View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Johannesburg Institutional Repository

African Journal of Business Management Vol. 5(31), pp. 12341-12355, 7 December, 2011
Available online at http://www.academicjournals.org/AJBM
DOI: 10.5897/AJBM11.1612
ISSN 1993-8233 ©2011 Academic Journals

Full Length Research Paper

Multiple usages of the distributed systems with numerical calculation

Eliza Consuela Isbasoiu[1] and Wilfred I. Ukpere[2]*

[1]Informatics Department, Faculty of Accounting and Finance, Spiru Haret University, 115100 Campulung Muscel, Romania.
[2]Department of Industrial Psychology and People Management, Faculty of Management, University of Johannesburg, Johannesburg, South Africa.

Accepted 15 August, 2011

The informatics society is a society in which the information, through its production and consumption, is the most important type of activity. The informatics society allows access to information and new ways of work and knowledge (Toma, 2009: 5). If within the scientific research, the interdisciplinary objective reflects the unity of the natural science, it is necessary to form a new didactic way of thinking, from autonomous sciences to border or interdisciplinary ones (Dagger et al., 2005: 6). The mathematical methods and language takes over from studies of the research problem (Koper and Tattersall, 2005: 67). The knowledge of the technology and the criteria a developer follows, assures a high level of the application. The web service is projected to support the interoperational interaction of machine to machine over netting; as such, other software systems can discover its definition and can interact with the web service in the way described by the definition.

Key words: Numerical calculation, graphic generator, web service, research.

INTRODUCTION

Within this article, we intend to demonstrate the utility of the distributed systems in the design of the studies and research software. The distributed systems are generally rich in structured knowledge, not so complex and do not imply behavioral relations among users (Isbasoiu, 2009: 4). However, all the existing architectures start their construction from the experts in the learning domain experience, but not from the user's perception and also from the studying material and the way the person constructs the mental concepts' schemes (Hansen, 2007: 234). We have to underline that there is an intrinsic connection between the results obtained by the users and the way of teaching. Owing to the restriction imposed by time and resources, the great part of the implementations is reduced to simple models for knowledge.

*Corresponding author. E-mail: wilfredukpere@gmail.com.

The web service is projected to sustain the interaction between machine to machine over netting: other systems can discover its definition and can interact with the web service in a way prescribed by the definition (Iyengar et al., 2008). There are numerous basic benefits of the web services such as interoperability in a heterogeneous environment, business services over web, integration with the existing systems, freedom in implementation, put up with different types of clients, with productivity and programming. The application is oriented towards different types of users such as managers, economic analysts, specialists in informatics, specialists in mathematics, specialists in physics and simulations.

It can be used as a working tool in studies of analyses based on numerical values, emanating from questionnaires, case studies, etc. It is recommended against other applications due to its possibility to be accessed online by many users simultaneously (Koren and Krishna, 2007). These users can as well receive the results in the real time. The application contains a multitude of

functions from which the users choose only the necessary ones.

MATERIALS AND METHODS

Although at present there are the hardware communication infrastructures necessary to exploiting the research at a large scale on the basis of the long distance soft, the usage of the new technologies is still at a low level (Press et al., 2007).

The efficiency of the accessible research platforms via web technologies can be proved, taking into account the following factors: instructors: the capacity of implying users, the degree of knowing the application; the computers network: hardware performances, transfer speed, the functioning state; the used platform: a facile usage both for instructors and for users, reducing the errors at a minimum number; the content in the digital format: the scientific accuracy, the attractiveness, the interactivity, attributed which deter-mine that the users remain connected to scan the contents The basic motivations for choosing the web services as a technology for implementing the distributed systems, which stand for the research platform are: the diversification of the offer: one can combine in simple or complex scenarios, services offered by different suppliers (from simulators which imply high performing calculation to simple graphics); re usage: they can embed existing applications, making the communications between them possible; adapting: different clients can be constructed: the teachers can customize the inter-faces for adapting to the students' or any other user's level (Nakhiovsky and Myers, 2004; Vasiliev, 2008).

The passing form an application based on local components to an application based on remote components implies a surplus due to the communications. This is only one of the disadvantages of passing from a centralized to a distributed system. We consider that as being the most important impediment of transforming in the realized case study, as opposed to other disadvantages, such as the network or components drawbacks or the system security.

So, the present study refers to the surplus introduced by the usage of the web services for presenting the components of a numerical calculation system. Another direction is the identification of the ways of presenting variant facilities introduced by the system, as simple as possible (Hurwitz et al., 2007; Prodan and Fahringer, 2007). This treatise as mentioned earlier intends to analyze the benefits of the new technologies for software in research; but the domain of the computational fields is based on intensive numerical calculation. It is necessary to consider as case studies the already existing numerical codes (Nagappan et al., 2003; Xiao, 2007).

There are a lot of codes for numerical calculation from where we can start the research. We mention the collection Netlib6 with over 0.6 billion visitors from its initiation or the collection NAG7. The great parts of these codes use C/C++, due to the superior speed in calculation with the current architectures.

The web services are naturally described by newer languages than C/C++. It is preferable, from the compatibility reasons, that the analysis be realized with Java native codes. So, for this study, we used the codes presented in the treatise, Numerical Library in Java for Scientists and Engineers (Hang, 2004: 45). The basic codes are not implemented, only their interface is changed. The passing from an interface with the user in a command line to an interface based on web services represents a great advantage, because it allows the combining and the re-usage of some procedures (routines) previously defined (Petcu, 2006: 75). The services vary from simple to complex and there are a great number of methods which are appealed by methods from other classes. Beneath the surplus study, we take into consideration the level to which a specific method or collection of methods is efficient to be exposed as a web service (McGovern et al., 2003).

Theoretical calculation

A starting point as concerned the construction of the services and the including of one or more methods in implementation is constituted by the analysis of the data structures from the elementary to the complex ones (Isbasoiu, 2009: 2). The great part of the literatures in numerical analysis use a specific subject "standard", such as linear equations, interpolation, extrapolation, ordinary differential equations, etc. Other discussed points are: the functions evaluation and especially mathematics functions, aleatory numbers, the Monte Carlo method, sorting, optimization, multidimensional methods, the Fourier transformed function, the spectral methods and other methods as concern the statistic descriptions and data modeling and relaxing methods.

In the treatise (Hang, 2004: 342), there are seven groups of thematic we focused on, namely vectors and matrices; algebraic evaluations; linear algebra; analytical evaluations; analytical problems; special functions; proximity and interpolation. Each group has the name 'Basic' followed by the appropriate number of every theme. A schematic representation of these groupings is presented in Figure 1.

In a first iteration, we consider that, due to the approached techniques, the aforementioned groups can each be viewed with a specific web service. At a first estimation, the surplus introduced by the services usage would be reduced only to that introduced by the interface of the 7 services; however, there is interdependence among the groups, this fact means an additional surplus when accessing a surplus by another one, which implies a more profound analysis of the groups versus services (Holdener, 2008).

As we have already said, the passing to an interface based on web services has the advantage of combining and re-usage of some previous defined procedures. For examples, the procedures analyzed in this study can be replaced by more performing implementations if they respect the same interface. There is a series of dependencies between classes and methods in our case study. We are going to realize a detailed study on these dependencies in order to see the dependence level. A weak coupling is for introducing the web services.

The aforementioned group contains an impressive number of functions. Due to an attentive analysis, we noticed the following: Basic1 stands for the basis of constructing the other basics; some functions of Basic1 are appealed even when the functions of the basic are constructed; every basic interacts with the others; there are also functions which do not interact or do not represent the basic for another one. These observations are clearly presented further; the graphic representation in Figure 2 showing the dependence of the functions in Basic 1.

The most functions do not depend on others; they are to be found in Basic1. Due to the fact they are in an impressive number, we intend to present only those found also in Basic2, Basic3, Basic5 and Basic7. So, in Basic2, we regain the function *Newgrn*; in Basic3, there are all the functions in Basic1, except those in Basic2, Basic5 and Basic7; in Basic5, there are the following functions: *Linemin, Rnk1upd, Praxis, Rnk1min, Flemin, Marquardt, Gssnewton, Multistep, Ark, Efrk, Efsirk, Liniger1vs, Liniger 2, Gms, Impex, Modifiedtaylor, Eft, Arkmat, Nonlinfemlagskew, Peide*; in Basic7, we find the functions *Sndremez, Minmaxpol* (Hang, 2004: 354). The graphic representation in Figure 3 shows the Basic2 functions dependence.

In Basic3, the function *Reaeig1* is found, in Basic4, there are the functions: *Reccof, Gsswts, Gsswtssym*, in Basic6 there are the functions: *Ei, Sincosint, Sincosfg, Inverseerrorfunction* and in Basic 7, the function: *Minmaxpol* (Hang, 2004: 360) appears. The graphic representation in Figure 4 presents the dependence of the functions in Basic3.

Basic4 presents the function *Gsslagwghts*, in Basic5, there are the functions: *Rnk1min, Praxis, Marquardt, Gssnewton, Multistep, Ark, Efrk, Efsirk, Liniger1vs, Liniger 2, Gms, Impex, Femhermsym,*

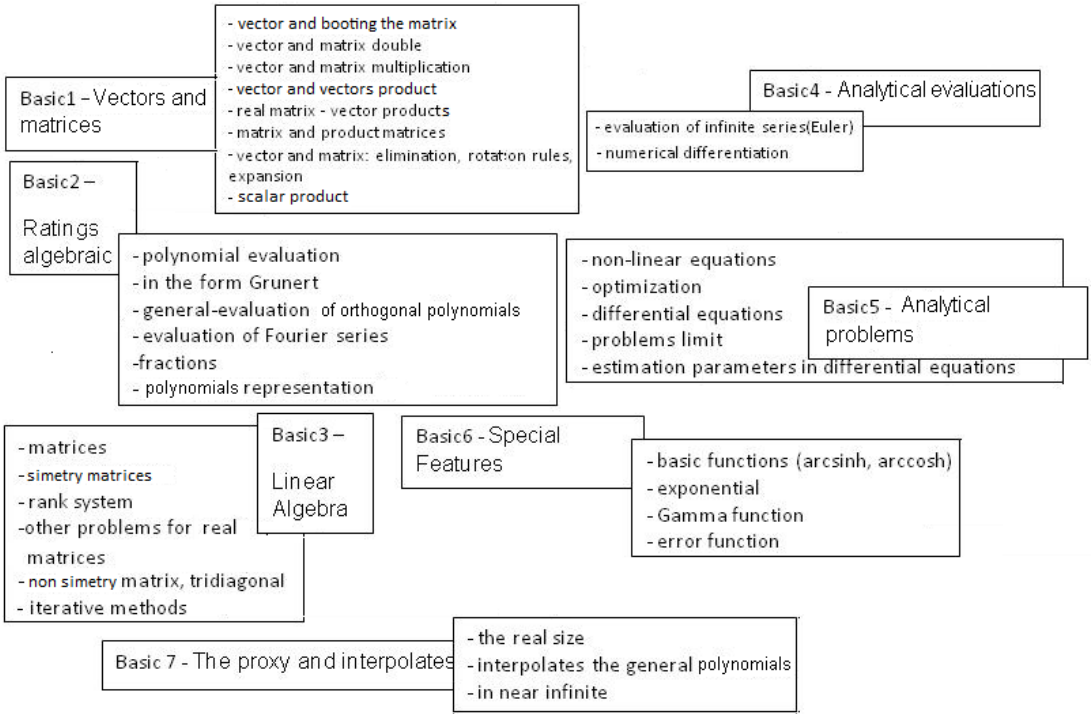

Figure 1. The basic grouping (Source: Isbasoiu, 2009: 3).

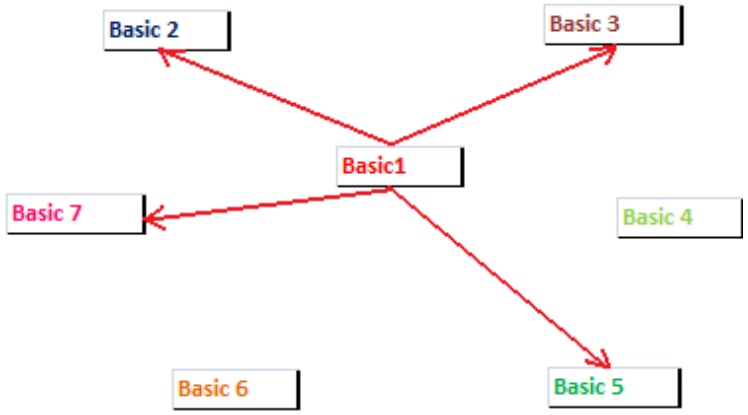

Figure 2. The groups' dependences in Basic1.

Peide. The graphic representation in Figure 5 shows the dependencies among these functions in Basic4.

In Basic5, there is the function: *Quanecobnd1* (Hang, 2004: 365). In the graphic representation in Figure 6, we present the independence of the functions in Basic5. There is only an internal interdependence with the *Rnk1min* function. In the graphic representation in Figure 7, we exposed the dependencies of the functions in Basic6. Here, we observe dependence with Basic7, realized with the *Minmaxpol* function. The last basic, Basic7, does not have any dependence. We try to conclude all the representations in a single one, so it is shown in the Figure 8. We then present a table (Table 1) with all the functions which appear in more than one basic (Isbasoiu, 2009: 4). A graphic representation of these dependencies after the number of the present function is realized in Figure 9 (Isbasoiu, 2009: 5).

The usage of the services in problems solving is justified by the major benefit of the web services usage, by the possibility of their re-usage in different combinations, independent of their implementation. In order to underline this advantage, we consider a few scenarios of usage, the services realized for each function in Table 1. This article presents the resolving of the systems with differential equations. Figure 10 presents the components implied in resolving. The following are the scenarios involved:

i. Used procedures: the production of two arrays; the production of an array and a matrix; matrices decomposition; particular cases,

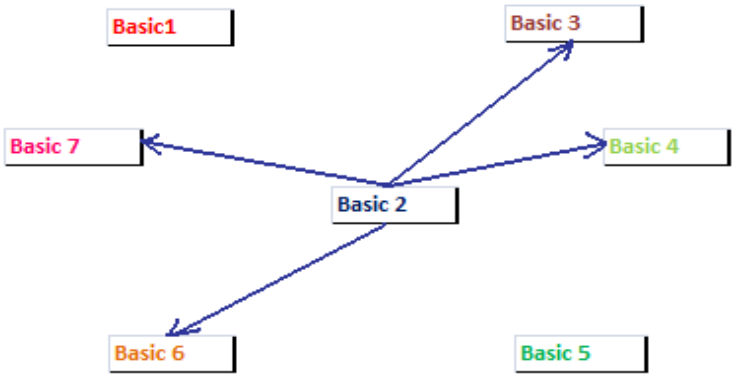

Figure 3. Basic 2 dependences.

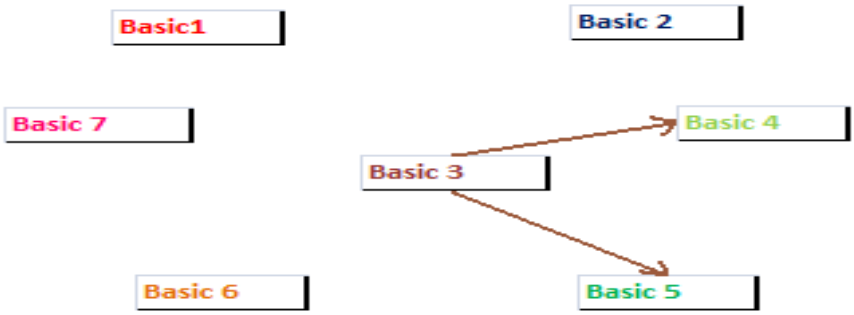

Figure 4. Basic 3 dependencies.

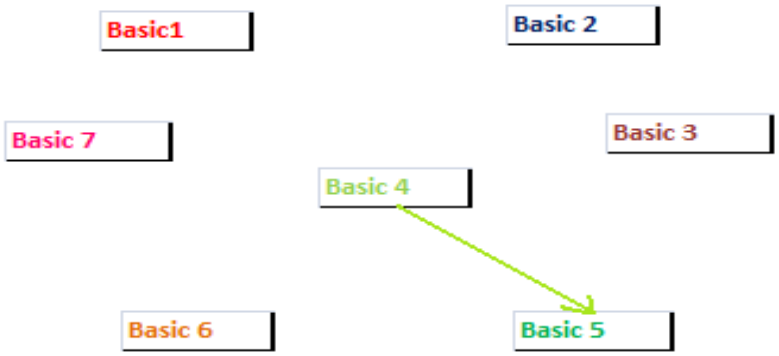

Figure 5. Basic 4 dependencies.

such as: a small number of variables as against the number of binary numbers; the resolving of the linear systems whose matrix was triangular decomposed
ii. The used service: Multistep

If one uses the web service with these scenarios, the following advantages are achieved: the possibility of resolving some problems of great dimensions, which, otherwise does not get into the client's computer memory; the reducing of the response time; the client can be involved into the problem's description.

As a consequence of the tests with encouraging aforementioned results, we propose a set of client applications with graphic interfaces based on web services, which allow the learning based on experiments.

A certain orientation of the solutions presented here is towards scientific applications based on numerical calculation (Newcomer and Lomow, 2004; Lawler and Howell, 2008). The knowledge and learning of the numerical calculation methods are practical, efficient and fairly elegant and the realization of our own architecture of web services will significantly contribute to this process (Hang, 2004: 245; Shneiderman et al., 2009).

The passing from an interface with the user in a command line to

Figure 6. Basic5 dependencies.

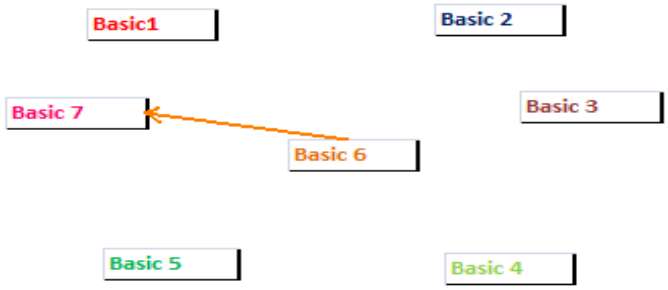

Figure 7. Basic6 dependencies.

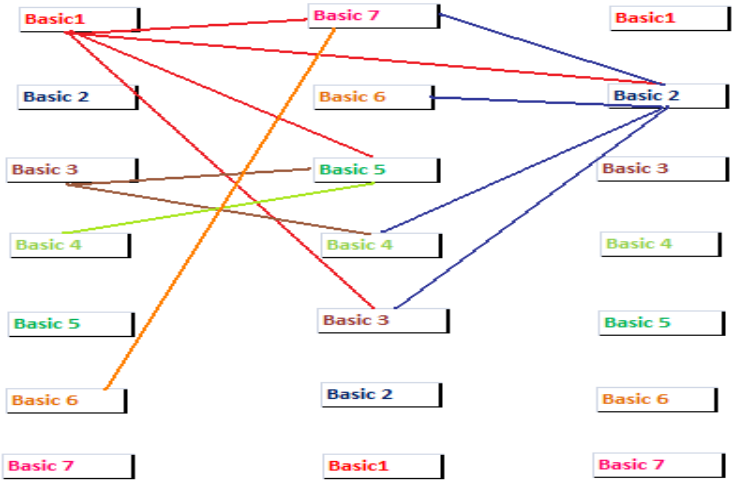

Figure 8. The representation of all the dependences (Isbasoiu, 2009: 4).

an interface based on web services represents a great advantage because it allows the combining and re-usage of some procedures or routines previously defined.

RESULTS

We consider those components which were already tracked down as being appealed by other components and we studied the impact on the response time in the case when the communication is realized through web service interfaces. It is to be noted that peculiar elements appeal more than other components before offering a response to the client.

We consider the elements in Table 1. The corresponding services were realized using the Eclipse platform. The presented example corresponds to *Rnk1min* function. Figure 11 presents the service interface (the WSDL interpretation in Eclipse).

Table 1. Functions which appear in more basics.

The function name	Basic 1	Basic 2	Basic 3	Basic 4	Basic 5	Basic 6	Basic 7
Rnk1min			*Rnk1min*		*Rnk1min*		
Praxis	*Praxis*		*Praxis*				
Marquardt	*Marquardt*		*Marquardt*				
Gssnewton	*Gssnewton*		*Gssnewton*				
Multistep	*Multistep*		*Multistep*				
Ark	*Ark*		*Ark*				
Efrk	*Efrk*		*Efrk*				
Efsirk	*Efsirk*		*Efsirk*				
Liniger1vs	*Liniger1vs*		*Liniger1vs*				
Liniger 2	*Liniger 2*		*Liniger 2*				
Gms	*Gms*		*Gms*				
Impex	*Impex*		*Impex*				
Peide	*Peide*		*Peide*				
Minmaxpol	*Minmaxpol*	*Minmaxpol*				*Minmaxpol*	

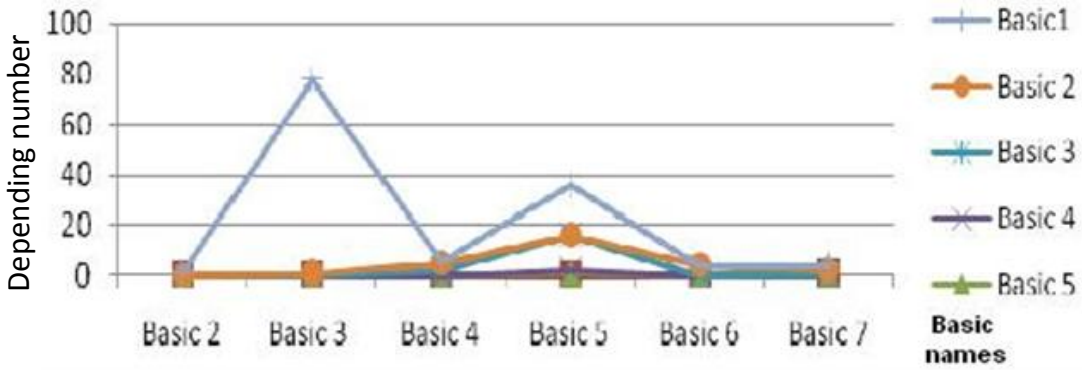

Figure 9. The graphic representation of the dependencies after the functions number.

An example of the service and manner interface of the client with graphic interface is presented in Figure 12. The response time is 16 min if the service was appealed locally (Figure 13). After testing, we noticed the results in the Table 3: according to every function there is the response time of the web service or of the component (without interfacing through a service). The times are the media of at least 20 tries. One can notice the considerable variations depending on the number of parameters transmitted by the interface (Isbasoiu, 2009: 6).

A more eloquent representation of the response time variations can be realized in Figure 14. As it follows, using the same method, we repeat the measurements, but for more complex input data, of the double type, not only for the integer type, for arrays and matrices, because the time depends on the input data. A more indicative representation of the response time variations can be seen in Figure 15. By comparing the two graphs in the Figures 14 and 15, and the data in the Tables 3 and 4, one can notice a clear dependence of the first half of the list with input data. The introduced surplus by the web interface is considerable and can get over 100% from the response time (for example Gms).

More than that, we can remark a surplus almost identical for the components with the same response time. In order to analyze the surplus introduced by the services, Table 5 and Figure 16 concentrate on the previous data (Isbasoiu, 2009: 2). By combining these two without WS, it results into Table 6 and Figure 17. The usage of the services in problem solving is justified due to the major benefit gained by using the web services, through the possibility of their re-usage in different combinations, independent on their implementation. In order to underline this benefit, we consider several scenarios of using the services mentioned earlier.

The following presentation is connected to the own created service and it is oriented towards client's interface. It is divided into two working areas: the former is for numerical calculation and the latter is for realizing the graphs for the cases selected and specified as

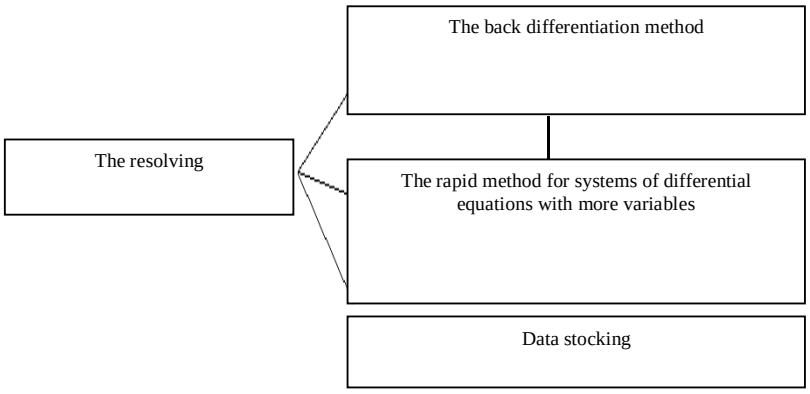

Figure 10. The graphic representation of the scenario for resolving systems with differential equations (Source: Hang, 2004: 670).

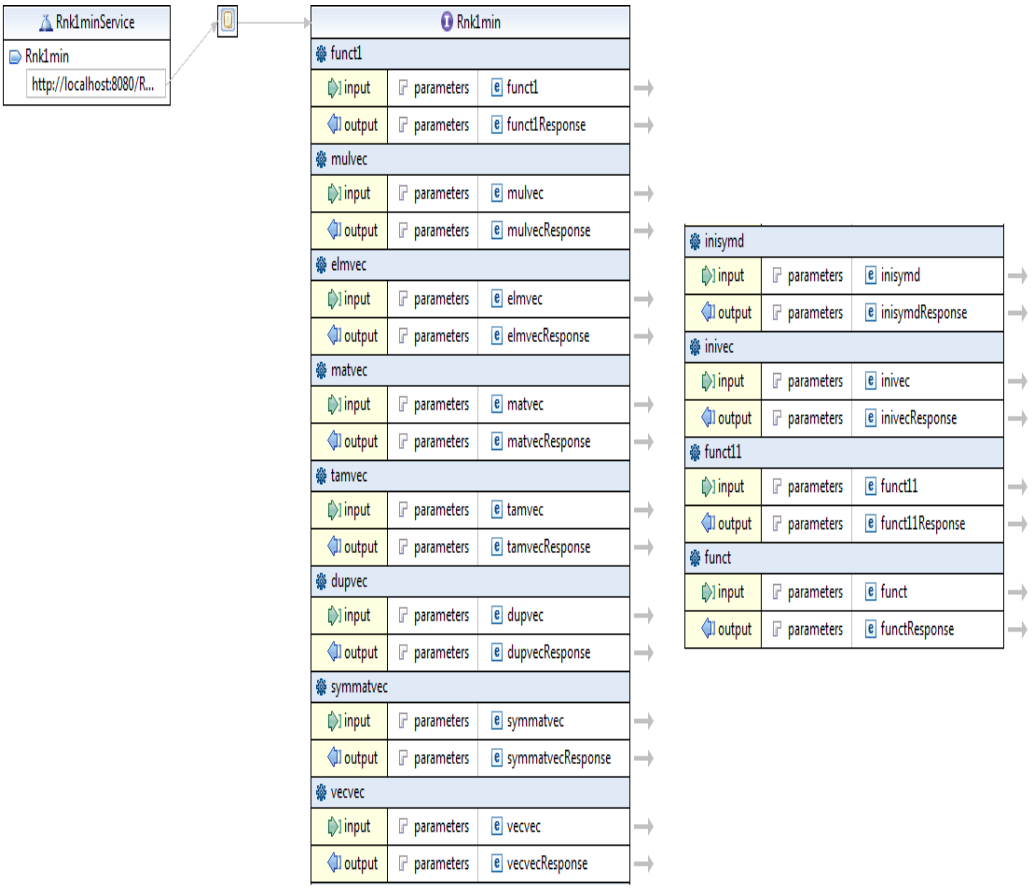

Figure 11. *Rnk1min* WSDL.

follows: in a static area where we present the selected studies, as in Figure 18 and 19 (Isbasoiu, 2010: 3). The variables introduced by the user which indicate the simulation parameters take values for a minimum and a maximum, chosen from the current web page. All the fields are seen as rows of values and treated in consequence. The own web could be a possibility to develop ideas and other applications on the basis of a well-known theory, thus generating other services in the future. Firstly, we realized a series of web services based on the

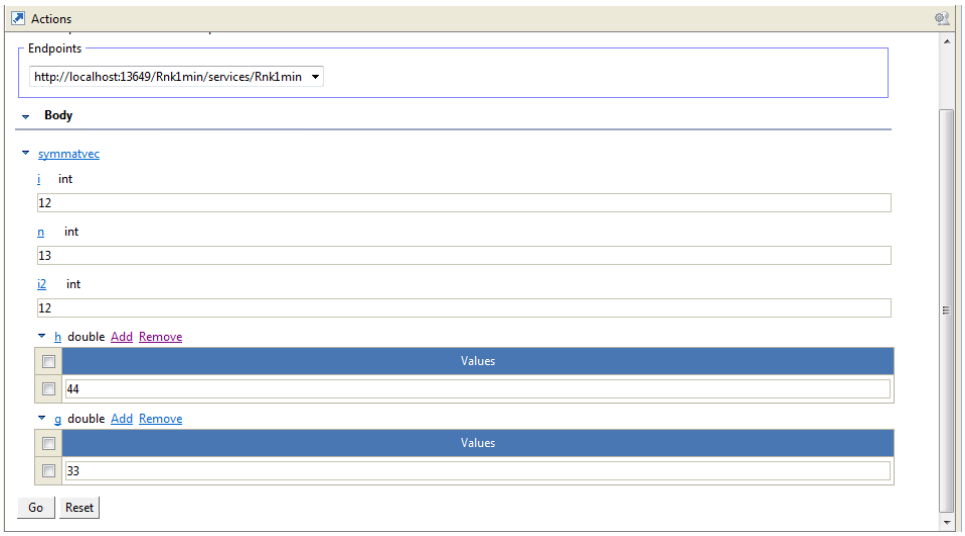

Figure 12. The launching of the web service for the *Rnk1min* function for an example.

Figure 13. The response time for the *Rnk1min* function for the respective example.

Table 2. The dependencies representation after the functions number.

	Basic 2	Basic 3	Basic 4	Basic 5	Basic 6	Basic 7
Basic 7						
Basic 6						1
Basic 5				1		
Basic 4				1		
Basic 3			2	14		
Basic 2		1	3		4	1
Basic1	1	77		20		2

numeric calculation (Landau et al., 2005; Michael, 2008). The passing form an interface with the user in the command line to an interface based on the web services is a great advantage, because it allows the combining and the re-usage of some procedures or routines previously defined (Abdul and Pilouk, 2008; Radhamani et al., 2007). The projecting of the graphic interface for the user follows the idea of other applications' further development, but the interpretations is a mix of numerical calculation finalized with the simulation of the phenomenon difference.

The own web service begins with another service previously described; we reorganized and developed it, focusing on its total completion. Thus we put at the users hands, the source code in order for him/her to intercede on the simulation and the realization of an equations editor, which generates the corresponding graph. The equations are to be found in the description made by the description of the phenomenon tool. Here is also the place for the graphs; the results are at the client's disposal on another page. The graphic must be defined within the text editor under the form equation (formula, min., max., step, precision):

i. the formula must be of x variable, enframed by double inverted commas. The x variable must have a space before and another space after in the equations, whenever it spears. For example: "sin(x) + cos(x + 4) + 2 * x "
ii. min, max represent the interval margins where the calculation is made;
iii. the step represents the incrementation value of the values for which the calculation was made;

Table 3. The response time of every component with and without web service.

The function name	The response time (value/m) without WS	The response time (value/m) with WS
Rnk1min	10	16
Praxis	8	11
Marquardt	9	13
Gssnewton	7	14
Multistep	8	10
Ark	11	15
Efrk	10	13
Efsirk	314	561
Liniger1vs	289	456
Liniger 2	293	473
Gms	348	605
Impex	311	562
Peide	362	628
Minmaxpol	353	611

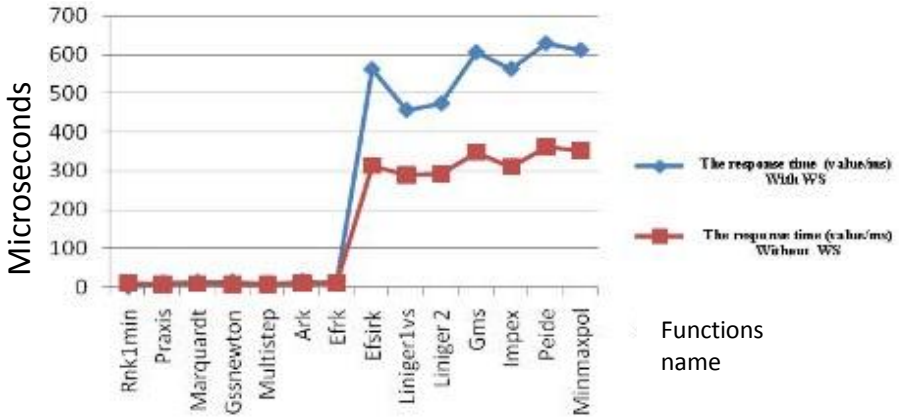

Figure 14. The graphic representation of the response time for every function with and without web service.

iv. the precision represents the ranging interval of the graphic. It is recommended to use 0.01; one can also use other values in order to see the results.

For the equation defining, all the mathematics functions are implemented:

i. abs: Absolute value
ii. acos: Arc cosine
iii. acosh: Inverse hyperbolic cosine
iv. asin: Arc sine
v. asinh: Inverse hyperbolic sine
vi. atan2: Arc tangent of two variables
vii. atan: Arc tangent
viii. atanh: Inverse hyperbolic tangent
ix. base_convert: Convert a number between arbitrary bases
x. bindec: Binary to decimal
xi. ceil: Round fractions up
xii. cos: Cosine
xiii. cosh: Hyperbolic cosine
xiv. decbin: Decimal to binary
xv. dechex: Decimal to hexadecimal
xvi. decoct: Decimal to octal
xvii. deg2rad: Converts the number in degrees to the radian equivalent
xviii. exp: Calculates the exponent of e
xix. expm1: Returns exp(number) - 1, computed in a way that is accurate even when the value of number is close to zero
xx. floor: Round fractions down
xxi. fmod: Returns the floating point remainder (modulo) of the division of the arguments
xxii. getrandmax: Show largest possible random value

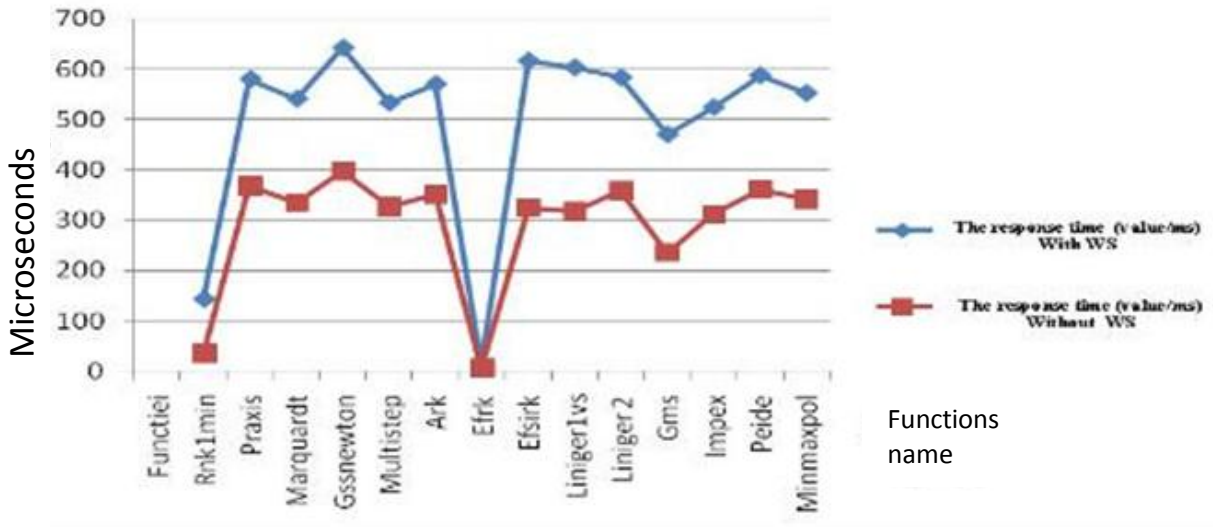

Figure 15. The graphic representation of the response time for every function with and without WS.

Table 4. The response time of every function with and without web service.

The function name	The response time (value/ms) without web service	The response time (value/ms) with web service
Rnk1min	36	143
Praxis	368	580
Marquardt	336	541
Gssnewton	398	642
Multistep	327	533
Ark	352	570
Efrk	7	13
Efsirk	326	616
Liniger1vs	318	603
Liniger 2	359	583
Gms	236	470
Impex	312	524
Peide	362	587
Minmaxpol	342	552

xxiii. hexdec: Hexadecimal to decimal
xxiv. hypot: Calculate the length of the hypotenuse of a right-angle triangle
xxv. is_finite: Finds whether a value is a legal finite number xxvi. is_infinite: Finds whether a value is infinite
xxvii. is_nan: Finds whether a value is not a number
xxviii. lcg_value: Combined linear congruent generator
xxix. log10: Base-10 logarithm
xxx. log1p: Returns log(1 + number), computed in a way that is accurate even when the value of number is close to zero
xxxi. log: Natural logarithm
xxxii. max: Find highest value
xxxiii. min: Find lowest value
xxxiv. mt_getrandmax: Show largest possible random value
xxxv. mt_rand: Generate a better random value
xxxvi. mt_srand: Seed the better random number generator
xxxvii. octdec: Octal to decimal
xxxviii. pi: Get value of pi
xxxix. pow: Exponential expression
xl. rad2deg: Converts the radian number to the equivalent number in degrees
xli. rand: Generate a random integer
xlii. round: Rounds a float
xliii. sin: Sine
xliv. sinh: Hyperbolic sine

Table 5. The response time with web service.

The function name	The response time (value/ms) with web service – small dimension of data	The response time (value/ms) with web service – big dimension of data
Rnk1min	16	143
Praxis	11	580
Marquardt	13	541
Gssnewton	14	642
Multistep	10	533
Ark	15	570
Efrk	13	13
Efsirk	561	616
Liniger1vs	456	603
Liniger 2	473	583
Gms	605	470
Impex	562	524
Peide	628	587
Minmaxpol	611	552

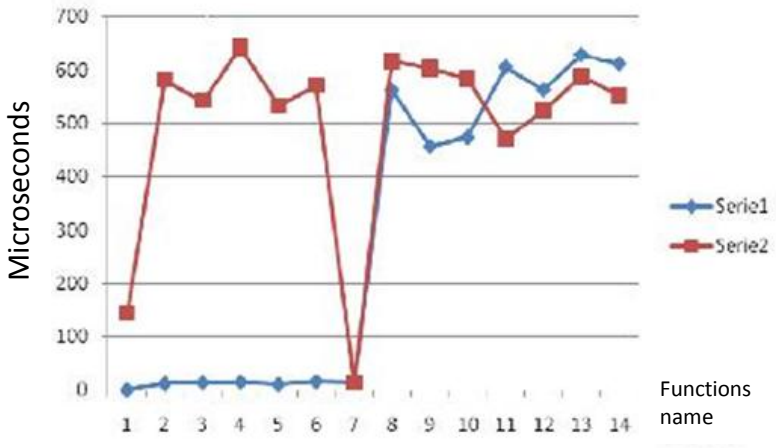

Figure 16. The graphic representation of the response time as compared to WS.

xlv. sqrt: Square root
xlvi. srand: Seed the random number generator
xlvii. tan: Tangent
xlviii. tanh: Hyperbolic tangent

In conclusion, one can define the graphs with formulae of the type I:

i. equation (" x ",0,50,1,0.01)
ii. equation ("sin(x) * x ",-3,3,0.1,0.01)
iii. equation (" sin (cos (x)) + 1",-3,3,0.1,0.01)

The web pages are generated on a linux/apache server. The application rule only on the linux server, but it can be adapted to function with other operating systems too. The main idea is represented by the way the equations are introduced in the administrative part and the fact that they are automatically generated on the interface. This is an original method which combines many original parts of the codes, in order to obtain graphs with the automatic processing of the equations. Another advantage is the idea that the generated text can be reprocessed and changed, in order to automatically generate the graphs. One has the possibility that every time after the visualization of every graph, to see also the values generated. A few examples of graphs generated by this method are shown in Figures 20 and 21. Figure 21 presents the movement of the same variable, depending on the entrance data.

DISCUSSION

By analyzing the pieces of information presented, we can

Table 6. The response time without web service.

The function name	The response time (value/ms) without web service – small dimension of data	The response time (value/ms) without webservice – big dimension of data
Rnk1min	10	36
Praxis	8	368
Marquardt	9	336
Gssnewton	7	398
Multistep	8	327
Ark	11	352
Efrk	10	7
Efsirk	314	326
Liniger1vs	289	318
Liniger 2	293	359
Gms	348	236
Impex	311	312
Peide	362	362
Minmaxpol	353	342

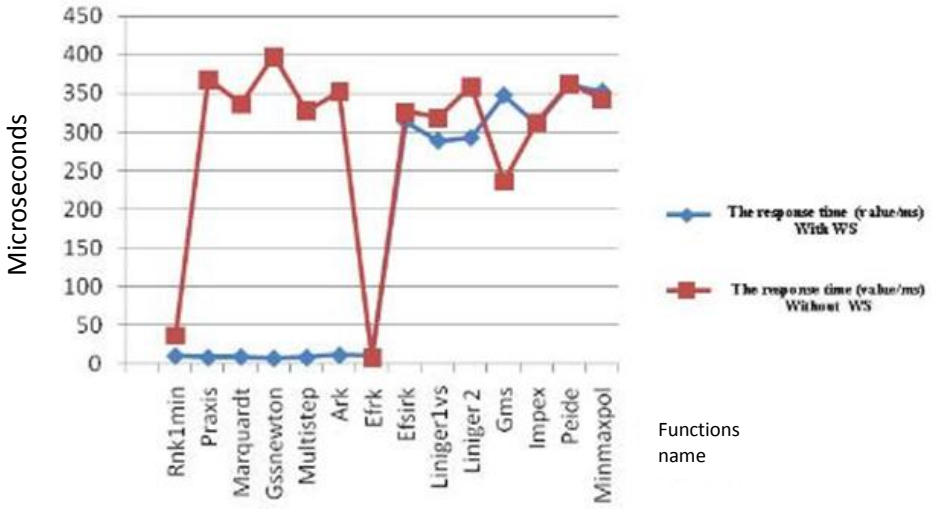

Figure 17. The graphic representation of the response time as comparing to without WS.

say the following: generally, the response time is greater for complex parameters and the main cause is represented by the value itself of these parameters; the longest response time is for those functions which resolve determinants, matrices and arrays; there are extremely close response time for some functions; it is obvious that the only function which has the smallest response time no matter the complexity of the input data is Efrk with 7 and 13 min; if we compare the time in the case of the launching of the web service, we notice greater differences, so, within these cases, the complexity of the input data are of great importance. If we compare the response time without the web service, we noticed greater value of the time, close time, but without a rule connected with the type of the input data, which is important for the function and its complexity

To round up this paragraph, we can say that the surplus introduced by the web services is considerably relative to the response time of the numerical calculation components, realized in basic procedures, especially in the cases when the data are numerous; the introduced surplus is kept under the level of 1 s; this fact justifies the usage of the services in order to take advantage from interoperability among the components written in different languages or the re-usage of some external codes. Thus, from modularity reasons, we consider as being reasonable the introduction of the services at the level of some basic components which were insignificantly

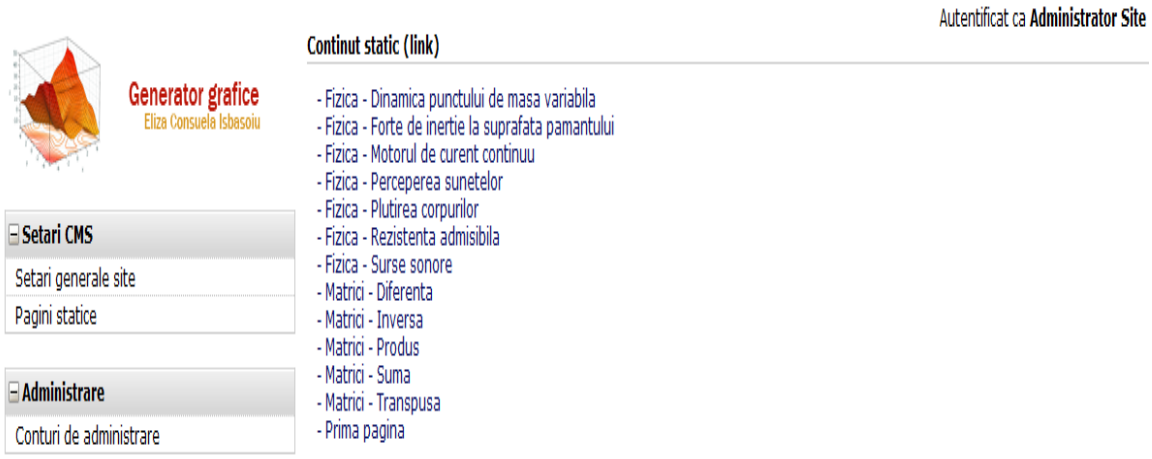

Figure 18. Static content (links) - image from the application realized in Romanian language, the unique version.

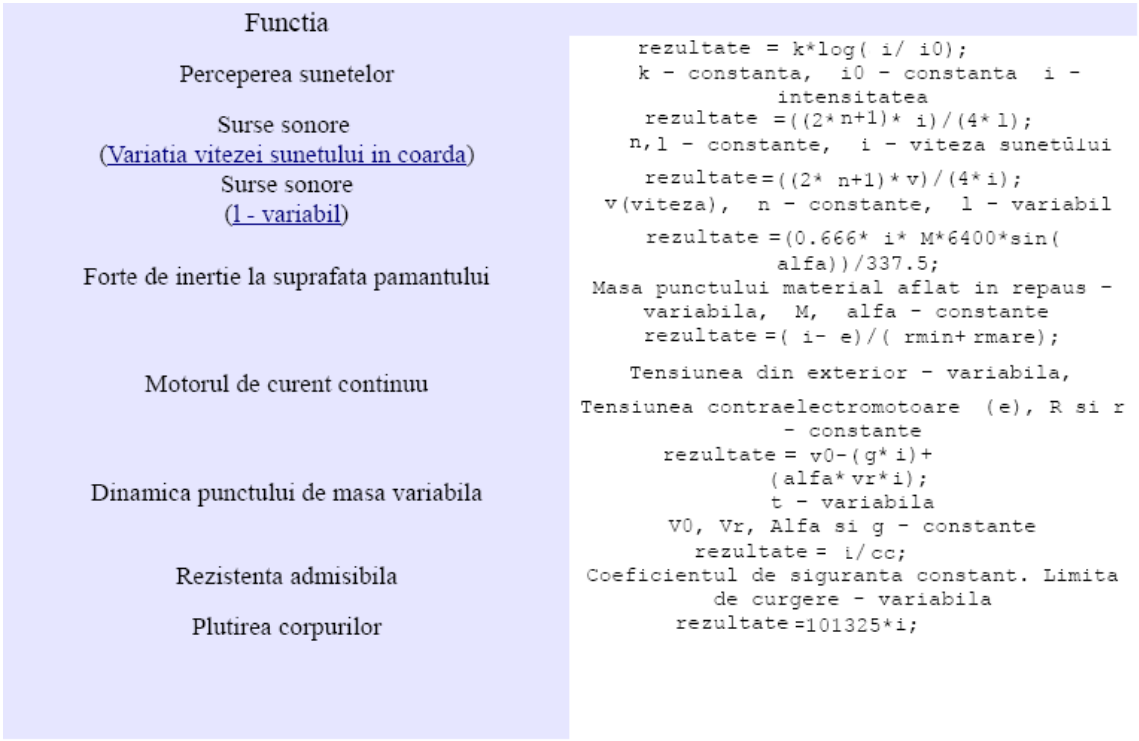

Figure 19. Example of calculation function - image from the application realized in Romanian language, the unique version.

influenced by the introduction of the web services.

From these representations, we can draw the following conclusions and observations: the necessity of knowledge; the possibility of their interpreting, depending on the values of the input data; simulations in order to find out other specific cases; the understanding of the physical phenomenon, at a practical level.

Taking into account the aforementioned, we realized that the service – in its first form – has a drawback, namely the impossibility to make comparisons between the input and output data, respectively the obtained graphs in the cases of the successive repetitions. In order to eliminate this disadvantage, we created the possibility to display simultaneously the two graphs, the initial and the second one, respectively, and introduce the values obtained. This is quite possible for every chosen

Figure 20. The calculation of a variable.

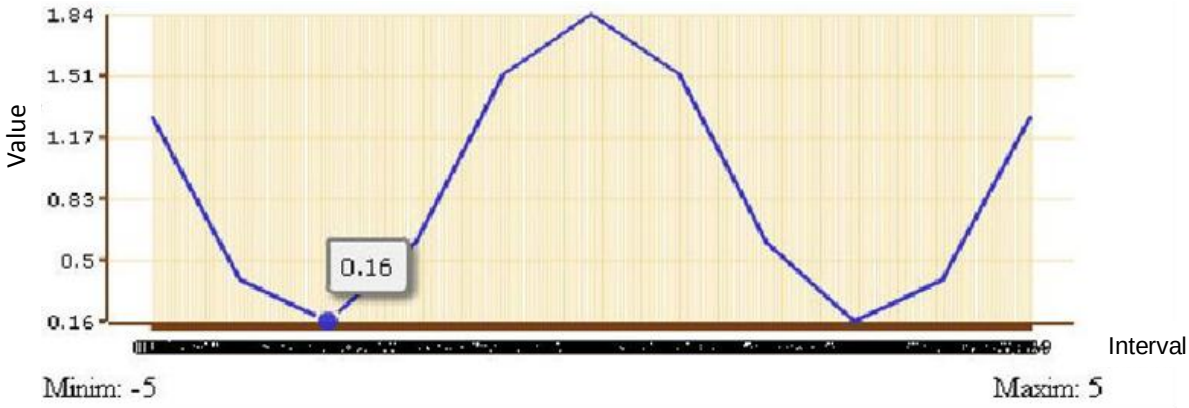

Figure 21. The movement of the same variable, depending on the entrance data.

phenomenon (Isbasoiu, 2010: 6).

Conclusions

We have presented the role of the web services and their benefits, but also the problems which occur when utilizing some web services. We systematically presented some concepts connected to the architecture of the distributed systems, exposing criteria for a good choice in using the proper technologies. The problem we dealt with consists in realizing some web services with the help of the numerical calculation. Many discussions about algorithms in the specific literature on the web, omit important details, which can only be uncovered by the coding or compatible with the code reading. The applications are created on the Eclipse platform and are realized with the help of the solutions presented in the treatise Numerical Library in Java for Scientists and Engineers. These solutions are oriented towards scientific applications based on the numerical calculation. They start with elementary structures and get some more complicated structures. By grouping them, and then, creating services for each of them, we noticed that many functions, for their construction, appeal to other functions; thus we realized a table with those functions appearing only one time, from them some others start. For each function, we created own service and followed the response time. It is imperative to remark that depending on the function complexity, the time differs. By comparing them, we realized a graphic representation with both the response time of the individual functions used as input data with simple values and the parameters with very complicated values, of different types. The next step of the research was to realize web services for education in physics and scientific research, using these services.

This activity can be seen from three different points of view namely the psychological point of view; the systematic point of view; and the organizational point of view.

From the psychological perspective, the application is a creative process, which requires knowledge in close subjects, such as the software engineer, the computer science, logics, cognitive sciences, programming languages, software methodologies for projecting and

knowledge in the applicative field.

From the systematic perspective, the project is seen as an activity which implies the finding of optimal solutions to a set of problems as long as the competition of the obstacles and forces is balanced.

The organizational perspective shows that there is the possibility that other software elements can be reused for other products. This following are reasons for which this web service was created:

i. There are a lot of people who pay attention to the software architectures, but they consider it to be a distinct profession and tend to separate it from the software engineer.
ii. The created web service would be a practical and useful solution for a medium level for research, without supplementary costs.

In the near future, we intend to undertake a more intensive study on sounds from the numerical calculation point of view and that of the simulations. Within this study, we have used the already created web service, modifying it by adding in conformity with the necessary calculation both for the numerical interpretations and simulations.

REFERENCES

Abdul AR, Pilouk M (2008). Spatial Data Modelling for 3D GIS, Springer.

Dagger D, Wade V, Conlan O (2005). Personalisation for All: Making Adaptive Course Composition Easy. Special issue of the Educational Technology and Society Journal.

Hang TL (2004). Numerical Library in Java for Scientists and Engineers, Chapman and Hall/CRC.

Hansen MD (2007). SOA Using Java Web Services, Prentice Hall.

Holdener AT (2008). Ajax: The Definitive Guide, publisher - O'Reilly.

Hurwitz J, Bloor R, Baroudi C, Kaufman M (2007). Service Oriented Architecture FOR DUMMIES, Wiley Publishing, Indianapolis, Indiana.

Isbasoiu EC (2009). Web Services for Physics – Graphics Generator, EMS 2009 UKSim Third European Modelling Symposium on Computer Modelling and Simulation [Online]Available: http://portal.acm.org/citation.cfm?id=1675163.

Isbasoiu EC (2010). Simulations of Some Physical Phenomena Using the Numerical Calculations, ISMS 2010 UKSim/AMSS First International Conference on Intelligent Systems, Modelling and Simulation [Online] Available: http://portal.acm.org/ citation.cfm?id=1731289.

Iyengar A, Jessani V, Chilanti M (2008). WebSphere Business integration Primer, Process Server, BPEL, SCA and SOA, International Business Machines Corporation.

Koper R, Tattersall C (2005). Learning Design – A Handbook on Modelling and Delivering Networked Education and Training.

Koren I, Krishna CM (2007). Fault – Tolerant Systems, Elsevier San Francisco.

Landau RH, Wangberg R, Augustson K, Paez M, Bordeianu C , Barnes C (2005). A First Course in Scientific Computing. Symbolic, Graphic, and Numeric Modeling Using Maple, Java, Mathematica, and Fortran90, Princeton University Press, Princeton and Oxford.

Lawler JP, Howell HB (2008). Service – Oriented Architecture, SOA Strategy, Methology and Technology, Auerbach Taylor & Francis Group LLC.

McGovern J, Tyagi S, Stevens M, Matthew S (2003). Java Web Services Architecture, Morgan Kaufmann Publishers.

Michael B (2008). Service-Oriented Modeling, Service Analysis, Design and Architecture, publisher - John Wiley & Sons, Inc., Hoboken, New Jersey.

Nagappan R, Skoczylas R, Sriganesh RP (2003). Developing Java™ Web Services, Architecting and Developing Secure Web Services Using Java, Robert Ipsen, Wiley Publishing Inc., Indianapolis, Indiana.

Nakhiovsky A, Myers T (2004). Google Amazon and Beyond: Creating and Consuming Web Services, Springer.

Newcomer E, Lomow G (2004). Understanding SOA with Web Services, Addison Wesley Professional.

Petcu D (2006). Arhitecturi si tehnologii Grid, Editura Eubeea, Colectia Informatica.

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007). Numerical Recipes – The Art of Scientific Computing, Cambridge University Press.

Prodan R, Fahringer T (2007). Grid Computing – Experimet Management, Tool Inegration and Scientific Workflows, Springer.

Radhamani G, Radha Krishna Rao GSV (2007). Web Services Security and E-Business, Idea Group, Multimedia University, Malaysia.

Shneiderman B, Plaisant C, Cohen M, Jacobs S (2009). Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison Wesley.

Toma S, Găbureanu S, Făt S, Novak C (2009). Impact Evaluation of Intel Teach Programme in Romania.

Vasiliev Y (2008). SOA and WS-BPEL, Composing Service-Oriented Solutions with PHP and ActiveBPEL, Packt Publishing Ltd.32 Lincoln RoadOltonBirmingham.

Xiao Y (2007). Security in Distributed, Grid, Mobile and Pervasive Computing Auerbach Publications.