Journal of Computational Science 10 (2015) 370–381

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Development of ontology-based multiplatform adaptive scientific visualization system

Konstantin Ryabinin*, Svetlana Chuprina

Perm State University, Bukireva Str. 15, 614990, Perm, Russia

ARTICLE INFO

Article history:
Received 3 October 2014
Received in revised form 27 February 2015
Accepted 13 March 2015
Available online 18 March 2015

Keywords:
Scientific visualization
Multiplatform portability
Ontology
Mobile devices
Adaptivity

ABSTRACT

In this paper, we propose methods and tools for multi-platform adaptive visualization systems' development that meets the specific visualization requirements of the computational experiments in the different fields of science. The proposed approach was implemented within the client-server rendering system SciVi (Scientific Visualizer) presented in this paper. This system provides multi-platform portability and automated integration with different solvers based on ontology engineering methods. SciVi is used in Perm State University to help scientists and researchers acquire the necessary multidisciplinary skills and to solve real scientific problems by means of adaptive visualization tools.

1. Introduction

The high variety and big volume of data is one of the main problems in scientific visualization. Nowadays, almost every scientific problem is related to measuring of the parameters of real objects or to modeling the processes that involve these objects. In both cases, there are hardware and software systems called solvers that manage the scientific experiment and produce output data in a special format. While a range of adequate software tools are used to analyze the solver results, the data do not have adequate tools of high-level visualization in a comprehensive and uniform way that, if necessary, may be easily integrated into data analysis system. Thus, the rendering of scientifically accurate 2D and 3D images and animations according to the data nature is often required for the meaningful analysis. However, the formats of output data as well as the interfaces of solvers differ significantly from one particular solver to the other due to the high variety of research objects in different domains. Often, this leads to the fact that it becomes impossible to reuse the visualizers across different solvers, even in the same field of science. The scientists have to develop a new visualizer for each new solver or to convert the data obtained from the new solvers to the existing visualizer requirements by special means or manually. Therefore, there is an actual need to develop the visualization systems that could be easily adapted to the different solvers to reduce the cost of program development and make conducting research easier, more efficient and more convenient.

Very often, the scientific solvers require a huge amount of computational power or special input devices like MRI scanners or genome sequencers. In these cases, there are complex problems to adapt visualizers to the features of supercomputers and special hardware systems in a uniform way. So, we suggest to use client-server technology to solve these problems. The visualizer as a client can run on the scientist's desktop computer or even on the mobile device obtaining all the necessary information from the solver via a local network or internet.

Many scientific problems require 3D visualization. Since the rendering is based on the discrete process of rasterization, the aliasing problem is inevitable. To improve the quality and realism of the image produced by 3D graphics rendering systems anti-aliasing is used. Although there are a lot of anti-aliasing algorithms, there are still challenges in this filed. The main ones are listed below:

1. Any kind of anti-aliasing reduces the performance. For example, the multisampling [7] anti-aliasing that is used as a standard built-in technique in many rendering systems and has hardware support on the most graphics cards reduces the performance in average in 3 times on mobile devices running under iOS and Android.
2. Very often the anti-aliasing algorithms produces artifacts on the image like unwanted blur of small details.

The goal of our work is to develop an adaptive multiplatform scientific visualization system based on client-server architecture to

* Corresponding author. Tel.: +7 9097281149.
E-mail addresses: kostya.ryabinin@gmail.com (K. Ryabinin), chuprinas@inbox.ru (S. Chuprina).

http://dx.doi.org/10.1016/j.jocs.2015.03.003
1877-7503/

automate the integration with different solvers adaptively, to distribute the rendering between client and server efficiently and to eliminate the aliasing problems. The client can run on the desktop computers and mobile devices, the server – on the desktop computers and high-performance computing systems. The highlights of our work include the following:

1. Adaptive integration with third-party solvers. To provide the integration of visualization system with any solver, we suggest the approach based on domain-specific ontologies about the features of third-party solvers. This approach is implemented in our scientific visualization system called SciVi.
2. Multiplatform portability. To solve the problem of multiplatform portability, we have developed the framework that allows building graphical user interface for different platforms based on the high-level declarative description automatically.
3. Efficient rendering. To improve the performance of rendering, we propose the special heuristics to adaptive distribution of rendering between client and server.
4. High quality rendering. To solve the aliasing problems we have implemented the adaptive algorithm of anti-aliasing that ensures the high quality of image rendered with animation's smoothness and no interactivity hitting.

We have successfully tested SciVi on the real interdisciplinary scientific tasks, such as visualization of:

- magnetic moments of nanoparticles (physics);
- prices on the stock market (economy);
- genetic sequences and phylogenetic trees (genomics);
- fluctuations of human body temperature (medicine).

2. Background

The analysis of the most popular scientific visualization systems and tools [4] has shown that there are only a few multiplatform solutions and they are not flexible enough to integrate with arbitrary third-party solvers. In our opinion the most versatile and powerful tools for scientific visualization are the VTK (Visualization Toolkit) and the systems based on VTK class library.

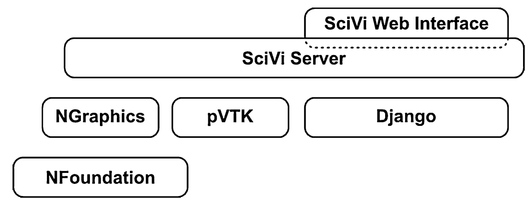

Fig. 1. Stack of SciVi server's frameworks.

VTK is an open-source, freely available family of libraries for 3D computer graphics, image processing and visualization VTK for desktop computers, pVTK for high-performance computing systems and VES for mobile devises.

VTK is designed to render complex 3D-objects using different techniques for rendering surfaces, volumes, slices, stereo images, etc. There are two most popular scientific visualization systems based on VTK ParaView for desktop computers and KiwiViewer for mobile devices. They allow rendering complex data but have no means for automatic integration with arbitrary solvers.

We decided to use the libraries from VTK family as a rendering subsystem for our visualization system. But we also have our own rendering subsystem that allows switching between different low-level graphical APIs. This is important when the rendering runs on MS Windows, because of the choice between OpenGL and DirectX. Support of DirectX ensures high performance on MS Windows and compatibility with MS WinRT.

A lot of popular interactive visualization systems use native operating systems' tools to build graphical user interface (GUI), but in the case of multiplatform visualization system portable solution is needed. As shown in [4], the most popular solution for building multiplatform GUI is Web application development. All the other

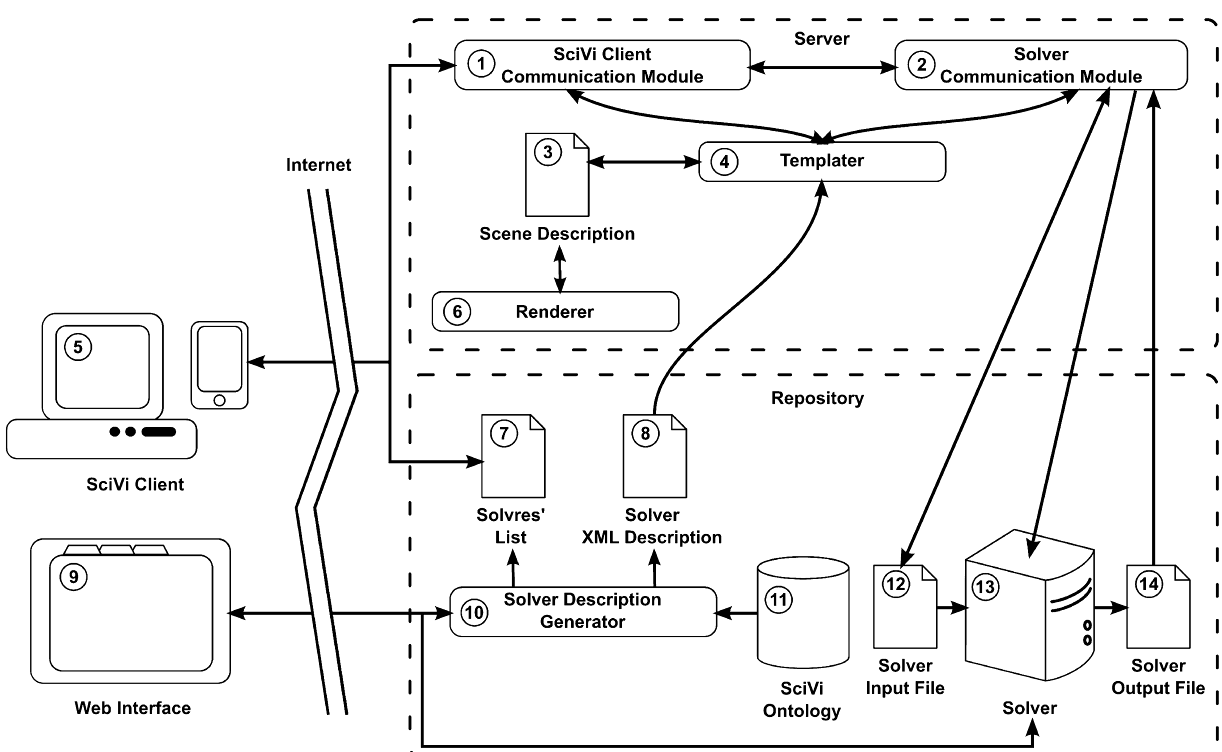

Fig. 2. Architecture of the SciVi server.