2nd International Conference "From Scientific Computing to Computational Engineering"
2nd IC-SCCE
Athens, 5-8 July, 2006

DEVELOPMENT OF A WEB SERVICE PLATFORM FOR REMOTELY DESIGNING MEDICAL IMAGE PROCESSING APPLICATIONS

Chikimtzis Chrysovalantis[1], Georgiadis Pantelis[1], Sidiropoulos Konstantinos[1], Daskalakis Antonis[2], Dimitropoulos Nikos[3] and Cavouras Dionisis[1]

[1]Medical Image and Signal Processing Lab., Dept. of Medical Instruments Technology, STEF,
Technological Educational Institution of Athens,
Ag. Spyridonos Street, Egaleo, GR-122 10, Athens,Greece.
email: *hikimtzis@yahoo.gr*, *cavouras@teiath.gr*, web page: *http://medisp.bme.teiath.gr/*

[2] Medical Image Processing and Analysis Group, Laboratory of Medical Physics,
University of Patras
26500 Patras, Greece
web page: *http://mipa.med.upatras.gr*

[3] Medical Imaging Department,
EUROMEDICA Medical Center,
2 Mesogeion Avenue, Athens, Greece.

Keywords: remote processing, web services, axis, workflow, internet, soap, java

Abstract: *The aim of the present study was to design and implement a web service (WS)-based image processing platform for providing medical image processing techniques remotely. Several categories of medical image processing algorithms were implemented and deployed at a server site using J2EE. Algorithms comprised grey-scale histogram modification, image segmentation, spatial and frequency domain image enhancement. The processing functions were exposed as web services, at the server side, employing the AXIS framework and the JBOSS platform. At the client side, the developer can invoke the remote processing methods to build an efficient medical image processing application in a quick and convenient way. The proposed system was tested by setting up the server at a site in Athens, Greece, and the developed client module at a site in Patras, Greece. Ten of the server-exposed algorithms at the Athens site were used by the Patras' staff to develop a particular application. Processing time depended on the network status between the two sites, the image size, and the processing algorithm invoked. For a 512x512 grayscale image the processing time was about 20 seconds for a typical convolution filtering, including the network overhead. The exposed image processing functions were proven to be flexible and convenient rendering the platform a helpful developing tool.*

1 INTRODUCTION

The distribution of processing is an important issue when computer processing time demands are high. Distribution of workload may be accomplished on different workstations (servers) that may be situated at remote sites. Cross-platform, web-based applications can be developed to provide services to remotely situated web clients. The cutting edge web services technology, can deploy a number of functionalities that can run on different servers over the Internet[1].

The purpose of the present study was to develop a web-based platform for distributing demanding image processing workloads to different powerful servers situated at remote sites. Additionally, users at different sites can develop their own image processing functions and, by using the proposed framework, they can easily share the functionality of those functions with other users of the platform. Finally, at each client, a workflow of consecutive processing tasks may be composed for processing a set of images residing locally, by functions provided by different servers. For accomplishing these tasks, state of art software tools, incorporating the functionality of web-services, were employed to perform remote image processing.

2 MATERIALS AND METHODS

Two servers were situated at different sites, one at the Medical Image & Signal Processing (MEDISP) lab at

TEI-Athens and the other at the Medical Image Processing & Analysis (MIPA) lab at the University of Patras. Both servers utilized the Axis 1.3 platform on the JBOSS 1.0 application server, for hosting the web-services functionality **[Fig. 1]**. The two servers (Intel Pentium 4, 3.2 GHZ, 1GB RAM) comprised the image processing web functions, implemented in Java 2 Enterprise Edition (J2EE). The web functions were described (name, return types and input parameters) by the WSDL (Web Services Description Language) and were deployed (published) on the servers.

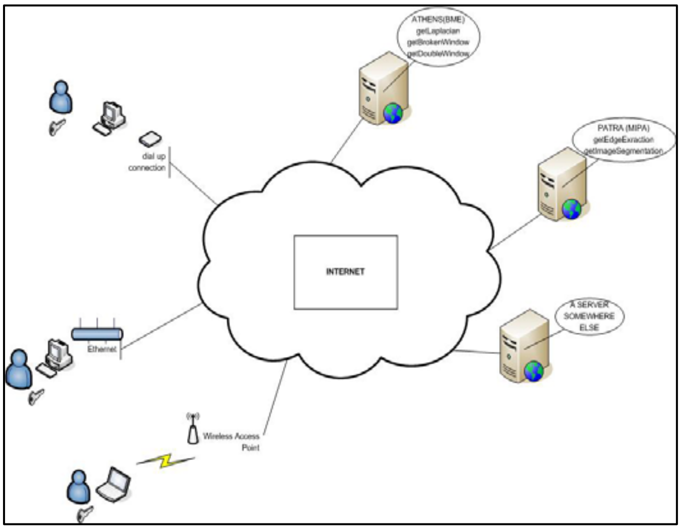

Figure 1: The System's Architecture

The WSDL language is an XML-based form, which describes the type of the routines to publish. The methods have as an input type and as a return type a double array String. The pixels of the images and the parameters are stored in that array and are transmitted through the internet in the form of SOAP[1] (Simple Object Access Protocol).

The client with a suitable graphical user interface (GUI) **[Fig. 2]** performs a UDDI (Universal Description, Discovery, and Integration) search to discover the URL (Universal Resource Locator) of the servers that host the image processing methods and register the available methods at its side[3]. After discovering the internet address (URL), the particular function is invoked by the client in order to retrieve the names, the return types and the description of the methods residing on that server. Both the description and name of every method is stored in the user's workstation.

The client is able to select a number of images for processing. The user can design a sequence of processing methods (workflow) to be applied on each image consecutively. Furthermore, the user can design a common workflow and apply it to a group of images.

The number of requests and responses between a server and a client is depended on the number of methods a workflow is composed of. The call between the server and client is synchronous. That is to say, the client can send one image at each time and invoke only one method at that time. The next method in the workflow will be invoked after receiving the response of a previously called method, therefore, creating a "processing chain".

The pixels encapsulated in the SOAP message are encoded in the Base64 encoding scheme, to ensure the validity of the XML message.

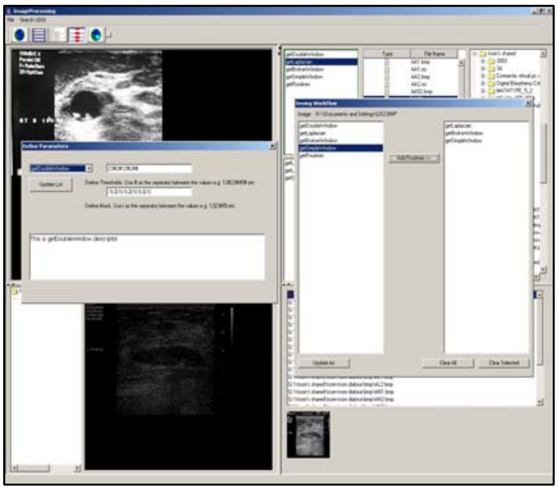

Figure 2 : The client – side application

Regarding security, the SSL (Socket Secure Layer) protocol was employed in order to provide encrypted exchange of information between all parties (clients and servers)[2]. Therefore, both the client and server used a set of certificates for mutual authentication, before any exchange of data. In particular, the client used its own private key for encryption, while data were decrypted at the server-side by means of the client's public key. The reverse procedure followed, in order to accomplish mutual authentication.

Using the developed Java–based client application, 20 US mammograms (7 simple cysts, 10 complicated cysts and 3 malignant tumors at 512x521x8 image resolution) were selected and processed by the Laplacian mask and the image segmentation algorithms (edge extraction).

3 RESULTS AND DISCUSSION

The use of Java in combination with XML and web services promoted interoperability and rendered the system platform independent[3].

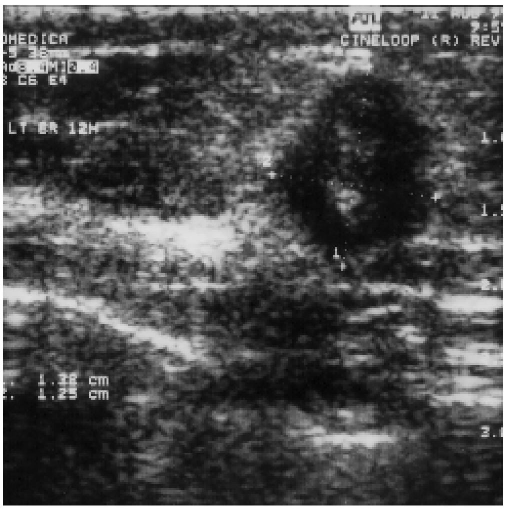

Figure 3 : Original image

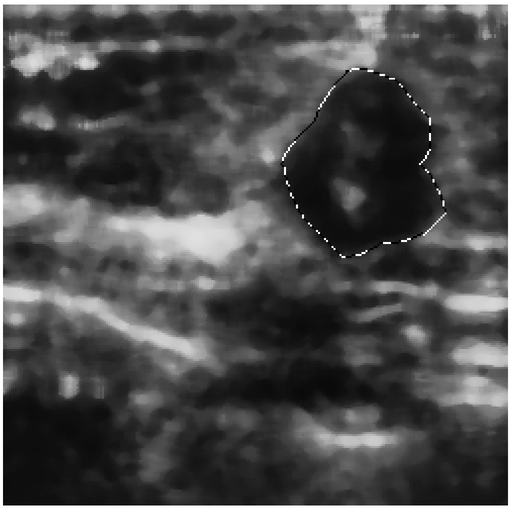

Figure 4 : Processed image

The time of UDDI search with a 56Kps speed connection is approximately 13 seconds including the time for the retrieval of the published methods. The typical time for processing a 512x512x8 image resolution was approximately 20 seconds **[Fig. 3-4]**. The response time varied in accordance with the size and type of the image; for an RGB image 1024x1024x24 the waiting time was approximately a minute. The speed of the connection is also depended on the current network speed and congestion status.

The Graphical User Interface (GUI) of the client-side application was designed to be simple and user friendly even for users unfamiliar with similar applications.

The implementation of the SSL protocol enhanced the security of the system by ensuring that only authorized users can connect to valid servers.

4 CONCLUSION

Web enabled remote image processing with cutting edge technologies, like web services, can ensure a reliable framework for future implementations exploited by researchers or physicians.

5 REFERENCES

[1] Professional XML Development with Apache Tools: Xerces, Xalan, FOP, Cocoon, Axis, Xindice by Theodore W. Leung, Wrox Press © 2004.

[2] IBM Systems Journal, Vol 41, No 2, 2002 “Securing Web services” M. Hondo, N. Nagaratnam, A. Nadalin

[3] IEEE Intelligent Systems “Semantic Web Services” Sheila A. McIlraith, Tran Cao Son and Honglei Zeng, *Stanford University*