Architecture of a Software System for Scientific Data Visualisation

Mykola Kolodnytsky, Ph.D., Andriy Kovalchuk
Computer Software Department, Zhytomyr Institute of Engineering and Technology,
Chernyakhovskogo 103, Zhytomyr, 10005, Ukraine
kolod@ziet.zhitomir.ua, sekito@ziet.zhitomir.ua

Abstract

In this paper after an enumeration of software for scientific computation and data visualisation we give a set of demands for this sort of software. We discuss architecture of scientific data visualisation software system entitled "Graph Server" which was designed and developed by authors. The examples of 2D/3D Hodograph Curve, 2D/3D Histogram, 3D Surface, 2D Area are presented.

1. Introduction

At present there exists a set of software tools for scientific computation and data visualisation. These tools are utilised in both industry and scientific research. Among them one should point out the following *universal* software systems: MathLab, Matematica, PV-WAVE, Statistica, TK Solver, SPSS, DPGraph, etc. [1–7] as well as some *specialised* ones: Application Visualization System (AVS) [8], Khoros [9], IBM Data Explorer (DX) [10]. The specialized tools are very powerful, but they are autonomous as a rule and should be integrated with some other software system. The universal software, especially the software for mathematical modelling and simulation (like MathLab, Matematica) have built-in tools for data visualisation. However, it should be noted that, as a rule, to do visualisation of data with this universal software systems a user has to expend too much effort. For instance, to use it a user is forced to learn some domain specific languages of these software tools; the user interface of these tools is not user-friendly, intuitive, and standard enough. From our point of view, these features cause the inconveniences and pitfalls even for experienced users. On the other hand, the up-to-date computer information technologies are making the new demands on software tools as well as on its user interface. One should regard that the most urgent demand is to utilise such technologies as follows: MS Active Technologies, data exchange automation for applications, multithreading, up-to-date graphics libraries, etc.

Therefore, there still exists a need to design and develop the easy to use tool for the scientific data visualisation which could be easily integrated into a software system for simulation. This tool has to have a uniform and friendly user interface.

We have developed such a kind of tool. This tool is the part of the software system "DSR Open Lab 1.0" for numerical analysis of the broad range of mathematical problems. (Due to the limited space of publication we do not present here a description of "DSR Open Lab 1.0"; however, it can be found on the Web-site of our laboratory http://scilab.ziet.zhitomir.ua/mathsoft/dsr.php). Our software system was presented at the series of recent conferences [11–13]. Here, we are going to discuss an implementation of architecture of its subsystem – the interactive software for data visualisation. This subsystem is entitled "Graph Server".

2. Architecture of Software

The software "Graph Server" is implemented as an Active Document Server based on Document/View architecture. The data sets are transmitted to the Document using Automation. Then it redirected to the view-window which, in its turn, transfers it to the window related with OpenGL rendering context (OGL RC Window). This window, using the special system of messages, sends data into the appropriate thread which contains the procedure of data processing and visualisation. Each view-window can have several OGL RC Windows. It means that the several different data sets can be depicted in the same view-window simultaneously – see fig. 1. Each OGL RC Window is a container of an additional thread and it is responsible for the thread creation, deletion as well as for the interaction with the thread.

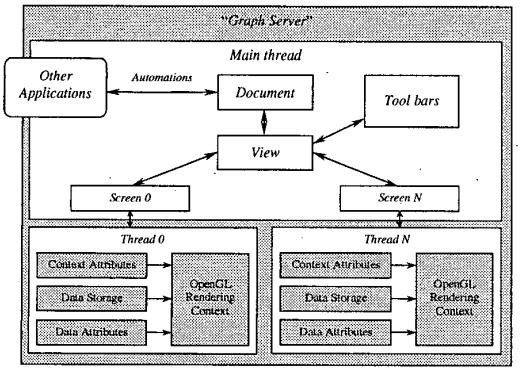

Fig. 1 General architecture of the software

Architectural features enumerated above provide the creation of an interactive (sic!) user interface which allows to do, simultaneously, such actions as follows: the scaling, scrolling, rotation, animation and so on. This architecture also gives the possibility to embed "Graph Server" into other software systems.

3. Implementation of Multithread Visualisation

We also implemented the multithread visualisation for "Graph Server" software. This multithreading means that the processes of calculation and visualisation of various data sets are running during the same time. This opportunity was achieved due to the usage of additional threads which are run simultaneously with the main thread.

In order to provide the interaction between the objects of main thread (*Document, View, Toolbars, OGL RC Window*) and the objects of additional thread (*Data, Attributes*) we implemented the system of inter-threads message processing. It has two different schemes: PostMessage and SendMessage (fig. 2).

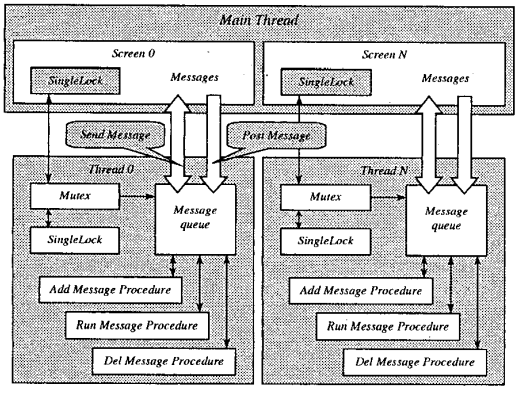

Fig. 2 Multithread architecture of the software

In accordance with PostMessage processing scheme, the messages are sent from the control objects (such as *Toolbars*) to the threads which make data visualisation. SendMessage processing scheme is used, as a rule, to get information about the state of visualisation process or about some peculiarities of data for visualisation.

4. Data Visualisation Procedure

All data are stored as a prime data, i.e. using that kind of data which was primary created by the client software. At present we implemented the storage of 1D/2D/3D areas of the following types: char, short, long, float, and double. The visualisation procedure casts it to the "double" type, calculates the scales and so on. After that, having chosen both the type of graph and dimension of data set the procedure runs the visualisation – see fig. 3.

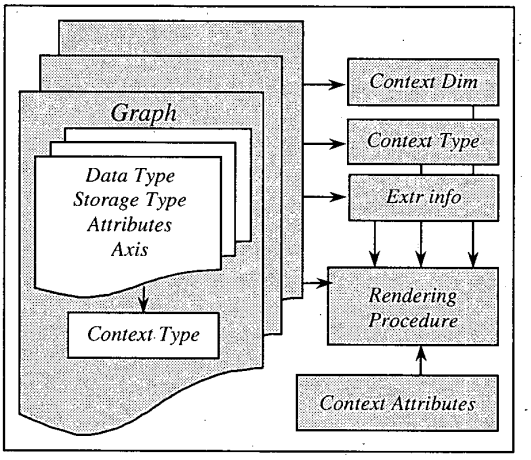

Fig. 3 Data storage and processing scheme

The data view presented on the screen will depend on both the type of data (which will determine automatically) and the type of context (which should describe by a user or using Automation). Thanks to this scheme of data storage and processing (fig. 3), the software system "Graph Server" effectively makes data visualisation for the following cases:

- 2D/3D Hodograph Curve,
- 2D/3D Histogram,
- 2D Area,
- 3D Surface.

5. Examples

Below we show examples of "Graph Server" user interface for data visualisation – fig. 4–7.

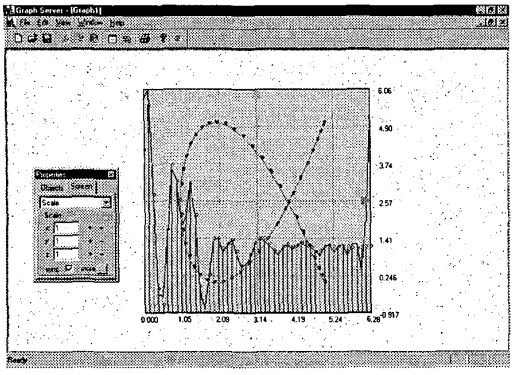

Fig. 4 Example of 2D Hodograph Curve

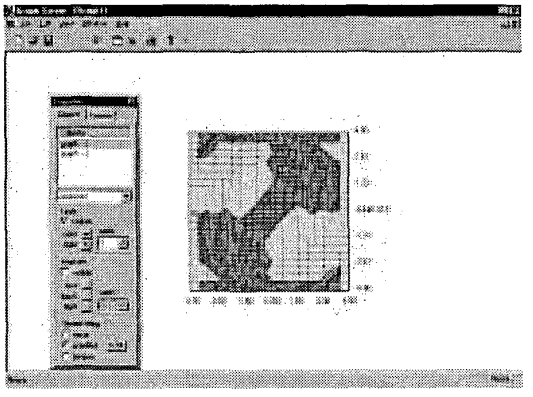

Fig. 5 Example of 2D Area

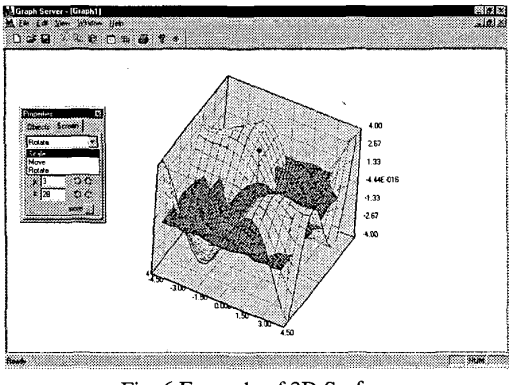

Fig. 6 Example of 3D Surface

6. Conclusion

We believe that the software "Graph Server" has the flexible, interactive, and context sensitive user interface due to the implementation of its architecture, namely: the multithread data visualisation and data storage and processing scheme. The methods of Automation provide to effectively use this software system as an automated server of data visualisation.

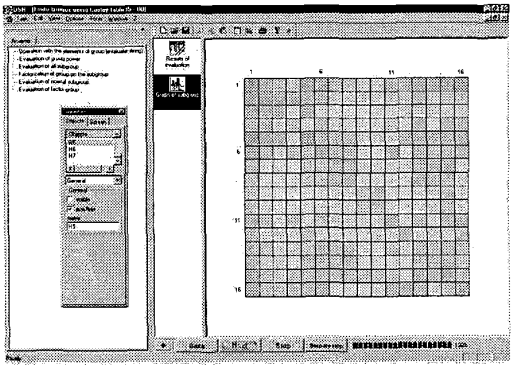

Fig. 7 Example of 2D Area

7. References

[1] MATLAB User's Manual, MATHWORKS Inc, MA, USA, 1999.

[2] Gaylord R.J., Kamin S.N. and Wellin P.R. Programming with Mathematica. 2nd Edition, Springer-Verlag, TELOS, Santa Clara, 1996.

[3] Stephen Wolfram. The Mathematica Book, Fourth Edition. Cambridge University Press, 1999, pages 1470.

[4] M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, Maple V Release 5. Learning Guide. Springer-Verlag, 1998, 284 p.

[5] Kolodnytsky M., Levitsky V. The review of the main mathematical software systems. // Computer-aided Design and Graphics. 1999. Vol. 10. p. 56 - 65.

[6] http://www.spss.com/

[7] http://www.statsoft.com/

[8] http://www.avs.com/

[9] http://www.khoral.com/

[10] http://www.research.ibm.com/dx/

[11] Kolodnytsky M., Ivanitsky I., Kovalchuk A., Kuryata S., Levitsky V. "DSR Open Lab 1.0" — software system for simulation // Proceedings of 21st International Conference on Information Technology Interfaces, Pula, Chroatia, 1999. – P. 31.

[12] Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V. "A Theory of Approximation of Functions" Courseware // Proceedings of the 5th Asian Technology Conference in Mathematics, December 17-21, Chiang Mai, Thailand, 2000. – – P. 86–95.

[13] Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V. The Mathematical Software Implementation for Computational Algebra and Number Theory // Proceedings of the 4th Asian Symposium on Computer Mathematics, December 17-21, Chiang Mai, Thailand, 2000. – P. 291–294.