Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1, ISSN 1726-9679
ISBN 978-3-901509-83-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2011
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2011

BASIC CONCEPTS TO DESIGN THE SOFTWARE APPLICATION OF A COMPUTER BASED MECHANICAL ENGINEERING MODEL

OANTA, E[mil]; PANAIT, C[ornel]; BATRINCA, G[hiorghe] & PESCARU, A[lexandru]

Abstract: *The paper is the result of more than 27 years of experience in software development and on an extensive literature review. Along these years turn-key applications in several fields were created (numerical programming, databases, graphical applications), consisting of more than 120000 computer code lines. This experience allowed us to conceive a 'blueprint' of the concepts to be used when a new software application must be created. The case study presented in the paper is dedicated to the mechanical engineering field which required customized applications which can be used in education, design and research.*

Key words: *mechanical engineering, analysis, criteria, new concepts*

1. INTRODUCTION

Computer based instruments become common nowadays, the users being able to solve problems in a fast and accurate manner, using a friendly graphical user interface.

Moreover, one can notice the extensive use of the computing capabilities in intelligent electronics. Several functionalities are added today to instruments which become 'smarter', day by day.

2. PROBLEM STATEMENT

Beside the instruments used to solve common problems, which are used by many people, there are special problems which require customized computer based solutions.

There is a wide range of engineering problems to be solved using dedicated methods.

The problem is to identify the set of criteria to be fulfilled by the original software applications to be developed, in order to create effective, useful and long-lasting instruments.

3. HISTORICAL PERSPECTIVE

A first approach to identify an answer to the aforementioned problem is to analyze the functionalities of the current software instruments and the way how they can be interfaced with other programs.

From a 27 years historical perspective and on an extensive literature review, we can notice that the information technologies of any age of the past had constraints as well as objectives which could be accomplished only using the high creativity and the intelligence of the analysts. The context of the current technology could be used very effective in order to solve the problems. Once the technology is changed, fewer constraints must be faced and more effective means to develop more intelligent instruments can be used.

An experienced software developer can notice that the problems in different IT ages are similar and, consequently, the solutions are alike.

The technological progresses in IT can be included in two main directions: communications and popularity.

Taking into account these trends we identify the most useful criteria to be fulfilled by the original software to be developed in a given topic of the mechanical engineering.

4. ANALYSIS OF THE PROBLEM TO BE SOLVED

We consider the mechanical engineering field and, more precise, the strength of materials topic. It could be considered a narrow topic, but it offers solutions in many other engineering topics such as: machinery design, ship strength, onboard equipment. It also must take into consideration other fields, such as heat transfer (thermal stresses), vibrations and fluid mechanics. To conclude, the strength of materials topic is analyzed as a support to understand general technical phenomena, and not to solve a particular problem.

The picture below presents the 'flow-chart' of the main topics in the classic approach, so to say.

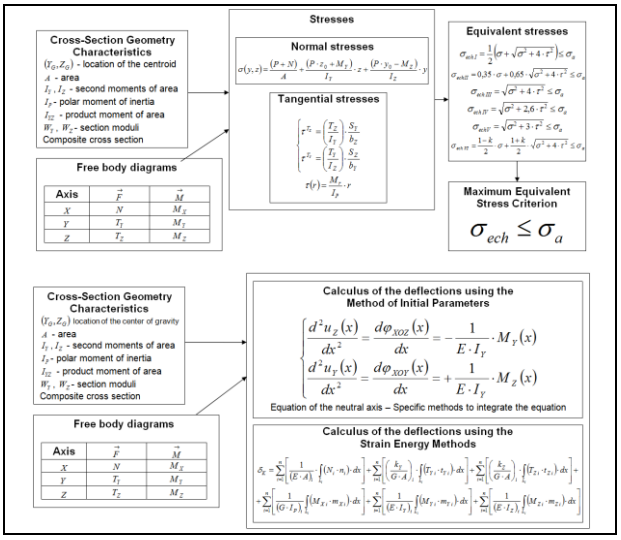

Fig. 1. Chapters are modules of the software to be developed

Another approach is to solve a particular technical problem, which means to use elements from each of the blocks of the previous figure. We chose to solve a problem related to the navigation field: the strength of the hull of a ship. The output of the software must consist of detailed computed data as well as refined information which offer the support to draw decisive conclusions if the structure withstands the loads or not.

5. CONDITIONS TO BE MET BY THE PROGRAM

Aside from the technical problems specific to the hull of the ship as a structure, we emphasize the ideas regarding the original software to be developed.

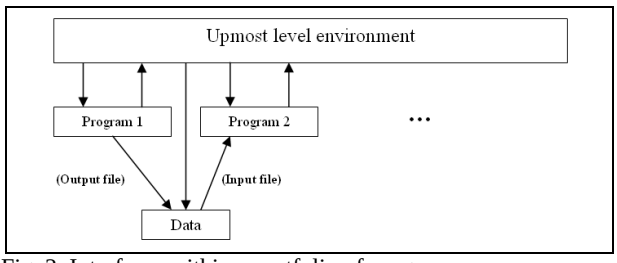

Fig. 2. Interfaces within a portfolio of programs

Handling the input and the output data in a flexible way is important in order to include this particular program in a portfolio of software applications which solve problems in this field of science and which transmit different datasets one to the other. We consider that the text input and output files are the best solution, because the text files can be created and accessed from any program or programming language. Thus, the input file stores information like:

- general information: number of intervals, number of subintervals in which the hull is discretized, flag parameters;
- interval-related values: length, geometrical characteristics, loads;
- boundary conditions based on common sense conditions which can result from practical measurements.

This input file can be created by another program which gathers data from output files created by other applications, libraries of cross-sections of the hulls, onboard sensors, etc.

There are several output text files, employed to:

- store the computed values in all the subintervals along the length of the hull in order to allow a detailed verification;
- store the problem formulation using a natural language in order to use these files for a textbook of problems;
- store the detailed solution of the problem, consisting of laws of variation, values at the end of the current interval and maximum values together with their location in this interval, to be included in the aforementioned textbook.

It should be also reminded that, for educational reasons, an *automatic data generator* should be included in the application in order to have instant presentations of the solved problems.

Choosing the appropriate *programming language* is paramount, because the particulars of the problem to be solved (numerical programming, databases, graphical facilities, communication) require a given set of facilities, which can be found in some 'candidate' programming languages, the most appropriate one being selected (Oanta & Tamas, 2009). However, it is advisable to have a 'background' database and the output data to be expressed in a standardized format, such as csv, xml, scr and html. Nowadays Java can be used to develop all the modules in figure 3 (Tanasa et al, 2003).

The *data structures* must be optimally designed in order to satisfy a set of requirements, such as: accuracy vs. amount of memory space needed, simplicity vs. flexibility (Oanta, 2000). These basic data structures are important because they are used to conceive the *data model* of the whole application.

The intelligent design of the *structure* of the application allows a facile maintenance, shorter execution time, fewer resources required. In this way there are modules dedicated to the general facilities, numerical methods modules and graphical facilities modules. For instance, if new graphical libraries are available, only the graphical modules will be upgraded.

For the numerical applications it is important to use the most *effective numerical methods* (Demidovitch & Maron, 1979). The criteria employed to make the decisions are: appropriateness with respect to the technical problem to be solved, possibility to embed the method or to create a general and reusable module, convergence of the iterative methods, speed, accuracy, simplicity.

Creation of the code is also important, at this level being identified the following criteria: concise style vs. explicit style, recursive technique vs. iterative technique, descriptive style vs. imperative style.

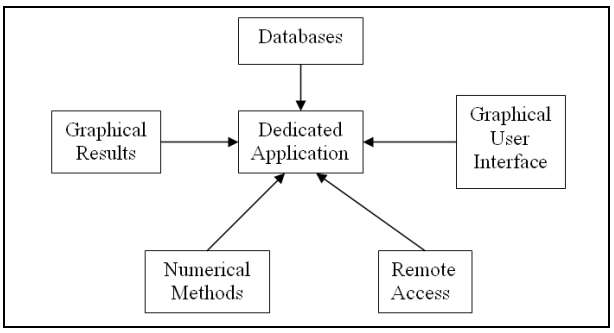

Fig. 3. Multiple requirements regarding the facilities of a given software application

Some other criteria are also important. For instance, a *cross-platform* application must respect additional constraints, but it can be used on a wide range of computers and devices. A *friendly graphical user interface* is always useful for the ergonomy of the application. The *documentation* of the software and the design of *embedded help* facilities are advantageous for the optimal use of the program.

6. CONCLUSION

Nowadays software applications offer intelligent suggestions in the decision making process. As well, in engineering the development of particular software applications offers fast, accurate and friendly solutions to various problems. The simple creation of the programs is not enough, several facilities being necessary in the development of versatile and intelligent software applications (Knuth, 2000). The set of criteria previously presented are important in the creation of the general design of the applications to be developed. The creation of intelligent software instruments is an important step on the path to the expert systems in that field of science.

7. ACKNOWLEDGEMENTS

Several of the ideas presented in the paper are the result of the models developed in the framework of the scientific research study '*Development of computer assisted marine structures*', Emil Oanta, Cornel Panait, Ghiorghe Batrinca, Alexandru Pescaru, Alexandra Nita, Feiza Memet, which is a component of the RoNoMar project, 2010.

8. REFERENCES

Demidovitch, B.; Maron, I. (1979), *Elements de calcul numerique*, Editions Mir, Moscou

Knuth, D. (2000), *The art of computer programming, 3rd edition, vol 1*, Teora Publishing House, ISBN 973-601-910-1, Bucharest

Oanta, E. (2000), *Basic theoretical knowledge in programming the computer aided mechanical engineering software*, Andrei Saguna Publishing House, ISBN 973-8146-04-6, Constanta

Oanta, E., Tamas, I. (2009), On the Path to the 'Ultimate Programming Language'?, *Proceedings of the 9th International Conference on Informatics in Economy, Section 6: K-Technologies*, May 7-8, 2009, Bucharest, ISBN 978-606-505-172-2, pp. 711-715, ASE Printing House, Bucharest

Tanasa, S., Olaru, C., Andrei, S.; (2003), *Java from 0 to the expert level*, POLIROM Publishing House, ISBN 973-681-201-4, Iasi