14. Quan Bai, Minjie Zhang and Khin Than Win, "A Colored Petri Net Based Approach for Multi-agent Interactions", 2nd International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, December 13-15, pp. 152–157, 2004
15. R. Cost, "Modelling Agent Conversations with Coloured Petri Nets", Working Notes of the Workshop on Specifying and Implementing Conversation Policies, Seattle,Washington, pp. 59–66, 1999.
16. D. Poutakidis, L. Padgham, and M. Winikoff, "Debugging Multi-Agent Systems Using Design Artefacts: The Case of Interaction Protocols", Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi Agent Systems, Bologna, Italy, pp. 960–967, 2002.
17. J. Hutchison and M. Winikoff, "Flexibility and Robustness in Agent Interaction Protocols", Proceedings of the 1st International Workshop on Challenges in Open Agent Systems, Bologna, Italy, 2002.
18. S. Paurobally, J. Cunningham, "Achieving Common Interaction Protocols in Open Agent Environments", *AAMAS* '02, Melbourne, Australia, 2002.
19. M. Hack, "Analysis of production schemata by Petri nets", Project MAC-94, Cambridge, 1972.
20. A. Zakrevskij, V. Sklyarov, "The Specification and Design of Parallel Logical Control Devices", Proceedings of PDPTA'2000, June, Las Vegas, USA, pp. 1635–1641, 2000.

Authors' Information

Dmitry Cheremisinov, **Liudmila Cheremisinova** – The United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganov str., 6, Minsk, 220012, Belarus, Tel.: (10-375-17) 284-20-76, e-mail: cher@newman.bas-net.by, cld@newman.bas-net.by

WEBCOMPUTING SERVICE FRAMEWORK

Evgenija Popova

***Abstract**: Presented is webComputing – a general framework of mathematically oriented services including remote access to hardware and software resources for mathematical computations, and web interface to dynamic interactive computations and visualization in a diversity of contexts: mathematical research and engineering, computer-aided mathematical/technical education and distance learning. webComputing builds on the innovative web*Mathematica *technology connecting technical computing system* Mathematica *to a web server and providing tools for building dynamic and interactive web-interface to* Mathematica*-based functionality. Discussed are the conception and some of the major components of webComputing service: Scientific Visualization, Domain-Specific Computations, Interactive Education, and Authoring of Interactive Pages.*

***Keywords**: web-access, mathematical user-interfaces, web computations, mathematical active learning.*

***ACM Classification Keywords**: F.1.2 Modes of Computation: interactive computation, online computation; G.4 Mathematical Software: user interfaces; K.3.1 Computer Use in Education: distance learning.*

Introduction

Internet and the World-wide Web make many kind of information and services easily accessible. However many Internet/Web technologies, so powerful in many areas, are not well suited to scientific computation; it is simply not their main focus. The importance of technical/mathematical communication on the Internet is underscored by the activities at the W3 consortium. Internet Accessible Mathematical Computations is part of these activities directed towards making mathematical computation or information accessible on the Web/Internet [3], [11]. The following topics are subject of investigation:

- Remote access to mathematical software over the Internet.
- Encoding of mathematical expressions (including text-based encodings, for E-mail and HTML embedding, and binary-based encodings for efficient communication between scientific applications).
- Interoperability between software that create/transform/display mathematical expressions (e.g. symbolic, numeric, graphics, text-processing packages) via ad hoc communication protocols & software architectures.
- Web-based mathematics education.
- Access and interoperability to mathematical knowledge bases.
- Protocols, APIs, URL schemes, metadata, and other mechanisms for system interoperability and standardization.
- Application of IAMC for practical purposes such as scientific publishing and archiving, distributed problem solving, etc.

Accessing a scientific computing service should be as simple as entering a command, accessing a Web page, or sending email. A good summary of Web tools for interactive computation is given in [2].

We call a session of mathematical computations *accessible through a Web-interface* if it is triggered by few mouse-clicks while browsing and reading related documents. The interactions happening within the session are related tightly to the actions the user is performing. One can use the computations and their visible results to understand, conjecture, or verify mathematical facts. Within learning environments such connections can be used for *active learning*, that is called also “exploration by doing”.

webComputing is a general framework of mathematically oriented services. These include remote access to hardware and software resources for mathematical computations, and web interface to dynamic interactive computations and visualization in a diversity of contexts: mathematical research and engineering, computer-aided mathematical/technical education and distance learning. webComputing was initiated by a Swiss NSF project [10] and further developed within the frames of the Bulgarian National Science Program “Information Society” under the grant IO-03/2003 “Web-Based Computations and Visualization”. webComputing services are supplied by the COSE server (cose.math.bas.bg); COSE comes from COmputational Science and Engineering. Presently, COSE server is a PC with Athlon64 3000+ CPU having 2GB memory but we hope that in a near future COSE will become part of a computational grid.

In this paper the conception and some of the major components of webComputing service are presented. We begin by overviewing the goals of webComputing. Then the underlying technology for developing and deploying dynamic and interactive web accessible computations and visualization is described shortly. Two major components of webComputing service are discussed in some detail. These are: Scientific Visualization – a service for dynamic and interactive generation of functional graphics, and visualization of data; and webComputing Authoring Framework (wCAF) which allows easy and remotely development of web interface to particular mathematical applications.

Goals of webComputing

webComputing aims to supply a wide range of mathematically oriented computing services over the Internet. Such services include:

- *From the end-user perspective*: accessing remote mathematical computations including access to both general purpose computations and highly specialized computing services; easy and intuitive web access to specialized calculators, general purpose and domain specific computations and visualizations; interactive educational web applications (online training, courseware, exercises, quizzes, etc.) that make a good use of dynamic access to mathematical computing. In these interactive web pages users select different input parameters and submit data to build up a sequence of results. There is no need to buy install and maintain software; users always have the most recent version; training time is considerably reduced.

- *From the developers perspective*: supporting interactive use of a remote compute server (COSE server where there are installed some scientific computing libraries [4], [5], [6] and *Mathematica* [15] as a computer algebra environment.); providing assistance and promoting research and developments on interoperability between different mathematical systems, environments and data bases; providing resources for easy and remote authoring of dynamic and interactive math oriented pages; hosting application sites using web*Mathematica* technology [13], [14].
- *From a general application perspective*: webComputing is intended to serve the areas of
  - mathematical research and engineering
  - computer-aided mathematical/technical education and distance learning.

We aim to provide a maximum of flexibility for ourselves, and others, in the context of continued development of Internet-based mathematical services.

Underlying Technology

Many Internet accessible mathematical systems have been developed so far. For an overview and general characteristics of these systems see [8]. The following are essential parts of such systems:

- a mathematical system to perform computations
- a user-interface to display the results of computations and to receive the input of the user
- a connection mechanism between the two.

webComputing services are based on the computational and visualization power provided by the technical computing system *Mathematica* [15]. *Mathematica* is a suitable development environment for working on code that models some physical process – code that can then be placed into a dynamic interactive site to enable people to run the model and use its results for their regular work. The system allows building technical computing web services, including numerical, symbolic, and graphical applications that solve daily technical computing problems quickly and easily. In addition, *Mathematica* connects readily to external services, which may be provided by languages such as Java, C, Fortran, Perl.

The connection mechanism and technology empowering webComputing is built upon web*Mathematica*, an innovative new product of Wolfram Research that allows *Mathematica* to run on a server to provide the necessary calculations and graphs [13], [14]. web*Mathematica* is an innovative technology that solves the problem of how to create and distribute solutions to technical computing problems quickly in today's networked environment. It is a server-based technology built on top of two standard Java technologies: Java Servlet and JavaServer Pages (JSP) technologies. The minimum technical components for web*Mathematica* are:

- A Servlet container supporting both the Servlet Specification 2.2 (or higher) and JSP Specification 1.2;
- A Java Development Kit (JDK) 1.2 (or higher), Java 2 Version 1.4 (or higher) is recommended.

web*Mathematica* provides a collection of tools that allow *Mathematica* commands to be placed inside HTML pages. When a request is made for one of these pages, the *Mathematica* commands are evaluated and the computed result is inserted into the page. This can be done by two different HTML templating mechanisms.

Mathematica Server Pages (MSP scripts) are the original form of web*Mathematica* interaction [12]. HTML tags that contain *Mathematica* commands in the classic MSPs are called Mathlets and have the form:

<%Mathlet *Mathematica* commands or web*Mathematica* functions %>

MSP scripting technology is superseded in version 2 of web*Mathematica* [13] by the standard Java templating mechanism, JavaServer Pages (JSP) making use of custom tags. JSPs use a special library of tags called MSP Taglib that work with *Mathematica*. JSPs support the embedding of Java into HTML, and are frequently used along with Java Servlets to develop large dynamic web sites. A web*Mathematica* page uses standard HTML tags as well as special web*Mathematica* tags; these have the form <*msp:tag*>. The web*Mathematica* tags are executed from the top of the page to the bottom. The `<msp:allocate>` tag causes a *Mathematica* kernel to be allocated to use for computations. The contents of the `<msp:evaluate>` tags are sent to *Mathematica* for computation with the result inserted into the final page. The `</msp:allocate>` tag frees the *Mathematica* kernel to be used for another computation.

Below we present the different stages that are involved in that how web*Mathematica* processes a request [12], see Fig. 1.

1. Make Request. The browser sends an HTTP request to the web server. The request references a particular *Mathematica Server Pages* (MSP) or *Java Server Pages* (JSP) script, includes variables and their values.
2. Forward Request. The web server performs any pre-processing steps, such as authentication, and forwards the request to the web*Mathematica* server (MSP server).
3. Acquire Session. The MSP server acquires a *Mathematica* session for the request from a pool of preinitialized sessions. Any variables and values are sent to this session, which is then instructed to load the script.
4. Process Page. The *Mathematica* session loads the MSP/JSP script and processes any msp tags. *Mathematica* kernel is initialized with input parameters, it carries out calculations, builds then returns the result.
5. Receive Response. The MSP server accepts the response and adds all the necessary HTTP headers for return to the browser. It then clears any temporary settings in the *Mathematica* session and releases the session to the pool of available sessions.
6. Return Result. The web server performs any postprocessing steps and returns the response to the browser.
7. Release Session. The browser accepts an HTML response, which may use applets, plug-ins, or other features of dynamic HTML. Alternatively, the response could be some other format such as MathML, TeX, or a *Mathematica* notebook.

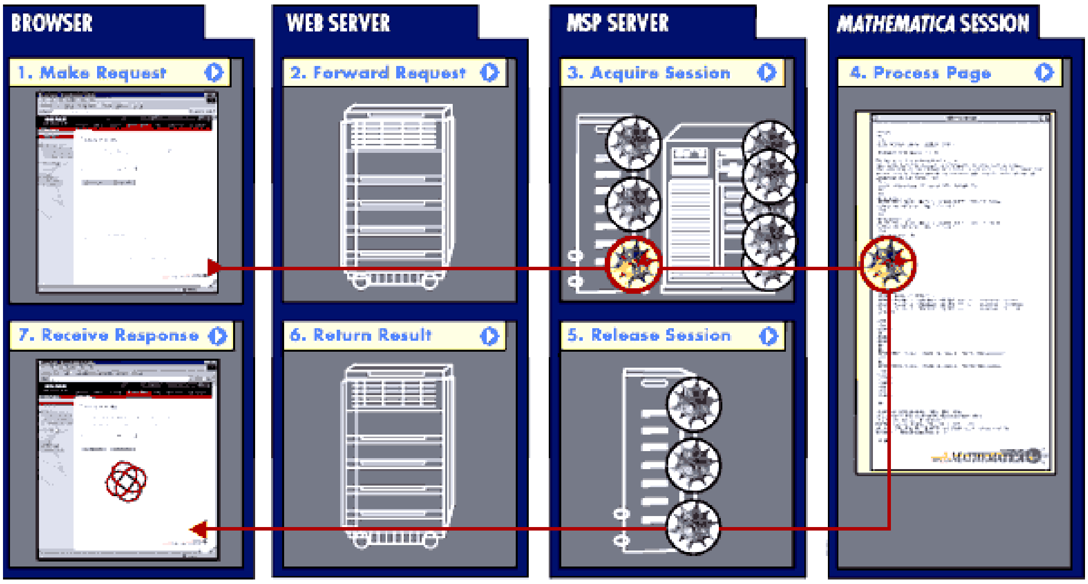

Fig. 1: How web*Mathematica* processes a request.

web*Mathematica* technology uses the request/response standard followed by web servers. Input can come from HTML forms, applets, javascript, and web-enabled applications. It is also possible to send data files to a web*Mathematica* server for processing. Output can be in many different formats such as HTML, images, *Mathematica* notebooks, MathML, SVG, XML, PostScript, and PDF.

An important part of web*Mathematica* is the kernel manager which calls *Mathematica* in a robust, efficient, and secure manner. The manager maintains a pool of one or more *Mathematica* kernels and, in this way, can process more than one request at a time.

webComputing Components

The focus of webComputing framework is on deploying dynamic computations and visualization in diverse contexts. Developed are sites in several domain-specific areas that demonstrate the dynamics in mathematical communication. In these sites users are taken through a sequence of web pages in which they select different input parameters and submit data to build up a sequence of results, Fig. 2. The main webComputing page (http://cose.math.bas.bg/webComputing) contains links to some pages including numerical, symbolic, and graphical applications that solve some technical computing problems quickly and easily.

Step By Step Differentiation page demonstrates how to use active learning strategies in computer-aided education and distance learning. Interactive exercises and activities change the role of the reader from passive to active working subject.

The research work directed towards developing techniques and algorithms for solving problems can now be deployed by live interactive web sites, vastly increasing the number of people who can use and learn from these results. **Interval Computations** web pages, presently accessible through the main webComputing page, present a typical web application delivering specialized calculations to technical professionals in the following forms: an interval web calculator, dynamic visualization pages, and interval problem solvers.

Teletraffic (http://cose.math.bas.bg/Teletraffic) is a web-based system that performs interactive computations and visualization in the field of teletraffic modeling. It is designed for computation and visualization of parameters of a model which examines the terminal teletraffic in communication systems characterized by (virtual) channels switching, without waiting, with finite number of homogeneous terminals, finite number of switching devices, generalized input flow and repeated calls.

Most of the developed web interfaces are multi-functional – allowing specialized calculations and visualization, and serving for computer-aided distance education involving active elements.

Scientific Visualization

Graphics is an indispensable component in any presentation, research or education material. Scientific Visualization is a web service for interactive graphics generation launched within the frames of webComputing [9]. Scientific Visualization contains a collection of interactive web pages for dynamic generation of 2D and 3D functional plots, visualization of data and some special objects.

Presently, this site provides the following functionality:

- *2D function plots*: two-dimensional graphics of functions of one variable; plots of parametric curves where x, y coordinates are functions of one parameter; plot of one or multiple functions filling the space between each successive pair of curves with a different color; plot of functions with logarithmically scaled x, y, or both coordinates.
- *3D function plots*: three-dimensional graphics of functions in two variables; one or multiple plots of parametric curves where x, y and z coordinates are functions of two parameters; 3D graphics of vector and gradient fields; 3D parametric plots with projections on the coordinate planes.
- *Contour and Density plots* of functions in two variables; plot of curves that are given implicitly as the solutions to equations.
- *2D data plots:* two-dimensional data visualization by points, error bars, labeled points, or in logarithmic scale of x, y, or both coordinates.
- *3D data plots*: visualization of three-dimensional data by density plots or contour lines.
- *Special plots*: 2D graphics of vector fields, bar charts, pie charts, and visualization of zonotopes.

In Scientific Visualization pages users can specify functions or submit data to generate a graphics image. A help-page displays the basic syntactic rules that should be followed in specifying the input. There are no restrictions on the quantity of the input data. The diagram presented on Fig. 2 shows a typical Scientific Visualization interactive page. Every page is equipped with a number of general and specific options that control the image: plot range, axes, axes origin, axes labels, ticks, frame, frame labels, plot label, image size, aspect ratio. The specific options include e.g., possibilities to control the number and color of the contours, or the size and color of the points in data plots. Each option has assigned a default value so that the user be facilitated as much as possible.

A 3D graphics can be displayed either static or by the LiveGraphics3D applet [7]. The latter gives a real-time rotation of the three-dimensional graphics object.

All the static graphics images can be exported to the client machine in a specified graphics file format. Scientific Visualization supports 12 of the most frequently used graphics file formats: GIF and animated GIF, JPEG, Encapsulated PostScript, *Mathematica* abbreviated PostScript, PDF, Microsoft bitmap format, Encapsulated PostScript with device independent preview, Encapsulated PostScript with TIFF preview, *Mathematica* independent raster graphics format, portable bitmap format, X window system bitmap format, TIFF.

Scientific Visualization is designed as a free and easy web service intended for general users, for distance education and active training, as well as for technical professionals. It will expand by more graphics functionality, static demo pages, and facilities for data uploading.

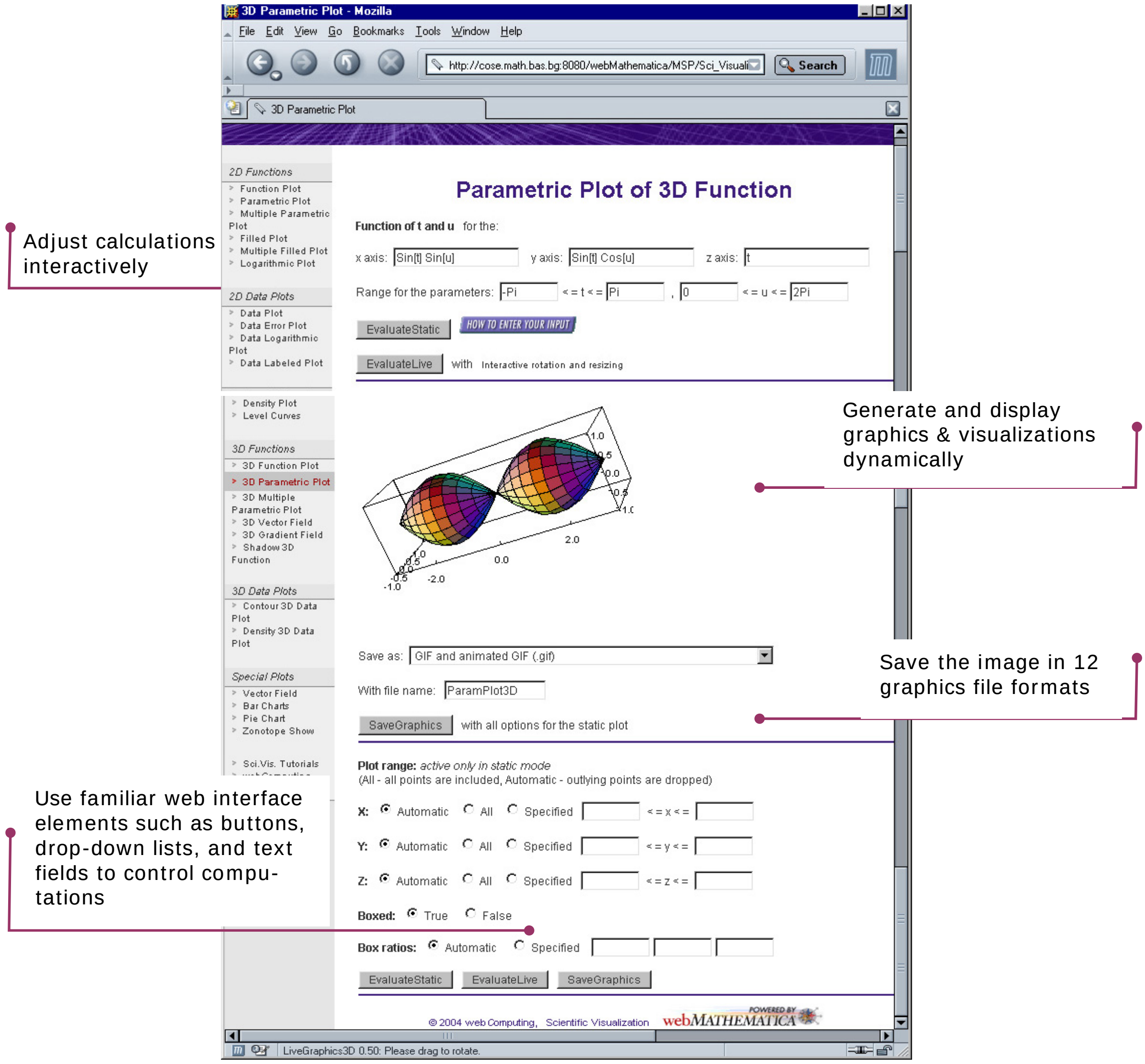

Fig. 2. A typical Scientific Visualization page.

webComputing Authoring Framework

webComputing provides hosting of sites for interactive computing, visualization and education that are built on web*Mathematica* technology. To this end, a webComputing Authoring Framework (wCAF) is designed. The goal of webComputing Authoring Framework is to allow *easy* and *remotely* development of dynamic and interactive web pages based on webMathematica technology. This service is provided for *Mathematica* users who intend to develop interactive courseware or other education materials based on the computational and visualization power of *Mathematica*, and for those who want to make their specific *Mathematica* software web accessible.

wCAF serves two groups of users: developers familiar with web*Mathematica* technology and beginners who need to explore and study the facilities provided by this technology. webComputing allows distance learning of webMathematica technology by a corresponding distance course providing:

- online (web accessible) static pages of web*Mathematica* documentation;
- live examples of interactive pages together with a display of their script code;
- access to script supporting directories of a web*Mathematica* server.

Thus, the content pages, the supporting background *Mathematica* packages, and other scripts can be developed independently but can interoperate from anywhere on the web.

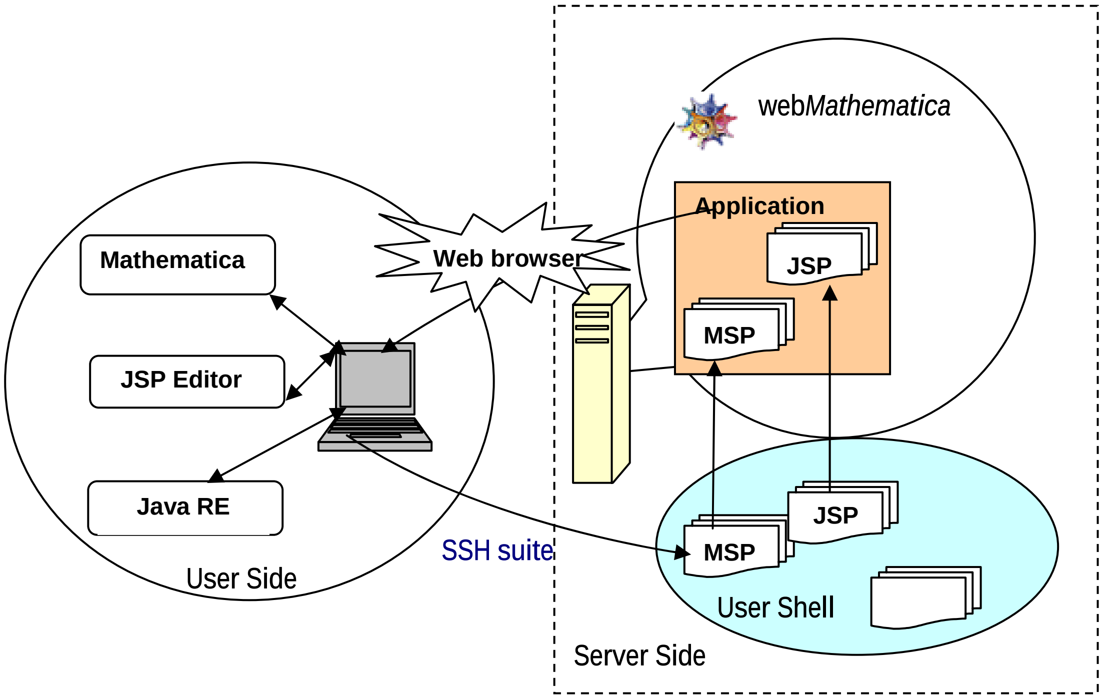

Fig.3. webComputing Authoring Framework concept and the overall architecture.

wCAF Architecture.Clearly, for wCAF to work, it must have front-end and back-end support. More importantly, such support must be delivered within an architectural framework that uses appropriate technologies to integrate components, allowing them to interoperate in a seamless manner on the Web.

- *Back-end support* – On the server-side wCAF architecture consists of three layers: user's home directories, webMathematica architecture connecting *Mathematica* to a web server, and other language, data base and technology support installed on the server. A user home directory has a structure corresponding to the tasks that will be performed and the server resources to be used. For developing interactive web pages e.g., two home subdirectories called MSP (containing web*Mathematica* ver. 1. script pages) and JSP (containing webMathematica ver. 2. script pages) are logically linked to corresponding domains of the webMathematica architecture. The latter are accessible by URL and can be processed by web*Mathematica* technology. Thus the Mathematica Server Pages and Java Server Pages, developed remotely by the user, can be uploaded on the server and invoked by any web browser to be processed by web*Mathematica*.

- *Front-end support* – The following common tools should be available on the client side: SSH protocol suite of network connectivity tools for secure remote login over the networks [1]. SSH file transfer should be used for downloading matherials from the server and for uploading the developed MSP/JSPs. Any available common Web browsers can be used to view and test the designed MSP/JSPs. Additionally, Java Runtime Environment will be necessary for running interactive 3D graphics. Depending on the Internet technologies that clients want to use and combine in their scripts, other front-end features can be provided through JavaScript, MathML, Java applets, etc. Any JSP editor, or some common type editors, can be used for developing MSP/JSPs. *Mathematica* system should be either installed on the client machine, or run remotely on the server.
- *Content-markup support* – it is provided by web*Mathematica* technology.

The remote MSP developing framework is presented on Fig. 3.

Usage Paradigm of wCAF.

- Users (both developers and students) have accounts, with appropriate structure, on the COSE server. Home directories are accessible via SSH/SSF protocol from anywhere on the Internet, at any time.
- web*Mathematica* documentation, live MSP/JSP examples and their script code are Web accessible from anywhere at any time. Users can study web*Mathematica* technology at their own. They can invoke and try the example scripts. Designing own interactive pages authors can easily include and mix powerful script features developed by others.
- Authors develop interactive pages remotely either on a client machine or on the server-side. They focus on the mathematical/technical content of the pages rather than dealing with the underlying Internet programming. Regardless of the size of the application that is created, wCAF cuts the development time and makes the application more robust as well as easier to use and maintain.
- Authors test/debug their own developed interactive pages by web access to a corresponding URL. In general, debugging anything running inside a server can be difficult. However, web*Mathematica* framework provides some ways for making this process easier. When authoring JSPs one can use messages and print statements to resolve a possible problem. web*Mathematica* provides specific functions that capture of *Mathematica* message and print output and return this output into the corresponding web page. In addition to message and print output, the developers of MSP/JSPs can use a more sophisticated tool called Kernel Monitor. It is a servlet that collects information on the running of a web*Mathematica* site. The kernel monitor is accessible through a web page which shows important status information for the *Mathematica* kernel which allows studying the performance of a site; any logs that have been collected and the input to *Mathematica*. The page provides also a number of controls that cause the kernels to restart, the monitor to reload, the logs to be cleared, reloaded, etc.
- The developed dynamic web pages – containing interactive computations, visualization or education materials – can be used from anywhere on the Web. Each author can provide well-defined set of capabilities within a particular scope.
- Authors deploy and maintain their interactive computations, visualization or courseware simply and easily. They do not have to worry about session management and error recovery. Authors can modify, revise, and change their webComputing pages anytime from anywhere.
- COSE server and web*Mathematica* are upgraded and improved without affecting wCAF user developed pages as long as equivalent contents are returned.

Conclusion

webComputing framework is an effort to deploy dynamic and interactive mathematically oriented services via web interface. The authoring framework, based on *Mathematica* and web*Mathematica*, allows the development of highly interactive mathematical web applications to be done easily and the applications to be more powerful. The overall framework is intended to be open, flexible, and usable as a test bed for research or a platform for application developments. Our prototype implementations represent a first step in this direction. As with any groundbreaking new technology, the capabilities open up an array of new possibilities: deploying calculators and problem solvers, showcasing *Mathematica*-based work in interactive web documents, publishing sophisticated courseware, papers, and book supplements that involve highly interactive web based education tools and materials. Work on an expanded interval web computation site is ongoing. We are also considering specialized education services that foster explorative learning. We hope to collaborate with more people interested in web accessible mathematics. The goal is to contribute to the creation of a rich interactive content and to make dynamic mathematical computations easily accessible in many contexts world-wide.

Bibliography

[1] D.J.Barrett, R.E.Silverman. The Secure Shell, The Definitive Guide. O'Reilly, 2001.

[2] L.Beccerra, O.Sirisaengtaksin, B.Waller. On Categories of Interactive Computational Web Tools. ATCM 2000 Proceedings, Thailand, 2000.

[3] IAMC: Internet Accessible Mathematical Computations, home page (1999-2004). http://icm.mcs.kent.edu/research/iamc/

[4] R.Hammer, M.Hocks, U.Kulisch, D.Ratz. C++ Toolbox for Verified Computing: Basic Numerical Problems. Springer-Verlag, Berlin/Heidelberg/New York, 1995.

[5] W.Hofschuster, W.Krämer. C-XSC 2.0 – A C++ Class Library for Extended Scientific Computing. In: Alt, R. et al. (eds.): Numerical Software with Result Verification. Lecture Notes in Computer Science, Vol. 2991, Springer-Verlag, Berlin/Heidelberg/New York, 2004, 15-35.

[6] R.Klatte, U.Kulisch, K.Lawo, M.Rauch, A.Wiethoff. C-XSC, A C++ Class Library for Extended Scientific Computing. Springer-Verlag, Berlin/Hiedelberg/New York, 1993.

[7] M.Kraus, M.: LiveGraphics3D: a Java applet which can display Mathematica graphics. http://wwwvis.informatik.uni-stuttgart.de/~kraus/LiveGraphics3D/

[8] P.Libbrecht. Mathematical Systems Accessed on the Web. Proceedings of IAMC'02 Workshop, 2002. http://www.symbolicnet.org/conferences/iamc02/

[9] E.Popova, M.Ivanova. Scientific Visualization: A Service for Interactive Graphics Generation. In: Mathematics and Education in Mathematics, Proceedings of 34 Spring Conference of UBM, Sofia, 2005. (in Bulgarian)

[10] SCOPES 7 IP 65642 project: Establishing Computational Science and Engineering in Bulgaria and Macedonia. Supported by Swiss NSF, 2002-2004. http://cose.math.bas.bg/

[11] P.S.Wang. Design and Protocol for Internet Accessible Mathematical Computation. Proceedings ISSAC'99, ACM Press, 1999, 291-298.

[12] T.Wickham-Jones. web*Mathematica*: A User Guide, Version 1.0. Wolfram Research, August 2001.

[13] T.Wickham-Jones. web*Mathematica*: A User Guide, Version 2.0, April 2003, Wolfram Research.

[14] Wolfram Res. Inc.: web*Mathematica*, the way the web calculates. http://www.wolfram.com/products/webmathematica/

[15] S.Wolfram. The *Mathematica* Book, 4th ed. Wolfram Media/Cambridge U. Press, 1999.

Authors' Information

Evgenija D. Popova – Institute of Mathematics and Informatics, Bulgarian Academy of Sciences; Acad. G. Bonchev str. Block 8, Sofia-1113, Bulgaria; e-mail: epopova@bio.bas.bg