Applying Provenance in Distributed Organ Transplant Management

Sergio Álvarez[1], Javier Vázquez-Salceda[1],
Tamás Kifor[2], László Z. Varga[2], and Steven Willmott[1]

[1] Knowledge Engineering and Machine Learning Group,
Universitat Politècnica de Catalunya. Jordi Girona 1-3, Barcelona, Spain
{salvarez, jvazquez, steve}@lsi.upc.edu
http://www.lsi.upc.edu/~webia/KEMLG
[2] Computer and Automation Research Institute, Kende u. 13-17, 1111 Budapest, Hungary
{tamas.kifor, laszlo.varga}@sztaki.hu
http://www.sztaki.hu/

Abstract. The use of ICT solutions applied to Healthcare in distributed scenarios should not only provide improvements in the distributed processes and services they are targeted to assist but also provide ways to trace all the meaningful events and decisions taken in such distributed scenario. *Provenance* is an innovative way to trace such events and decisions in Distributed Health Care Systems, by providing ways to recover the origin of the collected data from the patients and/or the medical processes. Here we present a work in progress to apply *provenance* in the domain of distributed organ transplant management.

1 Introduction

Cooperation among people using electronic information and techniques is more and more common practice in every field including healthcare applications as well. In the case of distributed medical applications the data (containing the healthcare history of a single patient), the workflow (of the corresponding processes carried out to that patient) and the logs (recording meaningful events) are distributed among several heterogeneous and autonomous information systems. These information systems are under the authorities of different healthcare actors like general practitioners, hospitals, hospital departments, etc. which form disconnected *islands of information*. In order to provide better healthcare services, the treatment of the patient typically requires viewing these pieces of workflow and data as a whole.

Also, having an integrated view of the workflow execution and the logs may become important in order to analyse the performance of distributed healthcare services, and to be able to carry out audits of the system to assess if needed, that for a given patient the proper decisions were made and the proper procedures were followed. For all that there is a need to be able to trace back the origins of these decisions and processes, the information that was available at each step, and where all these come from. In order to support this in this paper we propose to make distributed medical applications *provenance-aware*. Our working definition for *provenance* is the following: "the provenance of a piece of data is the process that led to the data" [1,2]. Provenance

L. Moreau and I. Foster (Eds.): IPAW 2006, LNCS 4145, pp. 28 – 36, 2006.

enables users to trace how a particular result has been achieved by identifying the individual and aggregated services that produced a particular output by recording *assertions* about a workflow execution in special assertion stores, the *provenance stores*. These stores, unlike standard logging systems, organize assertions in a way that complex queries can be executed to extract provenance information about individual aspects of a process or a full execution trace.

The contents of this paper are as follows. In section 2 we present the organ allocation scenario that we use as example and the applications we are developing for it. Then in section 3 we describe how provenance is handled in our applications. Section 4 presents related work and finally section 5 presents some conclusions.

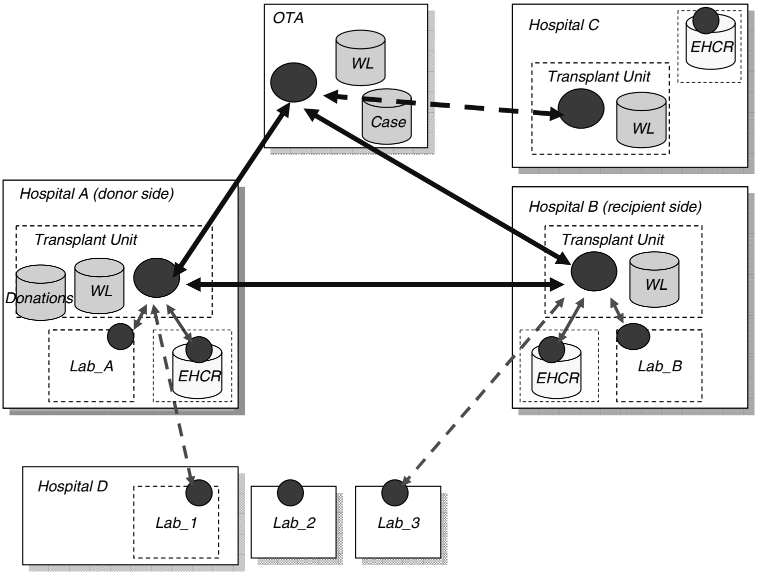

Fig. 1. The OTM application

2 Problem Domain

Patient treatment through the transplantation of organs or tissues 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, see Figure 1), a wide range of associated processes, rules and decision making. It is recognized worldwide that IT solutions which increase the speed and accuracy of decision making could have a very significant positive impact on patient care outcomes. Electronic systems that might be implemented for transplant management can be divided into two main types: a) systems for *distributed transplantation management* and b) systems for *medical record management*.

2.1 Distributed Transplant Management: The OTM Application

The Organ Transplant Management (OTM) Application aims to speed up the allocation process of solid organs to improve graft survival rates. Its policy implements the Spanish guidelines for organ and tissue procurement and Spanish regulations for allocation, as Spain is world leader in the area, followed as a model by other countries. OTM uses standard web service technology and has been adapted to be provenance-aware, by interacting with the provenance stores in order to keep track of the distributed execution of the allocation process for audit purposes.

Figure 1 summarizes the different administrative domains (solid boxes) and units (dashed boxes) that are modelled in the OTM application. Each of these interact with each other through Web Service interfaces (circles) that send or receive messages. The Organ Transplant Authority (OTA) is an administrative domain with no internal units. In a transplantation management scenario, one or more hospital units may be involved: the hospital transplant unit, one or several units that provide laboratory tests and the unit that is responsible for the patient records (which will use the EHCR application services, see section 2.2). The diagram also shows some of the data stores that are involved: apart of the patient records, these include stores for the transplant units and the 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.

By transforming OTM into a *provenance-aware* application, we augment OTM with a capability to produce at run-time an explicit representation of the process actually taking place (see example in Figure 2). Such representation can be then queried and analysed in order to extract valuable information to validate, e.g., the decisions taken in a given case, or to make an audit of the system over a period of time.

2.2 Medical Record Management: The EHCR System

The Electronic Health Care Record System (EHCR) provides a way to manage electronic health records distributed in different institutions. The architecture provides the structures to build a part of or the entire patient’s healthcare record drawn from any number of heterogeneous databases systems in order to exchange it with other healthcare information systems. The EHCR architecture has two external interfaces: a) a Web Service that receives and sends messages (following ENV13606 pre-standard format [3]) for remote medical applications; and b) a Java API for local medical applications that can be used to access the EHCR store directly. The application also uses an authentication Web Service to authorize request messages from remote health care parties.

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

3 Provenance Handling in the OTM Application Domain

The Provenance architecture developed within the PROVENANCE project [1] assumes that the distributed system can be modelled using a service-oriented approach. In this abstract view, interactions with services (seen as *actors*) take place using messages that are constructed in accordance with service interface specifications.

In the case of the OTM application, each organisational unit (the transplant unit, the ER unit, the laboratories) is represented by a service. Staff members of each unit can connect to the unit services by means of GUI interfaces. The provenance of a data item is represented by a set of *p-assertions,* documenting steps of the process, and they are stored and managed in *provenance stores*. The distributed execution of the OTM services is modeled as the interaction between the actors representing the services, and recorded as *interaction p-assertions* (assertions of the contents of a message by the actor that sent or received it) and *relationship p-assertions* (assertions that describe how the actor obtained an interactions'output data by applying some function to input data from other interactions). As in the OTM scenario a decision depends on a human making the decision, additional *actor state p-assertions* (assertions made by actors about their internal state in the context of a specific interaction) are recorded, containing further information on why the particular decision was made and, if available, the identities of the team members involved in the decision.

The application of the provenance architecture to the OTM system had to overcome two challenging issues: a) the provenance of most of the data is *not* a computational service, but decisions and actions carried out by *real* people in the *real* world; b) past treatments of a given patient in other institutions may be relevant to the current decisions in the current institution, so p-assertions about the processes underwent in those previous treatments should be connected somehow to the current p-assertions. An example on how we deal with both issues can be found in section 3.2.

3.1 Provenance Questions

In both the OTM and the EHCR systems, the provenance architecture should be able to answer the following kind of questions, related to a given patient (donor or recipient) or to the fate of a given organ:

- where did medical information used on each step of the process came from,
- which medical actor was the source of information.
- what kind of medical record was available to actors on each step of the process
- when a given medical process was carried out, and who was responsible for it.
- when a decision was taken, and what was the basis of the decision
- which medical actors were asked to provide medical data for a decision
- which medical actor refused to provide medical data for a decision

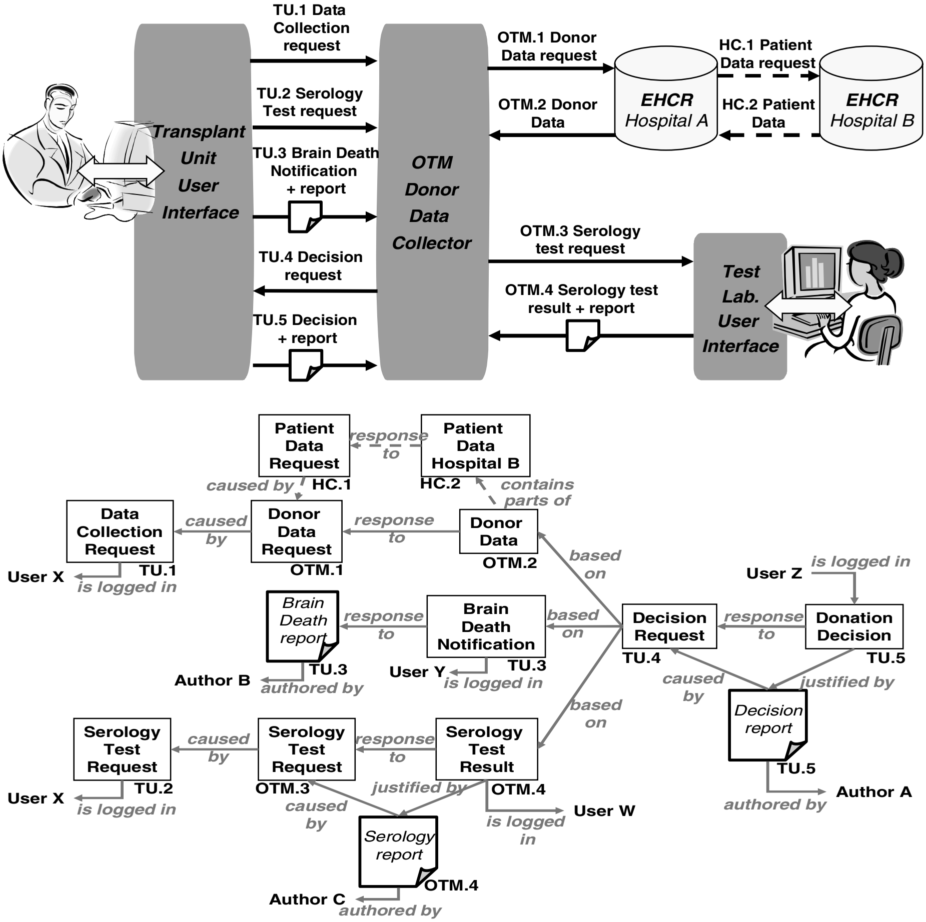

Fig. 2. Example scenario: (top) Interactions of the OTM components involved in a donation decision; (bottom) DAG showing the provenance of the donation decision

3.2 An Example

To illustrate how provenance is handled in the OTM application, let us see how the provenance of a medical decision is recorded and then queried. 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[1] through the OTM application. After brain death is observed and logged into the system (along with the report certifying the brain death), if all requested data and analysis results have been obtained, a

[1] A serology test is 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.

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 justification.

Figure 2 (top) shows the OTM components for this small scenario and their interactions. The Transplant Unit User Interface passes requests (TU.1, TU.2) to the OTM Donor Data Collector service, 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 service 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* OTM which are related to the donation decision. 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 has a parallelism to what happens in the real world, as we already said this is not enough to fully answer which is 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). Following back the p-assertions graph in Figure 2 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 from another hospital), 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 the provenance stores of the different institutions are connected, to solve the aforementioned problem is to solve the issue of discovering the different p-assertions related to the same patient. If this discovery step is done, then actors can make p-assertions that link together the separated sets of p-assertions to create a larger provenance document providing an integrated view of the healthcare history of the patient. The discovery can be done with the help of a patient identifier known to all actors. For privacy reasons the patient identity has to be anonymised. In the OTM application the EHCR system adds case identifiers (identifiers created at run-time) inside the p-assertions to create connections between sets of p-assertions related to the same patient. The result (not shown on Figure 2) would be that the provenance of *Patient Data Hospital B* would be added to the DAG as part of the provenance of the Donation Decision. Linking provenance stores in different administrative domains raises some challenging issues on privacy and security, though (see [4] for more details).

We had to find equilibrium between the amount of collected data and the level of interference such data collection may cause in the real medical process. The use of the reports and the information logged by the staff does not give full information about what happens in real world, but gives more than enough information to trace the individual or team involved, while not introducing an excessive increase of workload on the medical staff (we use the same reports medical staff already produces). It is important to note that the person who is logged in might not always be who authors the justifying reports (both are recorded in OTM), and the time when things are reported to the system might not be the time when things have happened (both also recorded in OTM). This is common practice in medical teams: most of reporting is delegated to a team member having the proper credentials and time to do it,[2] although the report may be later checked and even signed by a prominent member of the team.

4 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 [5], lineage was a kind of audit trail that traces 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 [6, 7]. There was also relationship to versioning [8] and data warehouses [9]. The provenance concept was later further explored within the GriPhyN project [10]. These techniques were used in [11] 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 [1] builds on these concepts to conceive and implement industrial strength open provenance architecture for grid systems, including tools for managing and querying provenance stores along with high-level reasoning capabilities over provenance traces. The price to pay for this is that applications should be adapted in order to provide high-quality p-assertions that not only record their inputs and outputs but also the (causal, functional) relation between them. The alternative would be the use of automatic data harvesting techniques such as RDF tuples harvesting [12], where RDF tuples include attribution (*who*) and time (*when*) information which is then processed by an external inference engine in order to construct RDF graphs by some kind of extended temporal reasoning. Another alternatives reduce to a minimum the adaptation step needed to make an application provenance-aware by adding a middleware layer or an execution platform capable to automatically create provenance assertions from the captured events and actions [13,14]. Problem is that, in automatic provenance collectors, it is very hard to infer causal relationships by only comparing sources and times [15], sometimes with the extra help of some derivation rules [16] or rigid workflow definitions. In the case of the medical domain this is not enough. Returning to the example on figure 2, let us suppose that at time *t1* system records a donor data request from user *X* for patient *P*, at time *t2* it records a serology

[2] Records of the process may be done asynchronously to avoid delays in critical steps: for instance, a surgeon should not stop a surgery to record through the GUI interface his last decisions and actions taken. If there is enough personnel in the surgery room, an assistant will record the events and decisions in parallel; if not, recording is done after the surgery.

test request from user *X* for patient *P*, at time *t3* it records a donation decision from user *A* for patient *P*, and *t1<t3*, *t2<t3*. Even if users *A* and *X* are the same, it would be unwise to directly infer that the donation decision was based on the result of the donor data request and the result of the serology test request just because i) all refer to the same patient *P* and ii) both requests happened before the decision was made, as this may not be true in all cases (e.g., anything terribly wrong in the serology test would lead directly to a donation rejection without having to take into consideration the rest of the collected donor data). Adding some generic rules to the temporal reasoner to express on which sources a donation decision uses to be made would not solve the problem either, as these rules can hardly handle all exceptional cases. The solution is to include in the provenance representation an explicit way to express relationships between recorded assertions (e.g. the doctor ticks on screen some boxes indicating which parts of the donor data and which test results he based his decision on, and this is automatically translated by the adapted provenance-aware OTM application into several, very precise, *based_on* relationship p-assertions, valid for that specific case).

In organ allocation management, there are few IT solutions giving powerful support to the allocation of human organs. The EUROTRANSPLANT system [17] is a centralised system where all information and decisions are made in a central server, and all activity is recorded in standard logging systems. The *Swisstransplant* system [18], is a distributed system 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. Up to our knowledge, the application of provenance techniques to distributed transplant management is novel.

5 Conclusions and Ongoing Work

In this paper we present an application of a service-oriented architecture for provenance applied in distributed medical systems. We used as example the domain of human organ allocation for transplantation purposes, where provenance is used to trace the actors that were involved in the important steps of the process (e.g., a medical decision) and to provide an integrated view of the medical history of a patient through the recollection of the medical treatment processes carried out in one or several institutions. 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 us feedback in the lasts steps of the development and fine-tuning of the application.

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.

References

1. The EU Provenance Project Enabling and Supporting Provenance in Grids for Complex Problems (IST 511085), http://www.gridprovenance.org/
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. CEN/TC251 WG I.: Health Informatics-Electronic Healthcare Record Communication-Part 1: Extended architecture and domain model, Final Draft prENV13606-1 (1999).
4. T. Kifor, L.Z. Varga, S. Álvarez, J. Vázquez-Salceda and S. Willmott. "Privacy Issues of Provenance in Electronic Healthcare Record Systems". Proceedings of the 1st Int. Workshop on Privacy and Security in Agent-based Collaborative Environments (PSACE 2006).
5. 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
6. Buneman, P., Khanna, S. and Tan, W.-C., "Why and Where: A Characterization of Data Provenance" in International Conference on Database Theory, (2001).
7. A. Woodruff,and M. Stonebraker. "Supporting Fine-Grained Data Lineage in a Database Visualization Environment", Computer Science Division, U. of California Berkeley, 1997.
8. A. Marian, S. Abiteboul, G. Cobena, and L. Mignet. "Change-Centric Management of Versions in an XML Warehouse", in 27th Int. Conf. of Very Large Data Bases, (2001).
9. Y. Cui, J. Widom and J.L. Wiener. "Tracing the Lineage of View Data in a Warehousing Environment", ACM Transactions on Database Systems, 25 (2). 179–227.
10. The GriPhyN Project. http://www.griphyn.org
11. I. Foster, J. Vockler, M. Wilde, Y. Zhao. "The virtual data grid: A new model and architecture for data-intensive collaboration. In: Proceedings of the First Biennial Conference on Innovative Data Systems Research, CIDR 2003, Asilomar, CA, January 5-8, 2003
12. J. Futrelle. "Harvesting RDF triples". International Provenance and Annotation Workshop, Chicago, IL, May 2005.
13. R. Barga. "Automatic Generation of Workflow Execution Provenance". International Provenance and Annotation Workshop, Chicago, IL, May 2006.
14. K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer. Provenance aware storage systems. In Proc. of the 2006 USENIX Annual Technical Conference, June 2006.
15. U. Braun, S. Garfinkel, D. A. Holland, K.-K. Muniswamy-Reddy, M. I. Seltzer. "Issues in Automatic Provenance Collection". In: Proceedings of the International Provenance and Annotation Workshop, Chicago, IL, May 2005.
16. I. Foster, J. Voeckler, M. Wilde, Y. Zhao: Chimera: A Virtual Data System for Representing, Querying and Automating Data Derivation. In Proceedings of the 14th Conference on Scientific and Statistical Database Management, 2002.
17. Eurotransplant International Foundation. http://www.eurotransplant.nl/
18. Swisstransplant. http://www.swisstransplant.org/