Integration of the original software applications for mechanical engineering

E Oanta
Constanta Maritime University, Faculty of Naval Electro-Mechanics, 104 Mircea cel Batran Street, 900663, Constanta, Romania
E-mail: eoanta@yahoo.com

Abstract. The paper presents some of the progresses recorded in the latest 35 years regarding the concepts, developments and applications of my original computer based research instruments. The identification of the original modelling concepts in mechanical engineering used original computer based instruments whose components are employed for: data visualisation, large matrices processing, parameterized calculus domains, parameterized solvers, arbitrary precision, interfaces between the software instruments and metaprogramming instruments. Data visualization was an important direction of development being used for the data validation and for the results’ interpretation. Several ‘sources’ of graphical facilities were identified and used over the time. Processing of the large matrices allowed the analyst to focus on the scientific problem to be solved, the data processing techniques being provided in background and leading to finer discretization and consequently, to more accurate results. Parameterized domains were used to define in a flexible way the calculus domain and the other input data. Moreover, the original interfaces allow the definition of the ‘common database’ concept, according to which several types of solvers may use the same ‘database’, i.e. the same set of input data. The parameterized solvers are original general software reusable solutions. The multiple precision allows the analyst to avoid some of the runtime errors, such as the overflow error and to get high accuracy results. The metaprogramming allows the user to easily integrate directly into the code the results of the previous data processing stage, which are expressed as a computer code. Several examples are given and some of the directions of future development are also presented.

1. Introduction
Performing calculi was a steady state demand along history, the first calculation instrument being considered the Sumerian abacus, while the rod calculus was the first mechanical method of algorithmic computation in ancient China. Mathematical methods which strongly supported the progresses in engineering were influenced by the computing instrument, the classic analytical approaches being based on simplifying hypotheses or assumptions, in order to minimise the volume of calculi. The first electronic computers were required in nuclear research and in space vehicles research, where fast and accurate calculations were necessary. The evolution of the computer science is the backbone of the developments of new computer based mathematical methods, which are currently used in nowadays commercial software applications useful in engineering. The continuous progresses in computing lead to the engineering concepts’ revise and to the identification of new research paradigms, such as the hybrid modelling where the analytical, numerical and experimental studies are deeply integrated using original computer based methods and interfaces. In the past 35
years one can notice an ‘explosion’ in computer science development, new generations of computers being quickly replaced by the newer technologies. This quick evolution was easily remarked in Romania where the students learnt programming using punched cards and mainframe computers and in less than 10 years they were exposed to new operating systems, such as RSX-11 (MIX), CP/M, DOS and Windows and to new programming languages, many of them being obsolete today, [1].

2. Theoretical background
Beside the commercial software applications used in mechanical engineering particular projects, the solving of the complex problems requires customized solutions and the according computer based instruments.

A first direction of development was suggested during the undergraduate studies by the usefulness of the diagrams employed to visually synthesize the data and to quickly analyse and interpret it. From this standpoint it became obvious that the visualisation techniques are paramount in engineering, the development of graphical interpreters being a constant concern. In this way, several resources were used for the data visualisation, [2].

![Figure 1. Integration of the software features necessary to create computer based original models.](image)

The mathematical approaches in engineering are extensively using matrices and matrix based calculus. Numerical applications are useful when a direct mathematical method cannot be conceived and they offer an approximation of the so-called ‘exact’ solution. The accuracy of the numerical methods often depends on the discretization of the calculus domain. In this way, a finer discretization of the calculus domain leads to higher order matrices which must be processed in an effective way. These remarks suggest the second direction of development, i.e. the creation of software modules dedicated to the matrices’ optimal processing. The resulting software libraries were useful to quickly implement software applications whose solutions are based on numerical methods, on the finite element method and on the finite difference method.

The quick development of applications in mechanical engineering requires both, solvers and interfaces. It results that other direction of development is to create various general solvers based on various numerical models, modern algorithms and data processing techniques, the resulting parameterized software modules being required to be flexible, for reusability reasons.

Data integration in a computer based model uses interfaces. In this way, the interfaces are useful to connect programming languages, software developing environments, commercial software applications and data processing stages in a complex project, designated as a hybrid model. Moreover, the application program interface facilities of the commercial software applications may be used to quickly link the application to the computing context of the project. This was another direction of development.

Analytical modelling of the calculus domain is useful for both analytical and numerical models of a given phenomenon. Using general methods to approximate the calculus domain may lead to the creation of versatile software instruments useful to generate the input data of the theoretical models.
This is why the development of software modules based on analytic geometry and on the approximation theory is another direction of development.

3. Original computer based instruments
The features presented in figure 1 were steady developed over the time, several solutions being completed, in various programming languages.

Data visualisation is paramount in the verification of the model’s results. In essence, two types of graphical output ‘generators’ were used. Firstly, there were used the graphical libraries of the programming languages in order to create customised graphical representations. Secondly, there were used applications which have graphical capabilities, that were employed to create sophisticated representations.

![Graphical output based on a programming language](image1)

Figure 2. Output data visualisations based on the graphical library of a programming language.

One of the early graphical libraries was in Turbo Pascal and it offered a set of functions and procedures useful to handle the display of the computer, which was initialised as a set of pixels. This library was used to develop customised graphics, [3], closely connected to the problem to be solved, i.e. particular graphical interpreters, figure 2.

![Graphical output based on automatic generation](image2)

Figure 3. Output data visualisations based on the automatic graphics generation.

Another data visualization solution was to use the pre-defined graphics of some programming languages and environments. Linking the output data of a particular application to the aforementioned pre-defined graphics, there were easily created data interpreters, for particular purposes. For instance, the left side of figure 3 presents a sketch of the deflections of a quarter-submodel of a plate, [4]. The right side of figure 3 presents the variation of the convective heat transfer coefficient with respect to the crank angle in an internal combustion engine, [5].
Graphical interpreters were also conceived starting from computer aided design commercial applications. There were used several techniques to process the outer data. An initial solution was to create AutoCAD ‘script’ files which contain AutoCAD commands that use the particular data of the model under development. This method was used for both 2D and 3D geometric models. In order to decrease the size of the resulting drawing, there were imagined criteria to consider only the outer surfaces of the geometric model. After the script file was imported in AutoCAD, the ‘VPOINT’ and ‘HIDE’ commands were used and some of the resulting visualizations are presented in the left side of figure 4. One can notice the FEM model of an energy absorber tank which was also investigated using the strain gage technology, [6], and a test problem regarding a plate whose opposite sides are simply supported, respectively supported on elastic fixed supports, i.e. beams along the sides, [7]. Other idea was to use one of the programming languages assigned to a given CAD environment in order to link the application to the outer data files resulted from the previous data processing stages. This method was further on developed in order to automatically create the FEM model in Femap/Nastran, [8]. It may be used for the CAD applications which have API, i.e. Application Program Interface facilities. This technique was also used in AutoCAD using the AutoLISP programming language and in GNU Octave where its specific scientific programming language and mathematical library functions were used.

Beside the data visualization, other concern regards the accuracy of the numerical methods and models which depends on various aspects, including the domain discretization. In this way, a finer discretization may lead to a higher accuracy of the results, being necessary to solve a problem with a larger number of unknowns. Because most of the mathematical solutions rely on matrix formulation, the readily handling of the large matrices was considered an important problem to be solved, [9]. The basic idea was to use the largest resource of a computer in order to store a matrix, i.e. its hard disk. In order to minimise the access time, the elements of the matrix were stored in a random access file. Other idea was to place a ‘matrix’ file in a RAM drive in order to access the file using the speed of the RAM. Several programming languages having specific strengths were used to develop distinct program modules which implement various facilities. The data exchange between these components was developed using the text version of the ‘matrix’ random access file. Because the structure of the record was the same for all the lines of the text file, consequently the lines have the same length, the text file was redefined as a ‘file of bytes’, this approach allowing the direct access in that given text file. Moreover, in each programming language were developed libraries which read from the ‘matrix’ text file and write a matrix into the ‘matrix’ text file. In this way an interface between several programming languages was developed. Another method to minimise the access time to a given element was to load a set of lines or a set of columns of the matrix into a one-dimensional array of doubly connected circular lists, the index in the array being used to identify the line or the column.
whose elements are stored in the current list. There was also developed an editor which easily handles 'matrix' files.

Another component presented in figure 1 regards the parameterized solvers. Parameterized solvers were developed based on the general numerical methods, the numerical aspects of the finite element method and of the finite difference method, figure 5. Several original software applications and software libraries consisting of reusable components were developed. In order to minimise the implementation time there were also used the GNU Octave library functions.

![Diagram](image.png)

**Figure 5.** Components of the parameterized solvers and ideas regarding the implementation.

The fast development of the applications for hybrid modelling requires solvers, as mentioned before, as well as interfaces. In this way, several types of interfaces were used. An initial solution was developed in order to link programming languages using the aforementioned ‘matrix’ text files. Other solution was based on the CSV format, the according text files being accessed from any programming language, including the spreadsheet applications. The API facilities of the commercial applications were also employed in order to read the according files used to link the software components of a program.

An interesting method to link the applications is to express the output data as a computer code, in a given programming language, i.e. to use metaprogramming. The resulting computer code may be instantly included in a library or in a chain of software applications. Other strength is the data encryption into the resulted code, which cannot be read after the code is compiled.

The flexible definition of the calculus domains may be accomplished in several ways, the main concern being to save time and effort when a model is upgraded several times. An initial approach was to conceive data generators, this idea being used to create the models presented in the left side of figure 4. However, in complex research projects several models which use the same calculus domain may use different types of solutions. In this case, there was developed and implemented the ‘common database’ concept according to which, a parameterized geometric model is developed and once a modification is operated, all the models of the research project may access the upgraded data, [10]. The 2D domains are conceived using a bool algebra which operates with simple geometric shapes which are added to the calculus domain or subtracted from the domain. A first method is approximating the boundary defined by a set of points with a set of spline functions, [11], while the second method approximate the domain with polygons, [12].

Accuracy of the results is a main goal in computer based modelling. However, there are certain cases when the maximum size of the variables isn’t large enough to assure the appropriate precision. In such cases there was used a solution based on libraries which offer multiple precision, also known as arbitrary precision, [13]. Using the types and functions in that given library, the user is allowed to define the variables’ sizes, i.e. the number of bytes of the ‘real’ type variables and of the ‘integer’ type variables. Usually, the analyst is identifying the sequence of the code where the errors are produced and replace that part of the code with a new version based on arbitrary precision variables. Using these multiple precision libraries, there may be defined small numbers useful in the particle physics models, or large numbers useful in astronomy.
4. Discussion
The previously presented ideas, methods and implementations were extensively used in the development of scientific research models. The rapid development of the software applications is a key factor in the creation of the hybrid models. Moreover, the software developed for research purposes may be reused for design, education and for the computer aided monitoring of various phenomena. Each component in figure 1 proved its efficiency in several scientific research projects and their upgrade in respect to the latest research concepts, programming languages and experimental technologies is a constant concern.

The data visualization solutions are very useful for the rapid and facile verification of the models. They are under a permanent development, being targeted data interpreters integrated in the structural calculus applications based on OpenGL. The geometric data is similarly structured in several CAD applications, therefore some of the original library functions may be reused. The analysis is focused on the common repetitive data processing stages which may be developed as reusable header files. Another direction of development is to use the virtual reality facilities in order to present in a more realistic way the results, [14]. According to the nowadays trends, the augmented reality is becoming a more and more widespread educational instrument.

The original libraries used to easily handle the ‘matrix’ files must be updated in order to support the specific implementation of the arbitrary precision libraries, such as GNU Multiple Precision Arithmetic Library (GMP) or GNU MPFR Library. According to the latest analysis, the fields for the indexes of the elements and the field where the value of the element are stored must have larger sizes in order to fit the requests regarding both, the large size of the matrices and the high precision of its elements. The next version software will be developed in C++ and, if possible, will use parallel data processing.

Regarding the original parameterized solvers, there were conceived several original solutions, beside the previously presented numerical solutions. An important project was based on a method which minimized the search space of the candidate solutions. Thus, at each data processing stage the algorithm starts with an initial domain where the solutions are searched, a given accuracy of the accepted solutions and it results a domain of the accepted solutions. The next stage considers a more accurate acceptance criterion of the solutions and the domain where solutions are searched is the resulting domain of the previous stage. This contraction method with successive improved approximations was implemented using a recursive algorithm and it was used to solve a ship hull equilibrium problem, [15]. Numerical methods principles were used to develop experimental data processing software. Thus, the bisection method principle was used to solve an interesting problem in strain gage technology used to measure the strains in the cylinder block of a running engine, [16]. In this class there must be also reminded the method used to deduce the analytical functions of the isostatics starting from the spline approximation of the isoclinic fringe pattern in photoelasticity, [17].

The parameterized definition of the calculus domains was used for more complex operations, such as the calculus of the geometric characteristic of the beams’ cross sections, [18]. In this case the analytic solution produced large numbers powered by 10, i.e. very large numbers, being required the use of the arbitrary precision libraries. The approximations were also used to discretize beams’ cross sections complex shapes, i.e. bulb flats for shipbuilding and to discretize ship hull cross sections. Another interesting problem solved using a parameterized definition of the domain was the calculus of the form factors for shear used in the calculus of the deflections using strain energy methods, [19].

To conclude this section, figure 1 presents not only accomplishments in the model development activity along the years, but also a strategy of development of new software instruments for research, which may be updated from several points of view. The wide range of problems solved using the hybrid modelling principles prove the correctness of the early development concepts when the computer programming used punched cards, 35 years ago. The high degree of integration in a complex model of all types of computer based studies, i.e. analytical, numeric and experimental, is another evidence regarding the future direction of development. Finally, one can notice that computer programming is the actual instrument used in the advancement of science and technology and its use in the development of intelligent instruments of research is a rational requirement.
5. Conclusions
The experience acquired during the steady state development of the computer based instruments of scientific research in the latest 35 years offers the grounds for a new vision regarding the concepts to be used in the modelling of the phenomena in our fast changing world.

Nowadays commercial/freeware software applications and fast advances in computer science offer the environment for the effective development of new research concepts and the according software applications. Two main development directions were identified: creation of new solvers based on modern algorithms and creation of interfaces between the solvers and the software applications.

Regarding the new solvers, as mentioned before in [20], in “the class of the meta-heuristic algorithms used to solve internal combustion engines related problems are included evolutionary algorithms, evolution strategy, evolutionary programming, genetic programming, differential evolution, estimation of distribution algorithm, ant colony optimization, particle swarm optimization, memetic algorithms and artificial immune system, a literature survey being given in” [21]. Beside these new algorithms, new computing concepts must be invented in order to employ in an effective way the experience earned in parallel computing, and in the use of the graphics processing unit.

The creation of new interfaces creates a synergic effect, all the components already developed being more and more useful in a higher complexity context and offering a metalevel of understanding. Hybrid modelling concepts and scientific investigation instruments may be applied in a wide range of engineering research problems [22-27] as well as in design and educational activities where the students must understand the various advantages offered by an analytic mind trained to learn by the use of computer programming principles.

6. References
[1] Oanta E 2019 Dobrogea’s Journal of Engineering 17/2019 https://agir-constanta.ro/wp-content/uploads/2019/08/JID-Nr-17.pdf.
[2] Oanta E, Panait C and Raicu A 2020 AGIR Workshop January 2020.
[3] Buzbuchi N and Oanta E 1996 Matrix Methods in Studying the Torsional Vibration of the Marine Engines Shafting Systems, Software Reference Manual, National Technical University of Athens, Greece.
[4] Kumbetlian G, Hreniuc V and Oanta E 1993 Proceedings of ELFIN2 Nat Conf, pp. 53-58.
[5] Oanta E 2001 Proceedings of the XI-th Romanian Conference on Thermodynamics Vol 3 pp 125-132 ISBN 973-8052-72-6.
[6] Kumbetlian G, Oanta E and Neamtu L 1992 Experimental and theoretical study of the strains and stresses in the energy absorber tank from Midia SA, Research project for industry, 1748-09/07/1992.
[7] Kumbetlian G and Oanta E 1993 Study of the strains and stresses in the thick rectangular plates with stress concentrators, research project 5050-07/07/1993.
[8] Oanta E, Panait C and Raicu A 2016 Proc. SPIE 10010 Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies VIII, 100102O (December 14, 2016); ISSN: 0277-786X; ISBN: 978-1-5106-0424-7, 978-1-5106-0425-4; doi:10.1117/12.2234000.
[9] Oanta E and Nicolescu B 1999 Proceedings the 19th Computers and Information in Engineering ASME Conference ISBN 9780791819722 2 paper DETC99/CIE-9059 457-464.
[10] Oanta E, Raicu A and Panait C 2017 IOP Conference Series: Materials Science and Engineering 227 ISSN: 1757-8981, doi:10.1088/1757-899X/227/1/012084.
[11] Oanta E, Panait C, Lazaroiu C and Dascalescu A-E 2014 Scientific.Net Publications Advanced Materials Research 1036 1017-1022 ISSN 102-660, ISBN-13: 978-3-03835-255-6, doi: 10.4028/www.scientific.net/AMR.1036.1017.
[12] Oanta E, Panait C, Raicu A, Barhalescu M, Axinte T 2016 IOP Conference Series: Materials Science and Engineering 145 ISSN: 1757-8981 doi:10.1088/1757-899X/145/8/082011.
[13] Pescaru A, Oanta E, Axinte T and Dascalescu A-E 2015 IOP Conference Series: Materials Science and Engineering 95 ISSN: 1757-8981, doi:10.1088/1757-899X/95/1/012125.
[14] Vucinic D, Deen D, Oanta E, Batarilo Z, Lacor C 2007 *Advances in Computer Graphics and Computer Vision. Communications in Computer and Information Science* **4** Springer, Berlin, Heidelberg Print ISBN 978-3-540-75272-1, Online ISBN 978-3-540-75274-5, https://doi.org/10.1007/978-3-540-75274-5_10.

[15] Oanta E, Tamas R and Paun M 2018 *Proc. SPIE* **10977** Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies IX 1097704 (31 December 2018); eISSN: 1996-756X, ISSN: 0277-786X, ISBN: 978-1-5106-2614-0; doi: 10.1117/12.2323164.

[16] Oanta E, Taraza D 2000 *Proceedings of the SAE 2000 World Congress*, ISSN 0148-7191, doi:10.4271/2000-01-0520, http://papers.sae.org/2000-01-0520/.

[17] Oanta E, Panait C, Barhalescu M, Sabau A, Dumitрache C and Dascalescu A-E 2015 *Proc. SPIE* **9258** Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies VII 92582A; doi:10.1117/12.2070409.

[18] Oanta E, Constantinescu E, Raicu A and Axinte T 2014 *Scientific.Net Publications Advanced Materials Research* **1036** 697-702 ISSN 102-660, ISBN-13: 978-3-03835-255-6, doi:10.4028/www.scientific.net/AMR.1036.697.

[19] Constantinescu E, Oanta E and Panait E 2017 *IOP Conference Series: Materials Science and Engineering* **227** ISSN: 1757-8981 doi:10.1088/1757-899X/227/1/012031.

[20] Oanta E 2018 *Hybrid modeling in mechanical engineering - Habilitation thesis* Doctoral School of Mechanical Engineering, Constanta Maritime University, https://drive.google.com/open?id=1sWRs_GetpD1svJfC2mQ-xjb497jcMdIX.

[21] Mohammad H, Tayarani N, Xin Y, and Hongming Xu 2015 *IEEE Transactions on Evolutionary Computation* **19-5** pp 609-629 Print ISSN: 1089-778X Print ISSN: 1089-778X Electronic ISSN: 1941-0026 doi:10.1109/TEVC.2014.2355174.

[22] Raicu A, Barhalescu M and Memet F 2019 *IOP Conf., Series: Materials Science and Engineering* **591** 012066 doi:10.1088/1757-899X/591/1/012066.

[23] Sabau A 2014 *Advanced Materials Research* **837** 471-476 https://doi.org/10.4028/www.scientific.net/AMR.837.471.

[24] Sabau A 2014 *Advanced Materials Research* **837** 477-482 https://doi.org/10.4028/www.scientific.net/AMR.837.477.

[25] Sabau A 2014 *Applied Mechanics and Materials* **659** 450-455 https://doi.org/10.4028/www.scientific.net/AMM.659.450.

[26] Deleanu D 2005 *Mechanics, Theory and Applications* (Constanta: Ex Ponto) ISBN 973-644-444-9.

[27] Deleanu D 2012 *Theoretical Mechanics, Theory and Applications* (in English) (Constanta: Nautica) ISBN 978-606-8105-89-5.

[28] Oanta E, Panait C, Nicolae B, Dinu S, Pescaru A, Nita A and Gavrila G 2007 *Computed aided advanced studies in applied elasticity from an interdisciplinary perspective* ID1223 CNCSIS Romania research project.

[29] Oanta E, Panait C, Lepadatu L, Tamas R, Batrinca G, Nistor C, Marina V, Iliadju G, Sontea V, Marina V and Balan V 2010 *Mathematical models for inter-domain approaches with applications in engineering and economy* MIEC2010, ANCS Ro-Md scientific research project.

Acknowledgement
The original ideas and implementations presented in the paper are originating in the “Computer Aided Advanced Studies in Applied Elasticity from an Interdisciplinary Perspective” 2007-2010 scientific research project supported by the National University Research Council (CNCSIS), Romania, [28], and from its follow-up scientific research project “Mathematical Models for Inter-Domain Approaches with Applications in Engineering and Economy” MIEC 2010-2012, under the supervision of the National Authority for Scientific Research (ANCS), Romania, [29].