

Maxima and Octave in Development of Online Applications: Service Based Approach

<http://dx.doi.org/10.3991/ijet.v9i5.3848>

K. Žáková

Slovak University of Technology, Bratislava, Slovakia

Abstract—The paper shows a way for fast development of interactive online applications that can be used in teaching process. It introduces the web service implemented in PHP programming language for two free computer algebra systems Maxima and Octave. It can be later used for building web applications that need to solve some mathematical tasks during their running. The example of such implementation is also shown.

Index Terms—Web services, open source software, simulation, XML, JSON, Maxima, Octave.

I. INTRODUCTION

There exist several web pages that are talking about the use of web applications in education. Larry Ferlazzo [1] tries to prepare a list with the best application examples. He does it already for several years. In order to include the application in the list, a site had to be accessible to English Language Learners and non-tech savvy users, free-of-charge, appropriate for classroom use and completely browser-based with no download required.

Growing number of web applications in fast every area of our life forces web developers to think how to prepare them in more effective way, how to reuse already developed components and how to create them more quickly. One way could be to use own or third party web services.

Different authors give different definitions of Web Services. Leidago Noabeb [7] introduces that “A Web service basically is a collection of open protocols that is used to exchange data between applications. The use of open protocols enables Web services to be platform independent. Software that are written in different programming languages and that run on different platforms can use Web services to exchange data over computer networks such as the Internet.” He also completes that since “different applications are written in different programming languages, they often cannot communicate with each other. A Web service enables this communication by using a combination of open protocols and standards, chiefly XML, SOAP and WSDL.” In spite of the fact that there already exist also other open technologies for building service based applications, Noabeb’s definition describes all main features of the web service.

Building web applications in Control Engineering area often requires accomplishing various computations. Many times there are calculations that are regularly repeated or that require similar procedures in solutions of various problems. Having web service (Fig.1) to an appropriate computer algebra system (CAS) can facilitate the whole design.

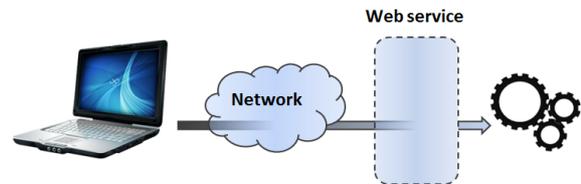


Figure 1. Communication via web service

We decided to prepare an API for online use of two open computer algebra systems – Maxima and Octave. The reason for this decision was to provide students (and/or interested web designers) with a simple tool that can be used as an engine for design of web applications illustrating selected topics from control theory. It is planned that applications will be developed by students in frame of their regular exercises accomplished in the subject “Design of Internet applications” or in web applications developed in frame of their bachelor thesis. The selection of CAS can be done according to preferences of each student. It is usually influenced by the knowledge of selected CAS.

Both considered CAS are described in Section 2. Section 3 presents one possible way for communication with web service. From pedagogical reasons (to be consistent with other study topics) we decided to use XML-RPC and JSON-RPC. Section 4 describes the created API that enables to write own web applications. The experience with the presented tool is introduced in Section 5.

II. COMPUTER ALGEBRA SYSTEMS

A. Maxima

Maxima [19] is free computer algebra system released under the terms of the GNU General Public License. It is suitable for the manipulation of symbolic and numerical expressions, including differentiation, integration, Laplace transforms, ordinary differential equations, systems of linear equations, polynomials, etc. Maxima yields high precision numeric results by using exact fractions, arbitrary precision integers, and variable precision floating point numbers. Maxima can plot functions and data in two and three dimensions.

The advantage is that the code and documentation is constantly updated. Since the community is quite large, one can also find various related projects and third party codes that offer additional functionality to Maxima computational kernel.

The Maxima source code can be compiled on many systems, including Windows, Linux, and MacOS X.

Originally Maxima environment was developed for the use on local computers. However, it can also be installed on the server where it can be used as an engine for other web applications. Maxima official web page introduces several web interfaces running Maxima. Most of them offer a possibility to calculate a selected type of examples that are mostly oriented to typical tasks from calculus. Our solution is based on the code of Bowo Prasetyo [9] who built a PHP program to access Maxima on the server interactively from a website.

B. Octave

GNU Octave [20] is a high-level interpreted language, primarily intended for numerical computations. It provides capabilities for the numerical solution of linear and non-linear problems, and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. The Octave language is quite similar to Matlab so that most programs are easily portable.

The Octave source code can be compiled primary on Linux systems. However, respecting some limitations it can also be used on Windows and OS X systems. The documentation is regularly updated since it is provided on the base of the Texinfo source files that are distributed along with every copy of the Octave source code. Similarly as Maxima, Octave is also provided mainly for local installation. However, there already exist installations that offer Octave engine via web interfaces.

III. REMOTE PROCEDURE CALL

Remote Procedure Call (RPC) is an interprocess communication technique that enables to request a service from a program located in another computer.

Its working principle is very well described e.g. in [3]: since RPC uses the client/server model, the requesting program is a client and the service-providing program is the server. First, the client sends a call message that includes the procedure parameters to the server process (Fig.2). Then, the caller process waits for a reply message. Next, a process on the server side, which is dormant until the arrival of the call message, extracts the procedure parameters, computes the results, and sends a reply message. The server waits for the next call message. Finally, a process on the caller receives the reply message, extracts the results of the procedure, and the caller resumes execution.

The server and client application logic are kept separate and multiple clients may be written using the same server API. According to the format of the sent messages it is possible to talk about XML-RPC or JSON-RPC.

A. XML-RPC

XML-RPC is a Remote Procedure Call method that uses XML passed via HTTP as a transport. With it, a client can call methods with parameters on a remote server (the server is named by a URI) and get back structured data. The body of the request is in XML. A procedure executes on the server and the value it returns is also formatted in XML. Procedure parameters can be scalars, numbers, strings, dates or also a complex record or list structures.

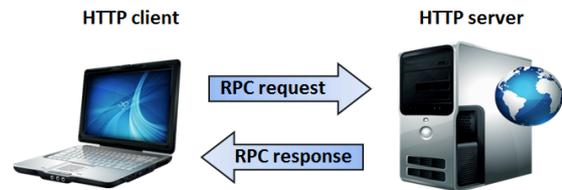


Figure 2. Remote Procedure Call communication

B. JSON-RPC

JSON-RPC is lightweight remote procedure call protocol similar to XML-RPC. Both request and response message is a single object serialized using JSON. JSON-RPC is usually more transparent than XML-RPC because of its compact form. It supports unicode by default and can be easily integrated into server or client scripting languages as php, python or javascript.

IV. REALIZATION

The similarity between XML-RPC and JSON-RPC motivated us to support both forms of the remote procedure call. The developed service allows to process requests in either XML-RPC or JSON-RPC, and the server provides responses according to the used RPC request protocol. Without loss of generality this paper describes only more compact JSON-RPC realization. XML-RPC communication is described more in [12].

A. Request and Response

For API implementation JSON-RPC 1.0 specification was used [21]. Then, the request object can contain three properties:

- *method* - a string containing the name of the method to be invoked.
- *params* - an array of objects to pass as arguments to the method.
- *id* - the request id that can be of any type. It is used to match the response with the request that it is replying to.

The created web service implements only one method – *eval* that evaluates the given code using the defined engine (i.e. selected computer algebra system – Maxima or Octave). The code and engine is specified among all other parameters where the web service user can also specify the format of results. Generally, results can be in two forms - either in the form of symbolical or numerical formula (defined in the parameter *formula_output*) or in the form of a graphical dependence (parameter *graph_output*). Using these parameters it is possible to specify several output formats. The parameter *formula_output* can have following values:

- *tex* – resulting expression is returned in LaTeX format,
- *image* – formula is returned as an image,
- *native* – formula is returned in native CAS format that can be used for future computations.

The parameter *graph_output* can be set to values:

- *png* – resulting figure is generated in raster graphics,
- *svg* – resulting figure is generated in vector graphics,
- *array* - returns graph as an array of values.

The following example illustrates the JSON-RPC request for solution of the set of two linear equations with regard to variables x and y . The request should be processed by Maxima computer algebra system using *linsolve* function. The resulting expression should be received both in LaTeX format and as an image, as well.

```
{
  "method": "eval",
  "params": {
    "code": "linsolve([3*x+4*y=7,2*x+a*y=13],[x, y]);",
    "engine": "maxima",
    "formula_output": [
      "tex",
      "image"
    ]
  },
  "id": 1
}
```

Figure 3. Example of JSON-RPC request

The response object also has three properties:

- *result* - the object that was returned by the invoked method. This must be null in case there was an error invoking the method.
- *error* - an error object if there was an error invoking the method. It must be null if there was no error.
- *id* - the same id as the corresponding request id.

The response to the request from previous example (Fig.3) is formatted according to the selected JSON-RPC specification (Fig.4).

```
{
  "id": 1,
  "result": {
    "result": "
(%i1) linsolve([3x + 4y = 7, 2x + ay = 13], [x, y])
          7 a - 52      25
(%o1) [x = -----, y = -----]
          3 a - 8      3 a - 8
",
    "graphs": null,
    "formulas": {
      "tex": [
        "$$\left[x=\frac{7a-52}{3a-8}, y=\frac{25}{3a-8}\right]$$"
      ],
      "image": [
        "http://.../092c75fceddbdc0781d5a0dc903e753d.png"
      ]
    }
  },
  "error": null
}
```

Figure 4. Example of JSON-RPC response

The result is available in the form of console output, LaTeX format and PNG picture that is placed in the temporary server folder. The picture (Fig.5) can be downloaded using the provided URL link.

$$\left[x = \frac{7a - 52}{3a - 8}, y = \frac{25}{3a - 8} \right]$$

Figure 5. Resulting expression in PNG format

B. Server Application

The whole server application is implemented in PHP programming language. It consists of several components that communicate together. The basic structure is illustrat-

ed in Fig.6. As it is evident from the structure the PHP server application has to perform several tasks:

- to receive, decode and process request from clients that can be either in XML or JSON format,
- to transform and send commands from client to the selected CAS system,
- to receive response with results from CAS system,
- to transform results to the required format,
- to generate response for client.

Graphs are generated directly in CAS. Both Maxima and Octave utilize open source program gnuplot for graph drawing therefore this program has to be installed on the server, too.

The server also requires an installation of LaTeX software together with freely available PHP library LaTeXRender that enables transformation of LaTeX based mathematical expressions to image format.

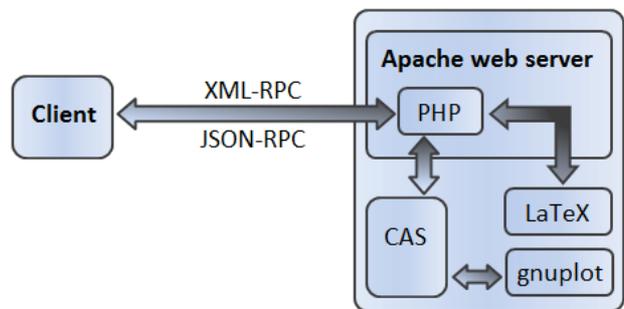


Figure 6. Structure of server application

V. SERVICE APPLICATION IN EDUCATION

Building the presented web service brings several advantages. Firstly, it can be completed by a graphical user interface and used in the education directly (see e.g. Fig.7). It helps students to understand some topics from Mathematics and Calculus also in subjects that are devoted to other topics.

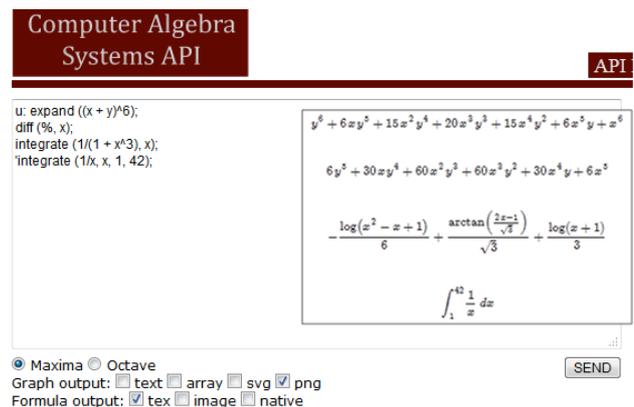


Figure 7. Graphical user interface for Maxima and Octave service

Then, it enables to prepare web applications that cooperate with the selected CAS system and can be used via institutional or personal web pages.

A. Modeling of Dynamical Systems

As it was already mentioned, the service was started to be used in the subject “Design of internet applications” that is devoted to the programming of client applications using mainly HTML5, CSS and JavaScript (no server side programming). As the final project students received the task to prepare animation of a given dynamical system. They could choose one of several available systems, but mostly the system was consisting of several elements representing a mass, a spring and a damper. Each system was described by a set of 1st order differential equations describing its dynamical behavior, e.g. in the form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{b}{m_1}(x_2 - x_4) - \frac{k_s}{m_1}(x_1 - x_3) - \frac{k_w}{m_1}x_1 + \frac{k_w}{m_1}r \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\frac{b}{m_2}(x_4 - x_2) - \frac{k_s}{m_2}(x_3 - x_1) \end{aligned}$$

where parameters m_1 and m_2 represent body mass, k_s and k_w – spring constants and b – constant of the damper.

This system [22] represents the automobile suspension of the quarter-car model. The resulting animation can look similar as the animation in Fig.8.



Figure 8. Example of a dynamical system animation

The animation behavior is influenced by the solution of differential equations that are depending on differential equation parameters, initial conditions and the force that affects the system. The moving position of objects in animation is described by variables x_1 and x_3 that denote actual position of the lower and upper mass. Variables x_2 and x_4 denote actual velocities of these bodies.

The animation was realized using HTML5 canvas tag whose content was influenced by JavaScript. Since in this phase of study students do not learn to use any server oriented scripting language, they just send values from the input form to one pre-prepared PHP script. It ensures processing of user’s defined data in regard to differential equations that are sent to server inside of hidden form input text fields. The animation is realized on the base of returned array-based results. After the animation is completed, students can verify its functionality by the change of system and simulation parameters and to try to understand the dynamical behavior of the system.

Till this moment the project was running just once. After its accomplishment we asked students to fulfill a short

questionnaire to have some feedback for the possible future realization of the project. The group consisted of 53 students. We learned that at the beginning almost one third of students were afraid of the project task since it was connected with the solution of differential equations. In spite of that most of students finished the project successfully. After they completed the code programming many of them tried to run the application several times. We asked them what the main reason of this multiple execution was. From the answers that are illustrated in Fig.9 is evident that at least 32% of students were trying to think about physical background of the problem. It can be considered as small benefit of the project since the subject is devoted to programming techniques and not to physics or mathematics.

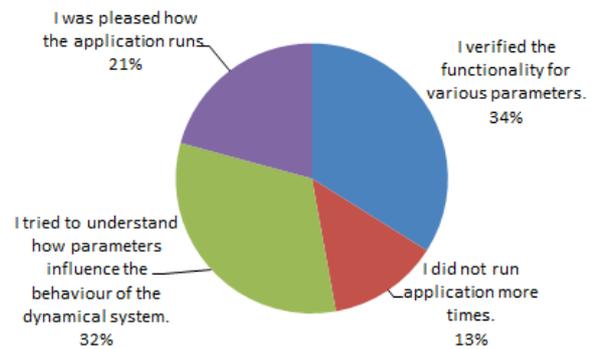


Figure 9. Distribution of answers to the question in questionnaire

In addition, in another question 28% of students introduced that solving project tasks they realized that differential equations can have practical use and another 23% of students have stated that the creative design of animation that behaves in similar manner as a real system was interesting for them. There are some opinions from students they entered to the questionnaire:

- After understanding the problem the task was interesting and I am glad that we had such a project.
- The task would be more interesting without a connection with physics since in this way it was more difficult.
- The task was quite difficult but I learned a lot of things.
- It was a good idea to have such a project.
- The task was interesting and it was not so difficult.
- The task enabled to show technical knowledge in the real use. It was good.
- I like the result.
- It was demanding task but I am content with the result and I am content that I was able to finish it successfully.
- It was a complex task where I could use everything what I learned in the subject. It showed me what I can expect in my future career.

It is to add that the subject is taught in the second year of study. In spite of the fact that the course is attended after the course of Mathematics and Calculus, students are not very familiar with the practical use of differential equations since they have not enough experiences from solution of real and applied problems and tasks. They are

simply missing practical skills. Therefore the connection of the practical preparation of web animation with the problem that was till this moment solved mostly only theoretically is really useful and effective.

B. Control Theory Applications

The developed animations can be used in the subject Control Theory since modeling of dynamical system behavior belongs to basic topics that are taught in this subject. However, in similar manner it is also possible to prepare other applications that can demonstrate various problems from the subject. In Fig.10 one can see the Nyquist plot drawn for the introduced linear system. It can help to determine system stability. Another application can illustrate e.g. Bode plots, step or impulse response, spate space representation, etc.

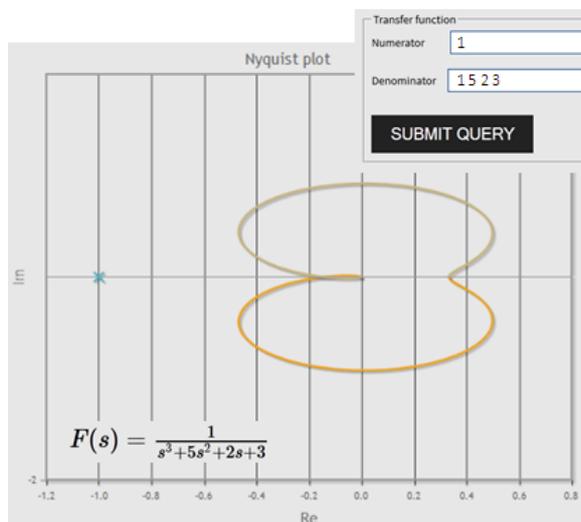


Figure 10. Nyquist plot of given system

VI. CONCLUSION

The achieved experience with students demonstrated that the connection of topics from various subjects (Mathematics, Physics, Design of Internet applications) can be interesting for students. Therefore we are planning to use the developed platform again.

The procedure of building the service based animation that is used by our students can also be used in design and preparation of more sophisticated web based simulations. In this way one can simulate not only dynamical behavior of a selected single linear or nonlinear system but also the whole control structure including control algorithm.

The advantage is that since the provided web service enables to use two various computer algebra systems (Maxima and Octave) the web developer can use that one which is more appropriate for given problem.

ACKNOWLEDGMENT

Author thanks to Peter Slivka for his help with the problem algorithmization and programming.

REFERENCES

[1] L. Ferlazzo, "The Best Web 2.0 Applications For Education In 2013 – So Far", [http://larryferlazzo.edublogs.org/2013/06/16/the-](http://larryferlazzo.edublogs.org/2013/06/16/the-best-web-2-0-applications-for-education-in-2013-so-far/)

- [best-web-2-0-applications-for-education-in-2013-so-far/](http://larryferlazzo.edublogs.org/2013/06/16/the-best-web-2-0-applications-for-education-in-2013-so-far/) [Accessed: April 15, 2014].
- [2] Huba, M., Šimunek, M. "Modular Approach to Teaching PID Control" *IEEE Transactions on Industrial Electronics*, ISSN 0278-0046, Vol. 54, No. 6, pp. 3112-3120, 2007.
- [3] Javvin Technologies, Inc., "Network Protocols Handbook", Javvin Press, 2007.
- [4] E. Králiková, M. Kamenský, J. Červeňová, "E-Learning Support in Engineering Education of Robotics Specialists", *Proc. of Int. Conf. Distance Learning, Simulation and Communication "DLSC"*, Brno, Czech Republic, IDET 2013, pp. 121-127, 2013.
- [5] C. P. Leão, A. E. Rodrigues: Transient and steady-state models for simulated moving bed processes: numerical solutions. *Computers & Chemical Engineering* 28(9): 1725-1741, 2004. <http://dx.doi.org/10.1016/j.compchemeng.2004.01.007>
- [6] P. Lutus, Symbolic Mathematics Using Maxima, <http://arachnoid.com/maxima/>
- [7] L. Noabebe, "Anatomy of a Web Service: XML, SOAP and WSDL for Platform-independent Data Exchange", http://www.webreference.com/authoring/web_service/index.html [Accessed: May 09, 2013].
- [8] K. Ogata, *Modern Control Engineering*, 3rd Edition, Prentice Hall London, 1997.
- [9] B. Prasetyo, MaximaPHP, <http://maxima.sourceforge.net/relatedprojects.html>
- [10] M. T. Restivo, J. Mendes, A.M. Lopes, C.M. Silva, F. Chouzal, A Remote Lab in Engineering Measurement, *IEEE Trans. on Industrial Electronics*, vol. 56, no.12, pp. 4436-4843, 2009. <http://dx.doi.org/10.1109/TIE.2008.2011479>
- [11] F. Schauer, M. Ožvoldová, F. Lustig, „Real Remote Physics Experiments across Internet – Inherent Part of Integrated E-Learning”, *Int. J. of Online Engineering (iJOE)*, 4, No 2, 2008.
- [12] P. Slivka, "Design of API for CAS service", Diploma thesis, FEI STU Bratislava, in Slovak, 2012.
- [13] E. Stuchlíková, J. Benková, M. Donoval, "An Easy and Effective Creation of E-Learning Courses", *Proc. of Int. Conf. Distance Learning, Simulation and Communication "DLSC"*, Brno, Czech Republic, IDET 2013, pp. 159 – 164, 2013.
- [14] A. J. Turgeon, "Implications of Web-Based Technology for Engaging Students in a Learning Society", *Journal of Public Service and Outreach* 2(2): 32-37.
- [15] E. L. Woollett, Maxima by example, <http://www.csulb.edu/~woollett/>
- [16] K. Žáková, "Maxima-An Open Alternative for Engineering Education", *Int. Journal of Engineering Pedagogy*, Vol. 1, No. 1, pp. 27-30, 2011.
- [17] I. Zolotová, R. Mihaľ, R. Hošák, "Objects for Visualization of Process Data in Supervisory Control", *Topics in Intelligent Engineering and Informatics 2: Aspects of Computational Intelligence: Theory and Applications*. - Berlin Heidelberg: Springer-Verlag, pp. 51-61, 2013.
- [18] jsMath: A Method of Including Mathematics in Web Pages, <http://www.math.union.edu/~dpvc/jsMath> [Accessed: 9 Mar 2009]
- [19] <http://maxima.sourceforge.net/>
- [20] <http://www.gnu.org/software/octave/>
- [21] <http://json-rpc.org/>
- [22] <http://control.ee.ethz.ch/~ifa-fp/wiki/index.php?n=Main.Modelling>

AUTHOR

K. Žáková is with Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia (e-mail: katarina.zakova@stuba.sk).

The work has been supported by the Grant KEGA No. 032STU-4/2013 and the Grant APVV-0343-12. This support is very gratefully acknowledged. Submitted 30 April 2014. Published as resubmitted by the author 14 August 2014.