Provenance in Agent-mediated Healthcare Systems

Tamás Kifor[1], László Z. Varga[1],
Javier Vázquez-Salceda[2], Sergio Álvarez[2], Steven Willmott[2],
Simon Miles[3], Luc Moreau[3]

[1] Computer and Automation Research Institute,
Kende u. 13-17, 1111 Budapest, Hungary
{tamas.kifor, laszlo.varga}@sztaki.hu;
http://www.sztaki.hu/

[2] Universitat Politècnica de Catalunya,
Jordi Girona Salgado 1-3, E - 08034 Barcelona, Spain
{jvazquez, salvarez, steve}@lsi.upc.edu;
http://www.upc.edu/

[3] School of Electronics & Computer Science, University of Southampton,
Southampton SO17 1BJ, UK
{sm, L.Moreau}@ecs.soton.ac.uk;
http://www.ecs.soton.ac.uk/

Abstract. Agent-oriented cooperation techniques and standardized electronic healthcare record exchange protocols can be used to combine information regarding different facets of a therapy received by a patient from different healthcare providers at different locations. Provenance is an innovative approach to trace events in complex distributed processes, dependencies between such events, and associated decisions by human actors. We focus on three aspects of provenance in agent-mediated healthcare systems: first, we define the provenance concept and show how it can be applied to agent-mediated healthcare applications; second, we investigate and provide a method for independent and autonomous healthcare agents to document the processes they are involved in without directly interacting with each other; and third, we show that this method solves the privacy issues of provenance in agent-mediated healthcare systems.

1 Introduction

Cooperation among people using electronic information and techniques is an increasingly common practice in every field, including healthcare applications. In the case of distributed medical applications, the data (containing the healthcare history of a single patient), the workflow (of the procedures carried out on that patient) and the logs (recording meaningful events in those procedures) are distributed among several heterogeneous and autonomous information systems. These information systems are under the authority of different healthcare actors such as general practitioners, hospitals, hospital departments, etc. which form disconnected *islands of information*. Communication and coordination between organizations and among members of a medical team, allowing the sharing of information and distributed decision making is often supported by agent-based techniques [1], because modeling application components as agents with some degree of autonomy easily reflects the decentralized nature of the network of healthcare institutions and can be considered as the natural extension to the notion of encapsulation in systems that are owned and developed by different authorities.

Even when using agent technologies, the distributed nature of healthcare institutions sometimes hinders the treatment of patients, because documentation of the healthcare history and therapy of a patient is split into independent healthcare institutions. In order to provide better, user-centered healthcare services, the treatment of a patient requires viewing the processes and data as a whole. Although agent-based cooperation techniques and standardized electronic healthcare record exchange techniques support the semantic interoperation between healthcare providers, we still face the problem of the reunification of the different pieces of the therapy of a single patient executed at different places. Currently there are some countries that have no unification method for patient healthcare records; each region in the country or even each institution inside a region may have its own medical record system, sometimes not even fully electronic, and with no automatic health care record exchange mechanisms. Therefore, it is not uncommon for doctors to depend on the patients themselves in order to include data from previous treatments and tests.

Making electronic systems *provenance aware* [2] enables users to trace how a particular result has been arrived at by identifying the individual and aggregated services that produced a particular output. In healthcare multi-agent systems (HC-MAS) there is a need to provide an integrated view of the execution of treatment processes, to analyze the performance of distributed healthcare services, and to be able to carry out audits of the system to assess that, for a given patient, the proper decisions were made and the proper procedures were followed. All of these tasks depend on being able to trace back the origins of decisions and processes, the information that was available at each step, and where that information came from. Note that the provenance of a piece of data is primarily about the causal dependencies of execution steps, although time sequences can also be handled. The provenance architecture of [2] focuses on making service-oriented systems provenance aware, but making healthcare agent systems provenance aware needs additional techniques described in this paper, because agents are autonomous actors and, unlike in service oriented systems, they may participate in the same process without directly interacting with each other.

Thus the main contribution of this paper is showing how healthcare agents interacting in an ad-hoc way can be augmented with the capability to produce at execution-time an explicit representation of the process actually taking place, without compromising the privacy protection in the original system.

First, we define the concept of provenance in Section 2. In Section 3 we describe an organ transplant management application and show how provenance can be applied to that application when agents directly interact. Then, in Section 4 we describe a method for independent and autonomous healthcare agents to document the processes when there is no direct interaction between them, and, in Section 5 we describe how this method helps to handle privacy issues of provenance in HC-MAS. Section 6 presents related work and we conclude in Section 7.

2 The Provenance concept

Trust in results produced by HC-MAS can be increased if we know the provenance of the particular result. As described in [2], the concept of *provenance* is already well known in fine art where it refers to the trusted, documented history of some work of art. This concept of provenance may also be applied to data and information generated within a computer system.

The provenance of a piece of data is represented in a computer system by some suitable documentation of the process that produced the data, called *process documentation*. This documentation can be complete or partial; it can be accurate or inaccurate; it can present conflicting or consensual views of the actors involved; it can be detailed or not. Provenance is investigated in open, large-scale systems typically designed using a service-oriented approach [3].

The technology-independent approach of the PROVENANCE project[1] to service-oriented architectures (SOAs) has formal foundations in the π-calculus[2] [4] and asynchronous distributed systems [5]. According to this view, messages are the only mechanism used to transfer information between actors. This view also allows formal definition of mappings between a) GRID applications, b)Web Services and c) Agent-Mediated Services and Applications. Therefore, *services* are regarded as components that take inputs and produce outputs. Such services are brought together by composition into a process to solve a given problem. In this abstract view, interactions with services (seen as *actors* in the process and realized as agents in a HC-MAS) take place using messages that are constructed in accordance with service interface specifications (agent messages in the case of HC-MAS).

Documentation of processes is represented in a computer system by a set of *p-assertions*, which are assertions made by the actors involved in those processes, documenting some step of the process. There are two kinds of p-assertions that capture an explicit description of the flow of data in a process: *interaction p-assertions* and *relationship p-assertions*. An interaction p-assertion is an assertion of the contents of a message by an actor that has sent or received that message. A relationship p-assertion is an assertion about an interaction, made by an actor that describes how the actor obtained output data or the whole message sent in that interaction by applying some function to input data or messages from other interactions. An interaction p-assertion, therefore links together the actions of two actors in a process, while a relationship p-assertion links together multiple actions by a single actor. In addition, *actor state p-assertions* are assertions made by an actor about their internal state in the context of a specific interaction.

The *provenance store* is the long-term facility for storing process documentation. A provenance store is used to manage and provide controlled access to process documentation. The *provenance lifecycle* is composed of four different phases. First, actors create p-assertions that are aimed at representing their involvement in a computation. After their creation, p-assertions are stored in a provenance store, with the intent they can be used to reconstitute the provenance of some data. After a data item has been computed, users or applications can query the provenance store. At the most basic level, the result of the query is the set of p-assertions pertaining to the process that produced the data. More advanced query facilities may return a representation derived from p-assertions that are of interest to the user. Finally the provenance store and its contents can be managed (subscription management, content relocation, etc). The provenance architecture is supported by a set of tools and a reference implementation of both the provenance store and the client-side libraries that provide connectivity between the actors and the stores.

By transforming a MAS into a *provenance-aware* MAS, the resulting system gets the capability to produce at execution-time an explicit representation of the distributed processes that are taking place (see example in Section 3). Such representation can be then queried and analyzed in order to extract valuable information to validate, e.g., the basis of decisions taken in a given case, or to make an audit of the system over a period of time.

In the case of a HC-MAS, by recording all the medical processes related to a given patient one can re-construct the treatment medical history of the patient. Therefore, making a HC-MAS *provenance-aware* provides a way to have a unified view of a patient's medical record with its provenance, i.e. to connect each part of the medical record with the processes in the

[1] http://www.gridprovenance.org/

[2] The π-calculus is of interest in this context because of its approach to defining events that are internal to actors as hidden communications.

real world that originated it and/or the individuals, teams or units responsible for each piece of data.

3 An example: the Organ Transplant Management Application

We will show how provenance can be used in HC-MAS through the application domain of Organ Transplant Management (OTM). The agent-mediated application we will use as example is the Organ Transplant Management Application (OTMA) that is a demonstrator in the PROVENANCE project.

3.1 Brief description of the OTMA system

The OTMA system aims to speed up the allocation process of solid organs to improve graft survival rates. It is an evolution from the CARREL Agent-Mediated Electronic Institution [6], developed not only with the help of the medical staff but also taking into account the recommendations of the hospitals' system administrators. Treatment of patients through the transplantation of organs or tissue is one of the most complex medical processes currently carried out, as it is a distributed problem involving several locations (donating hospital, potential recipient hospitals, test laboratories and organ transplant authorities), a wide range of associated processes, rules and decision making. Figure 1 summarizes the different administrative domains (solid boxes) and units (dashed boxes) that are modeled in the OTMA system. Each of these interact with each other through agents (circles) that exchange information and requests through messages. In a transplant management scenario, one or more hospital units may be involved: the hospital transplant unit, one or several units that provide laboratory tests and the Electronic Healthcare Record (EHCR) subsystem which manages the health care records for each institution. The diagram also shows some of the data stores that are involved: apart from the patient records, these include stores for the transplant units and the Organ Transplant Authority (OTA) recipient waiting lists (WL). Hospitals that are the origin of a donation also keep records of the donations performed, while hospitals that are recipients of the donation may include such information in the recipient's patient record. The OTA has its own records of each donation, stored case by case.

The Electronic Healthcare Record (EHCR) subsystem of OTMA provides a way to manage patient records distributed in different institutions. The subsystem provides the structures to build a part of or the entire patient's healthcare record drawn from any number of heterogeneous systems (the only requirement is that they follow the ENV13606 pre-standard produced by CEN/TC251 WG I.[3]). The EHCR subsystem also uses an authentication service to authorize request messages from remote health care parties.

[3] European Committee for Standardization (CEN), Technical Committee 251 (TC251): Health Informatics, Work Group I (WG I.): Information models, http://www.cenorm.be/

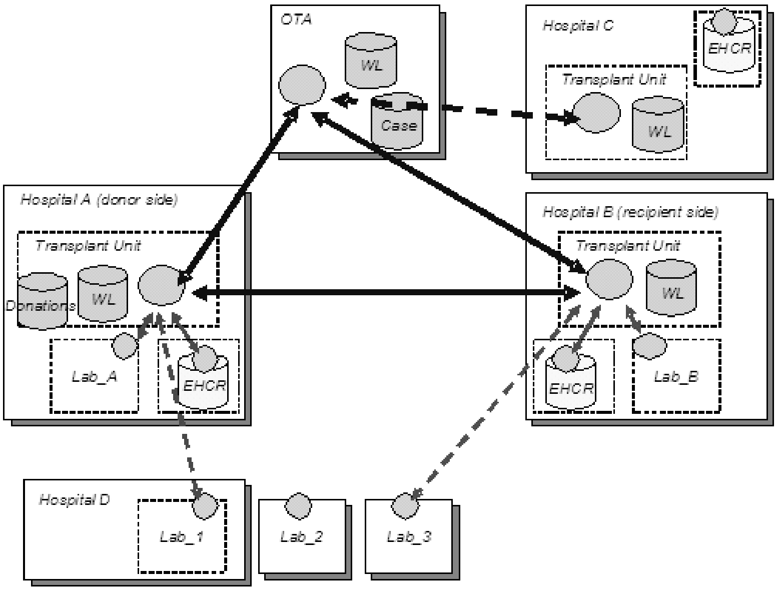

Figure 1 **The OTMA system.** Each medical unit is represented by an agent (circle in figure) which manages interactions with other units and with the EHCR subsystem.

3.2 Adapting the OTMA system for Provenance

Making the OTMA system provenance aware presented three challenging issues: a) the provenance of most of the data is *not* the execution of computational services, but decisions and actions carried out by *real* people in the *real* world (this is discussed in this Section 3); b) past treatments of a given patient in other institutions may be relevant to the current decisions in the current institution, so information of the processes undertaken in those previous treatments should be connected to the provenance information of a current process (this is discussed in Section 4); c) the agent with provenance information knows much more about the patient than any other agent in the system, so there are privacy risks to be mitigated (this is discussed in Section 5).

In the case of the OTMA system, each organizational unit is represented by an agent-mediated service. Staff members of each unit can connect to the unit services by means of graphical user interfaces. The distributed execution of the OTM services is modeled as the interaction between the agents, and recorded as *interaction p-assertions* and *relationship p-assertions*. As in the OTM scenario a decision depends on the human making the decision, additional *actor state p-assertions* are recorded, containing further information on why the particular decision was made and, if available, the identities(s) of the team members involved in the decision.

To illustrate how provenance is handled in the OTMA system, let us see how the provenance of a medical decision is recorded. Figure 2 (top) shows a simplified view over a subset of the donation process. We consider a patient who has previously given consent to donate his organs. As the patient's health declines and in foresight of a potential organ donation, one of the doctors requests the full health record for the patient and then orders a serology test[4] through the OTMA system. After brain death is observed and logged in the system (along with the report certifying the brain death), if all requested data and analysis results have been obtained, a doctor is asked to make a decision about the patient being a potential donor. This decision is explained in a report that is submitted as the decision's justification.

[4] A serology test is usually performed over blood samples to detect viruses (HIV, Hepatitis B/C, syphilis, herpes or Epstein-Barr virus), which, if present in the organ, can pass to the recipient.

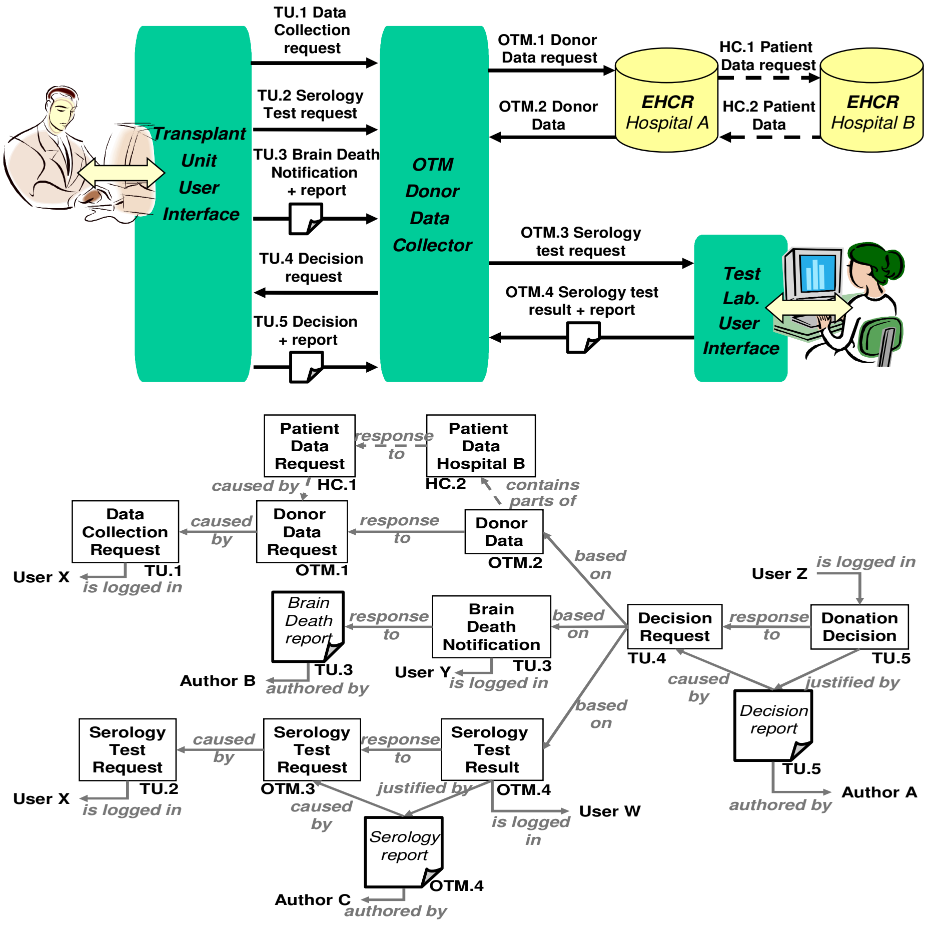

Figure 2 **Example scenario:** (top) Interactions of the OTMA agents involved in a donation decision; (bottom) directed acyclic graph showing the provenance of the donation decision.

Figure 2 (top) shows the OTMA agents for this small scenario and their interactions. The Transplant Unit User Interface Agent passes requests (TU.1, TU.2) to the OTM Donor Data Collector Agent, which gets the electronic record from the EHCR system (OTM.1, OTM.2). Sometimes all or parts of the record are not in the same institution but located in another institution (HC.1, HC.2). The Donor Data Collector Agent also sends the request for a serology test to the laboratory and gets back the result (OTM.4), along with a detailed report of the test. Reports are also passed in the case of the Brain Death notification (TU.3) and the final decision report (TU.5).

Figure 2 (bottom) graphically represents the subset of the p-assertions produced by the *provenance-aware* OTMA which are related to the mini-scenario described above. The part of the process that happens within the electronic system is represented by interaction p-assertions (regular boxes) for all interactions (TU.x, OTM.x, HC.x), and relationship p-assertions (*response_to, caused_by*, *based_on*) capturing dependencies between data. Even though what happens in the system parallels what happens in the real world, as we already said this is not enough to fully determine the provenance of a given decision. To solve this, we connect the electronic process to the real world by adding actor state p-assertions stating who logged the information in the system (*is_logged_in*) and when (not shown in picture), which are the reports that justify a given state in the system (*justified_by*), who are the authors of these reports (*authored_by*) and when the action reported was performed or the decision taken (not shown).

3.3 Analyzing an execution through the Provenance documentation

Storing provenance information instead of the, more common, standard log systems, has the advantage that the provenance representation is stored in a way that complex queries can be performed over it, which allows a provenance-aware system to extract valuable information to validate some of the steps taken into a (medical) process, or even to make an audit of the system over a period of time. In the OTMA system, apart from periodical audits, transplant coordinators also want to ask the following types of provenance questions, related to a given patient (donor or recipient) or to the fate of a given organ:

A. Where did the medical information used on each step of the process came from?

B. Which medical actor was the source of some piece of information?

C. What kind of medical record was available to actors at each step of the process?

D. When a given medical process was carried out, and who was responsible for it?

E. When was a decision taken, and what was the basis of the decision?

F. Which medical actors were asked to provide medical data for a decision?

G. Which medical actor refused to provide medical data for a decision?

All these kind of questions can be answered by querying the provenance store. A query will give as a result (a subset of) the provenance representation graph of the process related to the query. If we use as an example the graph in Figure 2, by following the edges from the "Donation Decision" p-assertion we can trace the provenance of the donation decision, how it was based in some data and test requests, how a brain death notification is also involved, who requested the information, where it came from (in some cases it might come from the EHCR of another hospital), and who authored the justifying reports in the main steps of the process.

In those cases (as in Figure 2) where the decision might be based on medical data coming from tests and medical treatments carried out in other institutions, another issue to solve is the following: how to find, retrieve and incorporate the provenance of the data coming from the other institution? If these institutions have also provenance-aware systems and the provenance stores of the different institutions are connected, to solve the aforementioned problem is to solve the issue of matching the different p-assertions related to the same patient. If this match is done, then actors can make p-assertions that link together the separate sets of p-assertions to create a larger provenance document providing an integrated view of the healthcare history of the patient. The result (not shown on Figure 2) would be that the p-assertions related to *Patient Data Hospital B* would be linked to the set of p-assertions already part of the provenance of the Donation Decision.

Collectively the p-assertions can be seen as describing a distributed process, spanning space as well as time. Every relationship described is causal, i.e. between the cause of something happening and the effect of it happening, and is therefore also temporal, i.e. causes always come before effects. Furthermore, extra information can be added to provide further detail. For example, an actor may record, as an actor state p-assertion, the time shown on their local clock. Together, the structured documentation of processes allow a rich set of questions to be asked about what occurred, why, when and by whom and, in the OTMA system, such a process may be a patient's healthcare history.

4 Process Documentation in Provenance Aware HC-MAS

As seen in the previous section, in order to create an integrated view of a patient's healthcare history, the process documentation created by each healthcare institution must contain interaction p-assertions and relationship p-assertions which link together the p-assertions of agents in the process. The way in which p-assertions provide this linking in usual service-oriented applications is by use of a common identifier, called an *interaction key,* for both parties, sender and receiver, in an interaction. From the fact that two agents have recorded documentation using the same interaction key, we can determine that their actions were part of the same process, and therefore both are part of the provenance of the process' output. However, to record p-assertions with the same interaction key, two agents must exchange that key, which means they must electronically interact.

In this section we show that there are processes without direct electronic interaction. Because processes executed by healthcare agents often belong to this category, for example in the OTMA system the processes of the EHCR subsystem or relevant previous treatment processes carried out in other hospitals, we describe how independent autonomous agents can jointly create process documentations of these processes.

4.1 Healthcare Processes as Weakly Connected Processes

There is a basic difference between a typical business process and healthcare applications such as the EHCR subsystem of the OTMA system introduced above. In a typical business or e-science application the agents participating in the process are in contact and connected by interactions between them. In this case there is exchange of documented messages, and we say that there is *direct interaction* between the agents.

In medical processes, however, the physicians treating the same patient may not be in direct contact. This is typical, but not necessarily specific to medical processes. The patient may be treated by one physician, be healthy for a while, and then go to another physician with another disease, in some cases as a consequence of the previous disease. In this case, the second physician is not in contact with the first, they do not know each other's identity and because the identities cannot be revealed for privacy reasons, the p-assertions belonging to the same patient cannot be linked together automatically. In this case we say that there is *latent interaction* between the physician agents. Note that the patient usually cannot determine the link between the current treatment and the previous one. Even if the patient remembers something informally, the formal link cannot be determined.

We can now define two types of processes: strongly connected and weakly connected processes. We view the processes as graphs where the nodes are the activities executed by agents alone and arcs are the interactions, either latent or direct. In *strongly connected processes* the whole process graph contains only direct interactions, whereas in *weakly connected processes* the process graph can be cut into two sub-graphs that are connected to each other only by one or more latent interactions. The full healthcare history of a patient is usually created by a weakly connected process containing strongly connected sub-processes. Although the transplant process of OTMA is strongly connected, it becomes "infected" with the latent interactions of the EHCR creation process when the EHCR of the patient is retrieved.

4.2 Process Documentation of Strongly Connected Processes

Figure 3, similarly to Figure 2, shows the model of strongly connected processes and their process documentation. Here, physicians are represented by agents 1 and 2. They are the actors of treatment processes 1 (treatmentp_1) and 2 (treatmentp_2). At some point, agent 1 sends the patient to agent 2 in a documented way. P-assertions about this interaction are recorded by agents 1 and 2. In case of medical or other secure applications, a global identifier for a patient in the local system is not used, because it could be used to determine the identity of the patient. Because the agents interact directly and electronically, they agree on an interaction key and both include it in their p-assertions. This way the process documentations of the treatment processes are not disjoint, therefore if some agent queries the process documentation using patient_local_name_1, then the provenance system is able to return all process documentation comprising the provenance of the patient.

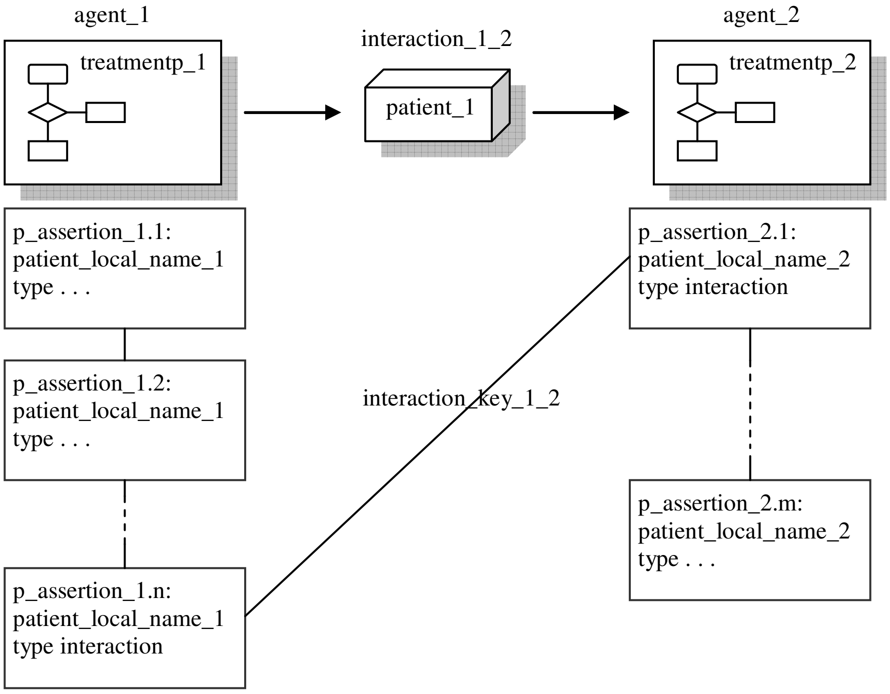

Figure 3 **Process documentation in strongly connected processes.** P-assertions of strongly connected processes are linked together by the p-assertions related to the interaction connecting the two processes.

4.3 Process Documentation of Weakly Connected Healthcare Processes

Process documentation creation is a bit more complicated in the case of weakly connected healthcare processes, when there is no direct interaction between the agents. The actors, processes and process documentation in a weakly connected healthcare process are similar to the one shown in Figure 3, but there is no link across the sets of p-assertions of the processes executed by the different agents. If we want to retrieve the complete provenance of the patient, then we are interested in both sets. Moreover if agent 2 finds out somehow that treatment process 2 is some way a consequence of treatment process 1, it still cannot find the relevant p-assertions made by agent 1, because there is no common identifier for the processes or the patient.

If there was interaction between the agents, then the common identifier could be the common interaction key which they share. However an interaction key is used to identify the flow of information between two steps/actors in a process, and is an "internal" identifier, in that it has no meaning outside of the process documentation and while, theoretically, it corresponds to one patient wherever the information exchanged in an interaction concerns just one patient, there is no mapping held in the system between an interaction key and any other patient identifier.

Although the patient could present its global identifier, such as its social security number, to the physicians, but this global identifier cannot be used in process documentation for privacy reasons as discussed in Section 5.

4.4 Method to Link Process Documentations of Weakly Connected Processes

We have described the problem of process documentation creation resulting from the lack of direct interaction between the agents. The solution to the problem is to use an intermediate institution in a higher hierarchical level, which is in contact with both agents and knows about the patient as well. This is usual in medical domains, as they are regulated by national and international bodies and there are services which give a global identifier to the patient, such as the national security number. But the global identifier should not be used in documentation of privacy-aware processes, because regulations ordain the separation of data, which means that medical information and personal identification cannot be stored together and anonymised identifiers must be used instead. Therefore, usually we should add an anonymisation service in HC-MAS to convert real patient identifiers to anonymised patient identifiers.

Figure 4 shows how this method works. In the first step of this method, we locate in the application an already existing service which is used to anonymise patient identifiers. If there is no such service, then we introduce it into the application. The service is called anon_service in the figure.

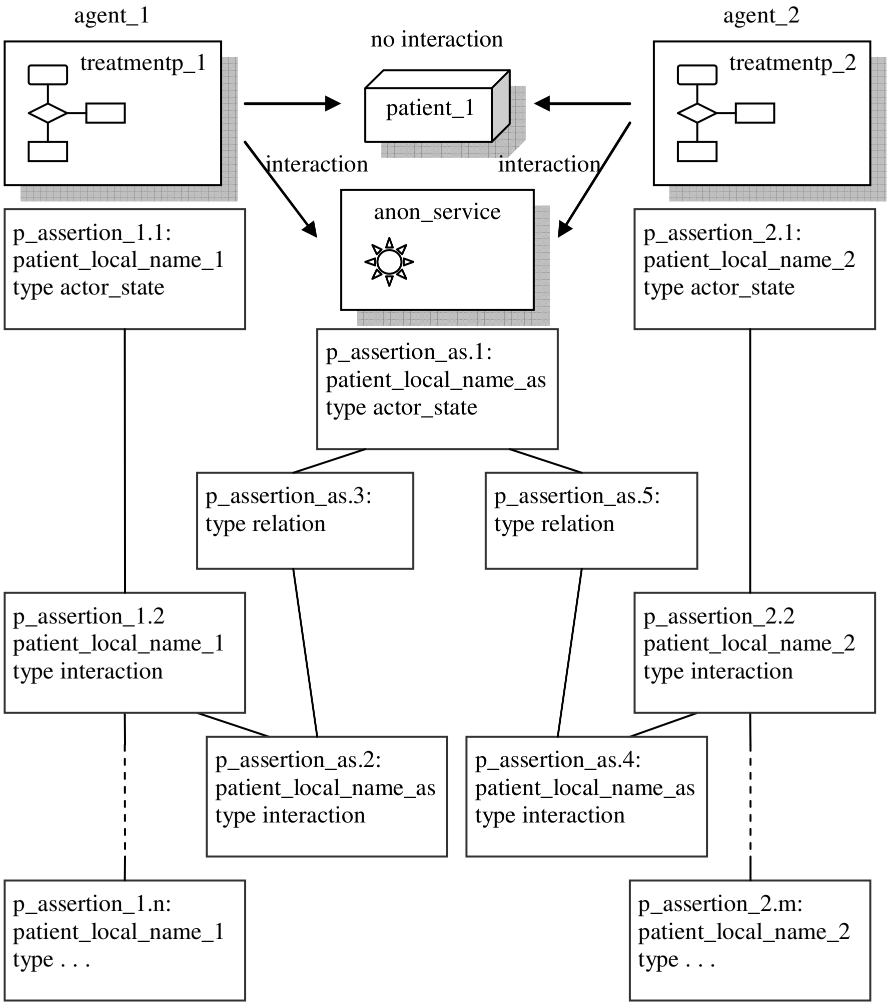

Figure 4 **Linking of process documentations in weakly connected healthcare processes.** P-assertions of weakly connected healthcare processes are linked together by the p-assertions of a higher level service.

The second important element of the method is that the anonymisation service documents its own processing. Whenever a new patient identity becomes known to the anonymisation service, then the anonymisation service puts an actor state p-assertion into the provenance store about the patient identity. Note that the provenance store does not contain the global patient identifier, only the anonymised identifier.

The third important element of the method is that the agents adopt the norm of notifying the anonymisation service when a new activity with a patient is started. In Figure 4, when agent 1 starts an activity on the patient, it makes an actor state p-assertion about the start of the activity and notifies the anonymisation service that the activity started. The interaction is recorded in the provenance store with interaction p-assertions on both sides. The anonymisation service asserts a relationship p-assertion between the p-assertion related to the anonymised patient identity and the p-assertion related to interaction between agent 1 and the anonymisation service. When agent 2 starts an activity on the patient, it behaves similarly, therefore there will be an indirect link between the two agent's processes, and the complete provenance of the patient record can be determined.

Although conceptually the anonymisation service becomes somehow a central interaction node in the system, scalability can be maintained. Agents communicate limited amount of data with the anonymisation service only when they start a new case. The anonymisation service creates a new p-assertion for the case, and agents link further p-assertions to the start case p-assertion without communicating with the anonymisation service. The functionality of the anonymisation service can be distributed in real implemented systems among cooperating services allocated to countries, regions, insurance companies, etc.

In addition to the ability to return the whole process documentation, the method described above has two other advantages: the agents can improve the quality of the process documentation and of their own activities.

The quality of the process documentation can be improved if the agents discover some relationship from the real processes (e.g. the current illness of the patient is a consequence of a problem in the previous treatment not discovered before). In these cases, they can augment the existing links documented by the anonymisation service with direct causal relationships. This is now possible, because, by following the anonymisation service based links, the p-assertions relevant to a single patient can be located, identified and linked.

The agents can improve the quality of their own activities using the linked process documentation, because when agent 2 executes its treatment process, it can already retrieve the p-assertions of agent 1. If the physician knows the details of the previous treatment of the patient, then he/she might use that information in the current treatment.

5 Protecting Privacy in the Provenance-aware Application

In healthcare applications, enforceable privacy rules are extremely important. Protection of individuals' health-related data has been a continued concern of the medical body from the very beginning of the medical practice, as reflected in the famous Hippocratic oath. There exist considerable efforts to put into practice a body of policies which ensure the protection of medical data in a scenario of massive use of computers in the health sector. Regulations[5] define guidelines about the adequate organizational and technical measures that must be taken in medical information systems. One of these guidelines is related to the separation of data: as a general rule, the design of data structures, procedures and access control policies must be such that they allow the separation of a) identifiers and data related to a person's identity, b) administrative data, c) medical data, and d) genetic data. Such separation must ensure that no unauthorized person can connect the identity of the patient with his medical or genetic data.

In EHCR systems, and in the OTMA system discussed above, a typical solution for the separation of identity information and medical data is the *anonymised identifier*. The anonymised identifier is generated from real patient identifier, and medical data is stored together with this anonymised identifier. If we know the real patient identifier, then we can find the corresponding medical data, but from the medical data we cannot find out the identity of the patient.

When we make agent systems provenance-aware, we introduce an additional data store into the system: the provenance store. There is a conflict between provenance and privacy. While for provenance we need as much information as possible about the whole process (*who* did *what* and *when*), for privacy we need to restrict as much as possible the information available, in order to avoid identification of patients and practitioners by unauthorized users.

[5] "Directive 95/46/CE of the European Parliament and of the Council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and of the free movement of such data," *Official Journal of the European Communities*, L 281/31 - L 281/39., October 1995.

In the provenance aware OTMA system two techniques are used to protect privacy: a) we do not store sensitive medical data in the provenance store, and b) we use anonymised patient identifiers in provenance stores.

In order to protect medical data, agents do not store sensitive medical data in the provenance store, but only references to such data. This way the provenance store contains only the linkage and the skeleton of the provenance of the medical data, and the healthcare data can be laid on the skeleton by retrieving it from the healthcare information system when needed. The retrieval is done via the EHCR system which is completely under the control of EHCR access rules. With this approach we keep the same degree of privacy of medical data as in the original agent system.

One might think that if we do not store medical information about patients in the provenance store, then there is no need to anonymise the patients and we can use real patient identifiers, because no medical information can be inferred about the patient. However this is not the case. Even the fact that the patient was treated can be sensitive information, and the reference to the place where the medical data of the treatment was carried out may contain sensitive information, because the type of institution can reveal the type of medical intervention. Therefore the patient identity has to be anonymised at least.

The anonymisation procedure should be irreversible: nobody should be able to tell the real identity of the patient by knowing the anonymised identifier. The method described in Section 4.4 satisfies this requirement. The irreversibility of the anonymisation is guaranteed by the way data storage is organized: the anonymisation service does not store the mapping from the real patient identifier to the anonymised patient identifier and computes the anonymised identifier each time it is needed using its own non-trivial algorithm. As a result, the real identifier and the anonymised identifier are not stored together anywhere in the system and the mapping from one identifier to the other cannot be found out without the algorithm of the anonymisation service.

6 Related Work

In those first investigations which started to record the origin and history of a piece of data, the concept was called *lineage*. In the SDTS standard[6], lineage was a kind of audit trail that traced each step in sourcing, moving, and processing data, mainly related to a single data item, a logical data record, a subset of a database, or to an entire database [7]. There was also relationship to versioning [8] and data warehouses [9]. The provenance concept was later further explored within the GriPhyN project[7]. The application of provenance in grid systems was extended in two respects: 1) data was not necessarily stored in databases and the operations used to derive data items might have been arbitrary computations; and 2) issues relating to the automated generation and scheduling of the computations required to instantiate data products were also addressed. The PROVENANCE project builds on these concepts to conceive and implement an industrial strength open provenance architecture.

To our knowledge, the application of provenance techniques to HC-MAS is novel. In organ allocation management, there are few ICT solutions giving powerful support to the allocation of human organs which keep records of the distributed execution of processes. The

[6] American National Standard for Information Systems. Spatial Data Transfer Standard (SDTS) - Part 1, Logical Specifications, Secretariat, United States Geological Survey, National Mapping Division, DRAFT for Review, November 20, 1997, http://mcmcweb.er.usgs.gov/sdts/SDTS_standard_nov97/part1b12.html

[7] http://www.griphyn.org

EUROTRANSPLANT[8] system is a centralized system where all information and decisions are made in a central server, and all activity is recorded in standard logging systems. The OTM system of Calisti et al. [10] is a distributed system (developed in collaboration with *Swisstransplant*) which combines agent technology and constraint satisfaction techniques for decision making support in organ transplant centers. In this case all activity is also recorded in standard logging systems.

7 Conclusions and ongoing work

In this paper, we have discussed the important issues of making healthcare agent applications provenance-aware. Provenance-awareness enables users to trace how a particular result has been produced by identifying the individual and aggregated services that produced a particular output. This helps users to get an integrated view of the treatment process executed by distributed autonomous agents, and to be able to carry out audits of the system to assess that, for a given patient, the proper decisions were made and the proper procedures were followed. We discussed the special techniques needed in agent systems to make the autonomous and independent actors provenance aware and produce joint process documentation. We presented provenance awareness through the example of the CARREL agent system in the organ transplant management application domain. We detailed a method of documenting processes by weakly connected autonomous healthcare agents and showed how this method helps to retain security and privacy of data within the process documentation produced by HC-MAS.

In the context of the PROVENANCE project we are building a first demonstrator of this application. Evaluation is planned with some hospital and transplant coordinators in Spain, who will give feedback in the lasts steps of the development and fine-tuning of the application. The method is expected to improve process documentation by allowing the creation and retrieval of complete processes. The improvement is attained in two aspects: the agents can improve the quality of the process documentation and the agents can improve the quality of their own activities.

8 Acknowledgements

This work has been funded mainly by the IST-2002-511085 PROVENANCE project. Javier Vázquez-Salceda's work has been also partially funded by the "Ramón y Cajal" program of the Spanish Ministry of Education and Science. All the authors would like to thank the PROVENANCE project partners for their inputs to this work.

9 References

[1] J.L. Nealon and A. Moreno, ed., *Applications of Software Agent Technology in the Health Care Domain,* Whitestein Series in Software Agent Technologies, Birkhäuser Verlag, Basel, 2003.

[2] P. Groth, S. Jiang, S. Miles, S. Munroe, V. Tan, S. Tsasakou, L. Moreau, "D3.1.1: An Architecture for Provenance Systems," *Technical report*, University of Southampton, February 2006; http://eprints.ecs.soton.ac.uk/12023/

[3] M. P. Singh and M. N. Huhns, *Service-Oriented Computing: Semantics, Processes, Agents*, John Wiley & Sons, 2005, p. 588.

[4] R. Milner, *Communicating and mobile systems: the π-calculus,* Cambridge University Press, 1999.

[5] N. Lynch, *Distributed Algorithms,* Morgan Kaufmann Publishers, 1995.

[8] http://www.eurotransplant.nl

[6] J. Vázquez-Salceda, J.A. Padget, U. Cortés, A. López-Navidad, F. Caballero, “Formalizing an Electronic Institution for the distribution of Human Tissues," *Artificial Intelligence in Medicine*, Vol. 27., No. 3., Elsevier, March 2003., pp. 233-258.

[7] P. Buneman, S. Khanna, and W.-C. Tan, “Why and Where: A Characterization of Data Provenance” in *Proc. 8th Int’l Conf. on Database Theory*, (ICDT 2001), LNCS 1973, J. Van den Bussche, V. Vianu, eds., Springer-Verlag, 2001, pp. 316-331.

[8] A. Marian, S. Abiteboul, G. Cobena, and L. Mignet, “Change-Centric Management of Versions in an XML Warehouse,” *Proc. 27th Int’. Conf. of Very Large Data Bases*, (VLDB 2001), P. M. G. Apers et al., eds., Morgan Kaufmann, 2001, pp. 581-590.

[9] Y. Cui, J. Widom and J.L. Wiener, “Tracing the Lineage of View Data in a Warehousing Environment”, *ACM Transactions on Database Systems*, Vol. 25, No.2, June 2000, pp. 179–227.

[10] M. Calisti, P. Funk, S. Biellmann and T. Bugnon, “A Multi-Agent System for Organ Transplant Management,” in [1]

Keywords:

I. Computing Methodologies
2. Artificial Intelligence
11. Distributed Artificial Intelligence
b. Intelligent agents,

J. Computer Applications
3. Life and Medical Sciences
c. Medical information systems

J. Computer Applications
8. Internet Applications
h. Health care

H. Information Technology and Systems
4. Information Technology and Systems Applications
1. Office Automation
g. Workflow management

E. Data
0. General
2. Data dependencies (provenance of data)

E. Data
m. Miscellaneous (provenance of data)

H. Information Technology and Systems
2. Database Management
4. Systems
16. Workflow management