant clinicians who have real life relationships with patients in clinical centres should access their records. This is documented in British Medical Association's security policy principles for clinical information systems (Anderson, 1996a), and the feasibility of adopting it has been evidenced in Denley and Smith (1999). Two major principles areas follows.

Principle of Access: "Each identifiable clinical record shall be marked with an access control list naming the people or groups of people who may read it and append data to it. The system shall prevent anyone not on the access control list from accessing the record in any way."
Principle of Control: "One of the clinicians on the access control list must be marked as being responsible. Only she may alter the access control list, and she may only add other healthcare professionals to it."

A named responsible clinician, possibly a patient GP, as in the UK or a primary care physician (PCP), as in the US, may setup a workgroup including the specialists who together deliver healthcare to the patient. According to the Principle of Access, it is the members of this group who will be in the patient access control list, as used by RBAC for files (Sandhu et al., 1996), have access to a subset of data they are responsible for, reflecting their job nature. The one who sets up the workgroup will let the system know the group members and their roles in the group, in accord with the Principle of Control. This implies a data ownership. Such a scheme decentralises management burden and increases scalability. The distributed environment and open access requirements suggest that a named doctor may involve specialists from other sites (remote consultants, temporary attending physicians, etc.) into healthcare procedures. For example, a medical opinion requested on a surgical patient may require a medical registrar, from other directorates, to exercise override access to that patient's notes (Denley and Smith, 1999). This is related with delegation (Zhang et al., 2002). Essentially, a responsible doctor grants access to local or remote users from trusted sites and occasionally, someone acts on their behalf, implying ownership transfer. A triangle relationship is described in Calam: 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 patient ("self-claimed" or "colleague-granted"/delegation).

3. Enhancing security in distributed healthcare

In fulfilling the requirements discussed in the previous section, we investigated a process calculus based messaging service that allows us to fragment and distribute clinical guidelines and protocols and a high-level knowledge representation paradigm to address knowledge/semantics interoperability issues. In the following, we first review existing approaches aiming at secure pervasive healthcare environments. We continue with a layered model and a brief discussion of its enabling technologies.

3.1. *Security domains and existing security solutions*

Security can be assured in different levels, operating systems, database systems, and applications. Some operating systems which take serious concerns of security include: OpenBSD, TrustedBSD, Trusted Solaris, Active Directory (as part of Microsoft Windows), and SELinux. Security-Enhanced Linux, or SELinux, is a Linux feature that provides a variety of security policies in the Linux kernel. It provides utilities to incorporate a strong, flexible mandatory access control (FMAC) architecture into the major subsystems of the kernel. In 2006, U.K. Cabinet Office backed SELinux and did experimental use of it to provide secure system access for the NHS's new finance system. Database security is the system, processes, and procedures that protect a database from unintended activity. Authentication, authorisation, auditing, and intrusion detection mechanisms are considered common database security measures. We will focus in this paper, however, the application level security where information sharing and resource access through the HealthAgents system must be under proper control. Access control at the operating system level provides the protection of "whose data is to be protected from whom" and a protection mechanism is the manner by which the operating system enforces the access control to its users. Access control at the HealthAgents system level requires the application to be designed in such a way that it recognises valid users, possibly from one site, and allow them to access resources, possibly from another site, under system-level constraints and mutually agreed policies between both sites, within an inter-connected network. This design provides an additional assurance on top of what will be secured in the operating system level and database system level, which must have been offered in the existing and standard environment but not under our control.

Two earlier access control models are *discretionary access control* (DAC) and *mandatory access control* (MAC). DAC is an access policy determined by the owner of an object. MAC is an access policy determined by the system, not the owner. An access control list (ACL), a list of permissions attached to an object, can be used by both models and applied in operating systems such as Windows. A newer access control model that supports efficient management is the widely accepted US National Institute of Standards and Technology model of *role-based access control* (RBAC) (Sandhu et al., 1996). All these models can be applied in the operating system level as well as the application level. Since no operating system can accommodate application-tailored requirements, adaptation of a suitable model for the system under consideration is required. In RBAC, 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. Privileges may be calculated as follows (M-Tech Information Technology, Inc., 2006):

$$Privileges = User\_Role * Role\_Definition \\ + Rules\_Function(User\_Attributes)$$

In addition to the static collection of rights accumulated by roles, a user can dynamically achieve extra rights if they expose certain attributes as defined by rules. This model is efficient when many users require the same set of rights in an organisation but otherwise unmanageable or even useless when roles vary in different conditions under which users act. In a hospital, roles can be defined for a number of classified groups to aggregate permissions, e.g. consultant, radiologist, nurse, who have static job functions. However, dynamic contexts exist in role playing, e.g. patients may be additionally assigned to or removed from a list for which a named doctor is responsible and this influences this doctor's role in caring these patients. RBAC has difficulties to capture such security-relevant contexts as patient, location, and time in healthcare environment (Zhang et al., 2002). Patient–doctor relationship is identified as a critical clinical security constraint to record access, described in Section 2.

The Community Authorisation Service (CAS) (Pereira et al., 2006) provides a solution to the management of user access control within Virtual Organisations (VOs) spanning over multiple sites in the Grid environment. It breaks the tradition of requiring each resource provider to maintain the mapping of individual users (across VOs) to its local database roles in order to authorise access to its resources. Using CAS, user memberships are instead based on VO roles and local resource providers only need to map these to local database roles. This dramatically reduces the number of mapping entries across resource providers and the duplicated maintenance burden put on them once a new user joins or a current user privilege changes. Such an approach requires no global user repository. However, a presumption of using the approach, as it is in RBAC, is that a large number of users can be grouped into several role groups requiring certain access levels in involved organisations. For the same reason that RBAC is infeasible to address the clinical requirement that information access or travelling may alter from patient to patient and user led as stated in the Principle of Access, the CAS is encountered with similar difficulties. Suppose clinicians A and B with the same speciality are from hospitals P and Q respectively. They will be categorised into the same VO role and the same access rights to data in P and Q. But in reality A shall have more privileges than B to certain data, e.g. of patients in P under A's care, and vice versa for B's privileges in Q.

Managing a resource access model is complex where there is a large number and various types of users, resource items, and access policies, user responsibilities being dynamic and ownership being distributed. The common practice of simply defining roles that aggregate all permissions required for the collection of resources to complete tasks is not realistic due to the diversity of individual needs which literally entails each individual having a distinct role. Even the burden of defining and maintaining a proper set of access control policies based on roles for automating authorisation could be considerable. A security solution must be able to cope with the complexity.

3.2. *Overview of a layered security model*

It has been pointed out that healthcare systems should be designed with multilateral security rather than multilevel security (Anderson, 1996b). Unlike some military systems which prevent information flow "down" from top secret to secret then to confidential, healthcare systems usually prevent information flow "across" from one clinician to another or from one hospital to another. This is evidenced by the requirements outlined in Section 2.4 and Section 2.5 where different access needs to cases and case partitions are distinguished due to distinct job responsibilities.

However, we argue a multilevel security model is more manageable, task availability being in the top level control and resource availability to tasks in lower level control. A multilateral security model resides in the lower level and complements the multilevel security model. The assignment of tasks to users is a business decision to be made by stakeholders, possibly explicitly in rules. It is sensible to regard the accessibility to tasks the organisational privileges with which organisation seniority is related and access to business functions restricted. Since tasks already exist in organisations and are routinely performed by specific user groups, they help to functionally decompose the system and ease security management. If a user can perform a specific type of task, then there must be certain resource items available to him/her to load into the task, if not all. Without the context of accomplishing one or more tasks in different privilege levels, information access makes no sense. The rationale of using a combined multilevel and multilateral model is further supported by the fact that a job responsibility is determined by the level of authority and the division of work (Crook et al., 2002). The former prevents information flow downwards and the latter prevents information flow across, being concerned about workgroup membership and job speciality under our further refinement. This forms a layered security architecture that addresses the healthcare security requirements.

1) Privilege of performing various types/levels of tasks and executing associated interaction models is determined by job title or grade/level. Users may upgrade their job titles occasionally and this is managed locally. Semantics of job titles and task collections must be globally defined and agreed among organisations.
2) Privilege of loading case instances for performing tasks (or enactment of interaction models) is determined by real life workgroup memberships or job boundary. This is managed by the locally named doctors, who shall be flagged as owners in case records' access control lists.
3) Privilege of accessing case record partitions (e.g. patient data, biopsy data, Microarray data, MRI and MRS data, diagnosis data, therapy data, surgery data, etc.) is determined by job nature or specialist one takes on in hospitals (e.g. oncologist, pathologist, radiologist, surgeon, etc.). This

is managed by system administrators when the account is setup and is maintained at a high level of stability.

This layered architecture can be seen as a hybrid of three different types of access control models that have been developed historically, DAC, MAC, and RBAC, as being defined and discussed in Section 3.1. Layer 1 of the architecture is a type of MAC, it is the system that constraints the subject to do various tasks, depending upon the actual level of that subject in the organisation. Layer 2 of the architecture is a type of DAC, it is the owner of the case records, or the named doctor, to restrict the access to the records based upon the identity of subjects or memberships to which they belong. Layer 3 of the architecture is a type of RBAC, it is the clinical job functions or roles that determine one's particular access privilege.

Thus, a user's overall privileges will be the sum of the user's access privileges in all tasks that the user is involved in (being a policy), each of which is decided by the particular cases he/she can operate as a workgroup member to deliver healthcare service (being a fact upon interaction instantiation) at the time of performing tasks, which in turn will be constrained by the accessible case partitions as determined by user professional roles (being a fact).

User Privileges $= \sum$ *(Privileged Interaction Model Type Selection * User Privileges in Interaction Model)*
User Privileges in Interaction Model = Interaction Model Function (User Privileges on Cases)
*User Privileges on Cases = User Workgroup Membership * (User Professional Role * Role Definition)*
→
User Privileges $= \sum$ *(Interaction Model Set as determined by job level * Interaction Model's Operational Cases as determined by job boundary * Case Subset as determined by job nature)*

Alternatively, the following meta-rule determines the prerequisite a user exercises privileges: a user has a title above the one required for running an interaction model, can load a case, that is under the care of a workgroup which the user is a member of, and perform operations on the case parts the user's specialists allow.

```
user_privilege (user, im, case, part, operation) ←
    job_title(user, title1) & executable(title2, im) &
    above(title1, title2) &member(user, workgroup) &
    responsible(workgroup, patient) & own(patient, case) &
    job_specialist(user, specialist) & rights(specialist, part, op)
```

Instances of this meta-rule include, a user can perform the operation of classifying case tumour types under their care, but not update the case profile (report, test, surgery, etc.) not in their specialist areas. Certain parts of the case, e.g. diagnosis results and treatment plans, may be updated by only a named doctor.

```
IF
    userA. responsiblePatientList. contains (patientB) &
    userA. specialiseIn (clinicalData. areaC) & areaC! = areaD
THEN
    userA. candoClassify (patientB. clinicalData) = = true &
    userA. canUpdate (patientB. clinicalData. areaD) = false
```

It is evident that any mechanism materialising the above multi-layered security model should facilitate the following. The first division, job-division, is based on job title which is normally enforced by healthcare providers to ensure a proper managerial chain and reporting hierarchy. Nurses naturally need to access different data than general practitioners and speciality registrars to carry out their duty. Either too much or too less data would render their effort sub-optimal or even futile. The second division, speciality division, is vertically among different specialities. As the medical domain is further divided, nowadays, neurosurgeons normally do not interpret biopsy slides directly which falls into the speciality of histopathologists. The third division, assignment division, is based on individual assignment. Each clinical staff has his/her own task-load. Unless there is a particular request, we assume that clinical staff do not normally have access to those cases that are not assigned to them directly. These three division inspire us to adopt a representation paradigm for capturing the security rules that focuses more on individuals' responsibilities than the actual persons carrying out such responsibilities. In the meantime, when searching for a proper formalism, we also need to bear in mind that in a distributed environment, applying rules is not straightforward. With more than one organisation involved in patient treatment and post-treatment management, centralised rule base is not strictly applicable. Fragmenting and allocating security rules to the concerned parties calls for new rule capturing paradigm. Combined with the division requests, this immediately suggests to us to take a process oriented view for system design and analysis. The formalism we use in the application is the Lightweight Coordination Calculus, LCC (Robertson, 2004) which is a logic programming language based on the low-level specification of something akin to Calculus of Communicating Systems (CCS) (Milner, 1980) or Pi-calculus (Milner et al., 1992). In pervasive healthcare environments, there is no single locus of control of task execution. Instead of the other resources existing merely to serve the control unit, these entities lead an autonomous existence and only undergo message induced transitions upon opening up access to each other – centralised control gives way to concurrent processes wherein each party accomplishes the tasks allocated to it and expose the results to accommodate the requests from the others. LCC prescribes concerned parties by specifying their responsibilities. Communication among different parties is regulated through messaging.

With the representation and rule capturing formalisms defined, we have to speculate on the rule reinforcement. A distributed environment introduces interoperability issues. On the one hand, different individuals participating in a data exchange task might maintain very different local vocabularies making a set of well-crafted rules invalid or falsely applied. On the other hand, the application of rules might result in a change of an individual's local knowledge by acting upon the status of a number of entities, e.g. the data that one possesses, the accessibility that one has on a particular part of the data, and the physical location of a piece of data. How to reflect and make explicit such changes becomes challenging when both data and data access are distributed. In order to address these two issues, in HealthAgents, we leverage a domain ontology as the common referencing point against which local views and vocabularies are juxtaposed. Rules, coded in LCC, are written in terms of the HealthAgents domain ontology and are interpreted thereafter. Rule segmentation and distribution is enabled by mechanisms native to LCC. We also propose using the Conceptual Graph (CG) based scheme to unify local views and offer a "reasonable" and "knowledgeable" interface to local data. Representing in CGs, apparently isolated data "islands" are interconnected together, waving into a landscape of one integral data network. Rule propagation and reinforcement among individual data holders subsequently can be carried out smoothly and seamlessly.

3.3. Lightweight Coordination Calculus (LCC) and secure interaction models

LCC, originally proposed in Robertson (2005), is a process calculus for specifying coordination among multiple participants. It does so by clearly stating what role an individual plays in a messaging process and thus what responsibilities that an individual should fulfil when interacting with others. An LCC model is built upon the principle that role-playing agents should obey the laws and/or protocols that are explicitly specified against the roles that such agents are expected to take. LCC ensures the fulfilment of roles by individuals through regulating the message-flows among them. These include: the messages that should be sent and are expected to be received and what constraints should be satisfied before a message can be handled. The full picture of LCC syntax is specified in Extended Backus-Naur Form (EBNF) as shown in Fig. 2.

In an LCC interaction model, we use predicate ***a***() to specify the role that an individual is playing, ⇒ and ⇐ to specify the direction of message flow, and ← for constraints. *Term* and *Constant* are implementation-specific. In the current version, *Term* is a well-formed formula in Prolog logic programming language and *Constant* is a Prolog constant starting with a lowercase letter. LCC also provides constructs for parallel (**par**), sequential (**then**), and switch branching (**or**) controls.

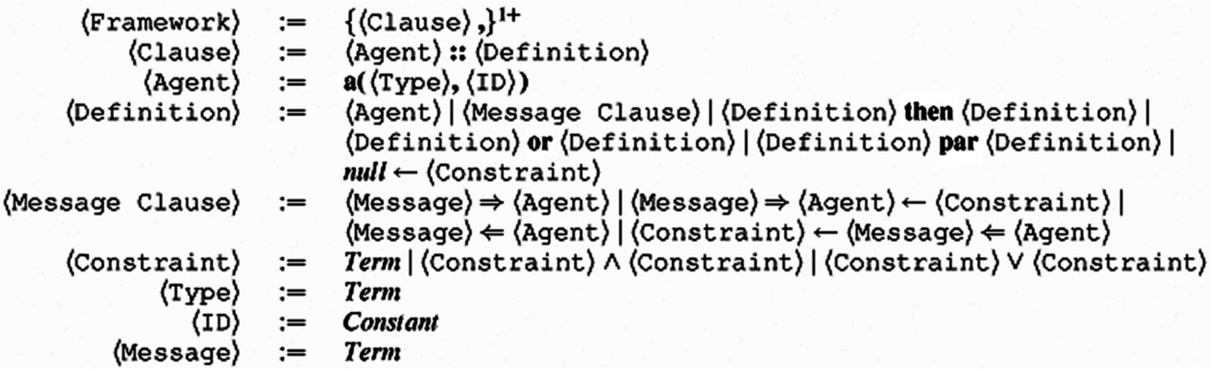
Interpreting LCC is tantamount to unpack LCC clauses, finding the next tasks that it is permitted to perform and updating the status of an interaction accordingly. A set of clause rewriting rules are introduced to ensure LCC constructs are interpreted in a consistent manner (Robertson, 2004). Let C_i be an LCC clause from a model M; I_i be a set of received messages currently queuing for an individual participating in an M-based interaction; C_{i+1} be the unfolded new LCC clause; $I_{i+1} \subset I_i$ be the set of remaining unprocessed messages; and O_i be the outgoing messages generated when processing C_i. An LCC model is interpreted by exhaustively unfolding clauses as detailed in Robertson (2004) to produce the following sequence:

$$C_1 \xrightarrow{I_1,I_2,M,O_1} C_2,\ \ldots,\ C_i \xrightarrow{I_i,I_{i+1},M,O_i} C_{i+1},\ \ldots,\ C_{n-1} \xrightarrow{I_{n-1},I_n,M,O_{n-1}} C_n,$$

The interpretation of LCC constraints depends on a particular implementation. In this paper, we assume Prolog as the underlying programming language and thus interpret the constraints in terms of a Prolog logic program. Nevertheless, this by no means denies the possibility of implementing LCC constraints with other programming languages, such as Java.

Pooling together the rewriting rules for LCC-specific constructs and the interpretation of a Prolog program, we obtain the semantics of LCC models. For instance, in the above LCC interaction model, the sequence construct **then** is unfolded by examining the first part of the sequence or, if it is closed (i.e. executed), unfolding the next part. After unfolding, the system tries to instantiate all the variables (e.g. *P* and *A*) to examine the satisfy-ability of LCC clauses. A narrative interpretation of the LCC model in Fig. 3, therefore, reads "when an on-call-doctor receives a routine check request on a patient (*P*), he/she first asks an arbitrary nurse (*S*) to take *P*'s body temperature. When the body temperature is done, he/she asks an arbitrary nurse (*T*) to take *P*'s blood sample if *P* has not been given blood test before." Note that whether nurse S and T are the same person is unknown from the context.

LCC lays down a nice framework wherein authentication, authorisation, data integrity and data encryption issues can be seen as constraints and message passing sequence among different parties. The role-playing nature of LCC interaction



```
      ⟨Framework⟩  :=  {⟨Clause⟩,}^1+
         ⟨Clause⟩  :=  ⟨Agent⟩ :: ⟨Definition⟩
          ⟨Agent⟩  :=  a(⟨Type⟩, ⟨ID⟩)
     ⟨Definition⟩  :=  ⟨Agent⟩ | ⟨Message Clause⟩ | ⟨Definition⟩ then ⟨Definition⟩ |
                       ⟨Definition⟩ or ⟨Definition⟩ | ⟨Definition⟩ par ⟨Definition⟩ |
                       null ← ⟨Constraint⟩
 ⟨Message Clause⟩  :=  ⟨Message⟩ ⇒ ⟨Agent⟩ | ⟨Message⟩ ⇒ ⟨Agent⟩ ← ⟨Constraint⟩ |
                       ⟨Message⟩ ⇐ ⟨Agent⟩ | ⟨Constraint⟩ ← ⟨Message⟩ ⇐ ⟨Agent⟩
     ⟨Constraint⟩  :=  Term | ⟨Constraint⟩ ∧ ⟨Constraint⟩ | ⟨Constraint⟩ ∨ ⟨Constraint⟩
           ⟨Type⟩  :=  Term
             ⟨ID⟩  :=  Constant
        ⟨Message⟩  :=  Term
```

Fig. 2 – Grammar of LCC.

$$
\begin{array}{l}
a(\text{on\_call\_doctor}, N) :: \\
\quad \mathit{routine\_check}(P) \Leftarrow a(,A) \ \textbf{then} \\
\quad \left(\begin{array}{l} \mathit{take\_temperature}(P) \Rightarrow a(\mathit{nurse}, S) \ \textbf{then} \\ \mathit{take\_blood\_sample}(P) \Rightarrow a(\mathit{nurse}, T) \leftarrow \neg \mathit{blood\_test}(P) \end{array} \right)
\end{array}
$$

Fig. 3 – An example of LCC.

models allows us to easily translate staff responsibilities into behaviour-specifying LCC clauses (as demonstrated in Fig. 3 defining the responsibility of an on-call-doctor). Security rules are then imposed on whoever is committed to fulfil these responsibilities. In the meantime, assignment division is enforced by treating different instantiations of an interaction model as independent cases. The clear separation between models and instances ensure that security rules only take effective on an instance and thus clinical staff "playing" in an instance. Finally, the speciality division can be implemented either explicitly through prescribing the behaviour of a role or indirectly in LCC Constraints. Synchronising through message passing ensure autonomy and transparency of role-playing individuals and in the same time provide systematic checkpoints that whether everyone fulfils his/her duty or whether a security constraint is properly checked and satisfied.

Here is an exemplar data transferring interaction model. Upon receiving a request of patient's data, one might check whether the data requester is what he/she claims to be by asking for an authentication message, whether the data requester has the privilege to view the entire patient record or part of it by looking up the access policy associated with his/her ID, etc. Fig. 4 illustrates fragments of an LCC interaction model that retrieves data based on the request submitted by an arbitrary domain specialist. It is evident that whether or not a particular specialist is qualified to receive the requested data can be crafted as data-specific evaluation using *is_qualified* (*E*, Patient). Meanwhile, this example interaction model also emphasises on the customisation of data transfer methods. We use *trans_method* (*E*, *M*) to state that the data transfer task is specific to a particular specialist.

The running of all above example LCC model specification for healthcare can be supported by the Openknowledge (Robertson et al., 2006) kernel. Next, we use LCC for the modelling of the HealthAgents system. The clinical decision support system has been implemented and tested, within the HealthAgents project. In this paper, we explore the use of LCC models in bringing better knowledge sharing capabilities for healthcare professionals in a decision support system and at the same time enforce better access control. The system prototype has been built and it is part of our future work to test the use of the developed Openknowledge kernel for the HealthAgents system.

4. Security in HealthAgents: a comprehensive case study

In this section, we present in-depth details of the HealthAgents system, the elicitation of interaction models, and their secure running in our layered security model for distributed healthcare applications. Meanwhile, as discussed in the previous section, our vision in secure pervasive healthcare systems relies on a mutual understanding of the case at hand. We elaborate an ontology and a conceptual graph based mechanism that work alongside with LCC interaction models.

4.1. *HealthAgents architecture and the aimed secure system access logic*

The HealthAgents system (Fig. 5) is a distributed decision support system that supports diagnosis and prognosis, employs a set of distributed nodes that either store patient case data, build classifiers that are trained upon case data and capable of classifying tumour types, or use classifiers for the diagnosis and prognosis of brain tumours. The magnetic resonance spectroscopy (MRS) data used by the system is built up using anonymous information from child and adult cases. Classifiers are created by the producer nodes that receive requests from the clinicians to generate classifiers for particular tumours. Clinicians with cases will employ classifiers to assist in the diagnosis of patients for particular tumours. The HealthAgents system consists of a variety of agents each charged with a different task. A more detailed description of the HealthAgents components and architecture can be found in Xiao et al. (2008). For the need of open access (a requirement

$$
\begin{array}{l}
a(\text{datahandler}, H) :: \\
\quad \mathit{patient\_record}(\text{Patient}, M) \Leftarrow a(\text{specialist}, E) \ \textbf{then} \\
\quad \mathit{register\_clearance}(E, \text{Patient}) \leftarrow \mathit{is\_qualified}(E, \text{Patient}) \ \textbf{then} \\
\quad \mathit{inform}(\text{Patient}) \Rightarrow a(\text{specialist}, E) \ \textbf{then} \\
\quad \mathit{get}(P, F) \Rightarrow a(\text{DataMart}, D) \leftarrow \mathit{registered}(D) \wedge \mathit{contains}(D, P) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \mathit{matches}(P, \text{Patient}) \wedge \mathit{trans\_method}(E, F) \\
\quad \ldots
\end{array}
$$

Fig. 4 – Ultrasound result evaluation.

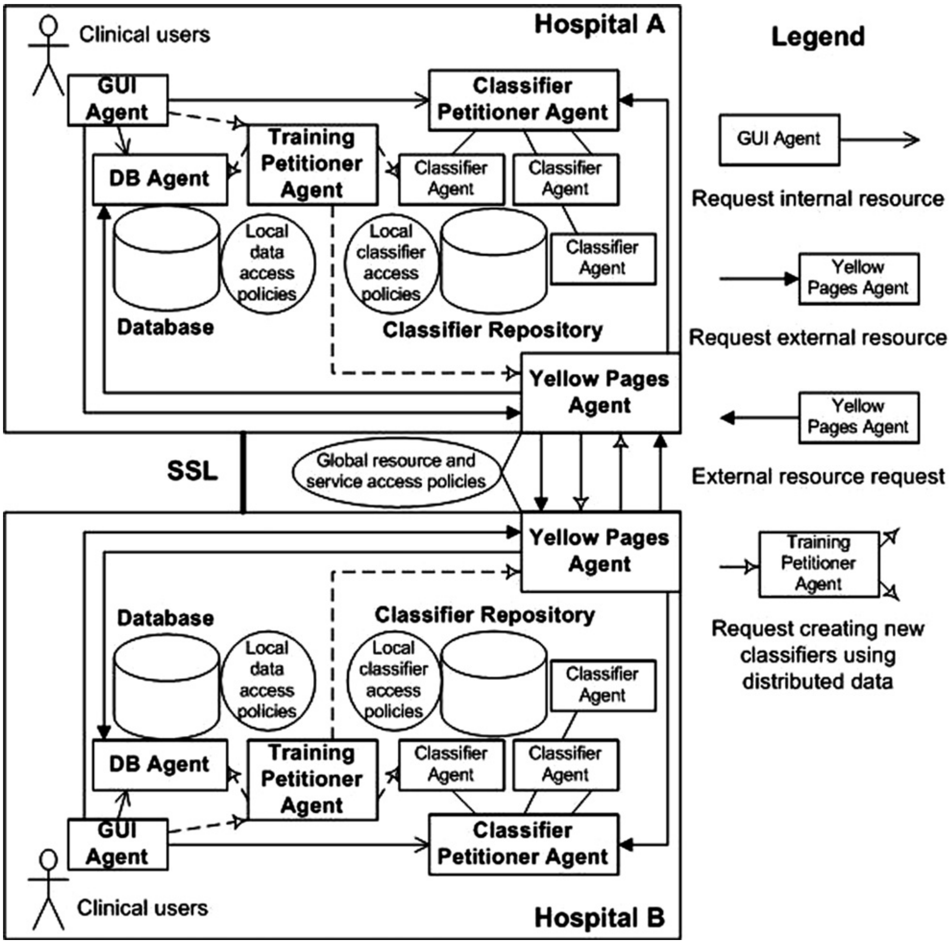

Clinical users
Hospital A
GUI Agent
Classifier Petitioner Agent
Training Petitioner Agent
DB Agent
Classifier Agent
Classifier Agent
Local data access policies
Local classifier access policies
Classifier Agent
Database
Classifier Repository
Yellow Pages Agent
SSL
Global resource and service access policies
Yellow Pages Agent
Database
Classifier Repository
Local data access policies
Local classifier access policies
Classifier Agent
DB Agent
Training Petitioner Agent
Classifier Agent
Classifier Agent
GUI Agent
Classifier Petitioner Agent
Clinical users
Hospital B
Legend
GUI Agent
Request internal resource
Yellow Pages Agent
Request external resource
Yellow Pages Agent
External resource request
Training Petitioner Agent
Request creating new classifiers using distributed data

Fig. 5 – The HealthAgents system architecture and resource access flow control.

described in Section 2.3), human readability and transparency should be minimised while resource availability maximised. This is achieved in HealthAgents via using classifiers. Knowledge extracted from cases is implicitly involved for decision making. Classifiers are developed and trained from relevant cases and classifiers instead of the actual cases will be used for facilitating the diagnosis decisions on cases coming up subsequently, the case profile statistics for training classifiers being referred for classifier selection. The HealthAgents can thus achieve its goal of facilitating brain tumour diagnosis by using the distributed knowledge base without compromising privacy.

When a user logs in, a patient case is retrieved and then, relevant classifiers will be invoked, after that the classification is performed upon the case and finally, the diagnosis results updated as well as the ranking of involved classifiers. Such a procedure for performing a task is as follows.

1) User account setup and his/her professional specialist and interaction model availability binding (preparation).
2) Login and authentication (locally).
3) User ID and task availability matching, accessible interaction model presentation.
4) User functional role determination and playing in a selected interaction model.
5) A set of cases the user is responsible for will be made available, presented and selected for interaction model execution.
6) A subset of case records may be visible and of manipulability to the user during the performance of the task.
7) Specific local policies may apply to add extra constraints to the particular access.

4.2. *Building an interaction model hierarchy with a goal-decomposition graph*

Four major interaction models, as shown in Fig. 6, are identified: create classifier, execute existing classifier, update classifier reputation value, and update case profile. They are elaborated as four sub-goals under the root goal of "tumour type diagnosis" via a goal-decomposition graph, useful for requirements analysis and interaction model identification. A detailed goal decomposition procedure and underpinning process elicitation can be found in Xiao and Greer (2009).

Table 1 describes a specific branch of the graph, where "Tumour type diagnosis" includes "Update case profile" which in turn includes "Classify case". It is identified in the table that, the job levels the users must reach in order to execute such interaction models or tasks; the participant components that

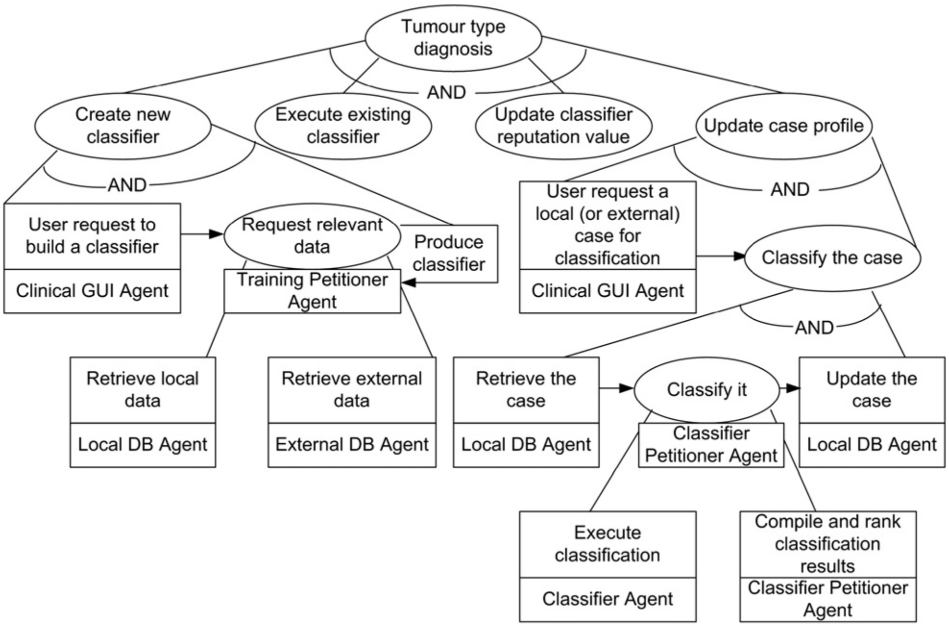


Fig. 6 – The goal-decomposition graph for HealthAgents.

form the interaction models, and the executing constraints. Further discussion of the interaction model ‘‘Update case profile’’ and its specification, based on this initial identification, will be given in the following sections.

4.3. *Secure interaction models and Lightweight Coordination Calculus (LCC)*

Bearing in mind the fact that existing organisational structure resists a common job level hierarchy, forcing different organisations to agree upon and change to the use of the same set of job titles is not an option. A similar issue was seen in managing user roles in local and global contexts. Mapping individual users across Virtual Organisations (VOs) to local database roles requires unnecessary but significant maintenance efforts for authorising resource access among multiple sites. The solution of CAS, as discussed in Section 3.1, introduces the mapping between VO roles and local database roles and this technique is adopted here. We introduce a global HealthAgents job title hierarchy and it is up to each individual organisation to map their internal job title structure to the items in the hierarchy. All security policies will be defined upon global job titles which will be mapped from individual local job titles. Consequently, there is no need that all participant organisations must assume the same set of job titles in order to make the scheme work. At the same time, a large number of mappings between clinicians to local job titles are avoided. In Fig. 7, for example, 5 job titles in VO1 and 3 job titles in VO2 are mapped to 3 global items, correspondingly. The mapping in VO2 is straightforward. In VO1, Levels 1&2 are mapped to a senior, Levels 3&4 to a principle, and Level 5 to a trainee. It would also be possible to map Level 1 to a senior, Levels 2&3 to a principle, and Levels 4&5 to a trainee. It is a business decision to do mapping in one way or another and grant access power to different levels according to the business strategies.

Assume in the global HealthAgents job title hierarchy, there are three job titles, senior clinical consultant, principal clinician, and trainee clinician, in that order, forms the existing clinical hierarchy, from top to bottom. Roles in a role hierarchy of RBAC have inheritance relationships. Likewise, a job title higher up in the hierarchy inherits task execution privileges from a job title further down in the hierarchy. Suppose the following rules in HealthAgents restrict task availability.

Table 1 – A high level view of selected interaction models.

Goal	Sub-goals (Interaction model)	Interaction model privileges	Interaction model participants	Interaction model constraints
Tumour type diagnosis	Update case profile, etc.	N/A	All	N/A
Update case profile	Classify case	Principle clinicians or above	GUI Agent, DB Agent, Classifier Agent, and Classifier Petitioner Agent	The clinician can update the specialised data areas
Classify case	N/A	Trainee clinicians or above	Classifier Agent, and Classifier Petitioner Agent	The clinician must be a workgroup member taking care of the case

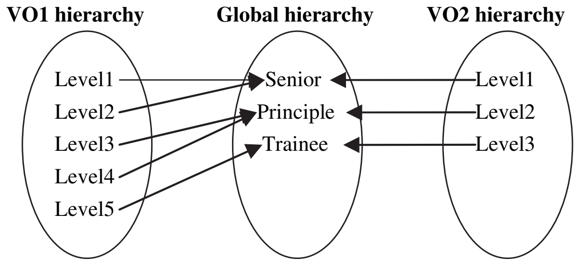


Fig. 7 – Mapping between different organisation job levels to those in a global hierarchy.

- Rule1: Senior clinical consultants can identify the need of new classifiers in the network and so are able to create classifiers, using all public cases and local private cases.
- Rule2: Principal clinicians have primary healthcare responsibilities and so are able to run classifiers, update case profiles and diagnosis results, as well as update classifier reputation values.
- Rule3: Trainee clinicians assist in healthcare and can run classifiers and be advised of classification results.

Gaia (Wooldridge et al., 2000) is a methodology for agent-oriented analysis and design, and has a view of a multi-agent system as a computational organisation consisting of various interacting roles. In Gaia, responsibilities and permissions are unified in a single role notion. It is also recognised in Omicini et al. (2005) that the coordination among agents/roles and resources must enable authorisation policy specification over interaction specification to achieve an expressive and safe interaction model. Thus, role, interaction, and constraint should be correlated. The descriptive interaction behaviour which consists of message passing and constraint solving have been defined in Lightweight Coordination Calculus (LCC) (Robertson, 2005) that can be transmitted, interpreted, and executed by agents in the network. The LCC language has been developed in the OpenKnowledge project (Robertson et al., 2006) and it uses logic expression to regulate the message exchange protocols among participant peers each of which plays a particular role.

The LCC language combines role functions and constraints in a single framework and this gives us the opportunity to express permission enforcement prior to responsibility fulfilment within role playing behaviour, in the context of running interaction protocols. The following LCC clauses describe the fundamental interaction pattern for resource access control.

$$
\begin{array}{l}
a(resource\_request, PRID) :: \\
\quad request(Resource, Operation, Context) \Rightarrow a(resource\_manager, RMID) \\
a(resource\_manager, RMID) :: \\
request(Resource, Operation, Context) \Leftarrow a(resource\_request, PRID) \\
\leftarrow grantPermission(PRID, Resource, Operation, Context, Policies)\ \mathbf{then} \\
\left(\begin{array}{l}
response(Grant\_yes) \Rightarrow a(resource\_request, PRID) \\
\mathbf{or} \\
response(Resource\_result) \Rightarrow a(resource\_request, PRID) \\
\leftarrow getOperationResult(Resource, Operation, Access\_result)
\end{array} \right)
\end{array}
$$

Briefly, a(resource_request, *RRID*):: Def_{RRID} and a(resource_-manager, *RMID*):: Def_{RMID} denotes that agents RRID and RMID play the roles of resource_request and resource_manager respectively as defined in the definitions follow. Def_{RRID} has a single and Def_{RMID} has a composite message passing behaviour. In the above role definitions, a message of resource access request is sent from the agent that plays the request role to the agent that plays the manager role. Upon receipt of this message, the resource manager agent applies appropriate security policies and responds by sending back a message either saying the request has been granted (or rejected) or by providing the actual resources (or the results of their usage) being requested. In the Def, ← $Cons_n$ denotes that a constraint must be satisfied (as some running code) before the clause prior to it.

The notion *a*(*id, role*) defines the role a certain agent should play and its identity can be bound with executable tasks, workgroup memberships, and professional specialists at runtime. The role playing behaviour defines the common responsibilities an entitled user supposed to fulfil, being in a position with/above a given title as are in Gaia, the organisational roles in well-defined positions associated with expected behaviour. Then the memberships and professional specialists further constrain the concrete resource usage in the role's interaction model participation, being identity-specific and role-independent. This layered architecture is discussed as follows, illustrated by a principal clinician updating case profile after classification.

4.3.1. *Level 1: interaction model constraints*

The first layer filters interaction model availability. A principal clinician (possibly a GP) can load cases for which they have caring responsibilities and later update its profile (diagnosis result, etc.). A junior clinician can perform classification but cannot do the update. Fig. 8 shows the interaction model. In the diagram, messages flow (represented by arrows) among agents (represented by rounded-corner rectangles) which digest and produce messages by playing roles (represented by circles). The role playing behaviour in the interaction model is as follows. In the beginning, a clinician requests patient data for classification. When the record is retrieved from database, it is requested to a petitioner for classification. Then, a set of relevant classifiers will be executed upon the case, and ranked classification results will be sent back to the clinician for decision support. After the real diagnosis result is known, the patient record will be updated, as well as the reputation of the executed classifiers.

The following LCC clauses show part of the specification of the interaction model. The clinician plays a role of classification (R1) and updating case profile (R5). The role changes when an accurate diagnosis result is known.

```
/*R1: classify a case */
a(clinician_classify, CID)::
    requestCaseRecordByID(I) ⇒ a(database, DBID) then
    caseRecord (R) ⇐ a(database, DBID) then
    requestClassification(R, C) ⇒ a(classifier_petitioner, CPID)
    then
    classificationResults(S) ⇐ a(classifier_petitioner, CPID) then
    a(clinician_followingdiagnosis, CID)
/*R5: update case record and classifier reputation following diagnosis */
a(clinician_followingdiagnosis, CID)::
   (updateCaseRecordByID(I) ⇒ a(database_update, DBID)
   then
     caseRecordUpdated(Y) ⇐ a (database_update, DBID))
   par
   (updateClassifier(I) ⇒ a(classifier_petitioner, CPID) then
     classifierUpdated(Y) ⇐ a (classifier_petitioner, CPID))
```

4.3.2. *Level 2: case level constraints*

An interaction model is uniquely defined and its running context varies, e.g. involved clinicians and cases. A resource manager must check the request (resource and operation) against the requester identity at runtime, in compliance with the access policies. Specifically, the clinician must be a member of the workgroup delivering care to the owner of the case before the case is allowed to be updated, being a meta-rule of healthcare access control. Additional local policy rule satisfaction must also be considered for extra constraints, e.g. a particular clinician can/cannot access particular resource items. A generic security policy schema for healthcare is described in Xiao et al. (2007) that can complement the meta-rule with any number of specific policies. The following shows the LCC constraints used by the database agent, being a resource manager, for permission checking before the actual role functions are carried out. The database agent issues a case record (R2) and updates the same record (R6), different levels of permissions being needed.

```
/*R2: send a case record for classification */
a(database_download, DBID)::
   requestCaseRecordByID(I) ⇐ a(clinician_classify, CID)
     ←grantPermission(CID, I, Read, Normal_classify_from_local_site, Local_database_read_policy_set) then
   caseRecord(R) ⇒ a(clinician_classify, CID) ← getCaseRecordByID(I, R) then
   a(database_update, DBID)
/*R6: update a case record after classification */
a(database_update, DBID)::
   updateCaseRecordByID(I) ⇐ a(clinician_followingdiagnosis, CID)
     ←grantPermission(CID, I, Update, Normal_update_from_local_site, Local_database_update_policy_set) then
   caseRecordUpdated (Y) ⇒ a(clinician_followingdiagnosis, CID)
```

It is at the point of checking the LCC constraint of “grantPermission” that user workgroup and case will be related (clinician identity of CID and case identity of I), and other locally set read or update policies applied, prior to the required operation. A clinician not in the right workgroup may be able to download a case but cannot update it. The running and execution of LCC specification is supported by the OpenKnowledge kernel.

4.3.3. *Level 3: case partition constraints*

Similarly with level 2, a user identity is bound with professional specialists at runtime and this will constrain further permission to case partitions, e.g. only the named clinicians may update or write major diagnosis results; certain specialists may write reports in their areas; others on the case care list may only read those areas. Thus, a three dimension resource request of (user, resource, operation) will be constrained in two dimensions: user-resource must match workgroup membership and user-operation match job specialist.

The layered security model empowered by LCC running in a distributed clinical environment, as discussed above, must be able to enable interoperability if different clinical sites have various ways of knowledge representation, e.g. different languages may be used to describe their resources, database schemas may vary from one dataset to another, policy descriptions may be annotated differently with different vocabularies. Even further, unless resource access requests can be precisely understood by the system as well as the corresponding related resources and associated regulation policies in a single intelligent framework, users will experience frustration due to the lack of mapping and reasoning capabilities in the system. We discuss in the following the extra power our security model posses in an interoperable environment offered by the HealthAgents domain ontology and the Conceptual Graph approach.

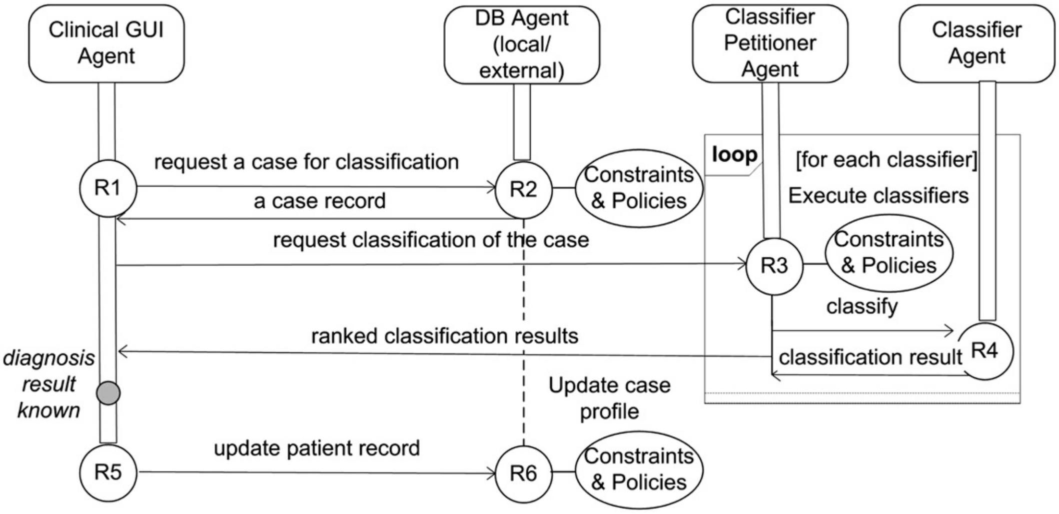


Fig. 8 – Interaction Model: update case profile (including case classification).

4.4. *Fuelling rules with a domain ontology*

In the following we will focus on the problem of representing, in a meaningful way, the knowledge involved in the HealthAgents project and the resulting security mechanism. We regard knowledge representation to be a (1) surrogate, (2) a set of ontological commitments, (3) a fragmentary theory of intelligent reasoning, (4) a medium for efficient computation, and (5) a medium of human expression. We will explain the reasons why our choice of Conceptual Graphs (Sowa, 2000) in the context of reconciling different perspectives of the domain of discourse in reinforcing security rules.

The problem of representing healthcare information (e.g. Electronic Healthcare Records, EHRs) about an individual has been a key research field in medical informatics for many years. Such information (Iakovidis, 1998) (which can include tests, observations, imaging information, diagnostics, patient identification, legal permissions) has either been stored in a structured document based format (e.g. relational databases etc.) or unstructured document based format (e.g. photocopied hard copies). EHRs are difficult to represent, in a consistent manner, due to their content complexity. However, information, in this paper we follow the work of Aamodt (2004) to distinguish between data, information and knowledge, interoperability (Brown and Reynolds, 2000) will benefit pervasive patient care as it will allow for exchange of data between multiple sites. This is important in the context of this project where we expect hospitals from different parts of the world to join the HealthAgents network and therefore, make the security issues crucial in the project development and the subsequent system deployment.

In order to address the interoperability shortcoming a number of standards have been proposed in the literature. A few examples that attempt to represent EHRs include Health level 7, Davis et al. (1993), Openehr and Clunie (2000). The aim is to structure the knowledge (using markup techniques) so that the clinical content is precisely identified. The ability to uniquely refer to a piece of information is denoted, in the context of these standards, as ‘‘semantics’’ since it allows the identification of the meaning of the knowledge. In this paper, however, we claim that this representation expressiveness is not sufficient for information retrieval. In the spirit of Mugnier (2000) we define semantics as the capability of inferring (reasoning) implicit knowledge from the knowledge base (based on explicit knowledge and given rules). This is important for HealthAgents as we seek to not only present information, but indeed to understand the information – brain tumour information could influence the patient diagnosis and prognosis.

In HealthAgents we developed HADOM (HealthAgents Domain Ontology) which conceptualises the parameters of the employed techniques (MRI, MRS, DNA Microarrays, etc.), the clinical information (age, sex, tumour location, etc.) and the known brain tumour classes compliant to WHO (World Health Organisation). For instance, the structure ‘‘medical control’’ contains information related to different MRI, MRS, etc. tests underwent by a patient. The HADOM ontology provides a basic terminology for the HealthAgents database schema and allows for interoperability at the terminological level. This is illustrated in Fig. 9. Furthermore, for managing security rules and appropriate reasoning we propose a Conceptual Graph based description of the different inter and intra hospital rules.

In HealthAgents we need to integrate medical knowledge from different sites and retrieve it in an intelligent manner. This retrieval has to be based on a set of rules that regulate the access to data. These rules have been explained in full detail in the previous sections. It is evident that we need a flexible mechanism for data representation and querying.

Primarily, the data in the HealthAgents system is stored in relational databases at the various participating European clinical centres. A uniform vocabulary needed for interoperability reasons is provided by means of HADOM. The patient concept is at the centre of HADOM (see Fig. 10(a)). Each visit of a patient is given a unique ID to be differentiated from other EHR regarding the same person. A particular patient instance, therefore, has several associated patient records. Tissue focus defines instances of the concerned areas under two sub groups, namely Primary_Focus and Secondary_Focus. Patient Record is linked up to main HADOM ontological concepts such as Symptom, Diagnosis, Clinical Centre, Clinical Intervention, Medical Control. This is visually represented in Fig. 10 by directed links between the nodes representing the concepts. Different colours have been solely used for visualisation purposes and have no semantics. In Fig. 10(b) one visit of a patient is depicted with the diagnosis further detailed by Tumour Grade, Daumas Duport Grade, Region of Interest and Histopathology. In both images the direction of arrows represents how the information is accessed and the concepts queried. A particular focus is related to the visit of a patient via Patient_Record in HADOM (see Fig. 10(b)). Many medical instruments and methods have been developed to diagnose brain tumour. In HADOM, we enumerate the following approaches and define them as sub-concepts of Medical_Control: Biopsy, HRMAS, Magnetic_Resonance and Microarray.

The problem with representing EHRs in this format is that certain rules that can help retrieve implicit knowledge are hard to represent. Indeed, mutual understanding among software agents is partially rooted in a commonly agreed vocabulary/terminology in the brain tumour domain when such agents need to communicate with each other to express things like ‘‘retrieve cases of all patients under age 5’’ and ‘‘fetch a case of glioma from Hospital A’’ where underlined words are concepts from HADOM. That is to say, the domain ontology captures only the static model rather than the inference procedures. We would like to be able to express statements like ‘‘due to the fact that [...] the tumour is malignant’’’ or ‘‘all peak areas with [...] characters suggest [...]’’. Such separation (static model rather than inference procedures) is based on both theoretical and practical considerations. On the one hand, such inferences are built using rules, machine learning techniques, etc. which, currently, are not ready to be combined with major knowledge representation and reasoning formalisms, e.g. Description Logic, Frames, Entity-Relationship Model, etc. On the other hand, a medical diagnosis is normally a complicated process with ambiguity and uncertainty which cannot be entirely and

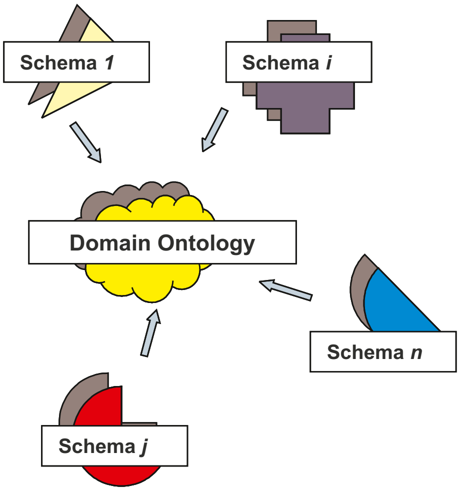
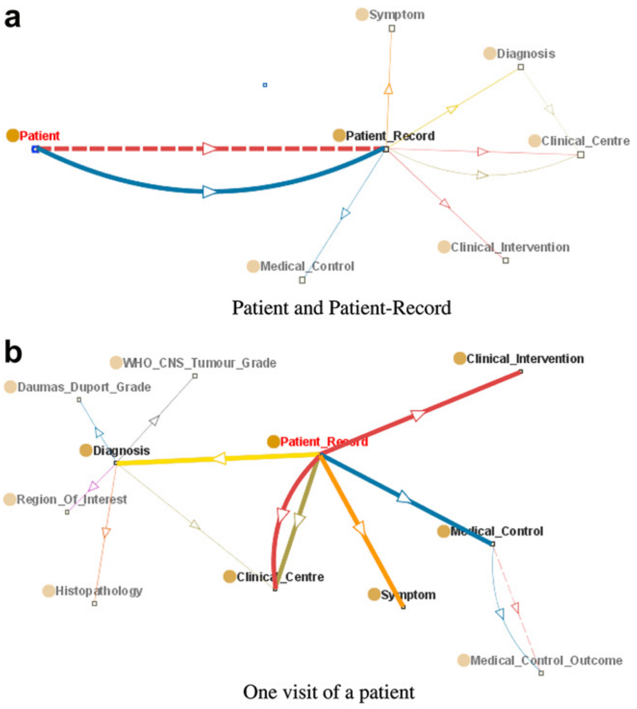


Fig. 9 – Ontological interoperability of HealthAgents database schema.

precisely formalised in an inference model based on taxonomic knowledge. This, however, does not deny the merit of building a reasoning system on top of HADOM to provide moderate suggestions and warnings to clinicians. Such reasoning capability would be more appropriate to perform simple and specific tasks. This sort of extra reasoning power will also allow one to check for consistency within the HealthAgents ontology.

The extra expressivity needs dictated by the necessity of security rules that should be enforced let to the proposal of using Conceptual Graphs for representing such rules. The advantage of this approach primarily lies in the ease of match checking between the local hospital rules and the global ones.



Patient and Patient-Record

One visit of a patient

Fig. 10 – Conceptual view of HealthAgents HADOM.

This is based on the querying mechanism for Conceptual Graphs, projection, and will be described below. In the next section we informally introduce Conceptual Graphs and further explain our choice of knowledge representation formalism also in the context of the five roles enumerated in the previous Section.

4.5. *Enhancing security model with inference: the conceptual graphs approach*

Conceptual Graphs represent background knowledge, i.e. basic ontological knowledge, in a structure called support, which is implicitly used in the representation of factual knowledge as labelled graphs. A support consists of a concept type hierarchy, a relation type hierarchy, a set of individual markers that refer to specific concepts and a generic marker, denoted by *, which refers to an unspecified concept. The support defines the main concepts and relations that exist in the world we are trying to describe. These concepts and relations are going to be linked together by the means of an ordered bipartite graph that will describe the facts we are interested in. The ordered bipartite graph is going to represent the "stencil" which is going to be "filled in" with the concepts/relations taken from the support. A CG can be viewed as a bipartite graph that provides a semantic set of pointers to two ontologies. This means that we can reuse sources' ontologies, database schemas etc. for the purpose of describing those sources by the means of a CG. Moreover, the attached semantics of Conceptual Graphs make them a powerful reasoning knowledge representation and reasoning formalism. CG reasoning mechanisms can be viewed as a powerful tool for the querying process.

Layered Conceptual Graphs (LCGs for short) is a rigorously defined representation formalism evolved from Conceptual Graphs. It allows highlighting a new type of rendering based on the additional expansion of concept/relation nodes. This way hierarchical knowledge can be represented in a mathematically sound manner. The semantics associated with layered conceptual graphs are based on the semantics of conceptual graphs.

LCGs preserve the bipartite graph structure of the original model by defining transitional descriptions which allow a successive construction of bipartite graphs. Unlike existing approaches the knowledge detailed on a level of a hierarchy is put in context by using descriptions for relation nodes as well. A transitional description of a bipartite graph G provides a set D of complex nodes in one of the classes of the bipartition, each complex node having associated a description. Complex nodes are visually depicted in bold. Their descriptions are disjoint bipartite graphs. The neighbors of complex nodes either have empty descriptions or are described as bipartite graphs. These bipartite graphs contain in one of the classes of the bipartition, (VC), all the atomic neighbors of the initial graph. The remaining nodes in each of these classes are new nodes or are taken from the descriptions of the corresponding complex neighbors of the initial graph. In other words, if we have a inter-connected world described by a CG and if we can provide details about both some complex concepts and their relationships, then we can construct a second level of knowledge about this world, describing these new details as

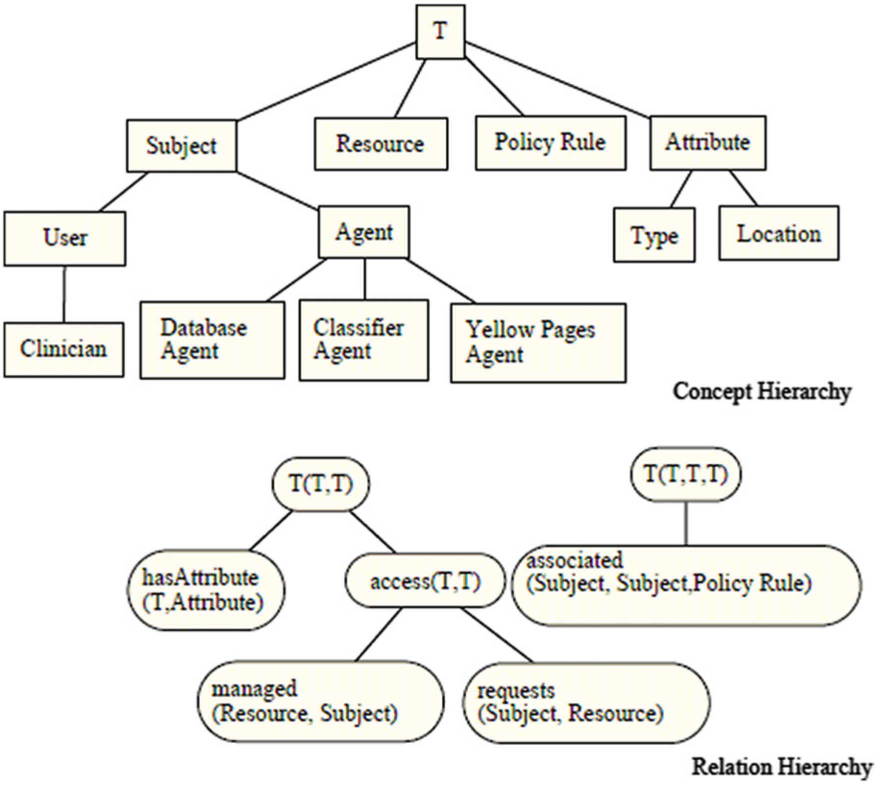


Fig. 11 – Support for the framework.

Conceptual Graphs and applying the corresponding substitutions. This process can be similarly performed with the last constructed level, thus obtaining a coherent set of layered representations of the initial world. We will use Layered Conceptual Graphs for representing the policy rules and then their associated "expansion" properties for highlighting the interdependencies between such rules.

Fig. 11 depicts the support for our framework. Please note that the support is not exhaustive, being intended for illustration purposes only. The concept hierarchy is comprised of the top, universal type, further refined as a subject, resource, policy rule or attribute. Policy rule is a stand alone concept as one of our aims is to represent their interdependencies. The agents are further specialised in database agent, classifier agent and yellow pages agent. The relation hierarchy is made out of binary relations: access and attribute; and ternary relations: associate. For simplicity reasons we only consider two very generic access relations: managed and requests.

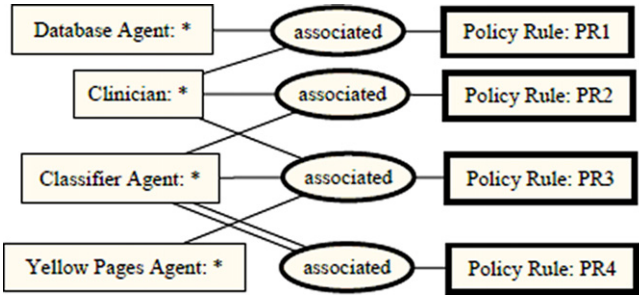


Fig. 12 – Example of rules.

In Fig. 12 a bipartite graph is depicted for four policy rules. The policy rules are depicted on the right hand side of the picture while the subjects are represented on the left. To increase readability the edges are not explicitly ordered in the diagram. The bolded out nodes stand for complex nodes, that is, nodes can be further expanded. The four agents from the interaction are:

1. Clinical GUI Agent: the clinician, working in a given hospital, requesting the d-DSS for a case to be classified. In Fig. 12 we used the term "clinician" for clarity purposes.
2. Database Agent: gives access to the data from a given hospital.

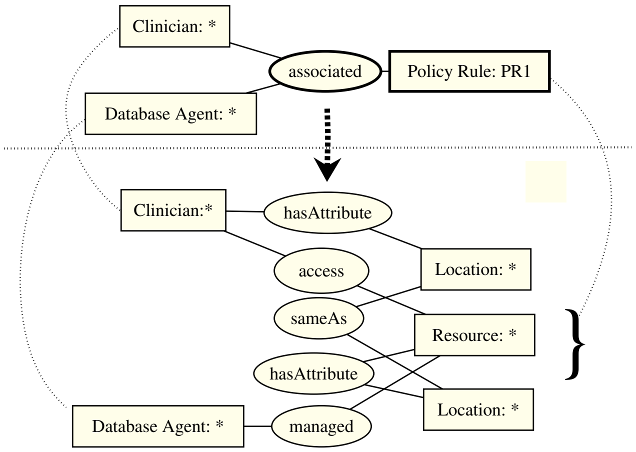


Fig. 13 – Local Policy Rule.

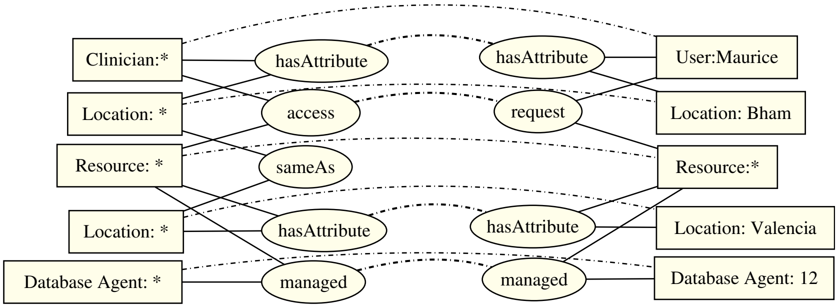


Fig. 14 – Local – Global Policy Rules Projection.

3. Classifier Agent: a software that classifies brain tumour cases based on their characteristics (MRS spectra, case meta-data, etc.)
4. Yellow Pages Agent.

The policy rules depicted in Fig. 12 address the following scenarios:

PR1: A clinician wants to view data from a hospital.
PR2: A clinician directly asks a specific classifier for a case to be categorised.
PR3: A user asks the yellow pages for a classifier and the classifier is found by the yellow pages.
PR4: Classifiers want to exchange information for combination.

We return to the previous example to intuitively explain our approach in this section. This section will only present the rationale for the expressivity provided by the Conceptual Graphs and detail the process of constraint matching.

$$a(resource\_request, RRID) :: \\ request(Resource, Operation, Context) \Rightarrow a(resource\_manager, RMID)$$

$$a(resource\_manager, RMID) :: \\ request(Resource, Operation, Context) \Leftarrow a(resource\_request, PRID) \\ \leftarrow grantPermission(RRID, Resource, Operation, Context, Policies)\ \textbf{then} \\ \left(\begin{array}{l} response(Grant\_yes) \Rightarrow a(resource\_request, RRID) \\ \textbf{or} \\ response(Resource\_result) \Rightarrow a(resource\_request, RRID) \\ \leftarrow getOperationResult(Resource, Operation, Access\_result) \end{array}\right)$$

The constraint *grantPermission(RRID, Resource, Operation, Context, Policies)* has to enforce that the resource requester RRID will have access to perform certain operation on a resource in a certain context based on certain policies. This means that there has to be a matching between the policies expressed on the intra level of the nodes in HealthAgents and the local policies described in each particular node. More precisely, we have to make sure that the logical formula associated to the local security restrictions subsumes the logical formula associated to the global restrictions applied for that particular node. Note that the logical approach is impetuous: due to the potential size of the system we need to be able to modularise the access and furthermore, to be able to automatically check for consistency.

The "*grantPermission*" will be satisfied by performing matching between the two Conceptual Graphs associated to the global rule and respectively the local rule. Let us consider a simple scenario, namely the clinicians accessing data from a hospital. We want to reinforce the fact that only clinicians within the same hospital as the data have access to them. This information is captured in Fig. 13. The bolded out nodes (the relation node associates and the concept node policy rule) will be expanded to capture this information in the Conceptual Graph depicted at the bottom of the page. At length, Fig. 13 represents the fact that a clinician, which has a certain location, is allowed to access a resource which is at the same location as him and is managed by a database agent. This information could be stored locally in one hospital as a local security policy rule.

Consider the example presented in Fig. 14. On the left hand side the local *PR_1* policy rule graph is depicted. On the right hand side we consider the query graph that wants to check if Maurice, a user from Birmingham is allowed to request data from Valencia. This could be the resource requester generated by the LCC. Checking whether the rules allow for that access is done by the means of projection, a labelled graph homomorphism between the query graph and the rules graph. More precisely, the relation nodes are projected into relation nodes and concept nodes into concept nodes. The structure of the graph also has to be preserved. We can see that, in this example, the answer to the query is "no". This is due to the fact that the structure of the query graph does not match the rule (more precisely, there is no "sameAs" relation in the query graph). Please note that information from the support is also considered while performing the projection. For example the concept type user from the query graph has been projected onto the concept type clinician (according to the concept type hierarchy). In the same way, according to the relation hierarchy, the relation node request was projected onto the relation node access.[2]

5. Conclusions and discussion

In this paper, we have analysed the general security requirements for clinical information systems and developed a layered security model, illustrated by its application to the HealthAgents system but which is also applicable to other healthcare systems. The interaction models being built will

[2] For a formal account of how Conceptual Graphs are defined and how the projection takes place see Croitoru and Compatangelo (2006a,b).

run upon our OpenKnowledge framework where agents are able to execute LCC protocols for interactions. Resource manager agents will govern the resource requests against the LCC constraints, reflecting clinical security policies. User agents will be allowed to have access and perform only what they need, reflecting their job responsibilities.

The major contributions and novelty of the approach is that it provides an enabling technology for bringing better knowledge sharing capabilities among healthcare professionals in a distributed environment and at the same time facilitating better access control. No global user account repository is required. Each clinician will get and only get the information they need in a collaborative decision support environment. Furthermore, our approach is enhanced with interoperability and inference capability with the use of a domain ontology and the Conceptual Graph technique. This enables the application of our model in clinical sites where resource and policy are defined in different languages. Overall, our healthcare knowledge sharing and security enforcement solution will be useful to large distributed clinical applications with separately managed users, resources, and access policies.

Organisational structure and context association are key assumptions to our privilege model. Organising authorisation at user level cannot realise cooperation and inter-organisational communication in extended health networks, as stated in Blobel (2004). The authors distinguish structural roles, describing prerequisites or competencies for actions and functional roles, being bound to the realisation of actions. Such a conjunctional perspective of role is in accordance with the privilege control in business processes and then their contextual constraint. The semantic similarity of clinical user group privileges and the business processes they can perform is described in Chandramouli (2000). In addition to that, access decisions need to be made on the exercise of privileges in business processes depending upon contextual information. Structuring business process (or task) context related constraints, e.g. attending relation between physician and patient as well as clinician speciality, as contextual parameters to task execution that affect access control decisions is expressed in (Hu and Weaver, 2004). Clinical task execution privileges, therefore, should be distinguished, and represented by the privileges of running interaction models in our approach. The layered security model authorises at a higher level, the users' task accessibility based on a static organisational structure and at a lower level, within task enactment, users' case and case partition accessibility based on dynamic functional needs in order to perform tasks.

This inevitably avoids the occasion that a junior clinician creates a classifier of poor quality or updates a classifier reputation value improperly. Next, higher level business function-based constraints are coupled with lower level data-based constraints. A limited set of data, determined by user workgroup memberships, will be allowed to be populated into the limited set of task functions. Finally, data-based constrains are additionally coupled with operation-based constraints. The available operations, determined by job nature and specialists, will be allowed, e.g. write (reports) or update (diagnosis results), upon particular data sections. These constraints, as well as individually defined local policies, must be satisfied prior to interaction model running. In sum, we constrain the availability of tasks to users, case availability to tasks, and further operations availability to cases, as the overall layered security architecture. The architecture is scalable since access rights are precisely controlled by the combination of these dimensions. For example, a senior pathologist doctor who is responsible for a patient can update the pathology part of this patient profile but someone who is a senior pathologist but not involved in caring for the patient cannot, or someone who is a junior doctor, or someone who is not specialised in pathology at all.

No global user account repository is required in our system. The necessary interaction models are globally agreed. The case to workgroup assignment is locally defined and user to workgroup possibly across organisations, for enabling interaction model running. When one user invokes an interaction model and this involves resources from other sites, the permission checking is determined by this user being involved in patient care or not, e.g. a remote clinician may perform a classification on behalf of a named doctor who is on holiday and delegates the responsibility to this clinician, in emergency situations, even the local hospital has not setup a local account for the clinician.

Interaction models can be publicly accessible since the descriptive interaction logic among peers reveals no secret information itself and so no issue exists such as alternative interaction model provision to certain users under certain conditions. Rather, alternative resource peers may be selected because the access to others is restrictive or, a subset or related/alternative resource items from query returned to the requester peer with a limited set of privileges. Such an autonomic query relaxation paradigm, as part of our future work, will avoid additional user interaction and frustrating experience. Another direction of future work is via monitoring unsuccessful resource access, an interaction model adjustment is advised if an access without satisfying constraints is encountered but considered necessary. It may be useful to let such requests be recorded and routed to responsible doctors or other delegated authorisers who may or may not approve the issuing of additional privileges, either permanently or temporarily. With better understanding of the necessity of such exceptional requests possibly after real life communication, critical and timely care aimed to patients will not be compromised.

Acknowledgements

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

REFERENCES

Aamodt A. Knowledge-intensive case-based reasoning in creek. In: Advances in case-based reasoning, 7th European conference, ECCBR 2004, proceedings, volume 3155 of lecture notes in artificial intelligence. Springer; 2004. p. 1–15.

Anderson R. Undermining data privacy in health information. British Medical Journal 2001;322:442–3.

Anderson RJ. Security in clinical information systems. British Medical Journal 1996a.

Anderson RJ. Patient confidentiality – at risk from NHS wide networking. In: Proceedings of healthcare 96; 1996b.

Blobel B. Authorisation and access control for electronic health record systems. International Journal of Medical Informatics 2004;73(3):251–7.

BMA – British Medical Association, http//www.bma.org.uk.

Brown N, Reynolds M. Strategy for production and maintenance of standards for interoperability within and between service departments and other healthcare domains. Technical report, CEN/TC 251. Brussels, Belgium: Health Informatics; 2000.

Calam D. Information governance – security, confidentiality and patient identifiable information, http//etdevents.connectingforhealth.nhs.uk/eventmanager/uploads/ig.ppt.

Chandramouli R. Business process driven framework for defining an access control service based on roles and rules. In: 23rd National information systems security conference; 2000.

Clunie D. DICOM structured reporting. PixelMed; 2000.

Croitoru M., Compatangelo E. A tree decomposition algorithm for conceptual graph projection. In: 10th International conference on principles of knowledge representation and reasoning; 2006a. pp. 271–76.

Croitoru M, Compatangelo E. Conceptual graph projection: a tree decomposition-based approach. In: Doherty P, Mylopuolos, Welty C, editors. Proc. of the 10th Int'l Conf. on the principles of knowledge representation and reasoning (KR'2006). AAAI; 2006b. p. 271–6.

Crook R., Ince D., Nuseibeh B. Towards an analytical role modelling framework for security requirements. In: 8th International workshop on requirements engineering: foundation for software quality; 2002.

Davis R, Shrobe H, Szolovits P. What is a knowledge representation? AI Magazine 1993;14(1):17–33.

Denley I, Smith SW. Privacy in clinical information systems in secondary care. British Medical Journal 1999;318:1328–31.

Hawker A. Confidentiality of personal information: a patient survey. Journal of Informatics in Primary Care 1995:16–9.

HealthAgents, http//www.healthagents.net.

http//www.hl7.org. Health level 7.

Hu J., Weaver AC. Dynamic, context-aware access control for distributed healthcare applications. In: 1st Workshop on pervasive security, privacy and trust; 2004.

http//www.openehr.org. Openehr.

Iakovidis I. Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare records in Europe. International Journal of Medical Informatics 1998;52(128):105–17.

Joint Computer Group of the GMSC and RCGP, GMSC and RCGP guidelines for the extraction and use of data from general practitioner computer systems by organisations external to the practice. Appenaix III. In: Committee on standards of data extraction from general Practice Guidelines; 1988.

M-Tech Information Technology, Inc.. Beyond roles: a practical approach to enterprise user provisioning; 2006.

Milner R. A calculus of communicating systems. Springer Verlag; 1980.

Milner R, Parrow J, Walker D. A calculus of mobile processes, i. Information and Computation 1992;100(1):1–40.

Mugnier ML. Knowledge representation and reasonings based on graph homomorphism. In: Proc. of the 8th int'l conf. on conceptual structures (ICCS'2000); 2000. pp. 172–92.

Omicini A, Ricci A, Viroli M. RBAC for organisation and security in an agent coordination infrastructure. Electronic Notes in Theoretical Computer Science 2005;128(5):65–85.

Pereira AL, Muppavarapu V, Chung SM. Role-based access control for grid database services using the community authorization service. Transactions on Dependable and Secure Computing 2006;3(2):156–66.

Pitchford RA, Kay S. GP practice computer security survey. Journal of Informatics in Primary Care 1995:6–12.

Robertson D. Multi-agent coordination as distributed logic programming. In: Proceedings of the 20th international conference logic programming (ICLP); 2004. pp. 416–30.

Robertson D. A lightweight coordination calculus for agent systems. In: LNCS, vol. 3476. Springer; 2005. pp. 183–197.

Robertson D, Giunchiglia F, Harmelen F, Marchese M, Sabou M, Schorlemmer M, et al. Open knowledge: semantic webs through peer-to-peer interaction. OpenKnowledge Manifesto, http//www.openk.org; 2006.

Sandhu RS, Coyne EJ, Feinstein HL, Youman CE. Role-based access control models. Computer 1996;29(2):38–47.

Sowa J. Knowledge representation: logical, philosophical, and computational foundations. Brooks Cole Publishing Co.; 2000.

Wooldridge M, Jennings NR, Kinny D. The gaia methodology for agent-oriented analysis and design. Journal of Autonomous Agents and Multi-Agent Systems 2000;3(3):285–312.

Weiser M. The computer of the 21st century. Scientific American 2001;265(3):66–75.

Xiao L, Greer D. Adaptive agent model: software adaptivity using an agent-oriented model driven architecture. Information and Software Technology 2009;51(1):109–37. Elsevier.

Xiao L, Lewis P, Gibb A. Developing a security protocol for a distributed decision support system in a healthcare environment. In: 30th international conference on software engineering. ACM; 2008. p. 673–82.

Xiao L, Peet A, Lewis P, Dashmapatra S, Sáez C, Croitoru M, et al. An adaptive security model for multi-agent systems and application to a clinical trials environment. In: 31st IEEE annual international computer software and applications conference. IEEE Press; 2007. p. 261–6.

Zhang L, Ahn G, Chu B. A role-based delegation framework for healthcare information systems. In: 7th ACM symposium on access control models and technologies. New York: ACM; 2002. p. 125–34.

Dr. Liang Xiao is a Postdoc Research Fellow in the HRB Centre for Primary Care Research at the Royal College of Surgeons in Ireland (RCSI), and joined the Centre in January 2009. Before coming to Ireland, Dr. Xiao was a Research Fellow in the School of Electronics and Computer Science at the University of Southampton in England. His research activities in Southampton included working on the EU Framework 6 projects of HealthAgents and OpenKnowledge. He obtained his BSc at the Huazhong University of Science & Technology in China, his MSc at the University of Edinburgh in Scotland, and PhD at Queen's University Belfast in Northern Ireland.

He has worked in the telecommunications industry as a Software Engineer following his graduation in China. His experience on solving real domain problems has stimulated his research interests in the area of Software Engineering in Edinburgh and Queen's. Specifically, his research work focuses on Software Adaptivity, Model Driven Architecture, and Agent-oriented Software Engineering. Later he applies his IT expertises into the healthcare domain in Southampton, where he and his colleagues developed a secure software framework for knowledge-intensive clinical decision support system. The results of his research have been published at many international conferences and journals.

Dr. Bo Hu is a researcher at SAP Research CEC Belfast. He received his PhD in Computer Science from the Robert Gordon

University, Aberdeen in 2004. Between 2002 and 2008, he worked as a Research Fellow in the Intelligence, Agent, Multimedia Group (IAM), School of Electronics and Computer Science, University of Southampton. During his days in Southampton, he was actively involved in UK EPSRC Advanced Knowledge Technology IRC and EU FP6 projects. His main research interest is in Knowledge Management (KM), KM in pervasive computing environments, Semantic Web, Web 2.0 and their applications in e-learning and e-healthcare.

Dr. Madalina Croitoru was born in Iasi, Romania in 1980. In June 2002 she graduated from FII with a thesis on Lineage in Data Warehousing. In October 2002 she started a part time PhD with the Department of Computing Science, University of Aberdeen while also working as a part time Teaching Assistant. Her research looked at improving Conceptual Graph applicability in Artificial Intelligence. After graduation she started working as a Research Fellow for the Department of Electronics and Computer Science, University of Southampton involved, part time, in two projects: HealthAgents and Open Knowledge. From September 2008 she started work at University Montpellier II as an Assistant Professor.

Paul Lewis is a Professor of Computer Science in University of Southampton. His main research interests are currently centred on the broad area of multimedia knowledge management. In particular he and his research team are addressing problems in image and video processing and analysis, multimedia annotation and semantic description of media. They are designing and developing novel facilities for multimedia information retrieval, navigation and browsing with a wide range of applications. The research is building on ideas from low level media processing, knowledge management and emerging semantic web technologies.

Dr. Srinandan Dasmahapatra is a lecturer in University of Southampton. His research interests are stochastic dynamics of gene networks, biomolecular networks and their state spaces, algebraic geometry and regulatory dynamics, and recommender systems, information in social networks.