Web-based Support of Time-critical Services for Image-based Intervention Planning

A. Hennemuth[1], H. Bourquain[1], W. Berghorn[2], I. Wetzel[3],
M. Lang[1], A. Schenk[1], H.-O. Peitgen[1]

hennemuth@mevis.de, bourquain@mevis.de, berghorn@mevis.de,
wetzel@informatik.uni-hamburg.de, lang@mevis.de, schenk@mevis.de, peitgen@mevis.de

[1]MeVis Research, Universitaetsallee 29, 28359 Bremen, Germany,
[2]MeVis Technology, Universitaetsallee 29, 28359 Bremen, Germany,
[3]Universität Hamburg, Vogt-Kölln-Straße 30, 22527 Hamburg, Germany

Abstract: Preoperative risk assessment for image based surgery planning often requires considerable technical and personnel resources especially for certain complex interventions, such as liver surgery. The purpose of the project SIMPL was to establish a remote service for specialized tasks in medical image analysis and the visualization of results. In this context we developed a web-based system to support the service process. Requirements included an adaption to the existing clinical infrastructure, time constraints, and quality management support. The software design is based on a process modelling concept and modern web technologies are used for an implementation that fulfills the demands regarding security and independence of special software. The prototypical system has been tested during the SIMPL project with international clinical partners and is now used in optimized form on a regular basis.

1 Introduction

Web-based applications are nowadays common and widely accepted in almost all areas of daily life. Especially in the medical sector, where patient information is often spread over different organizations, network-based information exchange plays an important role. The constraints for the exchange of medical data are complex and there already exist many special solutions for communication between medical organizations [ESM+05, KMKS05]. To manage this exchange and assure the legally required security, these systems use data exchange servers and special software installed on all participating computers.
External services such as for laboratory analysis are a cost-effective and viable option for many specialized tasks. The complex analysis of medical images requires very specific knowledge as well as an adequate hardware and software environment. Therefore institutes like *Heart Core* [BIT04] or *IXICO* [Hil06] have started to offer support for the analysis of large amounts of image data . These institutes work mainly on the analysis of study data and do not have to deal with short-dated requests. However, there are also projects supporting the diagnostic analysis of patient related data [HEE+04]. Many concepts supporting computationally expensive analysis services base on grid architectures which demand a relatively high connectivity of the involved computers [BBF+03, GBC+05].

Service Support for Image-based Intervention Planning The special requirements coming along with providing an image-based service for intervention planning support can be divided into three groups: First, the kind of data to be analyzed has to be taken into account. The analysis is based on DICOM, which normally contains private patient information that must not leave the hospital. The anonymization and sprotection of patient's privacy has to be assured. Furthermore, the establishment of a fast and reliable data transfer is important, especially considering that a corruption of data can lead to false results.
Another difficult task is to cope with the different hardware and software environments that are found in hospitals. Operating systems, type of internet access and firewall settings vary significantly making the development of a universally usable client software very difficult.
Finally the specific tasks of planning interventions bring along some requirements. In worst case the period from service request to the scheduled intervention covers only some hours.

2 Infrastructure

One major requirement is the unproblematic use with the hospitals' existing hard- and software. So we developed a solution with internet access via the secure HTTPS-Port 443, a webbrowser, and a Java Runtime Environment as only prerequisites on client side. To establish the support of service requirement and data exchange, a webserver and a database have been installed in the so called *demilitarized zone (DMZ)* outside the firewall of the service provider. All incoming and outgoing data passes this server. Management and logging of the service flow is done via the database.

3 Software Architecture

The software architecture arises from a combination of the requirements given by infrastructure with the conceptual software design. There are many approaches for modelling interorganizational cooperation workflows [GAHL00, vdA00]. We decided to use the so called *Serviceflow Concept* which has been applied to orthopedic intervention planning processes before [WK04].

3.1 Service Modelling

In the *Serviceflow Modelling* concept, a service is described as an entity floating between *servicepoints*. At these *servicepoints* users, who are assigned special roles, have to fulfill tasks. Results are added to the so called *servicefloat* which carries the process information associated with a case. *Servicepoints* can be related to spatial locations, organizational units or a package of coupled tasks. Every service is an instantiation of a service template which defines the included *servicepoints* and the services tasks with required roles as well as pre- and postconditions. For our implementation, Fig. 2 shows the initially identified *servicepoints* clustered by organizational units and possible paths of the *servicefloat*.

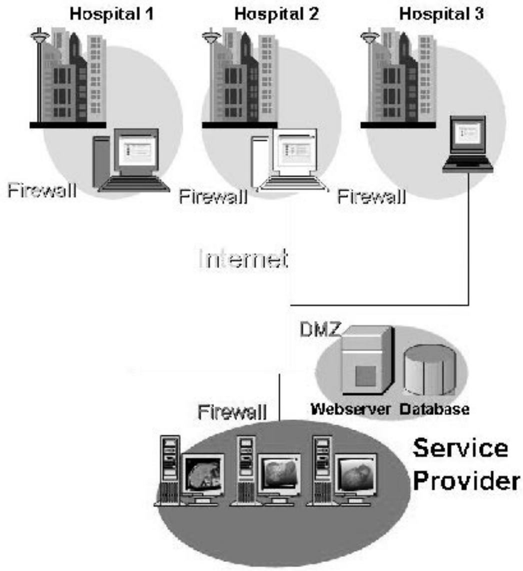

Figure 1: Infrastructure

The tasks to fulfill at a *servicepoint* can be described using standard UML diagrams as shown in Fig. 2.

3.2 From Model to Software Architecture

The final model derived from the one in Fig. 2 already describes a structured service template. As our infrastructure uses a central database to control the service process we directly mapped this model to database tables and classes instead using XML representations like proposed by Wetzel and Klischewski [WK04].

3.2.1 Data Model and Class Structure

The data model resulting from the requirements described in section 2 and 3 can be grouped into 4 parts:

1. **User Management:** This part of the database contains information about the system's users, their organizations and their possible roles in the process.

2. **Service Templates:** The tables in this part of the database describe the provided services. The developed model is mapped to tables determining tasks with assigned roles, pre- and postconditions grouped by *servicepoints* which describe functional entities.

3. **Service Process Management:** The instantiations of the templates consist of the selection of *servicepoints* and tasks according to the selected service. This data is

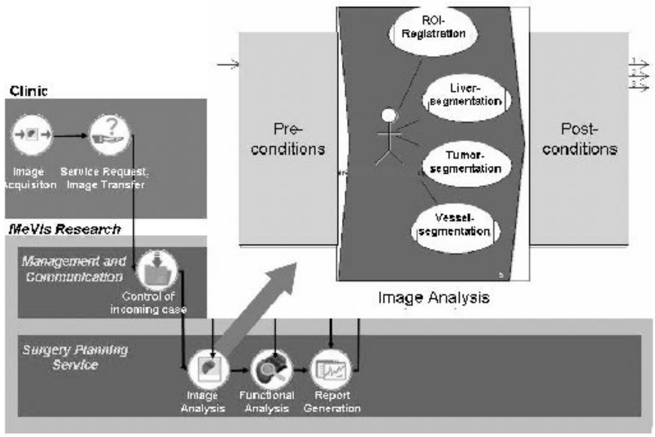

Figure 2: The service model

combined with information about the involved users and the status of the described process. Furthermore all events, errors and data transfer information logs are stored with a service.

4. **Service Data Management:** As described above, the data processed in the specific services requires special treatment. Thus a description of the type of data, the quality and the archiving status has to be stored. This also allows for a later use in scientific studies.

The classes managing the service flow consist basically of two groups, one managing the access to the external information stored in the database or in XML, the other one performing the manipulation of the *serviceflow* objects.

3.2.2 Implementation

The environment for the implementation of given classes is determined by the installed webserver. Using the Apache Webserver [Fou05] a combination of C++ and cgi-scripts can be used. With Tomcat [Fou06] we use Java to implement the core functionality and JavaServerPages for the dynamic interface generation. The main functionality of the software is implemented on the central webserver. Only for the upload and download of the input and result data the computer on the client side has to be accessed.

3.2.3 User Interfaces

There are two kinds of user interfaces to the system described above. The first one is used to define the static control information. Via this interface information about organizations, roles of users, and service templates can be accessed. The other one is the interface to the service instantiations for the processing of specific cases and will be described in the following.

Serviceflow Interface The websites constituting the user interface are built dynamically from a set of display elements and information, provided by the service control objects. The generic interface contains a login page, a page with an overview of cases the user is involved in, a page to access the case information and a page to access the service process information (Fig. 3). However the views on a *serviceflow* instance are adapted to the users role and the current service process state. While users in the role of service providers have a detailed view on single tasks and responsibilities, the client interface is dedicated to support the ordering process and the supply of result information. Furthermore, the dynamic generation of interfaces allows for a personalization and an adaption to an organizations ontology.

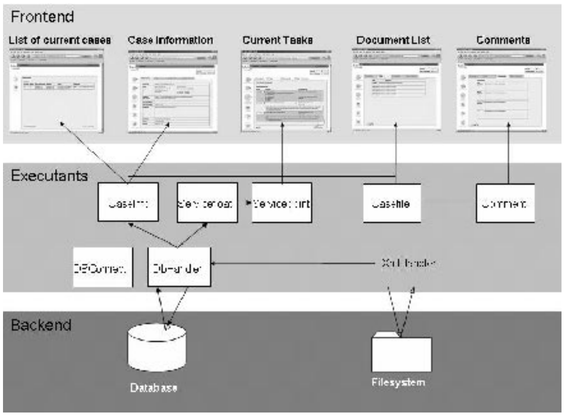

Figure 3: Structure of the user interface implementation

3.3 Management of Data Transfer

The reliable, fast, and secure transfer of the data to process poses a difficult problem, when all kinds of operating systems, transmission lines, and firewall settings have to be taken into account. Furthermore, a direct use without special installation should be possible from every computer. To achieve this, we developed a *Java Web Start* application [SM06]. With this technology, that is part of the Java Runtime Environment, our application can be started directly via a hyperlink to a *JNLP* file. The browser directs this MIME-type directly to *Java Web Start*. If no up-to-date version or the application is available on the client computer an automatic download and installation is performed. Subsequently *Java Web Start* runs the application in a secure environment. The user is shown information about the software provider based on a digital signature. He has to decide whether to trust this source before the local file system can be accessed. As in most clinical organizations network access is restricted, data is transferred via the HTTP protocol using port 80 or the HTTPS protocol on port 443. Data is divided into small portions, which are sent via a single socket connection and reassembled on the server. That way proxies with a data size restriction can be passed. Furthermore, interrupted transmissions can be completed. without transferring the whole dataset again.

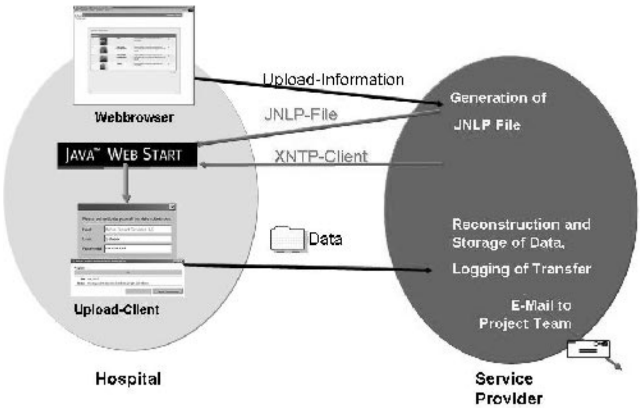

Figure 4: Datatransfer

4 Results

When the project SIMPL started, clinical partners had to download and install anonymization and compression tools on their computers. To send a case, manual anonymization and compression into packages $\leq$ 200 MB was necessary and the standard https upload took up to 5 hours from clinics with slow internet connection and sometimes resulted in faulty zip-files. A solution using a dedicated Java based upload client brought up problems arising from computer and network configurations in the clinical sites and showed many drawbacks. New users who wanted to send data sometimes needed the administrator to install the software, which means a loss of time, especially as this installation has to be done on all computers qualified for upload. Furthermore, the installation of software accessing the internet was sometimes not allowed and there occurred problems with path and file names in different languages.

With the introduction of the new software solution the effort for sending data were reduced to the export from the clinics PACS to a computer with internet access via https. Data can be sent by different people from different computers in one organization now. The upload time needed decreased to 30 minutes in institutions with slow internet connections and 15 minutes averaged. Problems occurred in two cases, in one of which a file could not be received because it extended the operation systems address space limitation of 2 GB. In the other case the transfer aborted due to a disk space problem on the server. At project startup 13 clinical sites participated. The number of cases came up to 11 per month with a turnaround time of 6,8 weekdays on average. In the course of the project 39 sites from all continents except for Africa joined the project. As shown in figure 5 the amount of processed cases per month rose to 45 while the turnaround time was reduced to less than 3 days.

While in the beginning the service was mainly dedicated to the support of liver surgery it was extended to the preparation of other interventions. New *serviceflows* were added and the number of analyzed cases concerning with these services amounts 75.

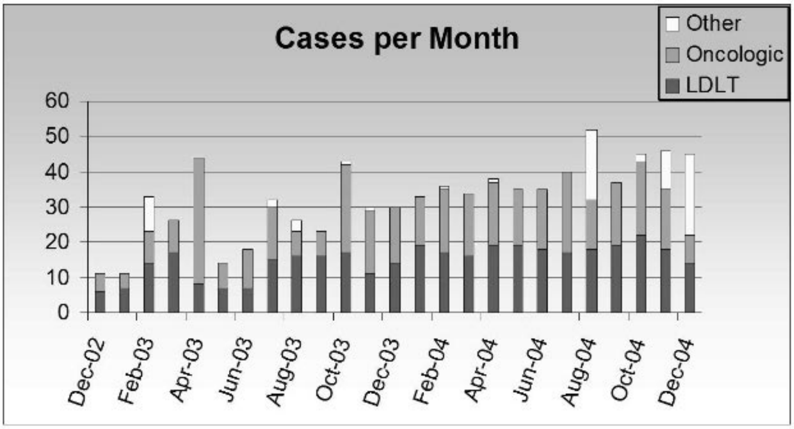

Figure 5: Development of case occurence during the SIMPL project

5 Conclusions

The idea of the SIMPL project was to establish a distant service for the support of intervention planning in liver surgery [ZBH+05]. This service should be provided without any overhead for the client apart from ordering and examining results, and take into account requirements regarding quality management, data security, and temporal constraints. The developed system is based on a central database which contains the templates for the service workflows as well as the status, logging information and data management information of service process instances. The core functionality is provided by cgi- or Java-classes running in the environment of the central webserver. To request and access services, user-adapted web interfaces are generated dynamically and data transfer is also started via weblink. Data is transferred in small portions via firewall enabled protocols using a robust method. The developed system has been successfully tested in the SIMPL project and an extended version is currently used by *MeVis Distant Services* [Bou05].
Results have shown that with the developed software the service workflow could be optimized especially regarding the turnaround time, reliability of data transfer, and the detachment of obstacles for clinical users.
The successfull application of the described software encourages further developments. The system is currently designed to work with one client, who orders the service, sends the data to be processed, and receives the results. In clinical practice the image acquisition and the intervention are carried out by different organizational units who do not necessarily belong to the same hospital. To simplify the process for all involved partners it would be desirable to integrate the radiologists and surgeons in a more comfortable way. While the image acquisition and transfer of incoming data could be agreed between the service provider and the radiologist the surgeon would only have to specifiy the required service and examine the provided results. As the underlying model of the systems architecture is general, this extension would mean no significant changes. Only the roles and service

templates would have to be adapted and extended.

References

[BBF+03] G. Berti, S. Benkner, J.W. Fenner, J. Fingberg, G. Lonsdale, E. S. Middleton, and M. Surridge. Medical Simulation Services via the Grid. In Sofie Nørager, Jean-Claude Healy, and Yves Paindaveine, editors, *Proceedings of 1st European HealthGRID Conference*, pages 248–259, Lyon, France, Jan 2003.

[BIT04] Inc. Bio-Imaging Technologies. Bio-Imaging Technologies, B.V. http://www.bioimaging.com/product/product.html, 2004.

[Bou05] H. Bourquain. MeVis Distant Services. http://www.mevis-distant-services.com, 2005.

[ESM+05] U. Engelmann, A. Schröter, H. Münch, T. Schweitzer, V. Braun, and H.P. Meinzer. *Von der bilateralen Teleradiologie zur Vernetzung von Regionen - der CHILI-Ansatz*. Jäckel, 2005.

[Fou05] The Apache Software Foundation. The Apache HTTP Server Project. http://httpd.apache.org/, 2005.

[Fou06] The Apache Software Foundation. Apache Tomcat. http://tomcat.apache.org/, 2006.

[GAHL00] P. Grefen, K. Aberer, Y. Hofner, and H. Ludwig. Crossflow: Cross-organizational workflow management in dynamic virtual enterprises. *International Journal of Computer Systems Science and Engineering*, 15:277–290, 2000.

[GBC+05] C. Germain, V. Breton, P. Clarysse, Y. Gaudeau, T. Glatard, E. Jeannot, Y Legré, C. Loomis, I. Magnin, J. Montagnat, J. Moureaux, A. Osorio, X. Pennec, and R. Texier. Grid-enabling medical image analysis. *Journal of Clinical Monitoring and Computing*, 19(4–5):339–349, Oct 2005.

[HEE+04] O. Hejlesen, B. Ege, K.-H. Englmeier, S. Aldington, L. McCanna, and T. Bek. TOSCA-Imaging–developing Internet based image processing software for screening and diagnosis of diabetic retinopathy. *Medinfo*, 11(1):222–226, Sep 2004.

[Hil06] D.. Hill. IXICO - Automated Image Analysis Solutions. http://www.ixico.net, 2006.

[KMKS05] M. Kämmerer, P. Mildenberger, G. Klos, and B. Schütze. SecTelMed - Teleradiologie durch Open Source Einsatz. In *50. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie*. German Medical Science; Düsseldorf, Köln, 2005.

[SM06] Inc Sun Microsystems. Java Web Start Technology. http://java.sun.com/products/javawebstart, 2006.

[vdA00] W.M.P. van der Aalst. Loosely coupled interorganizational workflows: Modeling and analyzing workflows crossing organizational boundaries. *Information and Management*, 37:67–75, 2000.

[WK04] I. Wetzel and R. Klischewski. Serviceflow beyond workflow- IT support for managing inter-organizational service processes. *Inf. Syst.*, 29(2):127–145, 2004.

[ZBH+05] S. Zidowitz, H. Bourqain, A. Hennemuth, A. Schenk, W. Berghorn, and H.-O. Peitgen. SIMPL: Service Center for Medical Image Analysis, Visualization, and Therapy Planning. In *Biomedizinische Technik*, volume 50, pages 151 –152, Sep 2005.