

Experimental Comparison of the Implementation of MVC in Java and C#

José David ALANÍS URQUIETA, Guillermo SÁNCHEZ FLORES,
Division of Technologies of Information and Communication, Technological University of Puebla
Puebla, Puebla 72300, México

Mariel Pamela MORALES RIVEROLL, Eduardo LÓPEZ MENDEZ
Master's Degree in Systems Engineering and Computer Intelligent, Postgraduate Department,
Technical University of Puebla, Cholula, Puebla, 72640, México

Bermúdez Juárez BLANCA
Computer Science Faculty, Autonomous University of Puebla
Puebla, Puebla, 72540, México

ABSTRACT

In this work it presents of experimental way a comparison of the implementation of design pattern MVC (Model View Controller), in where it is performed a simulation of MEMS accelerometer-type (Micro Electromechanical Systems by its acronym in English) with visualization, since it has a set of data coming from of the numerical solution of the type bi laplacian partial differential equation solved by the finite differences method, that represents the material's deformation with the device has been built.

In this problem it has evaluated the aspects of usability, functionality, portability, scalability and accessibility of the implementation of MVC in Java and C#. The OpenGL API (Application Program Interface) it can be implemented in many programming languages without changes neither important adaptation, this make possible to evaluate the applications of generic way.

The convenient use of the design pattern in the implementation of a tool to simulation with visualization results interesting of evaluate, moreover it expected obtain more than an absolute opinion, a rule to allow the evaluation of another design pattern to solve similar problems. At last this experimental process is underway already, however it expected to help in the choice of programming language to other simulations of MEMS devices

Keywords: Experimental Comparison, MEMS, MVC, Java, C#, Design Patterns.

1. INTRODUCTION

The implementation of MVC is related with the simulation with visualization made of a MEMS accelerometer-type, by physically based modelling as simulation technique as it can see at [1] [2]. The implementation consist basically in take a set of data that it has, and implement a simulation with graphical representation in the API OpenGL, of generic way, following the design pattern MVC and its rules, using as language of programming Java as the C#.

In order to obtain applications with a good design and structured, it is can use a convention, this is the correct use of models, views and controller, this usefulness and meaning of the pattern of design of MVC.

Each component has a special function, the controller answer to the calls or events and if is necessary ask information to the model and then it shall pass the data to the view to be visualized. The model, by other part, is defined as management the business logic and it does not implement no interface or well it can be

derivate from another class. Despite what above mentioned the rules are not absolutes or strict, it is considered this as a basic mechanism of work of the MVC pattern of design [4].

The importance of use the design pattern MVC lies in the advantages that it offer, among others it is find: the independence of code, the order in the distribution of the project, the reusing the classes, the planning of each one of the layers of isolated way to avoid coupling, so as the adoption of good practice of software development, that use the principle of design **soe** (separation of concern) that separate a program in different sections or topics to solve, etc.[2][4].

The objective of this job given the implementation of MVC in two different languages Java and C#, evaluate the aspects that allow to obtain a rule in order to evaluate of another design pattern to solve similar problems, taking in account the experimental focus.

The document organization is as follow, in the section 2 it describe the theoretical framework or state of art about the specific concepts necessary to understand the design pattern, the evaluation way and the API of OpenGL in brief way.

Then in the section 3 it is exposes the experimental comparison between the both applications made in Java and C, by the evaluation of the aspects afore mentioned such as portability, usability, among others, moreover the difficulties that appear in the building of the software.

In the section 4 it describe the results about the evaluation made in the last section and it try to obtain a general rule that it allow measure another type of related applications with numerical simulation and visualization.

By last the section 5 exposes the conclusions around the evaluation, the expertise gained, the knowledge obtained and the future works on the same research line.

2. PREPARATION OF PAPERS

This section describes the tools and methods used in the construction of the application mentioned in past section.

For this case it has three principal elements: MVC design pattern, OpenGL as a graphical API, and Software Engineering parameters to be measured.

2.1 MVC

In order to take advantage of solutions already tested in the practice and promote the reuse of code and additional flexibility in the construction of software.

Due to this the applications cited in the section 1 have been programed with this design pattern as it has shown in [2].

The design patterns are descriptions of class and objects related that are particularly specialized to solve a problem of general

design in a determinate context [2]. There are some categories of design patterns classified as follows: creational, structural and behavioral. The construction of a software with a specific purpose such as has described in the past section was been made using as an implementation rule of a pattern design called MVC (Model View Controller) [4].

The MVC is a design pattern into the category of creational patterns that in the construction of software solutions used by first time in 1979 to Small Talk and now is present in a lot of frameworks with multiples focuses as Java Swing, ASP .NET, GTK, Ruby on Rails among many others[4][2].

The proposal to use the design pattern cited emerge in order to construct a system where an interface of user is required and of the need to create software robust, with a life cycle very adequate, great power of maintenance, reuse and separation of concepts. To create software robustness with a life cycle much adequate, where power the easy to maintenance, software reuse and concept separation [4].

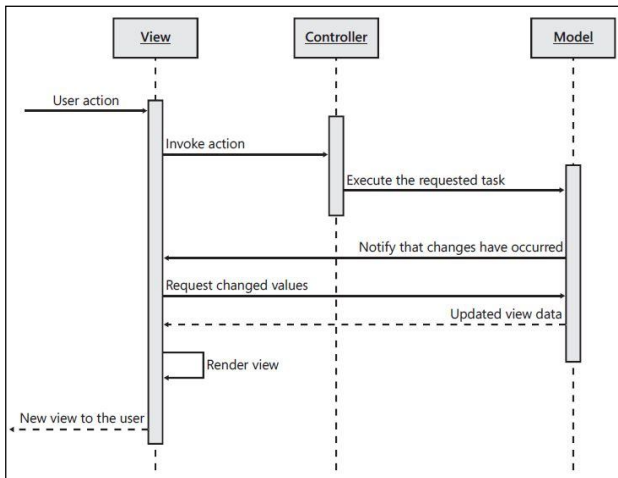


Figure 1. The sequence diagram of the Original MVC pattern [4]

In the figure 1 it shows a way as the functioning of the MVC pattern, then it is explain this.

The code is separate in three different layers, bounded by their responsibility, that they are called: Models, Viewers and Controllers [2] [4].

- **Models:** This is the layer where the data is worked, thus this layer contains the mechanism to access to the information and too update its state.

- **Viewers:** as its name says contains the code to show in the application that it produces the visualization of the interfaces of user, this is, the code allow make rendering of the state of application. To abbreviate this layer show the output of the system.

In way general in the view the data are worked, however, does not make access to the data. The views will ask data to the models and hers will generate the output as required by the application.

- **Controllers:** it contains the necessary code to answer to the actions requested in the application, as visualize an element, make a buy, look for information, etc.

In fact is a layer that serves of link between the viewers and models, it answering to the mechanism that can requested for implement the needs of the application.

2.2 OpenGL

OpenGL write originally in C, is a standard specification that defines a Multilanguage API and multiplatform to write applications that produces 2D and 3D graphics. It developed by Silicon Graphics Inc. (SGI) in 1992 and it widely used in CAD, virtual reality, scientific representation, visualization of information and flight simulators [2]. The use of OpenGL becomes a standard in the industry of the computer graphics by the capacity to adapting to the hardware and existent platforms. The order of execution it given as follows [2]:

- It builds the geometry primitives (dots, line, polygons, bitmaps, etc.)
- It positioned in the 3D space
- It selected the point of view
- It constructs the color of the objects (textures, conditions of the light indicated by the programmer)
- It generate the image of pixels (Calculus of Visibility)

2.3 Software Engineering Parameters

In this case the characteristics will be measured in both applications are usability, functionality, portability, scalability and accessibility, non-limiting as the aspects that it is considered in order to find a rule to evaluate this type of applications or systems.

Usability

Firstly the concept of the usability it is refers to the “user-friendliness” even this term is so very subjective tries to quantify the user-friendliness, whenever the functions that is performed are valuable, this characteristic can be evaluated by four aspects[7] as follows:

- Physical and intellectual skill required to learn the system
- The time required to become moderately efficient in the use of the system
- The net increase in productivity
- Subjective assessment of user’s attitudes toward the system.

Functionality

The next concept the functionality can be described as the result of the assessment by the evaluating the feature set and capabilities of the program, the generality of the functions that are delivered, and the security of the overall system. This means the skill of the system to protect itself against external attacks. In contrast the failure of the security it drives to loose of availability, data damage or leakage of information by non-authorized persons [9].

Even the security aspects are so very hard to measure those aspects are linked with the functionality.

Portability

The portability is in general de ability of one system to transposed form one environment to another [7].

Even the portability characteristics involve the use of the program on another system with reasonable cost and effort.

If this parameter is more than writing a different version of it in some aspect, definitely the system is not portable [5].

The sub attributes of adaptability, install ability conformance and replaces ability, is considered non-relevant for the systems rather, the end user perspective it is adopted in terms that means [6]:

- The Software can be moved across a wide range of systems with minimal changes.
- The program works with other applications on local or remote systems
- Users require little or no retraining on the program

The last point does not apply to this analysis since to the specification of the software is strong definite and it treat about one software with a specific purpose.

Scalability

The concept of scalability, defined as the ability to modify the configuration or size of the system in order to adjust to the change, [3] hence worth to measure this concept. The systems can be a network, a software application, among other, to increase the amount of work without lost the fluency and conserve the quality of the services.

The scalability process can be observed as a capacity of the systems of to increase its resources (hardware or software) and in consequence the performance of the systems improves in the same proportion than the increase aforementioned [3] [7]

In spite of this characteristic, as it is known, it results difficult to measure, due to the properties of the software and the system in general [7].

Accessibility

The accessibility of the systems is considered as the capacity of the software or system to be accessible by the greatest number of people possible, regardless of capabilities, over all the vulnerable groups. Exists a world initiative that is carried out by W3C (World Wide Web Consortium) to facilitate the universal accessibility [8].

The rules or requirements for the creation of accessible products in order to improve the quality of software, and taking in account to facilitate the access to the electronic products, among other the aspects that it is worth to mention are: No use text as images, it incorporate alternative phrases, it allow to increase or decrease letter size, images as references, among others [11].

In these days the concept of accessibility it is referred to the web systems in general, since the systems produced are oriented to this class of platform, in addition to be considered it a trend in technology [11]. The quality of the software has dependence with this aspect as it was already aforementioned, given that it is tried to make affordable the systems [7] [8].

For these both particular applications the access to the software are total and it is considered adequate for the type of visual application, for its driving, management, maintenance, etc. , since the visualization comes from a particular numerical simulation problem.

Security

Although that the systems are constructed with a specific topic or specific proposal, is very important to say that the security aspect must be considered in all application's build, as know, it is says that the system are secure if it is free of warnings or risk taking in a count the management of the information that it has or contains it [8] [9].

There are a lot of aspects around the concept of security in the systems, but the definition above mentioned considers the follows issues [9]:

- Integrity
- Confidentiality
- Availability
- Irrefutability

So that the indicators of security for these systems in particular, they must be described, not only as part of the software build, but involves all the aspects in the life cycle of the software [7]. In general for these both applications performed it is very important to consider the recovery of the system in case of disasters or contingency, as a hot key point, since in these days the vulnerability of the systems and the attacks are a lot spread

3. IMPLEMENTATION AND EXPERIMENTAL COMPARISON

In this section the experimental comparison between the implementation performed in Java and C# is presented. Moreover the difficulties in the implementation and the difficulties that appears in the building of the software.

3.1 Implementation of the Simulation with Visualization in C# and Java

As it is can see in [1] [2] as the theoretical support, it is treated of simulation of a MEMS accelerometer-type. MEMS is a non-deformable plate it can be square or rectangular, it subject to an extreme or in all its extremes. In this case, it is takes an accelerometer to subject by all its extremes and that it receives an electrical impulse in order to make it vibrate, in such a way that this impulse provoke the deformation of the material in the axis z, this is the main subject of study for almost all the MEMS.

In order to make this simulation with visualization, it is used a mathematical model that governs the deformation of the MEMS accelerometer-type, which it is: [1] [2]

$$\frac{\partial^4 \omega}{\partial x^4} + 2 \frac{\partial^4 \omega}{\partial x^2 \partial y^2} + \frac{\partial^4 \omega}{\partial y^4} = q \quad (1)$$

Where ω the deformation of the material and q is the intensity of the force applied to the plate. The equation (1) is called bi laplacian partial differential equation; due to it has a laplacian square operator. This force it is described in terms of the follow equation:

$$q = \frac{q_0}{D} \sin\left(\frac{\pi x}{a}\right) \sin\left(\frac{\pi y}{b}\right) \quad (2)$$

Where q_0 is the force in the center of the plate, the parameter D, is the rigidness of the material in opposition to the flexion and it is determined by [2]:

$$D = \frac{EI}{(1 - \nu^2)} \quad (3)$$

From equation (3) it is obtained the follow meaning [2]:

E: is the Young's modulus or traction's modulus, is a measure of characterization of the materials that measure the degree of rigidity in an elastic material.

I: is the moment of inertia, this is , the resistance of a material to the flex, in accord with this, the major moment of inertia is given before the material it is folded completely.

v: is the Poisson's ratio, that is the relationship of the contraction with transversal to the axis that exerted the tension or impulse

For this problem, it is used the follows parameters, in order to obtain the solution of the differential equation, it is worth mention that is not a results of others simulations, but rather comes from the experimental job in the ICBUAP, and below it is presents in the Table 1 [2].

	Material Silicon
Young's modulus	13×10^{10}
Inertia moment	2330
Poisson's radio	0.28

Table 1. Initial measures for the calculus of parameters D in the right side of the equation (1) [2]

Given the previous equations now it is shown the initial conditions and of border of the problem in question, firstly it is known that the accelerometer it is fastened by the four extremes, which describe the following border conditions to a plate simply supported [2].

$$\omega = 0, \quad \frac{\partial^2 \omega}{\partial x^2} = 0 \quad \text{para } x = 0 \text{ y } x = a \quad (4a)$$

$$\omega = 0, \quad \frac{\partial^2 \omega}{\partial y^2} = 0 \quad \text{para } y = 0 \text{ y } y = b \quad (4b)$$

In this case the dimensions of the sides are denoted by a and b, then the extremes are x=a and y=b, as it is said the last denote that the plate are simply supported and subject in its extremes: The border conditions come given as follow [2]:

$$\omega = 0, \quad M_x = 0 \quad \text{para } x = 0 \text{ y } x = a \quad (5a)$$

$$\omega = 0, \quad M_y = 0 \quad \text{para } y = 0 \text{ y } y = b \quad (5b)$$

Where M_x and M_y are the moments of flex that it is observed when the material is subjected to the force that already mentioned, in the sense of the z axis.

As it can see at [1] above mentioned equation has been solved by finite difference method, and performed the numerical simulation as it can be seen in [2]. Once it made this, the numerical simulation it was used in order to build a simulation with visualization, one implementation was performed in Java as it can see in [2] the another implementation was performed in C# for this work, in order to establish a comparison between two implementation, both following the MVC design patter. So the next table shows implementation in C# of simulation with visualization from the numerical simulation.

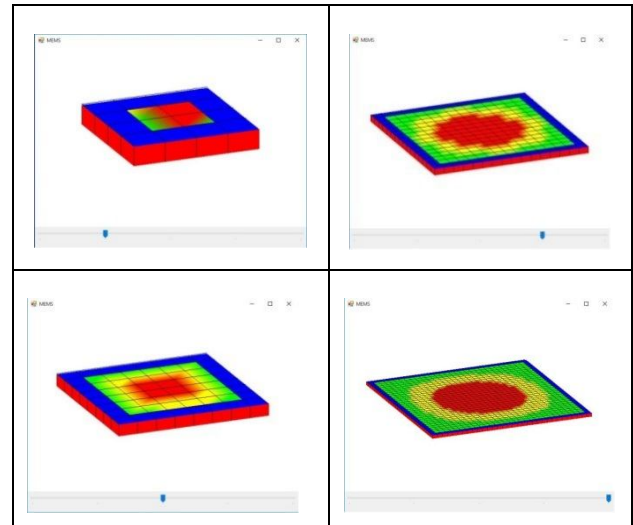


Table 2 Table of the implementation of simulation with visualization in C# by JOGL for 4,8,16 and 32 elements of size of step.

Now it presents the implementation using Java of the same problem, the simulation with v

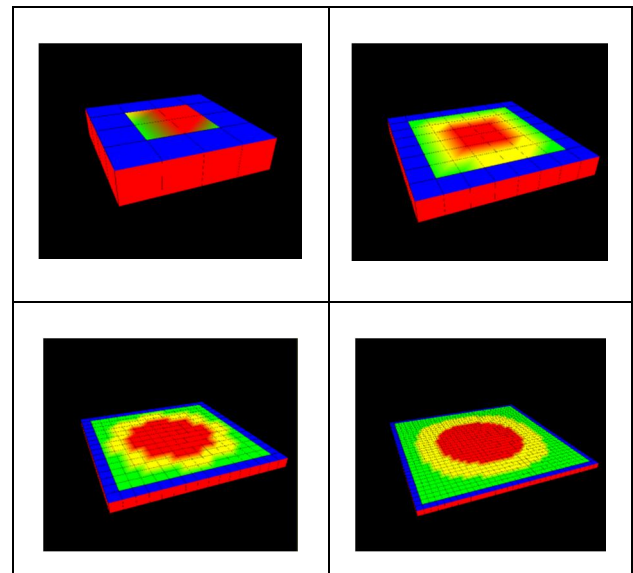


Table 3 Table of the implementation of simulation with visualization in Java by JOGL for 4,8,16 and 32 elements of size of step [2]

3.2 Experimental Comparison of Simulation with Visualization

Since to both problems has been developed following the rules of design pattern called MVC, with the same tool in order to visualize the results of numerical simulation, is possible to determine one comparison taking in account the concepts mentioned in the last section as follows:

Usability

Both systems have the same level of usability, taking in consideration the aspects of the skills that is necessary for domain the use of the system.

Moreover as it was said in the above sections a software was performed for users with expertise and background in the topic, due to this the usability level is high, in fact the software complied with the all needs of the users by now, therefore exists a subjective effective use of the system, independently of the tool of development (C# or Java).

Functionality

A brief analysis of the characteristics of the problem allows concluding that the system has a good level of functionality, given it the requirements asking by the specialized users.

In general, in both applications have the security's measures in order to prevent the intrusion of non-authorized users or persons oblivious to the system. In addition to the already mentioned, it is very important to say that both programming languages have themselves ways to protect the classes against the possible loss or leakage of information.

Portability

In this point definitively the Java's application has an advantage, but not so much, in comparison with the C's application. As it is known, while the JVM (Java Virtual Machine) is installed in a device, machine, or system, the portability is 100% ensured in order to run the programs. In contrast the C# application only can run into platform with Windows. Despite what has been said, it is not difficult overcome this disadvantage, in these days exists a lot of software for simulate the environments, among other solutions to overcome this disadvantage in the use of C#'s application

Scalability

Given that the applications and its processes don't use a large amount of memory in order to perform the visualization of data that it come from the numerical simulations, in this particular problem, the limit uniquely is established by the memory of the machine.

Both applications are able to increase the number of data almost indefinitely and their programming style has data structures adequate in order to support this class of solutions. As in past sections has said is difficult to measure this aspect, but in this case the solution has been bounded by the provided data of the numerical simulation, and this point was considered in both applications.

Accessibility

For the system is very important to take in consideration this topic, in order to improve the quality of the system and in general for the applications of render. The normative is almost non-existent for this class of problems, however, both software meet with the standard to make an adequate visualization, as it is can to observe in the system the parameters such as the perspective, colors (this point has not been specified which colors have to be used, but in the major of the applications basic colors are required), redimensionable window, among others.

The other aspect this visualization complies in adequate way is the use of the GUI (Graphical User Interface) to try to guarantee the most accessibility of users, for all class, even novice or experts in both applications.

The software can to run with a minimum of effort and the handled of the information is possible without complications. By last, the access to the application is considered easy.

The changes into the GUI or in the presentation of the simulation with visualization do not represent an obstacle significantly hard to overcome. In addition, despite of these applications were made for a special type of users, however, in both software the use of software is not exclusive for this type of users, on the other hand, tries to be an inclusive system,

Security

The aspect of the security in both applications is covered, in two aspects, in one hand, the applications were built with the paradigm of object oriented programming, and the OpenGL API, these tools have the capacity to hide the details that are not necessary in order to access to them. In other hand the program was developed into a window derivate of the Swing as GUI. Even if the application can be run into Web Browser there are not problems with the aspects of security, neither with of its four aspects. Somehow the application can be protected of ad hoc way in order to consider the users and its needs in easy way and almost without effort. In addition in this case does not matter the platform or in which place the application can be run, the security was developed careful in this sense in order to ensure the efficiency and effectiveness in this particular aspect and thanks to the paradigm of programming.

Trouble and Difficulties in the Implementation: Special Issue

In the implementation the team face with the follows troubles and difficulties:

- The implementation with C# with OpenGL for C# does not have allowed putting a Canvas object in the View component.
- The components cannot be built in C# application due to the rules in the develop of applications in .NET
- The data structure Array List does not exist in C#, so it is necessary to construct own structure to substitute it
- Adapting some properties of the frame, in C# to the methods such as repaint, or delete the pictures or image.
- The implementation of Java is so hard in the same point above mentioned.

4. EXPERIMENTAL RULE ABOUT THE DEVELOPMENT OF THE APPLICATION

Given all it previously mentioned and taking in consideration the aspects on the construction of the software, in this case with very specific requirements, it is concluded that:

The use of a design pattern offers the facility in order to build an application without taking in account the software platform. Maybe exists aspects for adapted when it is change of platform, but these efforts are not significant.

Exists certain "convergence" in the solution since the GUI is very similar in the two platforms performed, however, the process results are found almost without problems in the part of the implementation.

The structure of data that are own of the programming language, by the moment, result adequate in order to handle the security and the others topics that is needed in this type of application. Perhaps with another type of structures or structures made by the programmer instead programming language, can change the behavior of the application.

5. CONCLUSIONS

The experimental comparison of the implementation of design pattern MVC was presented, starting from the solve the problem of the simulation of MEMS accelerometer-type with visualization, taking a data set from numerical simulation of the behavior of the material's deformation of the device, represented by the numerical solution of one partial differential equation called bilaplacian.

An evaluation of the aspects such as usability, functionality, portability, scalability, accessibility and security was generated; moreover it was presented of the special topic about the troubles and difficulties that involve the implementation of the MVC in C and Java with the API OpenGL, in the simulation with visualization of the problem already mentioned.

For this particular problem was obtained an experimental rule in order to know how much effort could result the construction of another applications with the same characteristics, and was found that these efforts are not significant, in consequence the use of the design pattern results convenient.

The experience that it is obtained in this experimental evaluation can be used in another pattern derivate of MVC like Strategy pattern, observer pattern among others, or in the simulation problems like this with another conditions or change of the parameters.

By last, the election of the programming language that will be possible to make in the future, to perform simulations with visualization of MEMS devices does not affect the process for this class of applications.

9. REFERENCES

- [1] Alanís, José D., Bermúdez, Blanca, Hernández Rebolgar, José L. Solución de Ecuaciones Diferenciales tipo Bilaplaciano para la Simulación de MEMS tipo Acelerómetro, CICIC 2015, IIS, 2015
- [2] Alanís José, Morales Mariel, López Eduardo, Bermúdez Blanca, Visualización de MEMS tipo acelerómetro mediante JOGL, CCADET UNAM, Guatemala, 2016, ISSN 2395-8499
- [3] ARQHYS. La escalabilidad. Revista ARQHYS.com.2012, 12. Obtenido 03, 2017, de <http://www.arqhys.com/construcciones/escalabilidad.html>.
- [4].Esposito, Dino, Programming Microsoft ASP.NET MVC, Microsoft, USA, 2010
- [5].Ken, Garen, Software Portability: Weighing Options, Making Choices, CPA Journal, Vol. 77, No. 11, November, 2007
- [6].Mooney, D. James, Bringing Portability to the Software Process, West Virginia University USA, 2000
- [7].Pressman, Roger, Software Engineering: A Practitioner's Approach (Irwin Computer Science), McGraw Hill, USA, 2015
- [8].Ronda León, Rodrigo, Libro de Asignatura Tecnología II Productos Electrónicos: Principios y Pautas, Universidad de la Havana, Cuba, 2005
- [9].Somerville, Ingeniería de Software, Pearson Education, Mexico, 2011
- [10].Toni Granollers, Jesús Lorés, La ingeniería de la Usabilidad y Accesibilidad aplicada al diseño y desarrollo de sitios web, Universidad de Lleida, España
- [11].Trueba Irma, Los requisitos de accesibilidad en un proyecto de software. Implicaciones de usuario discapacitados en el proceso software, Tesis de Doctorado, Universidad Politécnica de Madrid, España, 2005