

Visualization Service Based on Web Services

Bojan Blazona¹ and Zeljka Mihajlovic²

¹Ericsson Nikola Tesla, Krapinska 45, 10 000 Zagreb, Croatia,

²University of Zagreb, Faculty of Electrical Engineering and Computing
Department of Electronics, Microelectronics, Computer and Intelligent Systems,
Unska3, 10 000 Zagreb, Croatia,

bojan.blazona@ericsson.com, zeljka.mihajlovic@fer.hr

Abstract. *Scientific visualization evolved up to the stage where collaborative work in virtual coexistence of various experts becomes unavoidable to identify and solve specific visualization tasks. These problems encouraged us to research and define key aspects of visualization service which will enable collaborative work over the Internet. In this paper we define architecture of the whole system and unified interface for all requirements in such environment. Unified requests are sent to visualization processing unit to execute required visualization routine. Collaborative work on the volumetric medical data is especially demanding task, so we present development of VMI (Visualization of Medical data over Internet) system, where segmentation for visualization is based on joint histogram.*

Keywords: Visualization, Web services, Service Oriented Architecture.

1. Introduction

Vendor adoption of web service's concepts over the Internet, with new enhancements in security concepts and fast computer networks is a reasonable base for development of the system which will give the ability of collaborative work on visualization task definition from distant locations over the Internet.

Data visualization and the routine distribution with ability to process request on distant place, we call Visualization Over the Internet (VOI). Concept of the visualization over the network is not new, but central data and visualization routine repository with ability to collaborative work on visualization routine is new, and architecture and interfaces for that purpose are proposed in this paper. Also, we focus our attention on visualization problems where amount of data is significant. Medical volumetric

data such as CT or MR sequences usually contain huge amount of data, but isosurface rendering could also generate significant amount of data.

Some pioneer concepts are presented as server-based visualization approaches developed for specific visualization needs like brain mapping [10] or building anatomical 3D scenes [2]. Our approach follows server based visualization that relies on the processing routine [7] where script is used to define processing routine and VRML is used to present generated visualized data.

Project behind this paper is focused on medical data visualization, but concept is applicable to the all visualization tasks from other areas of the science and human work. Problems about security, huge amount of generated data, privacy and interaction in any stage of the visualization process are new tasks in desired system and this is described in Section 2.

Development of distributed visualization systems for medical purpose can decrease overall cost of multiple desktop applications. It can be done using central system (medical visualization unit) over network communication channels. In all, medical staff and computer scientists can collaborate to produce better health care in such system using one central system from different geographical location.

Protocols such as SOAP (Simple Object Access Protocol) for data transfer and standards like VRML or X3D for virtual worlds are very helpful in definition of client side. Rendering of the objects using these standards is responsibility of the client viewers. These technologies are presented in Section 3.

In other way, new visualization patterns, from different areas of science, have to be simply adopted in system with respect to recent standard systems. New requirements in visualization have

to change standards and protocol first, or development of the custom client is necessary.

Needs like generating animation sequences based on visualization of anatomical human parts [2] in corporation with [1] can be mapped to similar process routine using base visualization functionalities (surface or volume rendering) with connectors to the Internet data centers where sequences are stored.

Emerging use of strict definition schema for processing routine with ability of enhancements can be found in XML as the routine script and XML Schema standard for the strict functionality set which can be simply checked and finally executed. Our definition of such schema can be found in Section 4. Finally, in Section 5, we present our VMI system. Developed system confirms to defined key aspects, following the proposed architecture and communication concepts described in first four sections.

2. Concept of the Visualization over the Internet

System that enables visualization over the Internet is a component based system. Identified components are client and visualization service. Visualization service is then granulated to interface, unified interface and visualization processing unit (Figure 1).

Client and visualization service are the main parts of the system that we present. Client's responsibility is sending volume or the other specific data, which must be accepted, processed and returned from visualization service.

Client or interface (for direct connection to standards e.g. DICOM or HL7 CDA) should provide a way to define processing routine for a given data to manage the visualization process. Client is responsible for the presentation of the visualized data. This can be provided by development of the new rendering software or integrating software applications to render specific responses from visualization service (3D viewer, graph generator, 3D printer).

At first, client sends data to the visualization service. Collaboration of scientists from different areas is evident in process routine creation on visualization service. More users can be members of the visualization project (medical case). Processing routine can be changed without resending data to visualization service. Visualization services are responsible to process data according to the created processing routine on the client's request. Processed data is retrieved on client, performing action of getting project by project's temporal ID (Figure 2).

To manage and process requests from client, visualization service has strict system architecture. This architecture consists of interface, unified interface and visualization processing unit (Figure 1). Communication interface to clients should be based on well used communication protocols and standards (such HTTP, HL7 or DICOM). All requests should be mapped to unified interface of the visualization service.

Interface is responsible for accepting of the requests according to implemented protocol (Figure 3). Therefore, requests go through unification process.

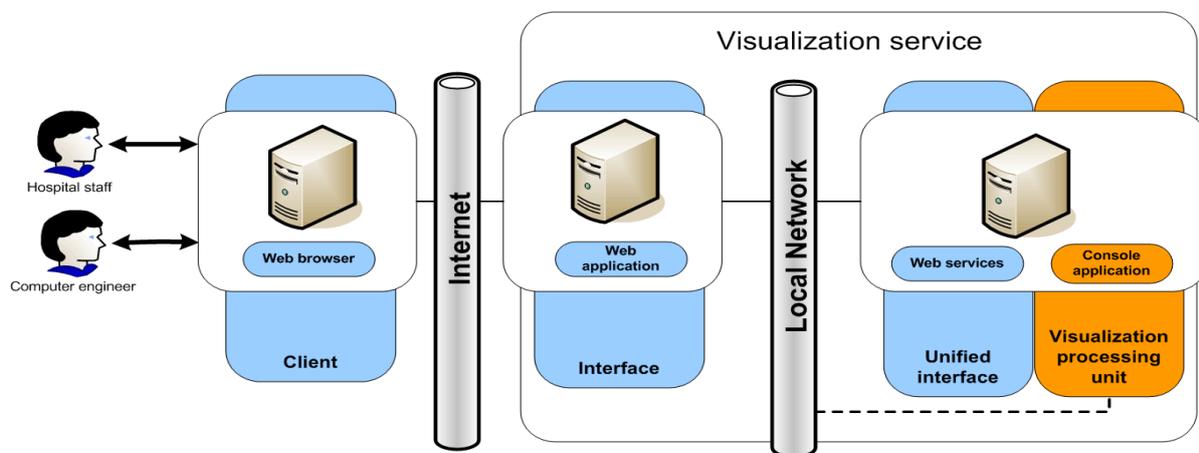


Figure 1. Overview of the system enabling visualization over the Internet.

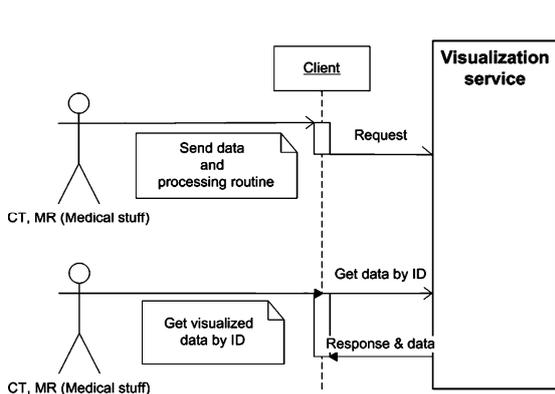


Figure 2. Client sends data and processing routine to visualization service.

Unified request is forwarded to unified interface. Temporal ID is returned to the interface after acceptance of the unified request on the unified interface. Requests are scheduled on unified interface if visualization processing unit is down or other process is running (this is the case when only one visualization processing unit is connected to unified interface). Time between invocations of processing request is set by operating system scheduling functionality. If visualization processing unit is ready, process starts and generated data are returned to unified interface where it should be saved. Client invokes get data by ID to see if specific request is processed through interface in event-based manner. Finally, generated data are returned if request is processed.

3. Visualization of the medical data over the Internet

Use of described VOI in the environment of medical data exchange and visualization, we call Visualization of Medical data over Internet (VMI). According to the defined key aspects, security is the most considerable task on the level of implementation.

Sending medical data, such as data sets or other reports, must go through process of anonymization. All personal data, that can identify patient, must be moved before sending data through communication channel. Identification should be managed on other way like temporal ID. Use of secured communication channel is demanded even if anonymization is used.

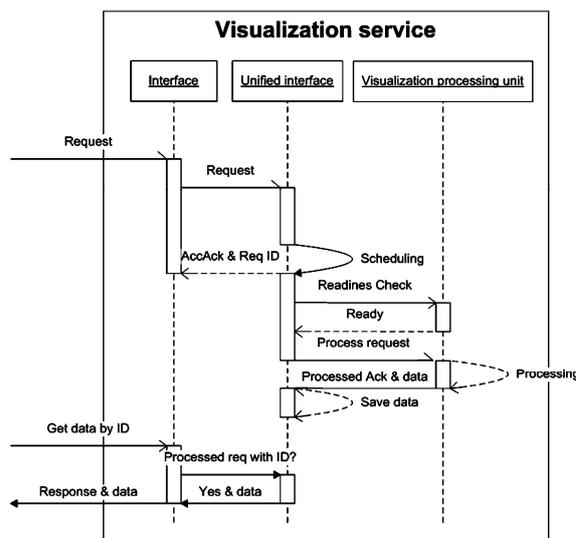


Figure 3. Client communicates with interface according selected protocol. Requests are unified and sent to unified interface.

Security threats are almost present because of the provisional server settings without advanced knowledge of the security concept of the underlying operating system.

Today's medical communication infrastructure consists of computer networks and other direct communication links. In all, DICOM [8] (Digital Imaging and Communications in Medicine) is widely used and accepted norm for storing medical data and transferring through computer network. DICOM's usage focused on radiology is overridden by new communication concepts, such as HL7 (Health Level 7) [9]. HL7 [3] can transfer all type of medical data through CDA (Clinical Document Architecture) [4]. Future system should implement DICOM or HL7 interfaces to VMI system. In this paper, we have chosen and developed Web Portal called VMI for sending DICOM files through HTTP interface (web application) to unified interface.

X3D [11] is open XML standard for exchanging 3D objects, scenes and for real time collaboration through network applications in 3D environment. Defined profiles encapsulate functionalities from fundamentals, like 3D object's and animation sequence definitions (Interchange profile), to NURBS (Non uniform rational B-spline) and humanoid animation (Full profile). Web3D Consortium has formed special group to develop MedX3D standard for medical use as enhancement to the X3D [12].

4. Processing routine and project definition

Today's visualization toolkits use processing routine defined as script in plain text file. This simple approach is good, but it can not accommodate the abstraction of the visualization process through the automatic imports of defined routines or special semantics according to the place of the visualization process subroutine definition. For example, generation of histograms must be implemented as the different functions with different parameters, but with our approach meaning of histogram is defined by its place in processing routine tree.

4.1. Project XSD schema

XML is the best standard to define processing routine. Strict schema for XML processing routine, defines allowed processing routine functionalities. It defines virtual object-oriented approach. Some tags define creation of objects and the other tags invocation of methods with specific parameters. Schema gives ability to validate lexical and semantically meaning of processing routine and demand strict implementation for vendors to implement defined functionalities.

Unification of all requests is verified by the Project XSD schema. Project file should define input file definition, input file data, security specific data, status and processing routine. Project files should be processed by visualization processing unit (Figure 1).

4.2. Web service interface

Web service implemented on unified interface should have methods such as Request (input is XML data according Project XSD Schema) and GetProject (Project ID). Exceptions are thrown if request is not valid according to Project XSD Schema. If exception isn't thrown, response parameter is project ID. Project ID identifies visualized data. Invocation of GetProject() method with returned ID will return visualized data.

Large datasets transferred through XML enlarge the size of starting dataset. Web services allow use of SOAP attachments to transfer binary data as they are without baseCode64 encoding. All binary files are attachments to SOAP message [13].

5. Technical overview of our VMI system

Client side of the system consists of web browser with plug-in to render X3D 3D objects and scenes generated by visualization service. Communication between client and visualization service is secured with SSL protocol.

Visualizations service is based on .Net platform developed by Microsoft. We have developed web application as interface to visualization service. Interface is available as web application through Microsoft's web server IIS (Internet Information Services). Unified interface is implemented as web service with functionalities specified by concept of VOI, presented in Section 2. Also, project schema follows definition for transferred data.

Unified interface is defined in WSDL (Web Service Definition Language). Visualization processing unit is C#.NET console application triggered by windows task manager which process requests stored in project repository (project repository is local or remote folder) managed by the unified interface. Processing visualization unit saves results in project's folder in the project repository. Client's request for getting visualized data is forwarded through interface to unified interface, according defined VOI sequence diagram for visualization service.

5.1. VMI XSD Schema and implemented functionalities

VMI XSD schema follows "object oriented" design. All objects and methods are defined in XML elements. Attributes of these elements are parameters of the object constructor or parameters of the method that invokes. We developed DICOM parser to extract text data like volume data size, pixel or color presentation of sliced images and other equipment's parameters from binary formatted DICOM dataset. Processing raw datasets requests input of the dataset volume size and real resolution.

5.2. Visualization and segmentation framework

In visualization processing unit, we have implemented segmentation framework. Segmentation framework can be used for surface rendering based on original Marching cube algorithm [6] with one isolation value parameter (Figure 4). We enhance segmentation framework

to enable surface rendering based on 2D area defined on 2D joint histogram. Idea is similar to idea presented in [5] for volume rendering. Modification of marching cube is necessary to enable the same usage in surface rendering (Figure 5).



Figure 4: Marching cube algorithm overview.

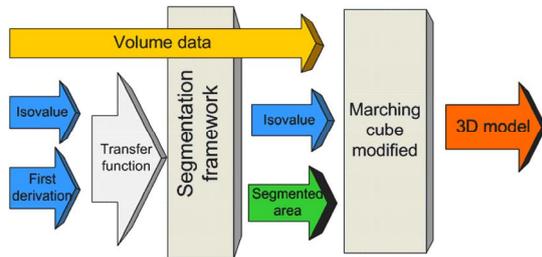


Figure 5: Segmentation framework and modified marching cube algorithm.

Processing routine could generate 1D histogram or 2D joint histogram (Figure 6; without area selection). Display of histogram is very important for correct segmentation and visualization. Selected area in 2D joint histogram defines transfer function as shown on Figure 6. Isovalue is not defined by one threshold value, but combined values of intensity and derivative are used for segmentation in transfer function.

Segmentation framework presents segmented area of same sized binary array. True presents vertex in given area (according 2D transfer function manifested as selection on 2D joint histogram) and false for vertex out of selected area.

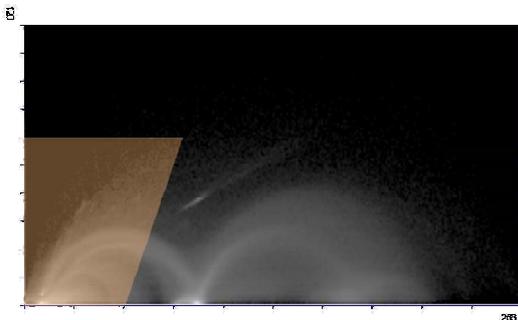


Figure 6: 2D joint histogram (2D Histogram) of volume data. X axis - intensity level, Y axis - magnitude of first derivative, area presents occurrences $N(x,y)$ in logarithmic scale. Selected area extracts sinuses and air.

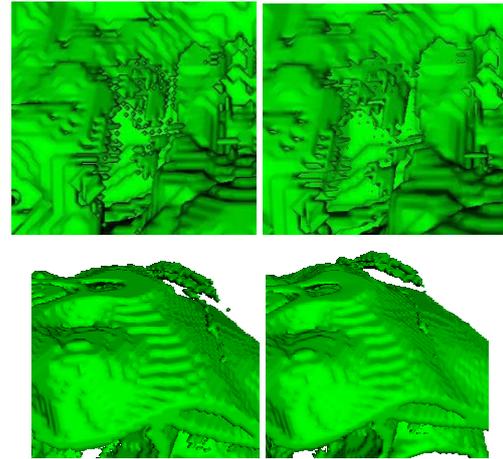


Figure 7: Left side- original object generated by marching cube; Right – Gauss separation.

This approach generates hashed vertex data that presents segmented area and should be somehow interlaced to produce 3D objects. Binary selection produces artifacts on the result, so we develop separation method which filters segmented area with Gauss filter kernel in three dimensions. This interpolation of segmented data results in smoother models (Figure 7). It is a good example of adding a new functionality to visualization processing unit.

According to specific segmentation, some specific operations are required such as area selection. X3D shape writer creates X3D object as result of Isosurface rendering on segmented volume data. Object writer creates obj file as simpler presentation without specific appearance definition.

6. Results

Logging to the VMI is required. User can select one of previously created projects to check project status and get processed data. Also user can create new project (Figure 8). Creation of the new project goes through the wizard to collect project details like definition and selection of input file type and file which will be uploaded to web application. Input file type can be “Raw data” or DICOM file. Raw data input file requires additional parameters. Project is then created and all data is saved on ASP.NET Application host server.

Finally, sample processing routine generates X3D scene consists of the X3D objects in 2 LOD levels. Sample generated X3D Scene presented in higher level of details consists of the 1 825 394 polygons (Figure 9).

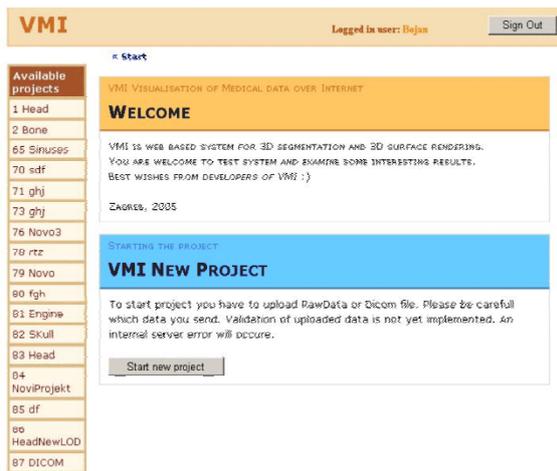


Figure 8: ASP.NET interface main screen with available projects (create new project option).

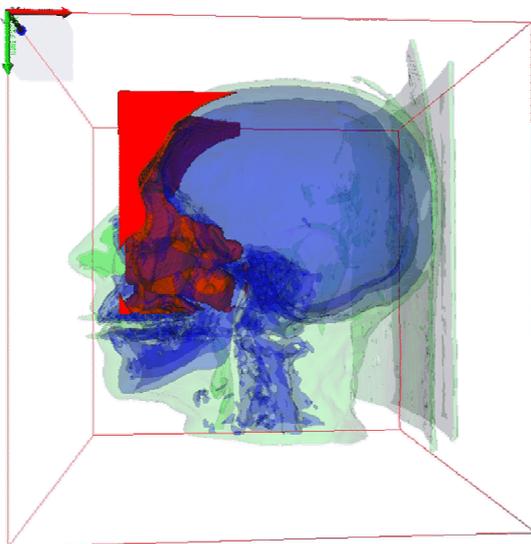


Figure 9: X3D scene generated from sample processing routine defined by VMI XSD Schema. Sinuses & air, bones and soft tissue are presented. 1 825 394 polygons in three high detailed X3D objects.

7. Conclusion

We have found that proposed architecture and given concepts of communication are acceptable according key aspects we define. Strict definition of communication based on presented architecture and XSD Schema processing routine definition should encourage research of the separated parts of the system to find out the most effective way of implementation and functionality enrichment. Our VMI can simply adopt new interfaces such HL7 or direct DICOM connectors.

Medical technicians or biomedical engineers can be introduced as human support in sequence of visualization process. If we define this

approach as new way of virtual medical environment the future research of presented architecture and communication concept of VOI is yet to be explored and developed.

8. References

- [1] Bhatti R, Shafiq B, Shehab M, Ghafoor A, "Distributed Access Management in Multimedia IDCs", IEEE Computer 38(9), 2005, pp. 60-69.
- [2] Evesque F, Gerlach S, Hersch RD, "Building 3D anatomical scenes on the Web", Journal of Visualization and Computer Animation, Vol. 13, J. Wiley, 2002, pp. 43-52.
- [3] HL7 Organization: HL7 Version 3, 2005 <http://www.hl7.org>.
- [4] HL7 Organization: The Clinical Document Architecture, 2005, <http://www.hl7.org/>.
- [5] Kniss J, Kindlmann G, Hansen C: Multi-Dimensional Transfer Functions for Interactive Volume Rendering, IEEE Transactions on Visualization and Computer Graphics, Volume 8 , Issue 3, pp. 270 – 285, 2002.
- [6] Lorensen WE, Cline HE: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Computer Graphics (Proceedings of SIGGRAPH '87), Vol. 21, No. 4, pp. 163-169.
- [7] Marovic B, Jovanovic Z, "Visualization of 3D fields and medical data and using VRML", Future Generation Computer Systems, Volume 14, Issue 1-2, June 1998, pp. 33 – 49.
- [8] NEMA: DICOM standard, <http://medical.nema.org/dicom/2004.html>
- [9] Oosterwijk H, "The relationship between DICOM and the CDA", 2nd International Conference on the Clinical Document Architecture, Acapulco, 2004.
- [10] Poliakov A, Albright E, Corina DG, Ojemann RF and Brinkley JF, "Server-based Approach to Web Visualization of Integrated 3-D Medical Image Data", Proc. AMIA2001, 2001.
- [11] Web3D Consortium: The X3D ISO standard, 2005, <http://www.web3d.org/>.
- [12] Web3D Consortium: "Medical real-time visualization and communication using X3D", <http://www.web3d.org/applications/medical/index.html>. 2005.
- [13] World Wide Web Consortium, "SOAP 1.2 Attachment Feature"