Journal of Achievements in Materials and Manufacturing Engineering

VOLUME 31
ISSUE 2
December
2008

Software independence: impact on automotive product development process

M. Annarumma*, A. Naddeo, M. Pappalardo
Department of Mechanical Engineering, University of Salerno,
via Ponte Don Melillo, 84084 Fisciano (SA), Italy
* Corresponding author: E-mail address: maannarumma@unisa.it

Received 15.09.2008; published in revised form 01.12.2008

Industrial management and organisation

ABSTRACT

Purpose: Currently, in automotive industries one of the most important works in product design is the evaluation of crash events using FEM simulation; in this context, software independence, that means to obtain the same simulation results on the same vehicle-model using different FEM solvers, will be useful for reducing virtual prototyping time and consequently Time To Market.
Design/methodology/approach: The carrying out of a software-independence translation methodology is the aim of this work, which is an input data translation by a methodology independent of the CAE calculation environment and allows reducing significantly the product development time. This methodology will be developed by routines written ad hoc in Matlab language, that carries back in LS-DYNA environment the cards written in RADIOSS environment, supported by TextPad editor and completed by pre-processor Hypermesh.
Findings: The translation could be made using the pre and post-processor Hypermesh and the RADIOSS model card manual correction, but this way working increases greatly the Time To Market, although today it is the only procedure used in the most important Italian automotive industries.
Research limitations/implications: Most of crash simulation models are carried out in RADIOSS environment, but the LS-DYNA environment potentialities exploration has increased the simulations demand using LS-DYNA software; therefore the development of a translation methodology like that satisfies this requirement.
Practical implications: Software independence in Virtual Prototyping could accelerate several processes, bringing many benefits such as the reduction in product development time and in real prototyping costs and the increase in quality, carrying out more competitive product on the market.
Originality/value: The carrying out of a translation methodology, that allows to carry back in a software environment the Know How developed for another software environment, will provide considerable industrial advantages for vehicle designers (We've to remind that often many design tasks of a project are made in outsourcing way).
Keywords: Translation routine; Software-independence translation; Product development time

1. Introduction

Recently, in order to identify efficiently the customer needs and to quickly create products that satisfy them and are manufacturable with contained costs, achieving the economic success in the global Market, only marketing or designing or manufacturing problems haven't to be tackled and resolved.

The winning product idea pass through a Product Development Process, carried out in effective and efficient way and involving all main business functions.
In this context, the reduction in costs and in time to market has not to be considered with aim to itself, because the full profit of designing is the goal of enterprise's management. The competition is won if a customer new need is satisfied and if answer modality is very interesting.

The information and virtual technologies, which allow, for example, the carrying out a “Software independence in Virtual Prototyping”, could help to increase speed in many processes.

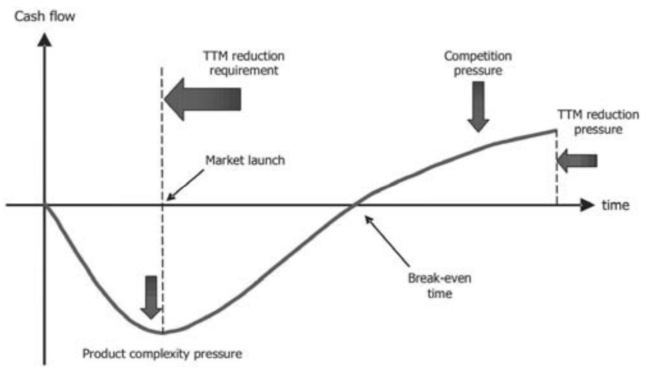


Fig. 1. The competitive and financial incentives to reduction in TTM

The expected benefits of using Information and Virtual Technologies are, therefore, the reduction in development time (it enables the concurrent engineering), the reduction in development costs and the increase in quality [8].

The need to reduce costs and to increase quality is obvious, but exist also other reasons that drive to reduce the time to market, like the need to bring in Market new products characterized, for example, by an actual style. The way in which the pressures, which the Market does on the car manufacturers, affect the design typical cash flow, has been synthesized in the Figure 1 [18].

In recent business experience, the most used FEM solvers in Automotive industries for crash simulation are RADIOSS, PamCrash and LS-DYNA. In particular, most of crash simulation models are carried out in RADIOSS environment, but the LS-DYNA environment potentialities investigation has increased the simulations demand using LS-DYNA software.

On the ground of previous remark, for vehicle designers, the development of a translation methodology, that allows to carry back in a software (LS-DYNA) environment the consolidate Know How developed for another software (RADIOSS) environment, will provide considerable industrial advantages [20-26].

The method for the whole translation [2,11] has been carried out by routines written ad hoc in Matlab 6.5 language [12], helped by textual editor TextPad and completed by pre-processor Hypermesh 6.0 [19].

1.1. Vehicle Product Development Plan

The total life of a product, in our case a car or, in general, a vehicle, has to be read beginning from its end, none on the contrary. If the development times of new car are much long, actually the product complexity needs at least three years from the product concept definition to the setting in production; the first element defined aren’t the product characteristics, but the business launching, that is when it has been considered that the model to replace will finish its life cycle, and when it is possible to dedicate to that design all the necessary resources.

Cars, also constituting a more and more unitary and integrated element, are composed by a high number of parts, some of that have an evolution different from the final model. The level ground and the engine, for example, have developments and life cycles very different and are identified with the new model characteristic part.

The entire vehicle development process requires a combination of many separate performances, how the Figure 2 shows [5]. These activities are frequently performed in separate departments with little internal communication. Automotive companies are now redesigning their internal processes to coordinate these diverse tasks and provide some logical structure for communication. In addition, a more formal design process is being implemented to ensure that separate design groups can work on different parts of the development activity in parallel and also ensure that the final combined activity produces a design that meets the performance targets initially set.

The vehicle development process has been influenced by increasing requirements for quicker and less costly development cycles, combined with reduced vehicle fuel consumption. These requirements are being addressed by new product and process development concepts.

The complete requirements for new vehicle development include the need for rapid prototyping and durability evaluation to achieve an accelerated vehicle development process. There is now a convergence of market and technical changes that directly affect this development process.

Modern vehicle development, in fact, requires a combination of many engineering disciplines. The demands for performance, durability, safety and other customer oriented factors must be combined in a cost effective process to create a new vehicle within a shorter time [1,14].

It is important to note that the new technologies that are being adopted, are changing the commercial and organizational side of the automotive industry. These commercial and organizational changes in turn are driving new technologies in engineering in an effort to "catch-up" with the changes demanded in the market environment [4]. Engineering isn't driving the changes; it's reacting to them.

In this context, the information technologies which allow the carrying out a “Software independence in Virtual Prototyping”, become an integral component of the product development process [16], that lies after the virtual model development and before its use for any testing operation.

Therefore, the planning of the product range in an automotive industry is much complex and lasts ten years at least. The setting in production of a new model has to be synchronized with all the others, especially with that in production.

Defining the right time for the business launching and assuming the development times, it goes on backwards in the start moment identification [3]. For this reason, the more a company is able to develop the design in short time, the more is probable that the final product comes up to customer's expectations [13]. However, in this case the conditional is required, because the fundamental factor is the perceptive ability to the customers' needs.

An interpretative error in the initial concept will be brought again in the final product, if it won't be recognized during the development plan. In this case the business ability to quickly develop a particular and to bring back it in the total product turns out fundamental [7].

Consequently, the organizational formulation for the new product development [9] is unavoidably conditioned by the strategic and economic formulation of the Time To Market, that is the time necessary to develop a new product from its conception to the business launching [17].

The comparison between car manufacturing companies on the Time To Market does make sense only if a common terminology is used whether in the output or in the activities contents, at least of most important that, in the initial and final steps of the development process [6].

Independently from the choice of one of the possible alternatives for the definition of the beginning of the process, several uncertainties remain about the information available to the initial step in the strategies and business plans definition. In fact, it has been considered that the Time To Market can be under the influences of total strategic formulation [10].

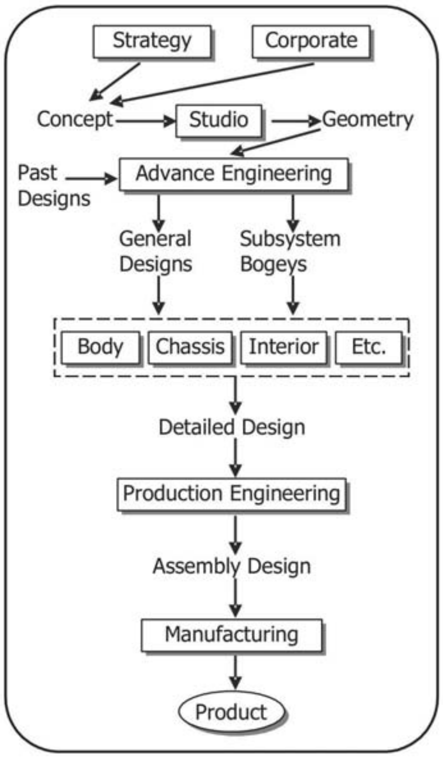


Fig. 2. Vehicle Development Plan

1.2. Virtual Prototyping

A short definition of Virtual Prototyping, extracted from various available definitions, is (Wang 2002):
"Virtual prototype is a computer simulation of a physical product that can be presented, analyzed, and tested from concerned product life-cycle aspects such as design/engineering, manufacturing, service, and recycling as if on a real physical model. The construction and testing of a virtual prototype is called virtual prototyping (VP)."

The proposed definition specifically stated that a virtual prototype is a digital mock-up; described the function of a virtual prototype; implied the importance of human-product interaction; excluded the virtual reality technique and design optimization from the definition; and differentiated the virtual prototype and virtual prototyping techniques. It is to be noted that the acronym VP stands for virtual prototyping and not for the virtual prototype.
From the definition of prototype, one can summarize the characteristics of a prototype as below:

- A model of a structure or apparatus (or a product)
- Used for testing and evaluate form, design fit, performance, and manufacturability.
- Used for study and training.

In a conventional product development process, a prototype is usually constructed to prove design concepts, evaluate design alternatives, test product manufacturability, and often just to present a product. With the goal of replacing physical prototype electronically, a virtual prototype must first serve the same functions and even more of a physical prototype, regardless of which techniques are used.

In light of this, a virtual prototype should be able to be used to "test" a product's form and performances, and be used for training and other studies. In addition, a physical prototype usually allows human beings' sensory evaluation of a product, such as color, form, aesthetic features, feel, fitness, and so on. Product's ergonomics is also of an increasing concern. To substitute these functions of a physical prototype, a human-product interaction component should be included in a virtual prototype.

Based on the proposed definition of VP, an elaboration of virtual prototype components is attempted. First, a computer simulation of a product is required.
At current stage, a 3D solid model is the widely accepted product presentation, usually parametric. Second, for a virtual prototype to be presented as a real physical model, a human-product interaction model is desired. Ideally, a virtual product can be viewed, listened, smelled, and touched by an engineer or a customer. This is the area that virtual reality techniques can play an important role.

More importantly, various perspectives of the designed product should be able to tested and evaluated. In summary, a complete virtual prototype should include essentially three types of models as follows:

- A 3D solid model,
- A human-product interaction model, and
- Perspective test related models (Wang, 2002).

The software design process includes many stages, including specifications, coding and integration. Each stage introduces bugs, and can greatly slip product schedules when performed serially. Virtual prototyping enables continuous iterative design and testing via simulation, allowing the developer to find errors in the system design and implementation earlier.

By bringing all the system pieces together early in the design cycle, architects and software developers have the most flexibility to adapt their designs and code to changing requirements, and ensure their products meet requirements for schedule, performance and power consumption.

2. Description of the approach to translation methodology

The translation methodology has been carried out writing ad hoc routines in Matlab 6.5 language, that carries back in LS-DYNA environment the most important cards written in RADIOSS environment, helped by textual editor TextPad and completed by pre-processor Hypermesh 6.0, which has the user section configurable according to used solver.

2.1. File ***.dyn description

The file ".dyn" is the ASCII file used in input by LS-DYNA solver.
The numerical data are brought in a "cards" rigorously subdivided in 8 fields, each composed by 10 characters, in which it's possible to take the most convenient format for numbers, without any limitation.
The Figure 3 tries to show the file structure and the way of linking inside its various entities.

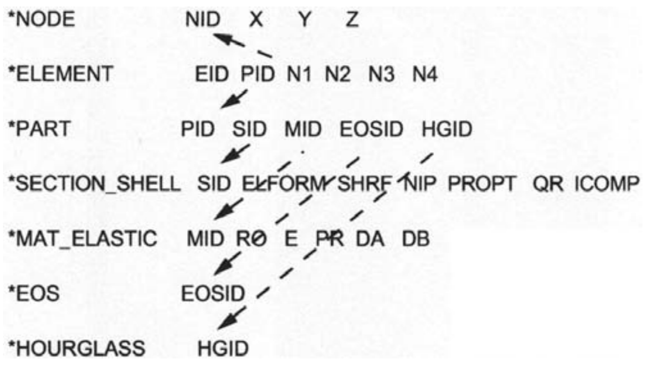

*NODE NID X Y Z
*ELEMENT EID PID N1 N2 N3 N4
*PART PID SID MID EOSID HGID
*SECTION_SHELL SID ELFORM SHRF NIP PROPT QR ICOMP
*MAT_ELASTIC MID RO E PR DA DB
*EOS EOSID
*HOURGLASS HGID

Fig. 3. LS-DYNA file structure

In that figure the arrows indicate the logical connections between the various keywords. So, in the *ELEMENT keyword an element identifier, a part identifier to which the element belongs, with the nodes identifiers, that delimitate the same element and are defined in the *NODE section, will be present.

The *PART keyword fields contain the identifier of the following elements: part, section to which the part belongs, material whose is composed (see *MAT section), state equation of material modeling (see *EOSD section), until *END keyword, which closes the file.

2.2. File ***D00 description

The file format in input "***D00" is of type reported in Figure 4.

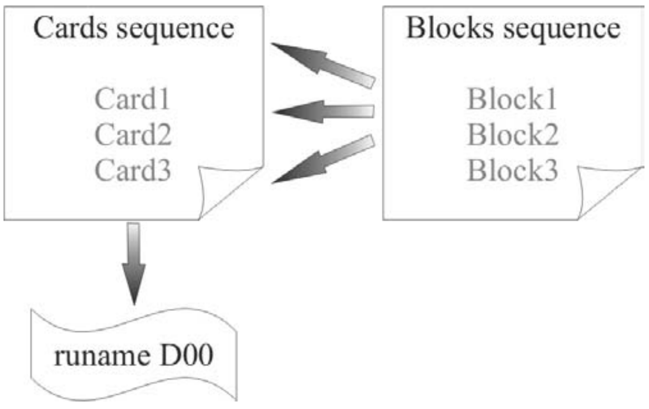


Fig. 4. Input file forma

Each block content is composed by 80 characters, subdivided in 10 fields everyone by 8 characters. Inside every block the numeric data, that characterize the simulated model, are let in a "cards".
A typical line is shown in the Figure 5.

1	2	3	4	5	6	7	8	9	10

Fig. 5. LS-DYNA typical line

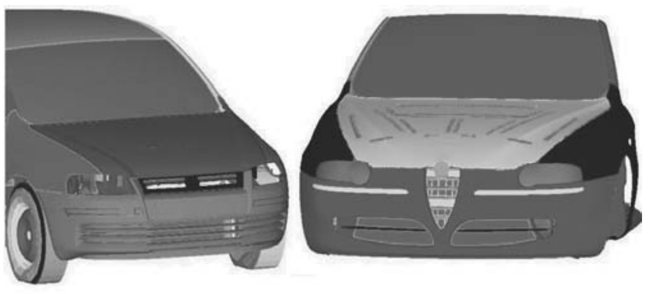


Fig. 6. Vehicle FEM model

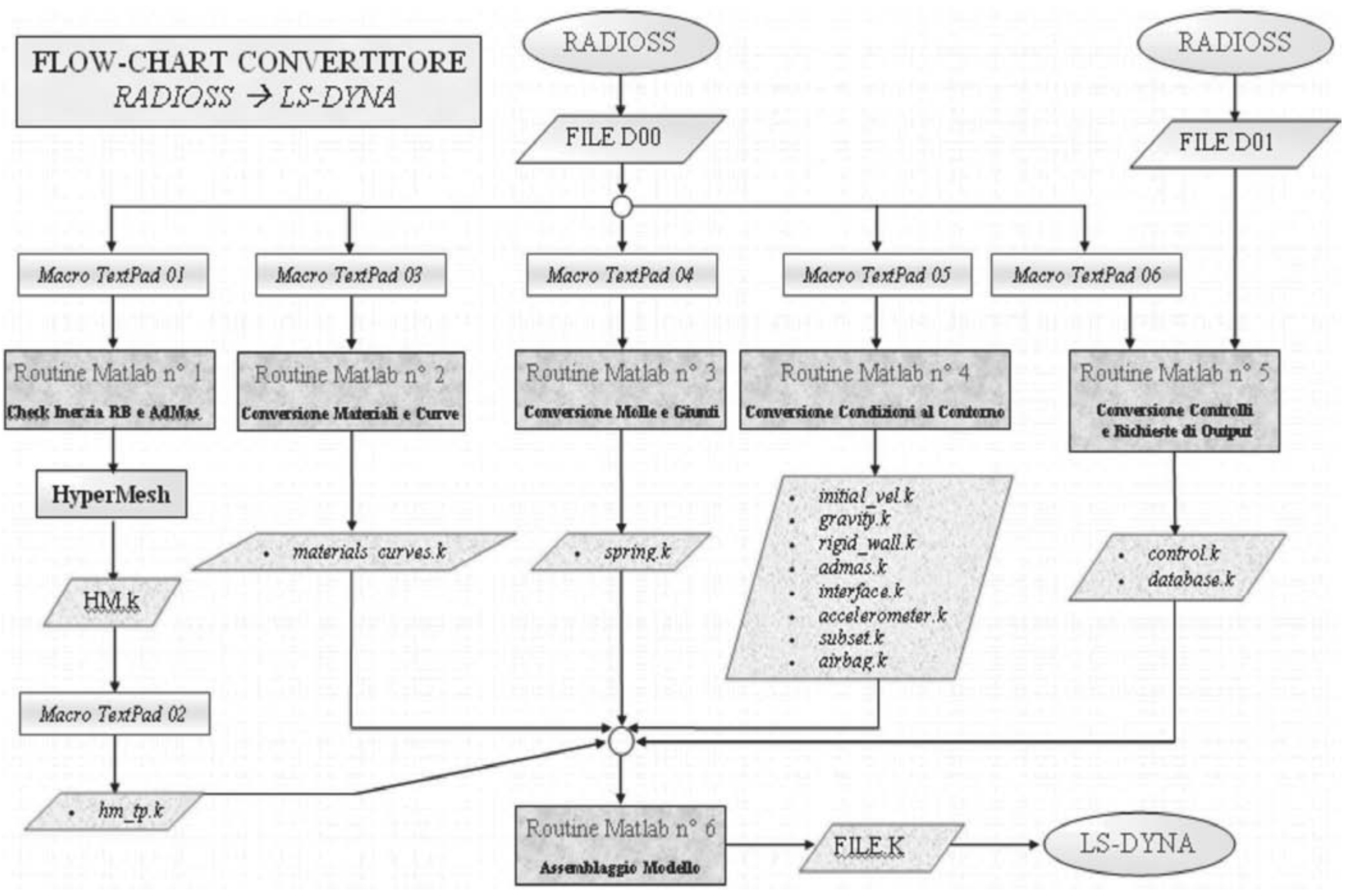


Fig. 7. Flow chart translator

3. Description of achieved results

3.1. General remarks

The carrying out of a translation methodology is the object of this paper: we've executed a complete software-independence translation, which is an input data translation by a methodology independent of the finite elements calculation environment; that routine automatically carries out the code translation, which allows to reduce significantly the product development time.

3.2. Translation routine

The translation has been tested on two car models got ready for a full frontal crash against USNCAP rigid barrier: Fiat Stilo and Alfa 147, shown in Figure 6. Although test-cases are few, the routine could be generalized to whichever RADIOSS model, on condition that the model's cards keep the files translated formulation.

The method for the software-independence translation was carried out with the support of TextPad editor, Matlab 6.5 calculation software and Hypermesh pre-processor.
The TextPad editor is essential in order to make an appropriate macro-operation on the text files developed in RADIOSS environment; indeed, the output files from a FEM pre-processor (input for calculation) are composed by millions of lines, and therefore they determine a memory occupation of 80-150 Mbyte and a high calculation time during the translation with Matlab.

In order to reduce the calculation time, and than product development time or time to market, the lines, that are not translation object, are sponged out from FILE.D00 by appropriate macro developed in TextPad, so that Matlab works up files of 15 Mbyte at most for each translation routine.

Also the lines that contain the cards which Hypermesh automatically translates, are sponged out.
In the Figure 7, the flow chart is illustrated; this describes into details the translation procedure from RADIOSS to LS-DYNA environment.

Control Routine

The routine has to execute a control on inertia of the rigid bodies and a control for the added masses.

The routine reads in input the file "checkD00", containing the rigid bodies and the added masses cards, and after its elaboration, writes in output two text files.

The text files are the following: "check_inerzia_rigidbody.txt" and "check_massa_da_ripartire_su_parti.txt".

Materials and relative Curves Translation Routine

The routine executes automatically a translation between characters lines, that represent so-called materials and curves cards .

The routine reads in input the text file “materials_curvesD00”, containing the cards of the RADIOSS materials and of representative curves of the behaviors of the same ones, carries out the due translation elaboration and writes in output the text file: materials_curves.k”, that is the file in LS-DYNA format.

Spring and relative Materials Translation Routine

The translation algorithm executes automatically the Springs card translation. The routine reads the RADIOSS text file “springD00”, containing the cards of the springs, parts and prop associated to the same springs, finally writes in output the text file: "spring.k” , after carefully prepared procedure.

Boundary Condition Translation Routine

The translation routine executes automatically the all remaining card translation, that are synthesized with the name of Boundary Condition. Initial velocity, gravity, rigid_walls, interfaces, added masses, accelerometers, subsets, databases, airbags and sections are part of them.

The routine reads the work file “CONDITIOND00”, that, beyond containing those over listed, introduces other cards associated to them, indispensable in order to carry back the read model in LS-DYNA environment.

The routine finally writes in output the text files: ”initial_vel.k, gravity.k, rigid_wall.k, admas.k, interface.k, accelerometer.k, subset.k, airbag.k, section.k”.

Controls and Output demands Translation Routine

This routine translates the file “FILE.D01” in order to obtain the translated model title definition, the control parameters description necessary to improve the calculation accuracy, the TimeStep definition and information on the model representation in the three-dimensional space. Moreover, with this algorithm, the database necessary to obtain in output files containing results information are defined.

The routine finally writes in output the text files: ” control.k, database.k”.

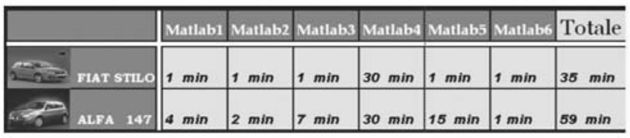
Assembly routine

The model has to be assembled, in order to be able to be launched. It’s necessary to link the several file ***.k obtained by the Matlab routine and that one obtained by the translation procedure in Hypermesh. Such operation is carried out by the last routine executed by the file “matlab6.m”, that generates a launch file called: “file_ls-dyna.k”.

Calculation time

Within every algorithm there is a temporizer that evaluates the execution time in each routine. In this way, at the end of each calculation, for the six translation routines, the number of seconds necessary to translation operation appears on video. Obviously the calculation time depends on CPU type and on applications opened during the routines execution.

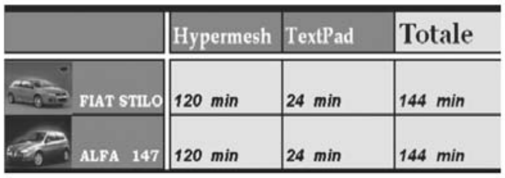
The processor on which the following times have been evaluated is an Intel Pentium 4.0, 2.6 GHertz. The Figure 8 shows the total computation time.



	Matlab1	Matlab2	Matlab3	Matlab4	Matlab5	Matlab6	Totale
FIAT STILO	1 min	1 min	1 min	30 min	1 min	1 min	35 min
ALFA 147	4 min	2 min	7 min	30 min	15 min	1 min	59 min

Fig. 8. Routine time

For the whole translation it is necessary to add the operations carried out with the TextPad editor and the Hypermesh pre-processor, whose computation times are shown in Figure 9, and to contemplate the execution times of that operations.



	Hypermesh	TextPad	Totale
FIAT STILO	120 min	24 min	144 min
ALFA 147	120 min	24 min	144 min

Fig. 9. Hypermesh and Textpad times

Following this way, the translation has been executed in a maximum time of 4 hours.

4. Conclusions

The complete translation could be developed without using Matlab, but only with the Hypermesh pre and post-processor and with the manual correction of the cards which constitute RADIOSS model. However it has been estimated that the necessary time to carry out this procedure is about 60 hours.

Therefore the developed methodology reduces a lot the product development times, in other words the Time To Market (TTM).

The TTM reduction implies, in fact, the shift from a tasks sequential flow to a simultaneous flow, in which more steps are carried out side by side, particularly on the Product Development Process beginning.

That means also an efforts concentration in a moment in which every change is little expensive, in order to evaluate several options. It is also evident that, from point of view of the costs, the maximum effort has to make during the development formulation step, since this is the moment in which the great part of product life total cost is defined.

Moreover, the development process first step concerns the realization of many activities in a virtual environment, in particular in that of the virtual prototyping, which is certainly made more intuitive, simple and fast by information methodologies. In fact, these methodologies cooperate with mechanical designers in the product development process, during virtual prototyping, greatly simplifying both design and redesign step. Thus, carrying out a new product as a vehicle, a drastic reduction in times and costs is allowed.

An example of reduction in time and then in costs is given just by this work, which influences on the calculation models' preparation time, that is the propaedeutical step to homologation tests realization, in crashworthiness optic.

In fact, such a result provides considerable industrial advantages, as the following:

- reduction in product development time (time to market),
- reduction in real prototyping costs,
- more competitive products on the market.

References

[1] W.J. Abernathy, The productivity dilemma: roadblock to innovation in the automobile industry, Johns Hopkins U.P., Baltimora, 1978.

[2] M. Annarumma, A. Naddeo, M. Pappalardo, Software independence: impact on Product Development Plan in Automotive Industries, Journal of Achievements in Materials and Manufacturing Engineering 18 (2006) 431-434.

[3] M. Bobrek, M. Sokovic, Integration concept and synergetic effect in modern management, Journal of Materials Processing Technology 175 (2006) 33-39.

[4] W.T Goh, Z. Zhang, An intelligent and adaptive modelling and configuration approach to manufacturing systems control, Journal of Materials Processing Technology 139 (2003) 103-109.

[5] P. Grote, M. Sharp, Defining the Vehicle Development Process, MTS Systems Corporation, 2000.

[6] A.J. Maunuksela, Towards a benefit measurement-based analysis method of product development process capabilities: an exploratory study, International Journal of Innovation and Learning 3 (2006) 127-143.

[7] I.P. McCarthy, Y.K. Tan, Manufacturing competitiveness and fitness landscape theory, Journal of Materials Processing Technology 107 (2000) 347-352.

[8] G. Monacelli, VR Applications for reducing time and cost of Vehicle Development Process, Proceedings of 8th International Conference and Exhibition Florence Vehicles Architectures: Products, Processes and Future Developments, Florence, (CD-ROM).

[9] L. Morel, V. Boly, 'New Product Development Process (NPDP): updating the identification stage practices', International Journal of Product Development 3 (2006) 232-251.

[10] N. O'Regan, A. Ghobadian, The strategic planning process: a navigational tool for competitive advantage, International Journal of Process Management and Benchmarking 1 (2005) 63-81.

[11] V. Pagano, Software independence impact on the product development times: application on explicit FEM solvers in automotive crashes, BD Final Project, University of Salerno, Italy, 2004.

[12] W.J Palm III, Introduction to Matlab 7 for Engineers, McGraw-Hill Professional, 2004.

[13] D. Spath, A. Agostini, A Flexible planning logic for technology planning, Journal of Materials Processing Technology 76 (1998) 76-81.

[14] K. Ulrich, S. Eppinger, Product design and development, McGraw Hill, New York, 1998.

[15] G.G. Wang, Definition and Review of Virtual Prototyping, Journal of Computing and Information Science in Engineering 2 (2002) 232-236.

[16] H.P. Wiendahl, H. Stritzke, Logistic orientated product design, Journal of Materials Processing Technology 76 (1998) 12-15.

[17] X.Y. Xu, Y.Y. Wang, Multi-model technology and its application in the integration of CAD/CAM/CAE, Journal of Materials Processing Technology 129 (2002) 563-567.

[18] R. Zambrano, V. Pagliarulo, New design methodologies and supplier partnership: impacts on Vehicle Development Process (VDP), International Journal of Automotive Technology and Management 1 (2001) 321-334.

[19] HyperMesh Theory manual, Altair®.

[20] LS-DYNA Theory Manual, LSTC.

[21] LS-DYNA 960 manual k vol. 1, LSTC.

[22] LS-DYNA 960 manual k vol. 2, LSTC.

[23] PAM-CRASH THEORY NOTES Manual V2000, Pam System International, France.

[24] PAM-CRASH SOLVER NOTES Manual V2000, Pam System International, France.

[25] PAM-CRASH SOLVER Reference Manual V2000, Pam System International, France.

[26] RADIOSS Theory manual, Mecalog, 2004.