XXXVI Ibero-Latin American
Congress on Computational
Methods in Engineering
CILAMCE
Rio de Janeiro, 22-25 Nov 2015

AN OBJECT-ORIENTED COMPUTATIONAL PLATFORM FOR FINITE ELEMENT METHOD IMPLEMENTATION

Daniele M. S. Paulino

dani.mspaulino@gmail.com

Federal Institute of Maranhão, 65030-005, Maranhão, Brazil.

André S. Muller

andre.muller@ifma.edu.br

Federal Institute of Maranhão, 65030-005, Maranhão, Brazil.

Abstract. *Computational mechanics solves engineering problems by modeling a structure and using a numerical method implemented through a computer system. Using this concept, this research aims to develop a robust computational platform to structural analysis supported by the Finite Element Method (FEM), which has been established as the dominate discretization method in the Computer Aided Engineering (CAE) technology. The platform has its foundation on an Object-Oriented Programming (OOP) language, Java, allowing strong programming techniques such as Design Patterns and Unified Modeling Language (UML). Concerned to code efforts, those approaches provide an easier program to modify, supporting the implementation of different types of analysis, such as dynamic, plasticity and non-linearity, meeting the objective of this framework: be an academic tool for civil engineering analysis. Moreover, it allows the user to introduce a new application or a new FEM formulation using a format very close to their corresponding mathematical descriptions. The use of software in the industry is widespread, however the engineer has a responsibility to interpret and check computer results. Hence, designing a computational platform and learning how to apply the mathematical concepts to its scope can be very helpful to undergraduate courses.*

Keywords: *Computational platform, Objected-oriented programming, Finite element Method*

1. INTRODUCTION

The evolution of programming techniques and software developing allowed the use of programming languages to solve real-world problems faster and with more precision. The Objected-Oriented Programming (OOP) has been established, in the last years, as a powerful development model, which it can be decomposed into a complex system by objects and then use the resulting structure as framework. This technique takes advantage of important concepts: encapsulation, hierarchy, abstraction and modularity. Learning how to use those elements will help us to write a natural, reliable and reusable software.

While trying to achieve an OO design, a significant effort in the process, not only the final product, must be considered. For this reason the phases so-called Objected-Oriented Analysis (OOA) and Objected-Oriented Design (OOD) play a huge role in Objected-Oriented Paradigm. The preliminary analysis aims to identify the problem and then obtain the system's objects, thereupon during the design phase the relationships among system components are established.

Research has been conducted during the last years in order to provide techniques that could define a solid process to allow the development of engineering software. Hence, in order to afford standards and assist the designer to target a strong Object-Oriented structure, methodologies like Design Patterns (DP) and Unified Modeling Language (UML) were developed.

The objective of this work is to present the conceiving process of an ongoing finite element analysis platform GoFEM, based in the concepts of object-oriented paradigm. The architecture proposed has as purpose a flexible, robust and simple platform for FEM teaching and research activities.

2. METHODOLOGY

When it comes to adopt an OO approach, a good amount of languages are available, such as Object Pascal, C++, Ada, Eiffel and Java. The Java language was chosen for the implementation of GoFEM since presents a solid OO foundation and offers a complete set of components for Graphical User Interface (GUIs).

According to Cook (2002), "solving a practical problem by Finite Element Analysis involves learning about the problem, preparing a mathematical model, discretizing it, having the computer do calculations, and checking results". The different types of problems involving FEM (e.g. linear static, linear dynamic, nonlinear analysis and buckling), the model decomposition in nodes and elements, the numerical analysis, corresponding to the computation of element vectors and matrices required to assembly the global vectors and matrices that will be used by analysis algorithms, all represent aspects that characterize the object-oriented approach appropriate for FEA.

The overall organization of GoFEM platform will be discussed afterward.

2.1 Definition of packages

Aiming a clear and well-formed development structure, the platform was decomposed in four packages: the graphical interface package (ggui), the mathematical support package (gnumlib), the numerical method package (gfem) and the analysis package (ganalysis).

2.2 Definition of objects and classes

The principle applied to the development process was the domain analysis, one type of Object-Oriented Analysis, which focus on the identification of classes and objects that are common to all applications within a given domain.

All the important requirements to have a well-formed OOA were applied, such as building CRC (class – responsibility – collaboration) diagrams and use cases.

- **Package ggui.** This package represents the interaction between user and the platform. The pre-processor classes are GInput and GBasicInfo. The class GGUI is responsible for correlates the package ggui with the others, executing the analysis. A design pattern, Facade, was used to simplify the GGUI, becoming easier to use. The graphical implementation of this package will be developed in the future, improving the platform.

- **Package ganalysis.** During the process, different types of analysis can be executed, such as linear, nonlinear and time-dependent. A superclass, GBVP, controls the access to those classes, applying the concept of inheritance. Furthermore, this package is responsible for the representation of the problem model, both geometrical and physical. A class GModelBVP contains all this information.

- **Package gfem.** This package represents the FEM abstraction, modeling the real structure and discretizing it into finite elements. The first step in the conception of this package is to identify all the necessary *methods* comprehending the analysis. There are three main classes in respect to the mechanical model*: the model, the element and the nodes*.

- **Package gnumlib.** This package contains classes that will be used by other packages, which represent the mathematical tools necessary throughout the software process, either for the numerical method or to simplify mathematical operations of data.

2.3 Interaction between classes

During the OOA e OOD phases, the use of abstraction and hierarchy concepts, as well as other elements that characterize the object model, aims the development of a strong interaction between classes and objects. This arrangement was established through design patterns components, such as *Factory* and *Concrete Factory*.

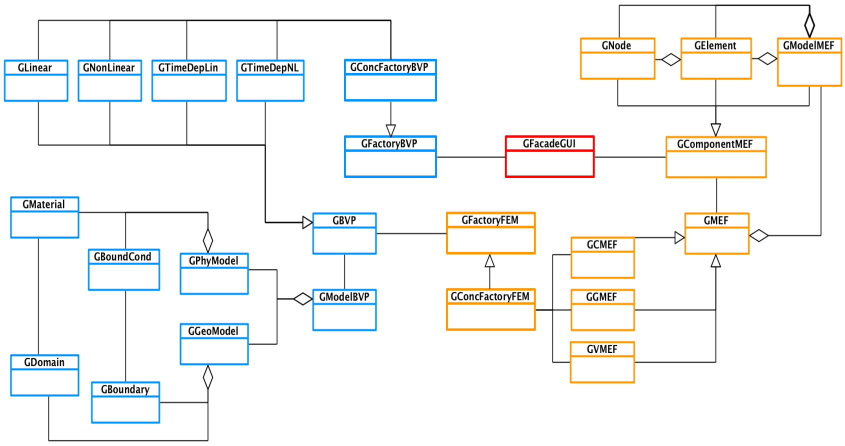

Figure 1. Relationship between packages gfem and ganalysis.

3. RESULTS

All these concepts applied during the state of art allowed the result of a clear, robust and efficient structure, aiming future modifications. Currently, the GoFEM performs only linear-elastic analysis. In order to demonstrate the working process, a linear-elastic problem can be used to validate the results, through Ftool, a structure analysis software designed by Tecgraf/PUC-Rio.

The following figure presents the simple frame model, with the loads and boundary conditions (left), and the resultant reactions values (right), as displayed by Ftool.

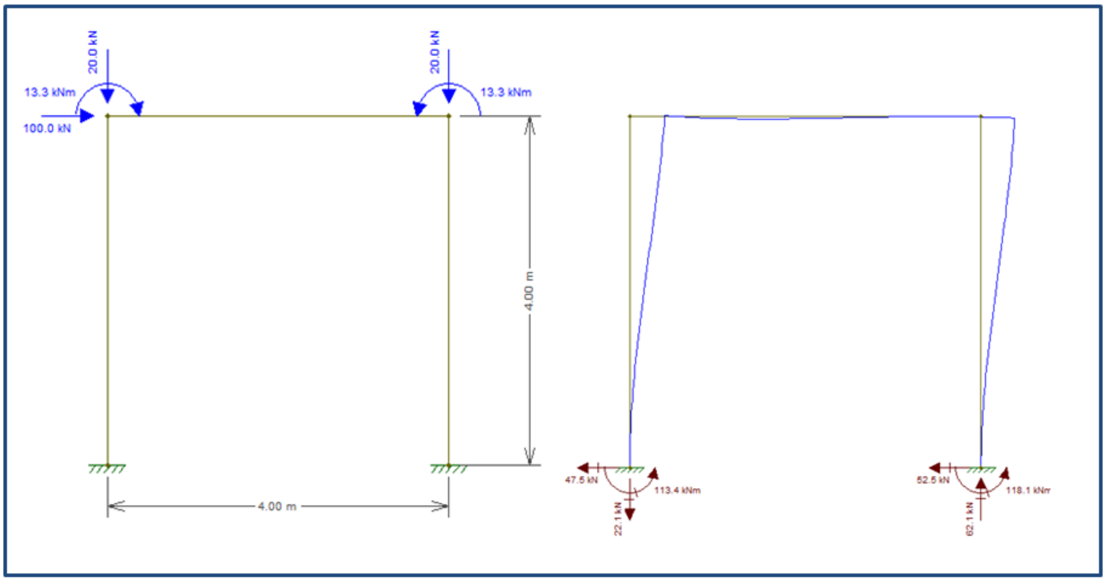

Figure 2. The frame used to validate the results – Ftool software.

The platform's configuration, accomplished using the exposed techniques, and the obtained results to the input model executed by the Java application developed using Netbeans IDE are shown as follows.

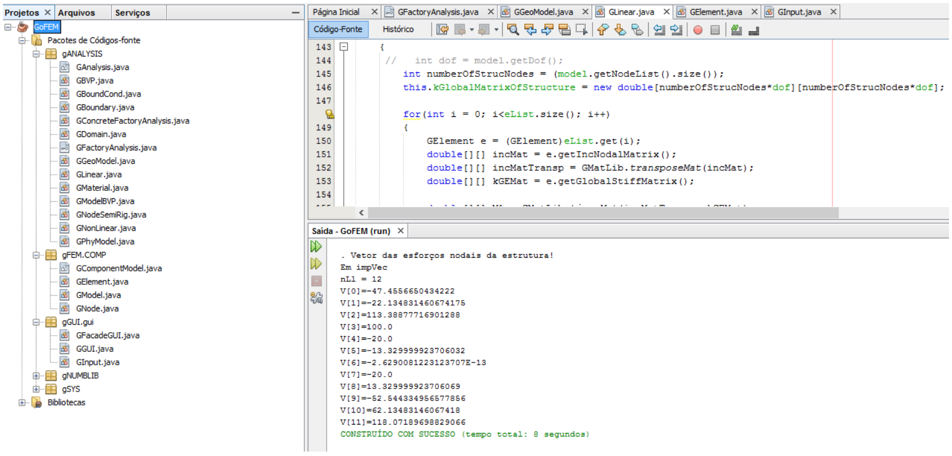

Figure 3. GoFEM Netbeans IDE project and the resultant reaction values for the proposed example.

4. CONCLUDING BENDMARKS

It is important to notice the relevance of Object-Oriented concepts during the platform's design. After this initial stage, the following phases, concern with the programming conception, can be developed more effective, once such techniques allow code extensibility, supporting the incorporation of new features related to FEM analysis.

The authors are now devoting time to development of algorithms according to the scope presented in this paper. The intention is use the GoFEM platform to educational purpose and to incorporate its coding and mathematical concepts to undergraduate engineering courses.

REFERENCES

Booch, G., Rumbaugh, J., & Jacobson, I., 1999. *The Unified Modeling Language: user guide*.

Cook, R. D., Malkus, D. S., Plesha, M. E., & Witt, R. J 2002. *Concepts and Aplications of Finite Element Analysis*.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J., 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*.

Noronha, M., Muller, A., & Pereira, A. M., 2005. An Object-Oriented Computational Platform Based On Design Patterns For The Bem Implementation. *Conference on Mechanics and Materials*.

Martha, L. F. 2002. An Object-Oriented Framework for Finite Element Programming. *Fifth World Congress on Computational Mechanics*, pp. 1-11.