

Molecular Visualization with Supports of Interaction, Immersion and Collaboration among Geographically-Separated Research Groups

Moacyr Francischetti-Corrêa¹, Luis Carlos Trevelin¹,
and Marcelo de Paiva Guimarães²

¹ Departamento de Computação – Universidade Federal de São Carlos –
São Carlos – SP – Brasil

² Faculdade Campo Limpo Paulista/Instituto Federal de São Paulo – SP - Brasil
moacyr@spacnet.com.br, trevelin@dc.ufscar.br,
marcelodepaiva@gmail.com

Abstract. This paper explores the possibilities of integrate two or more geographically-separated research groups in the same molecular visualization. It is presented the JViewer (JAMP Molecular Viewer), which is a molecular visualization application with interaction, immersion and collaboration. The interaction and immersion are provide by the use of Virtual Reality technology; and the communication and synchronization using JAMP platform (Java Architecture for Media Processing), which is a distributed programming environment for multimedia applications.

Keywords: Molecular visualization, collaboration, immersion, interaction, Virtual Reality, JAMP Platform.

1 Introduction

Technological advances become feasible to use computer as laboratories, in which students immerse themselves within interactive contexts that challenge and extend their understanding. In 1965 was developed by Cyrus Levinthal the first interactive molecular visualization system (IMVS) [1]. Throw a plastic "globe" (the predecessor to the torque ball) the system was manipulated. This device enabled the user to orient a line drawing of a three-dimensional molecular model. By placing a light pen over the computer display, it was possible to select menu options, choose an object or execute zoom into parts of a molecule.

Nowadays the researchers are using Virtual Reality technologies to improve the molecular viewers. Virtual Reality offers solutions to build computer generated environments which combines immersion, interaction, multi sensing and a three-dimensional world. A molecular viewer based on Virtual Reality allows investigate the structure, properties and dynamics of a molecular system which are extremely complex and comprises millions or billions of atoms in an immersive and interactive environment. The user can interact with the molecules in real time (rotate, resize, measure, verify distances between atoms, angles, adjust rendering parameters) and he

can be immersed in the environment using stereoscopic vision. There is plenty of molecular viewers available (VRDD [2], Distributed PaulingWorld [3], Jmol adapted for collaborative use [4], Interactive Molecular Visualization System [5], AMMP-Vis [6], KinImmerse [7], MICE [8], VrMol [9], Gesture-Based Molecular Visualization Tool [10]). Some of them were projected to support local collaborative work. However, the author's don't know none of them that support collaborative work where the users are geographically-separated. Collaborative work means interdependence of activities. The users are interdependent in their work in the sense that one user's actions will change the state of a set of objects and processes and, in turn, this change of state has implications, directly or indirectly, for the work of the other members of the ensemble, and vice versa [11].

This paper describes the application JViewer which is an immersive, interactive and collaborative environment for visualization of molecules, where two or more geographically-separated research group can work together at the same time. The user can get immersion using stereo view. The interaction is reach throw the use of devices as Wii Remote and mouse. The collaboration was obtained throw the integration of geographically-separated research groups working in the same molecule simulation.

This paper is organized as follows: the section 2 present the JAMP Platform [12-16], which is the software infrastructure used to develop the JViewer (JAMP Molecular Viewer); in the section 3 is presented the JViewer, which is the molecular viewer based on JAMP (Java Architecture for Media Processing) Platform and Virtual Reality technologies; conclusion and references are, respectively, in sections 4 and 5.

2 JAMP Platform

The Fig. 1 presents the JAMP (Java Architecture for Media Processing), which is a set of frameworks and services that facilitates the development of distributed multimedia applications. JAMP applications can use any one of the available frameworks and service. Each framework provides a service: the JBroker framework is the base to develop new JAMP applications and provide the service of localization of other services; the AUDIO/VIDEO frameworks allow to add audio/video support in the applications and they offer the service of audio/video streaming; and the COLLABORATE framework is the foundation to collaborative application and it provides groupware services, as the concurrency control.

Although extensibility and code reuse are complex for distributed multimedia applications, it is important to achieve a way to make it easier. JAMP Platform aims to find a solution. It is part of its conceptions requirements as software extensibility and code reuse, providing a way to avoid software rebuild.

JAMP Platform uses RMI framework as a basic layer in its architecture, responsible for remote method invocations and object serialization in a local area network. Moreover, the platform has extra functionalities for distribution and it is available through the Broker Framework (JBroker). The JBroker is a program that contains a database of available servers at a moment. It enables clients to find distributed servers (remote objects) through the network. The use of server services occurs by remote method invocation (RMI) and after the successful search by its remote reference in the JBroker.

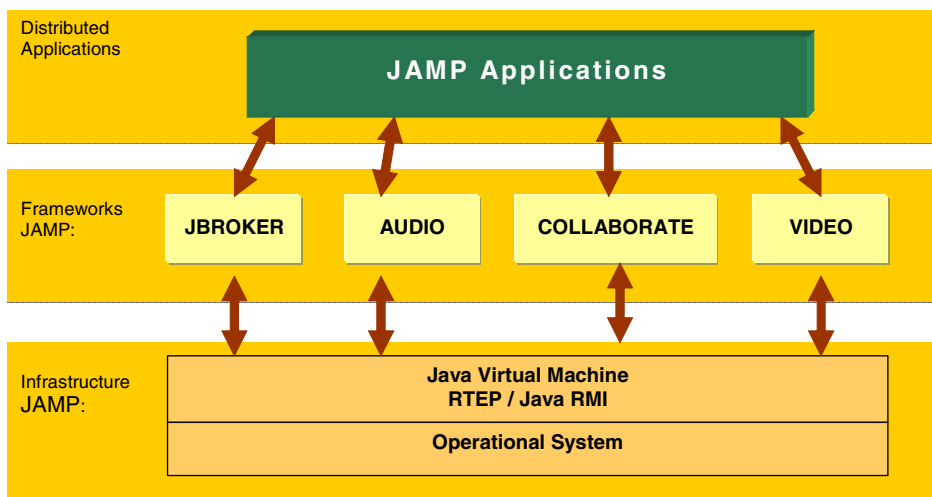


Fig. 1. JAMP Platform architecture

Middleware frameworks are basically used for integrating distributed components and applications. For instance, RMI (Remote Method Invocation) is capable of integrating distributed Java objects. It allows that an object invoke methods located in remote objects, which can take place in a computer with a completely different Java Virtual Machine, running on heterogeneous hardware architecture. It is also possible to send and receive objects through the network easily [17].

However, RMI performance over the internet is not good for a collaborative and immersive molecule visualization environment. When the application is distributed geographically, as the JViewer, the problem relies on transmission latency. So, it was developed a new communication protocol to supply the JViewer performance requirement. It was created the RTEP (Real Time Events Protocol) [18]. It is a protocol implemented at the application layer; it uses RTP (Real-time Transport Protocol)/RTCP (RTP Control Protocol) and has mechanisms to minimize the latency time issues, especially by managing the event queues and quality of connection of each server. These allow an efficient event distribution to the servers.

3 JViewer (JAMP Molecular Viewer)

The base of JViewer were the JBroker and the Collaborative frameworks with their respective services form the JAMP Platform, and the free Jmol [19] viewer. Fig. 2 shows the JViewer interface which displays molecular structures from files written in different formats: MOL,V3000, SDF, CTFile, CIF, mmCIF, CML, PDB, XYZ, XYZ+vib, XYZ-FAH, MOL2, Alchemy, CSF, GAMESS, Gaussian, Cube, Gchemical, MM1GP, HIN, Jaguar, MOLPRO, MOPAC, MGF, NWCHEM, odydata, xodydata, QOUT, SHELX, SMOL, spinput, GRO, PQR, Amber, JME, CASTEP, FHI-aims, VASP, DGrid, ADF, XSD, AGL, DFT, AMPAC, WebMO, Molden and PSI3. The formats are automatically recognized. The JViewer is extensible, so support to new formats can be added easily in the core.

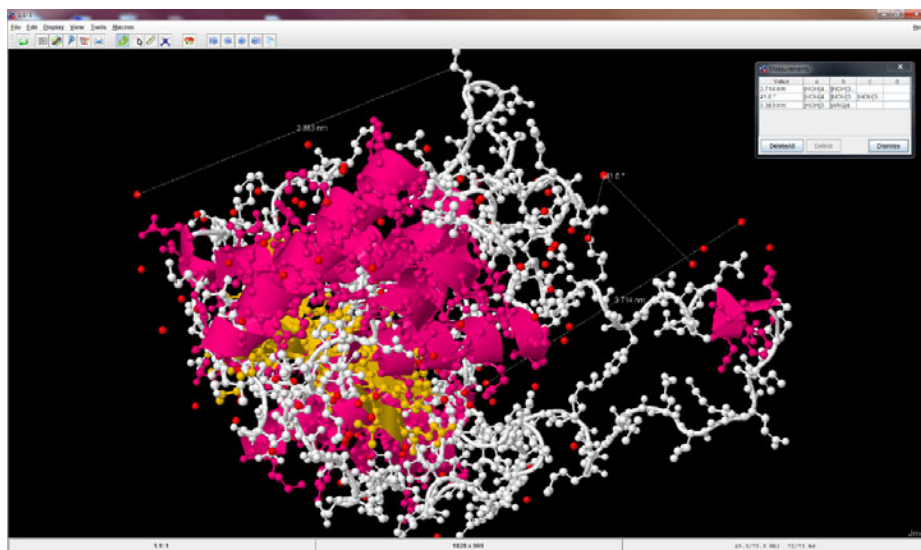


Fig. 2. System screen showing a protein molecule. Menu options allow changing the visualization properties.

As well as JAMP Platform the JViewer was written in Java, then inherited the multiplatform flexibility (it runs on Windows, Linux and MAC OS). Java provides a high-performance 3D rendering without specific hardware. The JViewer displays animations, vibration, orbitals and surfaces ; measure distances, angles and torsion angles; supports RasMol/Chime language scripts; and exports structures to various file formats (jpg, png, gif, ppm, pdf, POV-Ray, Gaussian, Maya, vrml, x3d e idtf).

The JViewer can be used in different ways by a teacher during a class. He can execute instruction on the fly or he can start the class from a specific simulation state. The system can read instructions that were done before. Some of the instructions are: rotation, translation, rendering type, colors, select molecule portions, animations, etc. Model position can be set and retrieved using these instructions. It is also possible make stereographic rendering for stereo displays and anaglyphic rendering.

JViewer recognizes all chemical elements of the periodic table. Each element receives a single color and its own radius. The atoms are rendered as cylinders and their colors can be chosen by the user. They can be identified with labels. The ability to simplify the molecule structure instead of showing all atoms and bonds is for great importance for the biological macromolecules study. JViewer possibilities to obtain this simplification viewing backbone, trace, ribbons, strands and cartoons. It accepts van der Waals radii percentages, as well as absolute values. Ionic radii are also supported, as an alternative to atomic.

Fig. 3 presents the JViewer hardware architecture. Each research group has a local server that controls the communication and synchronization of them. The local hardware infrastructure is flexible, it can be just a single personal computer or it can be a multiprojection system. When an event is generated by a client, it is send to the local server. When a local server receive an event, it replicate de event to the local and to its server, which propagates it to the others local servers.

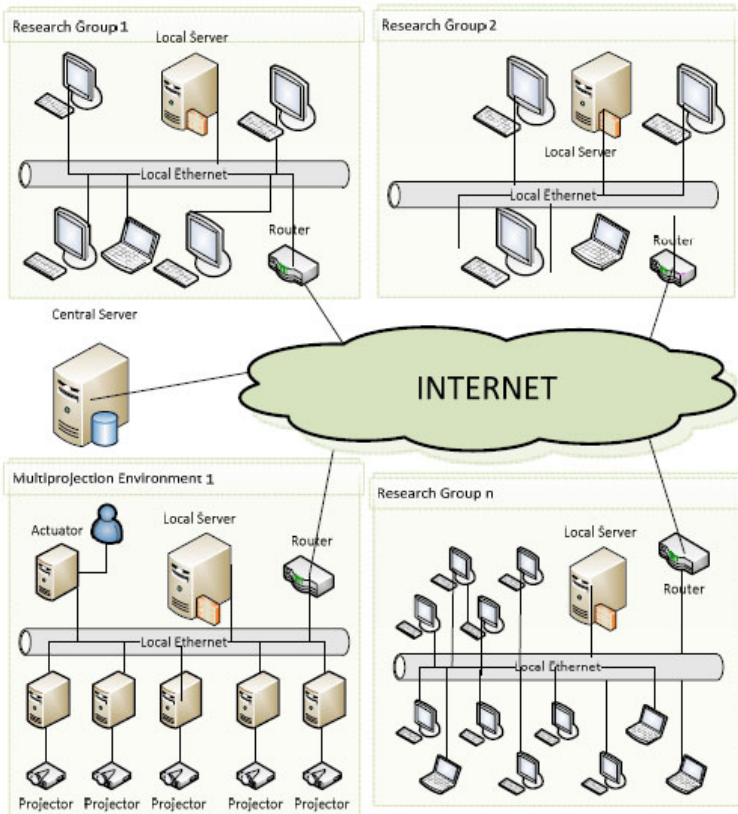


Fig. 3. JViewer hardware architecture

It was necessary to employ the framework JBroker and the Collaborative with their respective services in the JViewer development. After JViewer be started, it communicates with the JBroker to localize the Collaborative Service. After that, it communicates directly with the Collaborative Service which is implemented in the central server (Fig. 4). The Collaborative Service implements a centralized concurrency control, to manage all interaction with JViewer objects. JViewer implements WYSIWIS ("What You See Is What I See") paradigm where share the same visual perception of the work area. The disadvantages of the centralized concurrency control is that the central server is vulnerable to the failure (either computer or network), and it could become a network and processor bottleneck as all locks must pass through it. Local server at each site are responsible only for passing requests to the central server, and to sending any output sent to their clients from the central server. The advantage of a centralized scheme is that synchronization is easy, as state information is consistent since it is all located in one place. Events will never be received out of order (they are usually handled first-come, first-served).

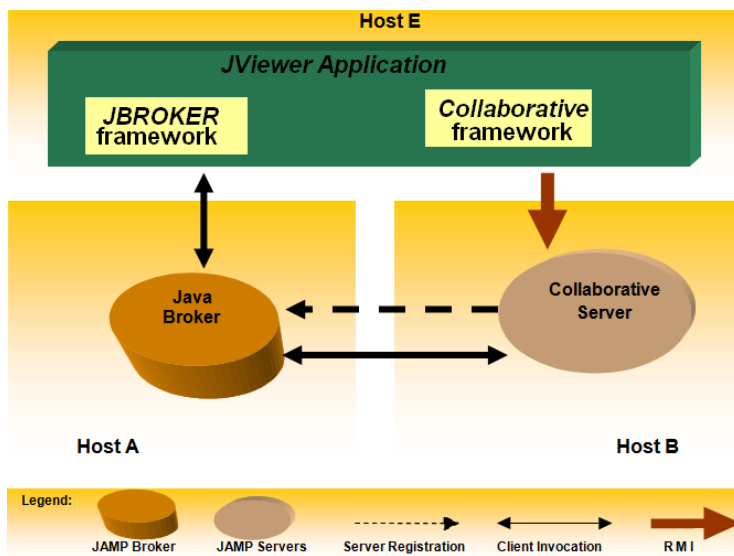


Fig. 4. JViewer software architecture

Collaborative work in JViewer resulted at least two benefits to the dispersed research groups:

1. A time-efficient method, where research team members must otherwise travel long distances to meet;
2. Meeting with expertise, where it may otherwise be logistically impractical or expensive.

Table 1 compares immersion and collaboration of some molecular viewers.

Table 1. Comparison of molecular viewers

Molecular Viewer	Immersion	Local Collaboration	Remote Collaboration
JViewer	Yes	Yes	Yes
VRMol [9]	Yes	Yes	Poor
IMSVE [5]	Yes	No	No
VRDD [2]	Yes	No	No
Kinimmerse [7]	Yes	No	No
Gesture Based Tool [10]	Yes	No	No
AMMP-Vis [6]	Yes	Yes	No
Colaborative Jmol [4]	No	Yes	Poor
Distributed Pauling World [3]	Yes	Poor	Poor
MICE [8]	No	Yes	Poor

The comparison presented at Table 1 shows that just the JViewer offers immersion, local collaboration and remote collaboration features .

4 Conclusions

The teaching of Chemistry has found allies in Virtual Reality and in Collaborative Work, which offers immersion, interactivity and collaboration. The feeling of immersion is created by the output devices and by immediate response of user interaction. JViewer supports monoscopic or stereoscopic output devices and can be extended to a multiprojection system, as a CAVE [20]. The collaboration between dispersed researcher groups in JViewer is like a single-user program.

As future work, we plan to port JViewer to run in a CAVE and add multimedia capabilities, including video and audio.

References

1. Levinthal, C.: Molecular Model-Building by Computer. *Sci. Am.*, 41–52 (1966)
2. Anderson, A., Weng, Z.: VRDD: Applying Virtual Reality Visualization to Protein Docking and Design. *Journal of Molecular Graphics and Modeling* 17, 180–186 (1999)
3. Su, S., et al.: Distributed Collaborative Virtual Environment: PaulingWorld. In: 10th International Conference on Artificial Reality and Telexistence, pp. 112–117 (2000)
4. Marchese, F.T., Mercado, J., Pan, Y.: Adapting single user visualization software for collaborative use. In: Seventh International Conference on Information Visualization (IV 2003), pp. 112–117 (2003)
5. Luo, Y., et al.: An Interactive Molecular Visualization System for Education in Immersive Multi-projection Virtual Environment. In: Third International Conference on Image and Graphics (ICIG 2004), pp. 485–488. IEEE Computer Society, Los Alamitos (2004)
6. Chastine, J.W., et al.: A Collaborative Multi-View Virtual Environment for Molecular Visualization and Modeling. In: Coordinated and Multiple Views in Exploratory Visualization (Cmv 2005), pp. 77–84. IEEE Computer Society, Washington (2005)
7. Block, J.N., et al.: KinImmerse: Macromolecular VR for NMR ensembles. *Source Code for Biology and Medicine* 4(3) (2009)
8. Bourne, P., et al.: A Prototype Molecular Interactive Collaborative Environment (MICE). In: *Symp. Bioconput.*, pp. 118–129 (1998)
9. Rodello, I.A.: VRMol - um ambiente virtual para visualização e análise de moléculas de proteínas. Universidade de São Paulo, São Carlos (2003)
10. Kim, J.-I., et al.: Development of a Gesture-Based Molecular Visualization Tool Based on Virtual Reality for Molecular Docking. *Bulletin of the Korean Chemical Society* 25(10), 1571–1574 (2004)
11. Cooperative work is seamless: Integrating the support of the many modalities of articulation work, <http://www.itu.dk/~schmidt/papers/seamless.pdf>
12. Trevelin, L.C., et al.: Development of Object Oriented Distributed Systems (DOODS) Using Frameworks of The JAMP Platform. In: ICSE 1999 - International Conference on Software Engineering - Workshop on Web Engineering, Los Angeles - Califórnia (1999)
13. de Guimarães, M.P.: Projeto e Implementação de Suporte para Trabalho Cooperativo na Plataforma JAMP. UFSCar, São Carlos (2000)
14. Trevelin, L.C., Vessoni, M.D.: Projeto e Implementação de um Ambiente de Escrita Colaborativa Baseado em Uma Plataforma de Distribuição. In: XVII Simpósio Brasileiro de Redes de Computadores (1999)

15. Trevelin, L.C., Mendonça, S.P.: Serviço Multicast na Plataforma JAMP para Suporte a Aplicações Multimídia Cooperativas. In: XXV CLEI 1999 - Conferência Latino-Americana de Informática (1999)
16. Trevelin, L.C., Guimarães, M.P.: Um Framework na Plataforma JAMP para suporte ao Trabalho Cooperativo. In: CLEI - Conferência Latino-Americana de Informática, Assunción (1999)
17. Ferreira, M.M., Trevelin, L.C.: Java Broker: concepts & java Language - porgrammer's guide. Release 1.0, UFSCar (1998)
18. Francischetti-Corrêa, M.: Arquitetura de Alto Desempenho Para Integração de Ambientes Interativos e Imersivos Remotos para Visualização Molecular. UFSCar, São Carlos (2010)
19. Jmol, <http://www.jmol.org>
20. Cruz-Neira, C., Sandin, D., DeFanti, T.A.: Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In: International Conference on Computer Graphics and Interactive Techniques, pp. 135–142 (1993)