

# iGraf: an interactive graphical software for CSET education

Reginaldo do Prado, Anarosa A.F. Brandão, and Leônidas de Oliveira Brandão  
University of São Paulo, rprado@ime.usp.br, anarosa.brandao@poli.usp.br, leo@ime.usp.br

**Abstract - An important discipline in the basic education of CSET students is related to mathematical functions and associated issues. Therefore, technology advances on building and delivering software to support such subject had been benefited students all over the world. However, most of such software are stand alone systems that don't provide support for their integration to Learning Management Systems – LMS. In this paper we present iGraf - interactive Graphics on the Internet, a new software that provides the basic resources for general-purpose plotting tools, as well as resources that make possible its easy integration with LMS. Moreover, it provides a simple model for authoring and automatic assessment features. Both, authoring and automatic assessment were designed based on a systematic study of the literature related to Calculus and K-12 math courses. In order to produce a proof of concept, it was integrated to Moodle with the iAssign package and some experiments with K-12 math teachers and undergraduate students were conducted. iGraf can be freely downloaded from <http://www.matematica.br/igraf>.**

*Index Terms* – Automatic Assessment, Distance Learning Education, Functions, Graphical Viewer.

## INTRODUCTION

Two important contributions from the *World Wide Web* (which for now on we will denote by **Web**) is to allow students to interact with digital educational content, anywhere and anytime, and to enable teachers to distribute such contents to a huge number of students. Its importance was completely established with the advent of the *Learning Management Systems (LMS)*, which allowed teachers to monitor the students' activities and provided several communication tools for both actors, teachers and students. LMS are key systems for the new *Distance Learning (DL)* model, that is delivered through the Web.

Several references pointing out advantages of using *Information and Communication Technology (ICT)* in education were found in the literature. The possibility of interaction with the content (without third part intervention) and the capability to promote the reuse of digital contents (from other authors or previous use) could be cited as important advantages of such a use. From the learning perspective, the immediate feedback to students is very positive to keep them motivated to study, as pointed by many authors -.

However, whenever the study of functions is considered, the lack of tools that can be integrated to LMS and provide mechanisms for automatic assessment is a reality.

In this sense, the major contribution of this paper is to present **iGraf - Interactive Graphics on the Internet**, a new system that aims to deal with the shortcomings mentioned above. By using iGraf, teachers can develop activities involving functions (such as drawing graphics, derivatives and integrals of functions), and students can interact directly with iGraf activities, even through the Web. Besides, it implements the concept of *interactive Learning Module (iLM)*, formerly known as e-LM, which allows iGraf to be used in an integrated manner to any LMS, particularly under *Moodle*<sup>1</sup>, which already has an iLM integrator. Another important aspect of this proposal is to encourage the production of educational content directly by teachers. However, as this task is laborious, as indicated by the experimental results reported here, it is convenient that teachers use iGraf integrated with some repository to exchange and enhance digital learning content.

This paper is structured as follows: in the next three sections some background issues and related work were presented, followed by the iGraf overview. After, a section about the authoring and automatic assessment resources of iGraf is given, followed by some experiments and their associated results. At last, we present our conclusions.

## BACKGROUND

The subject *function* is common in books to middle and high school and in introductory books to undergraduate courses in sciences. The importance of the study of functions is emphasized by Lu, “Developing an understanding of the concept of functions is central to secondary mathematics education”. Another evidence of its importance is the number of books on the subject in several areas such as Financial Mathematics, Nutrition, besides the classic courses of Exact Sciences.

Another important topic that is constructed upon the concept of function is the *differential and integral calculus*. For example, Stewart notes, “The fundamental objects that we deal with in calculus are functions”, while Avila says that “all the Differential and Integral Calculus are developed around two fundamental concepts: the concept of function

<sup>1</sup> Modular Object Oriented Dynamic Learning Environment: <http://moodle.org>

and concept of limit"<sup>2</sup>. In conclusion, Barnett argues that "the concept of function is one of the most important ideas in mathematics" [6].

In general, the "...study of mathematics beyond the elementary level requires a firm understanding of a basic list of elementary functions, their properties and graphs" [6] and the mastering of these topic requires that the student faces several different exercises. This need is explicit in Dahl conclusions, "All pupils say that doing exercises is important in learning", but it is also a tacit conclusion of various books of Calculus in which we can find a large amount of exercises.

In this context, the use of Web-based system emerges as a good option for teachers applying a more extensive list of exercises. Moreover, if the system has features to streamline the understanding of the objects studied, as is the case for graphic viewers, ICT works in another important direction that is to stimulate the student to learn.

Another important aspect to student learning motivation is providing quick feedbacks on the activities undertaken by pupils. It is highlighted, for instance, in -.

Thus education can be significantly enhanced under the use of computer systems that employ some automatic evaluation technique. In one hand, the quick feedback can reduce the students frustration - and on the other hand, the system may also reduce the teachers workload. Another interesting aspect that must be emphasized is the potential to enhance digital contents through summary reports of students performance. Having them, teachers can quickly identify and fix flaws in activities or realize the lack of prerequisites for any of the released activities. These corrections being fast have a direct effect on the future performance of the students.

At last, again considering the point of view of teachers, there is another difficulty they face: the production of good quality activities, being them digital or not. Here, environments in which teachers can work collaboratively, exchanging and refining digital learning content are ICT tools that can help teachers to circumvent this problem.

#### RELATED WORK

There are several approaches to support the learning of functions using ICT, specially the ones which use graphical viewers to explore their properties. Computer Algebra Systems (CAS) are example of systems that provide these functionalities and that were already used for such purpose. CAS were built to support the manipulation of mathematical formulae, since they were primary intended to automate tedious and sometimes difficult algebraic manipulation tasks. Therefore, they are "heavy" systems that provide much more functionalities that were need to support the learning of

<sup>2</sup> Our translation from the original Portuguese "todo o Cálculo diferencial e Integral se desenvolve em torno de dois conceitos fundamentais: o conceito de função e o conceito de limite".

functions and most of them didn't present authoring and automatic assessment resources.

Exceptions are Maple TA<sup>3</sup> and WebMathematica<sup>4</sup> since both present tools for exporting content as HTML files and Java applets that allow some interaction with the content. Nonetheless, Maple TA can be integrated to the LMS Blackboard<sup>5</sup>, but both are commercial systems with high acquisition costs. This could be a hard constraint for their large scale use, specially in under development countries. Also, since they are big systems, they need more computational resources than iGraf.

Moreover, there are several web portals that offer educational resources associated with functions, such as Wessa.net<sup>6</sup> e WebGraphing.com<sup>7</sup>. In addition, there are stand alone systems to draw and manipulate function graphs, such as Winplot<sup>8</sup> (free distribution), Graphmatica<sup>9</sup> (shareware) and Autograph<sup>10</sup> (commercial). Such systems are platform dependent and didn't run on Web browsers.

iGraf differs from the aforementioned systems due to its flexibility (it can be used in a stand alone version and as an applet Java, running in Web browsers) and availability as free software. Also, since it is an iLM, it can be integrated to LMS. Currently, there is a Moodle package called iAssign that provides integration of iLM to Moodle enabling teachers to benefit from all Moodle features.

#### iGRAF OVERVIEW

iGraf is a system to support the process of teaching and/or learning of mathematical functions. It was fully developed in Java and it has a Web version. iGraf is freely available at <http://www.matematica.br/igraf>. Its development started in 2004 as part of the research activities of the Laboratory of Mathematics Teaching of the Institute of Mathematics and Statistics from the University of São Paulo – Brazil. Three prototypes were created until its deployment, in 2010. In the following it is presented an overview of the system.

##### I. iGraf Architecture

The modular approach was adopted for the definition of the iGraf architecture. Such an approach allows the integration of new functionalities to the system with little effort. The architecture is composed of seven modules: *broadcaster*; *file manager*; *drawing manager*; *user input manager*; *common utilities*; *auxiliary frames* and *JCM Package* (see Figure 1).

The broadcaster is the module responsible for dealing with event communication. It receives events communication from each module separately and broadcasts them.

<sup>3</sup> <http://www.maplesoft.com/products/mapleta>

<sup>4</sup> <http://www.wolfram.com/products/webmathematica>

<sup>5</sup> <http://www.blackboard.com/>

<sup>6</sup> <http://www.wessa.net>

<sup>7</sup> <http://webgraphing.com>.

<sup>8</sup> <http://math.exeter.edu/rparris/winplot.html>

<sup>9</sup> <http://www.graphmatica.com>

<sup>10</sup> <http://www.autograph-math.com>

Therefore, after receiving the event communication each module decide if that was an event to be treated or not. Such a decision is transparent since each module is described using the Model-View-Controller (MVC) paradigm .

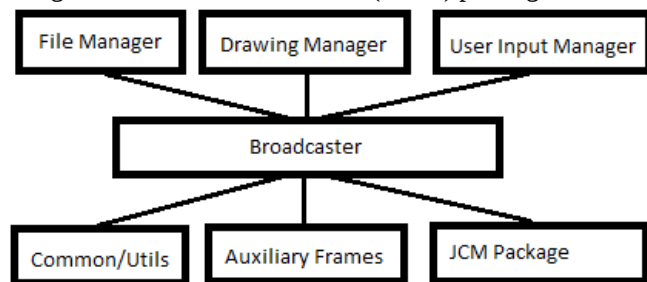


FIGURE 1  
iGRAF MODULAR ARCHITECTURE.

The file manager is responsible for dealing with events related to the creation/deletion/saving of files; the drawing manager is responsible for dealing with events related to drawing objects from the main interface's drawing area (Figure 2). The user input manager is responsible for dealing with events related to input data and if such data is a math expression it is sent to the broadcaster, in order to be treated by the Java Components for Mathematics (JCM) Package module. The common utilities is the module responsible for dealing with the system configuration events, including the system internationalization according to ISO 639 and ISO 3166 . Finally, the auxiliary frames module is responsible for dealing with events related to all the auxiliary windows that are popped up during the system use, for instance, while in an authoring activity.

## II. iGraf Main Interface

The iGraf main interface was designed to be clean and simple. It is composed of five parts: at the top the *title bar*, the *edit area* and the *buttons' menu*; in the center, the *drawing area* and at the bottom, the *message bar* (see Figure 2). The *title bar* identifies the system and, in its stand alone version, it presents, in its left side, the menu *File*, for opening existing or new files and saving files. Users input data to interact with the system using the *edit area*. Such data are, generally, function expressions that would be plotted in the *drawing area*.

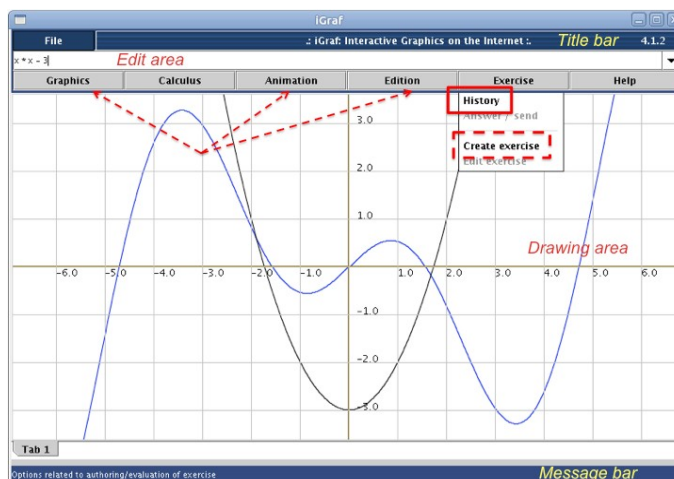


FIGURE 2  
iGRAF MAIN INTERFACE

Additional resources were available from the *buttons' menu*. For instance, when the mouse is placed over the *Exercise* button, a combo with related functionalities is opened. In this example, the functionalities are *history*, *answer/send*, *create exercise* and *edit exercise* (Figure 2). The *message bar* is responsible for presenting to users the systems state. Figure 2 shows the screenshot of the system while opening the *Exercise* button and the message bar contains information related to the systems functionalities that are available because of such action.

## III. iGraf resources

iGraf presents features for text editing, zooming, display options of the Cartesian axes, among other common functionalities of several graphical viewers. Nevertheless, iGraf also has other sophisticated options, such as: animation of functions through the use of parameters (such as the parameters  $a$  and  $b$  in the expression  $a*x^2+b$ ); tools for Differential and Integral Calculus (animation of tangent lines for graphs; drawing multiple tangent lines, and the differential function); generation of Web pages; and the register of the session *history*.

The *history* resource (highlighted with a rectangle in Figure 2) allows teachers to review all actions performed by a student to achieve a certain result, improving the teacher understanding about the student's reasoning path. Also, teachers can also build tutorials by performing several actions and saving them in a file that could be followed by students as well as they want. Nevertheless, such a resource can also be used by the system itself to include intelligent tutoring features. Finally, other advanced features that are noteworthy in iGraf are the authoring and automatic assessment tools. They will be further explained, in a section for this purpose.

An interesting characteristic of iGraf is the possibility of integrating it into LMS in order to get the benefits from both systems during the process of teaching and learning functions. Currently, it is possible to integrate iGraf to SAW and Moodle .

## AUTHORING AND ASSESSMENT RESOURCES

The growing number of people with access to formal education in high school, in higher education and in distance learning courses in general, has led to an increased number of students per class, indicating that automatic evaluation tools for exercises can produce great benefit to the education process. As Sondergaard and Thomas say, "Since cost-effectiveness may grow with class size, some types of investment into Web-delivered feedback become attractive. We already mentioned the model of broadcast fair".

Besides the likely time-saving for teachers, such tools can enhance positive behaviors for teachers and students. For instance, teachers can quickly identify problems in one activity (identified from a poor performance class) and students may keep motivation and challenge with available activities due to the rapid feedback in each activity.

In order to set the iGraf model for automatic evaluation, it was done a detailed survey about the most frequently used types of questions related to functions in books adopted by high school and CSET introductory undergraduate courses in Brazil. Exercises from 6 books were analyzed: 2 from high school curricula; 2 from Integral and Differential Calculus; and 2 related to computational activities to support learning functions. Moreover, two Web portals related to the subject were also analyzed.

The data analysis was divided into two stages. Initially, all the exercises were read to determine categories for the exercises types. The criterion for the category delimitation was the answer pattern requested, generally defined explicitly in the statement. Finally, the categories were defined as: numerical results (a value), a set (usually an interval), algebraic expressions (usually a function) and graphics (the drawing of a function). Since the last two types were associated with a function itself, they were aggregated into a single category, therefore, resulting in 3 categories: *numeric*; *interval* and *expression*. As indicated in the Table 1, just a few exercises were not included in one of the established categories.

Once defined the categories, the exercises were categorized by book. For instance, Iezzi (a book) presented 274 exercises categorized as numeric, 164 as interval, 104 as expressions and 74 as other (Table 1). It were analyzed 2.991 exercises.

TABLE 1:  
CATEGORIZING EXERCISES FROM BOOKS AND WEB SITES

| Sources    | Numeri<br>c | Interval<br>s | Expression | Others | Tota<br>l |
|------------|-------------|---------------|------------|--------|-----------|
| Iezzi      | 274         | 164           | 104        | 74     | 616       |
| Paiva      | 174         | 75            | 57         | 33     | 339       |
| Ávila      | 235         | 26            | 317        | 85     | 663       |
| Boulos     | 83          | 0             | 189        | 40     | 312       |
| Silva      | 163         | 139           | 132        | 119    | 553       |
| Figueiredo | 67          | 6             | 109        | 55     | 237       |
| Barufi     | 10          | 8             | 50         | 19     | 87        |
| Taylor     | 78          | 23            | 74         | 9      | 184       |
|            | 1084        | 441           | 1032       | 434    | 2991      |

## I. Authoring an exercise in iGraf

An exercise in iGraf must have a statement clear and objective, describing what the student must do to solve it. Such statement can be made using the iGraf or the LMS editor. In order to use the automatic assessment features of iGraf, the authoring activity will be complete after defining the desired answer in the model answer configuration panel (Figure 3). The exercise could have any number of items, each one based on the 3 categories previously presented. For instance, Figure 3 shows an exercise answer panel during an incomplete authoring activity. Moreover, such exercise is expected to receive answers composed of 4 items (a real number, an ordered pair, an interval and an algebraic expression), whose expected values and expressions must be written in the blanks to finish the authoring activity.

## II. Automatic assessment in iGraf

Having an exercise ready to be solved, the student will have the same panel for configuring its answer as the one presented in Figure 3. Therefore, he or she must fill the items with his/her answer and confirm them, in order to receive their automatic evaluation. The automatic evaluation for exercises in iGraf is based on the comparison of values provided by the student against the correspondent ones provided by the exercise author.

FIGURE 3  
ANSWER MODEL CONFIGURATION PANEL

In *numerical answers* that include real numbers and ordered pairs (item 1 and 2 of Figure 3), iGraf compares the value registered in the answer model with the value provided by the student. If the item is *interval* (item 3 of Figure 3), it is possible to represent open, closed or mixed intervals, just using brackets “[” and “]”. For instance, the closed interval from -1 to 3 must be represented as [-1.3]. To open an interval end, just a click on the corresponding bracket and its *direction* is changed. A click on the left bracket, for example, would result in ]-1.3]. Thus, the evaluation method for this type of item compares the bracket and the values considering the student answer and the author model.

The third type of item, algebraic *expressions*, is a little more complex, since iGraf must accept as correct any variant of the same expression. For instance, if the response model is  $2*x+x^2$ , it is perfectly right the answers  $x*x+x+x$  or  $3*x+x*x-x$ . So, an heuristic was adopted to compare the expression of the student's answer,  $A(x)$ , with a corresponding template registered for the item,  $G(x)$ , for every value of  $x$  in the range defined by the visible portion of the abscissa axis on the screen. The item is evaluated as correct if, and only if in no more than 5% of the cases the difference  $|A(x)-G(x)|$  is less than  $\xi > 0$  (this is an acceptance parameter, for now, internally defined in iGraf).

EXPERIMENTS AND RESULTS

Aiming to analyze the didactical use of iGraf, three experiments were carried out, all of them in a computer lab, one participant per computer, none of them having previous knowledge about iGraf. At the beginning of each experiment its dynamics was explained to participants. It was like: presentation of the basic features of iGraf; a period of time was fixed to accomplish a series of activities using iGraf; and the last half hour was used to fill out a questionnaire. The final questionnaire were the same for all experiments, focusing the usability of iGraf. All questions were of multiple-choice, for easy quantification. Questions were related to: Q1) *visual appearance*, Q2) *simplicity of use*, Q3) *quality of graphics*, Q4) *amount of features*, Q5) *simplicity of the menus*, Q6) *usage of the automatic assessment feature* Q7) *usage the message bar*, Q8) *help interface*, Q9) *satisfaction of the teacher/student in general*.

I. Experiment 1: math teachers in 2008

25 math teachers took part of the first experiment, occurred in January of 2008, during the summer program of IME-USP. The experiment last 1 day: 2 periods of 4H. It was used the version 3 of iGraf, integrated with SAW. In the first 4H, they were asked to do about 20 exercises involving simple concepts. The first exercise was to draw the graphic of affine functions  $f(x)=x+a$  (for different values of  $a$ ) and the last was to draw the derivative and integral for  $f(x)=\cos(x)*x$ . During the last 4H they used the stand alone version of iGraf to construct their own examples and exercises and the final activity was answering the questionnaire. The authoring activity was free, without the need of registering their tests.

II. Experiment 2: math teacher candidates in 2010

24 students from the Federal Institute of Education, Science and Technology (math teacher candidates) took part of the second experiment, in 2010. There were used two sessions of 4H each, in 2 days. The script were the same of the first experiment, except by the number of exercises in the first 4H session (14 instead of 20 exercises). Also, in the authoring session they were required to produce one exercise for elementary courses, with a clear statement. This experiment was considered as part of the activities of their regular undergraduate course and they got a grade for it.

III. Experiment 3: math teachers in 2011

The last experiment occurred under the same basis of experiment 1, but now using the iGraf integrated with Moodle through the iAssign package. There were 15 math teachers, 12 of them answer the final questionnaire. The results of these experiments are indicated in figure 3.

Discussion

Each question had 5 options, following the Likert scale, as shown in Table 2.

TABLE 2  
LIKERT SCALE FOR QUESTIONS

| Very bad | Bad | Regular | Good | Very Good |
|----------|-----|---------|------|-----------|
| 1        | 2   | 3       | 4    | 5         |

In order to grade the questionnaires results from 0 to 10 to facilitate their analysis, weights were uniformly assigned to each answer. Compilation of results are shown in Figure 4. To analyze data from the questionnaires, we separate the views of teachers (experiments 1 and 3) and students (2). Analyzing data from the first group we see that teachers, in both experiments, quite liked the visual solution of the program (Q1, Q3), but they think it should be easier to use (Q2). They also liked the amount of tools (Q4), the clarity of the menu items (Q5) and using the system for automatic assessment (Q6). The level of acceptance of the program among teachers was therefore quite satisfactory.

The students (experiment 2) liked the system, in general, but they graded questions 2 and 6 significantly smaller than teachers in experiments 1 and 3. Reason for this could be the level of difficulty between experiments 1-3 and 2. In experiment 1 and 3, the same script was followed, there wasn't the obligation of authoring exercises and their scores in the final questionnaire was very similar. In experiment 2, despite following the same script of experiments 1 and 3, there was the obligation of authoring an exercise and they were graded for that. But it is worthwhile the significant smaller grades they gave in question 2 (*simplicity of use*) and 6 (*usage of the automatic assessment feature*). These questions are related to difficulties to deal with exercises.

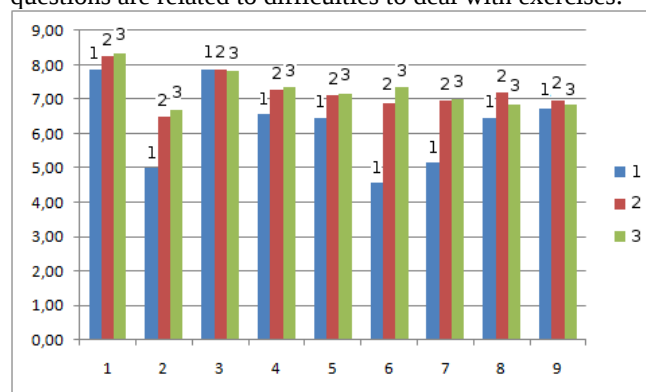


FIGURE 4  
RESULTS FROM EXPERIMENTS 1, 2 AND 3

CONCLUSION

This paper presented the iGraf system, a free software to support the teaching and learning of functions and related issues. iGraf can be integrated to LMS, particularly it can be used with Moodle through the iAssign package. Three experiments using iGraf were described and their results presented, two of them with math teachers and one with math teachers candidates. iGraf was well evaluated concerning several aspects related to its usability.

An interesting observation emerged from these experiments: the level of difficulty on using it to solve exercises and to produce digital content. These result in a significant difference when considering questions 2 and 6. Analyzing that, we noticed that the difficulty of authoring an exercise could be the explanation of the difference. In fact, to produce educational content is not an easy task, even without using computational support.

In the next year iGraf will be tested in real classes of elementary and high school. Considering the iGraf features, its automatic assessment tools will be improved by the incorporation of intelligent tutoring techniques. A long term goal is to implement 3D graphics in iGraf.

#### ACKNOWLEDGMENT

Leônidas O. Brandão is partially supported by FAPESP.

#### REFERENCES

- [1] Hara, N; Kling, R. Students' Frustrations with a Web-Based Distance Education Course. *First Monday: Journal on the Internet* 4(12), 1999.
- [2] Hentea, M., Shea, M. and Pennington, L. *A perspective on fulfilling the expectations of distance education*. Proceedings of the Conference on Information Technology Education, 2003, 160-167.
- [3] Søndergaard, H., Thomas, D. *Effective Feedback to Small and Large Classes*, Frontiers In Education, 2004.
- [4] Moura, J.G., Brandão, L.O. and Brandão, A.A.F. A web-based learning management system with automatic assessment resources. 37th Annual Frontiers in Education Conference, Milwaukee, WI, EUA, 2007.
- [5] Rodrigues, P., Brandão, A.A.F. and Brandão, L.O. Interactive Assignment: a Moodle Component to Enrich the Learning Process, 40th Annual Frontiers in Education Conference, Washington, DC, EUA, 2010.
- [6] Lu, Y.A. The use of Mathematical Software in Secondary Mathematics Teaching: a New Challenge for Young Teachers, 11<sup>th</sup> International Congress on Mathematical Education (ICME) 2008.
- [7] Stewart, James, *Calculus – Early Transcendentals*, 4<sup>th</sup> ed, 2001
- [8] Ávila, Geraldo, *Cálculo I: diferencial e integral*, LTC,1978
- [9] Kayan, F. A study on preservice elementary mathematics teachers' mathematical problem solving beliefs, Master Thesis, Graduate School of Social Sciences, Middle East Technical University, Turkia. 2007.
- [10] Dahl, B. Analysing Cognitive Learning Processes through Group Interviews of Successful High School Pupils: Development and Use of a Model. *Educational Studies in Mathematics*, Vol. 56, No. 2/3, pp. 129-155, 2004.
- [11] Abu, N.A., Hassan, M.D., Sahib, S. *Mathematical animations: the art of teaching*, Frontiers In Education, 2001.
- [12] Dan Ginsburg, Brian Groose, Josh Taylor, Prof. Bogdan Vernescu (Advisor), The History of the Calculus and the Development of Computer Algebra Systems, Computer Algebra Systems, Chapter 5, available at [http://www.math.wpi.edu/IQP/BVCalcHist/calc5.html#\\_Toc407004380](http://www.math.wpi.edu/IQP/BVCalcHist/calc5.html#_Toc407004380)
- [13] Java Programming Language, <http://www.oracle.com/technetwork/java/index.html>, 2011.
- [14] Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Design patterns – elements of reusable object oriented software* Addison Wesley Longman, Inc. 1995.
- [15] Java Components for Mathematics (JCM) <http://math.hws.edu/javamath/> 2001
- [16] ISO 639 <http://ftp.ics.uci.edu/pub/ietf/http/related/iso639.txt>
- [17] ISO 3166 <http://ftp.ics.uci.edu/pub/ietf/http/related/iso3166.txt>
- [18] Goulart, Linda T, O desafio de universalização do ensino médio, <http://www.inep.gov.br/imprensa/artigos/2006/universalizacao.htm>, 2006
- [19] Pinto, José M. R., O Acesso a Educação Superior no Brasil, *Revista Educação e Sociedade*, v. 25, 2004, 727-756
- [20] Associação Brasileira de Educação a Distância, Anuário Brasileiro Estatístico de Educação Aberta e a Distância, 2007
- [21] Kislenko, K. and Grevholm, B., *The Likert Scale Used in Research on Affect - A Short Discussion of Terminology and Appropriate Analysing Methods*, IEEE Internacinal Conference On Multimedia and Expo, <http://tsg.icme11.org/document/get/415>, 2008.