Integrated modeling with Top-Down approach in subsidiary industries

†N. Aleixos[1]; P. Company[1] and M. Contero[2]

1 Dept. of Technology, Jaume I University of Castellón (Spain)

2 Dept. of Engineering Graphics, Polytechnic University of Valencia (Spain)

Abstract

This article considers how conceptual design of industrial products is supported by current CAD systems. The case of subsidiary industries, or first tier suppliers, that must simultaneously deal with different customers and CAD platforms, receive special attention. Conceptual design is critical, since the large variety of fundamental product data managed (not just geometry) would be specified, modeled and interrelated (i.e. functional relations), to both simplify and ensure correctness and efficiency of the next design phases of current design, and make them easy to reuse, modify and redesign in the future.

We give an approach to introduce conceptual design through top-down methodology and integrate it with final geometry. In this context, and in order to help subsidiary industries to improve their model quality, we propose the elaboration of product-oriented modeling guidelines, or "best modeling practices", instead of CAD-oriented modeling guidelines. The

† Corresponding author:
Dr. Nuria Aleixos Borrás
Departamento de Tecnología, Universidad Jaume I
Avda. Sos Baynat, s/n
Campus de Riu Sec, E12071 – Castellón (Spain)
Tel.: +34 964 72 81 23
Fax.: +34 964 72 81 06
e-mail: naleixos@tec.uji.es

approach has been validated by testing the conceptual design tools of two commercial high-end CAD systems at use in many subsidiary automotive industries.

Keywords

Conceptual engineering design. Top-down design. Integrating fundamental data structure. Product-oriented modeling guidelines. Product data quality.

Acknowledgements and grants obtained:

This research was partially funded by the FEDER CICYT project 'Implementing Design and Manufacturing Advanced Technologies in a Concurrent Engineering Environment. Application to an Automotive Components Manufacturing Company' (Project No. 1FD97-0784), and a grant by the Education, Universities, Research and Development Secretary of the Regional Government: 'Development of New Methodologies in the Projectual Design to Solid Modeling of Digital Mockups' (Ref. POST00-14-126).

The first author's three-month sojourn in Elasis (from FIAT) and Federico II University (both in Naples, Italy) was also very important to the research. Thanks are due to Prof. F. Caputo (from the Federico II University) and to Eng. Carmine Di Martino, responsible of the I.C.T. CAD-CAE (Information & Communication Technology) at Elasis, in the development of this research.

1 Introduction

The diversity of partners and software tools involved in the product development process leads to a complex flow of product data, and the hierarchical structure dominant in many industrial sectors increases this complexity; because different data requirements, design responsibilities and CAx tools appear according to the tier in question. Subsidiary industries suffer particularly the data exchange problem, because their customers are different original equipment manufacturers (OEMs), each one forcing them to adopt their particular CAx systems. In addition, the lack of standards supporting the capabilities of current commercial CAx applications, the variety of downstream applications making use of product data and the heterogeneous nature of work teams lead to poor quality CAD models. In a previous work, we proposed a Linguistic Model of product data quality (Contero et al., 2002), where three levels of quality were distinguished: morphological, syntactic and semantic/pragmatic level. The morphological level is related to the geometrical and topological correctness of the CAD model, while the syntactic quality evaluates the use of the proper modeling conventions. Those levels have been subject to intense work in the last few years (VDA 4955, 1999), and commercial applications to automatically verify the quality criteria in CAD models do exist (i.e.: Parametric Technology: ModelCHECK, Trascendata: CAD/IQ, Avatech Solutions: DesignQA, etc.). However, in our approach, semantic/pragmatic quality is closely linked to the conceptual design phase, and so-called *design intents*. Because if those design intents are managed together with the geometrical model, we get models of better semantic quality, and this greatly simplifies modifications and redesigns. In other words, the semantic/pragmatic level takes into account the CAD model's capability to reuse and modify.

Information related to this level is difficult to find, because it belongs to the enterprise's know-how. Hence, while many enterprises have developed their own internal *modeling guidelines* where the *best practices* for improving CAD model quality are recorded, they are related to the domain of knowledge management, so they have a strategic value and remain hidden from public diffusion. Furthermore, many CAD vendors (and some researchers) claim that implementations for the capture of design intents are now direct and easy. According to our experience, those implementations are definitely not so straightforward. They are poor and rudimentary and, in most cases, they are almost inexistent. Only in some current high-end CAD systems the design intents can be indirectly introduced through the so-called skeletons or control structures.

Those problems are especially encouraging for subsidiary industries, because their particular position forces them to deal with different clients and their different CAD platforms. In fact, some OEMs have developed their own mandatory modeling guidelines, where the best practices for improving the way design intents can be introduced via fundamental structures are described (BMW, 2002).

We have studied two widely used high-end CAD systems, in order to find out how they support design intents, and how they transfer all these critical data and functionality into the final geometry of the model. We agree that a *generic modeling guide* is necessary to fix good practices and reinforce cooperation and standardization providing support for semantic quality in the conceptual design phase. Hence, we propose a generic methodology as a frame to help subsidiary industries in the creation of such modeling guides. But we claim for the development of *product-oriented* modeling guides, since this is an efficient way to keep the know-how about each product independent from any particular CAD system. In other words, in a subsidiary industry, guides must not be application-oriented, but product-oriented, in order to preserve the know-how implicit in every design for future reuse, when working for a different client in a different CAD platform.

The next section reviews the current state of the art in the conceptual design subject. Then we focus on our methodology to integrate the know-how and the detailed design, afterwards we test it studying how some commercial CAD tools support this methodology with the modeling of two real automotive products, and finally we illustrate the generic modeling guide outlined. To conclude, we summarize the research work presented here and discuss some main conclusions.

2 Conceptual design

While the old fashioned design-by-drawing methodology is being substituted by the *design-by-digital-prototypes*, the ideal engineering design support system is still an academic challenge. According to Ullman (2002), the conceptual engineering-oriented CAD tools should have "friendly and intelligent tools to assist in the creation of *functional product models*; to allow a fast and easy introduction of design intents and serve to check the completeness of design". Unfortunately, functional models are now in the earliest stage of their development. In fact, functional meaning is not only required in design but also in all other disciplines involved in the life cycle of the product. This is not, however, the specific subject of this article and more details can be found in Kraker et al. (1997), Bronsvoort et al. (1997) and Holland et al. (2000).

While functional modelers are still an academic challenge, old CAD systems based on CSG and B-Rep representations lacked two capital requisites to fit the needs of designers and how they work. First, design intents were not preserved (even a simple tangent condition is lost after resizing the circle radius). Second, their semantic level was too low. In other words, geometrical description was based on extremely simple elements (or "primitives").

Feature-based models improved the amount of functional information contained in the models. In fact, many CAD systems have introduced *form* features (detailed specifications of all individual parts), and even some *fit* features (detailed specifications of assemblies), in order to assist the designer in the creation and maintenance of real links between form and function. However, current feature-based modelers do not properly link the functions to the final geometry. At present, once the form is defined, some functionality is automatically or manually incorporated into the model. But we are still a long way from achieving automatic form definition from functional requirements. In other words, we can add functionality to models, but we cannot create models from their functionality.

Finally, Constraint-based models, provided by parametric and variational CAD systems, improve the modeling of form and allow re-usable and re-definable models to be created. However, modeling large complex assemblies with those systems is a difficult task (Anderl et al., 1998). Of course no single parameter controls almost any single functional characteristic, and design intents can be introduced only after being drastically converted into roughly equivalent parameters.

In this context, the current state of the art considers two *engineering design methodologies* as being capable of coping with two opposite scenarios: The bottom-up modeling approach is commonly used in most industrial sectors, while the top-down approach is not yet in widespread use because it is only possible in long established (or mature) sectors, where the functional features of the product and the relations among them and their geometry are completely defined. We could say that traditional design is well fitted to bottom-up, while conceptual design is closer to the top-down approach. In fact, bottom-up methodology is available in feature-based CAD systems, where the user builds part models from scratch and later on generates the corresponding assemblies. Nevertheless, the bottom-up modeling approach is a detailed design methodology with a posteriori consideration of design intents. In the top-down design approach some main criteria and product specifications are initially fixed in the higher hierarchical level. These specifications must be successively inherited by every lower level product component without loose of their explicitness, i.e., the domain elements adopted in the higher lever of the model, which represent decisions and conceptual engineering, must affect the occurrence of the domain elements at lower levels. Yet, it is not clear at all how to perform the transfer from conceptual design to top-down. Fortunately, the term *form* implies both the final shape and dimensions, and can be divided into "architecture" and "shape", where the former has come to mean the fundamental structure that maps the function to the form, and the latter refers to the final, refined form. Hence, the transfer from design intents to parameters may be carried out through the fundamental product structure, made in the early phases of the project from higher to lower levels. Such fundamental structure may contain the whole project criteria and the basic elements. This structure is advantageous because it allows modifications to functional features to be propagated throughout the whole product in an automatic and consistent way. In fact, some CAD systems allow a large range of relations to be introduced through fundamental structures, constraining the system to accept just those designs that accomplish the required functionality.

Nevertheless, it is the user who, through the particular filter of each enterprise philosophy, captures the *design intents* (the know-how: design methodology, manufacturing criteria, organization, etc.), and defines the fundamental structure (Brunetti et al., 2000; Whitney et al., 1999). In this way, conceptual engineering design methodology has been introduced in big companies through particular top-down methodologies and parametric models tailored to cope with their particular needs and requirements. But this process is too complex and expensive to be mimetically reproduced in small and medium enterprises. The problem is even worse in subsidiary industries, because they are forced to work *simultaneously* for many main companies that use *different* CAD systems. And this means that they are even forced to inherit the conceptual engineering methodology developed by each of their customers. In other words, the relation with the big companies subordinates the technological level of the subsidiary industries.

To sum up, subsidiary industries need a modeling design methodology compatible with the tools and methodologies used by their customers, and simultaneously tailored to take advantage of their own capabilities and size. In addition, and due to the small size of their design teams, many of them do not have the capability or the human resources to develop their own modeling design methodology to force good practices, while transition from traditional design to design based on new conceptual methodologies must be done with no damage to the production process, and the efficiency of this new approach must become apparent in a short space of time. Hence, as far as the small design teams of subsidiary industries must develop their own modeling design methodology ensuring compatibility with all their customers, the proposed solution is to do this with the help of an external "firing team", where academic specialists or consulting firms provide the required support. Alternatively, a generic methodology may serve as a frame to help subsidiary industries in the self-creation of such modeling guides.

3 Functional modeling. Integrating conceptual and detailed design

It is well established that, for theoretical functional modeling to be applied early on in design, the workflow must always go from global or abstract to detailed or particular design specifications. Information inheritance and propagation are other desired qualities of an organized structure of a product model (or assembly). Inheritance and propagation must be done in such a way that when a modification happens on a certain level, this action is reflected on all its subordinate (lower) levels. Hence, the changes are instantaneous and automatically propagated throughout the created structure.

In actual top-down oriented parametric CAD systems, inheritance and propagation are to be accomplished through a fundamental structure available to all designers at any moment. This fundamental structure, or *product skeleton*, is supposed to contain all the criteria and fundamental elements, such as assembly conditions for components, housing of the different parts and sub-assemblies, parameters, reference surfaces and curves, and style surfaces. This procedure is very useful when design team collaboration is required in the modeling of the same product (Vila, 2000), because the problem can be divided into simpler parts that can be modeled separately, while interrelation between individual components or sub-assemblies will later guarantee the consistence needed to create a complex project formed by such parts. In other words, this design scheme encloses several advantages, such as the improvement in managing of complex assemblies and organization of complex projects, the work distribution for the concurrent modeling of the different parts of the product, the flexibility of models during modifications, modeling autonomy for each project designer and reuse of data for future new products.

The skeleton behaves as a three-dimensional scheme, allowing the sharing of critical project information through all the sub-assemblies and linked parts, and allowing the interdependencies to be controlled within the same sub-system. Hence, the skeleton allows the top-down approach to set and manage the functional relations and interactions among the whole components of a system.

Briefly, a fundamental structure may be used to introduce design intents by way of a top-down oriented tool of a parametric CAD system. But, the fundamental structure is neither easy to define, nor trivial to manipulate, and, hence, a methodology is proposed to convert design intents into generic fundamental structures, not tied to a particular CAD system. In other words, a generic (and, hence, particularly useful to subsidiary industries) guide to use current CAD systems in an environment of top-down design is established.

According to Pahl and Beitz (1996): *it is useful and common to divide this flow of work into four main phases: Planning and clarifying the task, Conceptual design, Embodiment and Detail design*. These main phases or *working steps* were described in order to suit the flow of work during the process planning and designing to the terms in VDI (Verein Deutscher Ingenieure) Guidelines 2221 and 2222 (VDI, 1987). Other related design methodologies have been proposed too (Shah, 1995; Cross, 2000), but according to our own experience, we propose a new five steps approach, tailored to take the maximum benefit from top-down conceptual design tools, whose descriptions and methods are outlined following:

1. *Earliest product planning and organization*
2. *Establishment of the product conceptual design*
3. *Arrangement of specification principle into a hierarchical-fundamental structure*
4. *Integrating the hierarchical-fundamental structure to a CAD structure*
5. *Integrating modeling detailed design*

Figure 1. Near here

The approach distinguishes two different tasks embedded in what is usually known as *embodiment design* (figure 1). This division is aimed to separate the management of conceptual information, typical of the embodiment design phase, from the task of transferring and integrating this conceptual data into a commercial CAD system tool. This strategy ensures that the conceptual information is always kept aside from the constraints imposed by particular software in the creation of CAD models. Hence, we have included this as a separate phase to provide the most of know-how into the CAD model, giving as well the possibility to try other alternative solutions without having last or detail geometry. A detailed description of these five steps methodology is included next. It is illustrated through two practical experiences, which were used to check its validity:

- An **automotive radiator** model was designed with Pro/ENGINEER from PTC (Parametric Technology Corporation) using its design intents tools.
- A fundamental control structure for a first virtual prototype of an **automotive gearbox** was developed with UG/Wave from UGS (Unigraphics).

3.1 Earliest product planning and organization

Is in this phase where the basic ideas of the product: the sketches, links and the specification of the project dimensions, functionality, accessibility and the company standards to manufacture the product, are elaborated. The product design process can be structured as a set of activities interrelated by means flow of information to produce a specific output. In this way, the whole design task of a new product can be structured or organized through several activities, supporting the study of other solutions and providing early knowledge, and allowing iteration to the best/optimal solution for the design task. Figure 2 shows the IDEF0 model for the design process of a car radiator product.

Figure 2. Near here

PDM (Product data management) systems can be used for managing sketches, links and other project specifications, supporting the development, updating plans, and providing the management of issues that could come up during the development of a project. PDM systems suit well for this purpose, allowing us to integrate nearly any sort of information with the design process.

The OEM begins the process by sending the tentative specifications of the new project (dimensions, functionality, accessibility and the company standards to manufacture the product). These specifications usually suffer minor changes during the initial phase of the design process until they are fixed. To isolate and define the functions of the feature/part, we need to break the part down to its simplest functions. The two particular specifications of our test are briefly described below.

Case of study: Radiator

The most obvious functions in the radiator that are taken into account are: cooling a given volume of water, providing piping connections with the engine cooling circuit, providing fixations that guarantee immobility, providing fixations that guarantee easy assembly and disassembly, and respecting housing requirements.

Case of study: Gearbox

A special three-axis car gearbox (common car gearboxes have only two axes or shafts) was designed. The requirements to its design are: one main shaft, two secondary shafts (main and auxiliary secondary shaft, respectively), one differential shaft, and respecting housing requirements.

All this structure was to be fixed to the engine by means of a particularly shaped box. For the sake of vehicle habitability, a very important design criterion was the minimization of the gearbox housing inside the engine housing. The basic data used in the control structure are taken from an external calculus application particularly designed to obtain optimal parameters to construct a gearbox (Mosca, 2001).

3.2 Establishment of the product conceptual design

This conceptual phase starts with determining the overall functions and the main subfunctions to obtain the so-called *principal solution* (Pahl and Beitz, 1996) as a combination of the solution principles established in the conceptual design. The functional model for a particular product is organized to obtain the simplest sub-problems.

Conceptual design is mainly modeled through design parameters. Additionally, some sort of "divide and conquer" is applied to obtain simpler sub-problems controlled by particular parameters. Throughout this phase, we use as knowledge resources both the history of technical problems of products that have been marketed previously, and the record of technical decisions that have been made in prior projects.

Here, we need to prioritize the functions, identify trouble spots and be specific. ldeally, only one function should have top priority. This step can be one of the more difficult, especially it creative design principles are used and the parts are designed to incorporate many functions. This is probably the most overlooked step, but it is also the most important. All parts can be broken down into their basic functions, and several solutions can be considered.

Case of study: Radiator

The problem of the car radiator was divided into simpler problems, so its assembly was separated into four main "zones", as shown in Figure 3: zone 1 (central body), zone 2 (output dump), zone 3 (entry dump sub-assembly) and zone 4 (fan housing).

It is worth noting that the successful zone division was amazingly similar to the level of "sub-assembly" in the traditional design-by-drawings methodology of three levels of assembly, sub-assemblies and parts.

Figure 3. Near here

Case of study: Gearbox

Unlike the car radiator, the gearbox is not only determined by requirements of space, but dynamic requirements are also necessary to manage the movement transmission. Both, the gear shafts parameters (their location in the three dimensional space), and the fundamental parameters to manage the movement transmission and the positioning and size of the gear wheels were considered (Table 1).

Table 1. Near here

3.3 Arrangement of specification principle into a hierarchical-fundamental structure

This is the phase where the functional relations between components are first designed, and after they are organized within a hierarchical-fundamental structure. In this phase, decisions about solutions obtained in the previous phase are made, and embodiment design is here carried out to conclude with the solutions that match better with the requirements established in the previous phases. That is, the ideas and requirements of functionality and housing concluded in the previous phase with the *principal solution*, are arranged into a hierarchical organization to provide several possible solutions that match those requirements. From all possible solutions, the more optimal form design variants of assemblies are taken into account to converge into a so-called *definitive layout*, where this definitive design defines the arrangement of assemblies, components and parts, as well as their geometrical shape, dimensions and materials.

3.4 Integrating the hierarchical-fundamental structure to a CAD structure

Is in this phase where fundamental data is introduced and converted into CAD data with tools provided for design intents. One important characteristic, according to top-down approach, is that the design work can be carried out with incomplete geometry (just important details are represented). This approach has been also implemented to integrate the conceptual information into the CAD systems used. As an output from this phase, the definitive layout is obtained.

Since generic CAD data need to be selected in order to match project specifications, we need to convert the functionality and housing requirements established in the *definitive layout* from earlier phase, into understandable information for the CAD system where the product design is going to be modeled.

Fundamental CAD data structures can be created from different elements. Choosing the appropriate type of elements is important, because this greatly improves the links between the fundamental data structure and parameters that control the conceptual design. Our main proposals are listed below:

1. Reference datums to limit the housing of the model.
2. Reference axes to fix axial elements in the model, such as cooling liquid input and output pipes, or brackets to hook different engine components.
3. Reference curves to create complex shapes or limit new geometry.
4. Reference coordinate systems to benefit the assembly phase and other future operations related to the manufacturing phases (i.e. creation of moulds).
5. External carry-out from other models to limit the housing and determine the shape or location of new geometry.
6. Inter-part relations in order to link mathematical information between components through complex programming conditions, to prevent absurd solutions due to wrong values of certain parameters.

Case of study: Radiator

The radiator skeleton was mainly determined by datums, delimiting the model housing (Figure 4). The datum PL_SUP separates the water entry dump from the central body, the

PL_POST and PL_ANTE set the limits of the central body separating the component that houses the fan, the PL_INF separates the central body from the output dump, and the PL_SX, PL_DX, PL_SUP, PL_INF, PL_ANTE determine the size of the central body.

Figure 4. Near here

General levels were kept as simple as possible and skeletal details of each zone were hidden from the rest of the structure. To achieve this, the critical project data was divided into sub-skeletons created in successive steps. The organization of the whole skeletal structure in sub-skeletons linked to particular zones improves and simplifies the management, because not all information is needed in all the zones, so it is better to organize it in such a way that only the required information for each zone is present at a particular level. Subdivision prevents local information of a particular zone from being defined on a more general level. However, mutual relations among sub-skeletons must be guaranteed and defined in a complete and consistent way. This is why a tree structure is required. Pro/ENGINEER allows the same model structure to be used to link each skeleton to its own zone, and to establish the required mutual relations.

The data in a sub-skeleton can be transferred to the rest of the elements on lower levels. This information can also be used by elements of higher levels, but this transfer must be done through the global skeleton, to avoid circular references. The transfers allowed in hierarchical structures are performed from the higher skeletons to the rest of the assembly parts (paths A-b, A-c, A-B, A-C, A-D in Figure 5), from each part to the rest of the parts from the same level (path b-a-c), and from any parts to any other parts belonging to a higher or lower level through the global skeleton (path c-A-D).

Figure 5. Near here

The water output dump and its skeleton is shown in Figure 6. As before, many references are datums to determine its housing, but location axes were also added to some features, such as pipes or brackets for fixing it to the car engine. It is worth mentioning that datums to control unmoulding were considered. The output dump sub-skeleton also contains the references to link it to the main skeleton. The sub-set of references (in Figure 6b) represents the imported reference data from the highest-level skeleton.

Figure 6. Near here

The skeleton used for the entry dump contains fundamental data inherited (through a *copy external geometry*) from the main skeleton, from the output dump sub-skeleton, and other criteria created only for this component. Figure 7a shows the exploded view of the two main components that make up the entry dump sub-assembly, and Figure 7b shows the skeleton for the upper component. Figure 8 shows how the fundamental data are shared from global skeleton to entry dump sub-assembly, which is represented by the path a-b. Fundamental data, shared from output dump component (zone 2) to the entry dump sub-assembly (zone 3), is represented by the path b-a-c. And lastly, paths c-d and c-e represent fundamental data inherited from two components of zone 3 to create detailed geometry. The fundamental data or control information from the output dump is now contained in the main skeleton (path b-a) and can be referred to by any component or sub-assembly in any lower level.

Figure 7. Near here

Figure 8. Near here

Case of study: Gearbox

A typical *control structure* (CS) by Unigraphics consists of a geometrically simple assembly model that defines the high level geometric layout of the product. This layout represents the location and size of the key components and the interfaces between the main sub-systems.

For instance, main dimensions such as variables, datum planes or axes specifying key locations, 2D or 3D curves representing critical mating interfaces, and surfaces or solid volumes defining envelops within which sub-systems must be confined. The assembly model provides a set of global, product-level geometric constraints that can be used as a control framework for more detailed sub-system and component design.

UG/Wave separates global relationships that may affect several sub-systems in the product design from local relationships (which are applied only to particular sub-assemblies). Global considerations are fitted into the *control structure*, while local inter-part relationships are isolated within the detailed sub-system design (Gott, 1999).

The scheme that UG/Wave proposes to transfer conceptual project data consists in creating a link between the control structure (CS) and what is the final development or *product assembly* (PA), and to do so some special parts of the CS are created. They are called *start parts*, and in the CS tree levels are the last results of each branch. So, the high-level layout geometry in the CS is divided into different sub-systems and fed into the detailed product design via *start parts*, which provide the interface between the current state of the overall CS and the detailed part geometry modeled at component level. *Start parts* typically contain a subset of the CS geometry, providing a set of external design constraints for the part designer. The link operation thus consists in the creation of a certain number of *linked parts*, this time belonging to the PA tree structure.

In our particular case, this scheme results in Figure 9. The reference set contains the geometry selected from the ANT (Assembly Navigator Tool) or from the graphics display. Several reference sets can exist in the same component, and the same feature can lie in different reference sets. This is useful for the creation of the *start parts* of the control structure of a project.

Figure 9. Near here

It is important then to state the steps to start performing a new project in Unigraphics: 1) the Control structure creation (a critical step due to its relevance in the project); 2) the definition of the *start-parts*, representing the future output of the control structure; 3) the product assembly assessment; 4) *Start parts* and *linked parts* linkage; 5) insertion of the *linked parts* in the *product assembly*; 6) the detailed modeling of *product assembly* parts; and 7) the verifications of the created links.

The correct propagation of the modifications can be tested by modifying the key parameters of the CS, the transmission of such modifications being evaluated in the Product Assembly. High-level product specifications are passed down and are ‘imposed’ as constraints on the lower level sub-assembly and component design process.

Figure 10 shows the entire structure of our gearbox product control structure. It has been divided into four sub-assemblies, with one component in each. In this particular case, fundamental project parameters were included in the assembly *Cs_gear_3_axis* defined at top level of the control structure. In this control structure assembly, the double triangulation of the shafts and the interface to locate the shafts and gearboxes are represented. The axes representing the main shaft, the main secondary shaft, the auxiliary secondary shaft and the differential shaft, govern the rest of the control structure at top and lower levels. This three-dimensional layout was designed to create the interface to configure the gear housing, and other parameters required to manage the whole assembly.

Figure 10. Near here

To set the original diameter of the $\mathbf{X}_i$ main shaft wheels, which depends on the input data of transmission τ, it was necessary to define the simple top-level relations shown in Table 1.

Figure 11 shows the parameters in the visual editor and the control structure for the triangulation of gear shafts at the top level, and for the main shaft wheels with their original diameters, as the result of the relations in Table 1.

Figure 11. Near here

In Figure 11, the low-level detail geometry has been represented, showing only the housing wheels. In Figure 12, the control structure for the other shaft wheels is obtained in a similar way, i.e. by introducing tangency conditions, defined by means of basic geometry in the control structure.

Figure 12. Near here

The expressions of tangency have been defined as relations in the immediate lower level, denoted by *Cs_Secondary_Shaft* and *Cs_Aux_Secnd_ Shaft* in Figure 11, which appeared above.

3.5 Integrating modeling detailed design

Is in this last phase where the translation from fundamental data structure to final geometry is performed. All details of geometry are created, involving the fundamental CAD structure, output from the preceding phase. The project documentation is generated and the new product is achieved.

At this point, we need to manage, in a well-organized way, the interdependency of different components that make up the whole project. In other words, when a designer modifies a component, those components of the assembly affected by such modification ought to be automatically updated.

These updates between different components take place by means of *external references* or *links*, that is, one component takes as reference points certain entities that contain another component, allowing the changes to be transferred to other components on a master-slave based philosophy (Bigelow, 1995).

When an assembly is very complex, the data need to be managed in order to see just the relevant information in the graphics display (so you do not see datums, sketches or other simple components used to create the final geometry), which can be achieved using a level structure. Furthermore, tools exist to control the amount of conceptual information in a subassembly or component to keep only the information relevant for detailed geometry, which can now be achieved easily using the fundamental data stored and organized into the control structure, by simply modeling as usual with the CAD application.

4 Managing the fundamental data structure

Checking that changes, reuses and redesigns can be done is a heuristic but straightforward way of determining the validity of the proposed models.

Case of study: Radiator

Once the radiator was finished, some changes to the skeleton were made in order to check the final model. These changes had to be made by the project leader because they did not affect detail geometry but the datums contained in the skeletons of the project. So, values of length, height, width, draft angles and others were changed. The changes were automatically propagated to the rest of the geometry, and a valid model was obtained (see Figure 13).

Figure 13. Near here

Case of study: Gearbox

In order to evaluate the performance of our fundamental control structure in UG/Wave, some variations in the values of fundamental parameters were tested. Figure 14 shows the initial configuration of the car engine gearbox. Meanwhile, in Figure 15 we can appreciate how the geometry of main shaft wheels changes when parameters of motive transmission are modified, and how the rest of the shaft wheels accommodate their central position (i.e. fixed to the top-level control structure) to their tangency conditions. In Figure 15 some parameters of motive transmission are changed, forcing one variation in the diameters of the related wheels compatible with tangency conditions. In particular, due to the direct proportionality between the τ parameter and the diameter of the wheel implied, the value of the differential wheel diminishes when τ_{fin} decreases, while the diameter values of the first and second wheels of the main shaft increase when parameters τ_1 and τ_2 decrease. Additionally the positions of the first and third wheels of the main and secondary shafts are automatically modified (Figure 16), the first wheel becoming nearer with respect to the differential wheel, and the third wheel further from the fourth wheel.

Figure 14. Near here

Figure 15. Near here

Figure 16. Near here

Some minor differences in the way the design intents are implemented in *Top-Down* technique of each CAD system were underlined above, but, to conclude, we must highlight a big difference in the implementation of *Top-Down* technique: the independence of the fundamental structure from the product assembly. While Pro/ENGINEER has its *skeleton* highly tied to the product assembly, UG keeps its *control structure* out of it. The main advantages provided by this independence are:

- The upper levels can be isolated. Hence, the leader of the project can manage the "conceptual model" without having any solid model in memory (model detailed geometry is not required to see changes in the product, because simplified geometry is implicit in the fundamental structure).

- The flexibility and robustness of creating other similar products with some common structure (different products use part of the same fundamental structure), allowing a common management of critical data at the upper levels, and avoiding successive modifications of the different products. This way of working allows the rapid development of product variants or families from common product architecture, obtaining a better integration and management of entire complex products.
- Allows simulating a tentative behavior of the simplified and schematic product (without changing final geometry of the model). This checking of critical aspects on the fundamental structure is a low cost strategy for early preventing crashes on the model, and even makes it reasonable the exploration of alternative solutions, that otherwise would not be considered.

5 Modeling design guidelines for design intents

As we have noted, fundamental structures differ greatly from one CAD system to another. And, to gain real advantage from using them, specific design modeling guidelines must be used. From our experience in capturing design intents from the fundamental data project, we can suggest some generic tips and advice for good modeling practices in the conceptual modeling phase, in order to create the fundamental structure within a CAD system. The main tips suggested are the following:

- **Create reference data inside every feature**, in order to avoid collecting data in the model that can overlap with the model geometry.

  Also, the **reference data should be organized into different levels**, in such a way that there is one level structure assigned to the reference features and another to the geometric features.

- **Refer the reference features to a global coordinate system**, in order to make later modifications to the fundamental structure easier.

  This will benefit the assembly task, avoiding the regeneration or updating of the model, whenever a component is modified or substituted. And it will also make it easier to create the deformations of the model geometry required to obtain its mould.

- **Choosing the "proper" reference in the creation of new features**, to avoid undesirable interdependencies that can provoke model crashes in later modifications.

  So, global reference features are always preferred to geometry features.

- We should always **use reference curves and surfaces to create complex shapes**. This will avoid the loss of the parameters of a complex surface or solid feature when the function cannot be performed or causes aborts, like crashes. In this case, complex reference parameters will still be kept available inside the fundamental structure.
- **Be extremely careful when choosing among different types of features** (such as *revolution* or *sweep* features), because subsequent transformations or operations on them (*shell* or *offset* features) can provoke crashes in the regeneration of the model. This is highly dependent on the geometric kernel of each application, so we need to know how it works internally to suggest the best particular strategy.
- **Perform the "finishing" operations at the end** of the fundamental model tree, particularly those finishing operations best accomplished by current 3D geometrical kernels (like rounds and chamfers).
- **Promote the use of the level of independence property** of the fundamental structure as far as possible. The creation of fundamental structures is very complex and they are highly related with the product they belong to. So, we should start by creating simpler

fundamental structures and gradually add complexity, in such a way that they can be reused, even separately, in the creation of other products.

In spite of their particular value, the above tips constitute an example of the contents that a style guide must contain to meet the general objective of *data quality*, interpreted as "fitness for use". In this way, the style guide becomes the main support on which the strategy of semantic quality is constructed. It covers two different objectives, first it acts as a *repository* of those good practices on CAD system management, which constitute the know-how of engineers/designers, and, simultaneously, it acts as a *training tool* for freshman designers, helping them to rapidly acquire and take benefit from the experience accumulated in the firm they work for. It is a sort of rudimentary "knowledge management" strategy, in the sense that a regularly updated style guide favors the dissemination of such "know-how" and "best practices" by pointing out the way towards the generation of models with a high semantic quality.

6 Conclusions

Subsidiary industries that must simultaneously deal with different customers and CAD platforms have serious problems to support such heterogeneous environment in their product development process. In this context, conceptual design phase is critical to ensure correctness and efficiency of the next design phases. Current commercial CAD systems are not specially tailored to support conceptual design. For that reason, we have performed a study to improve the integration of conceptual design activities with available CAD capabilities. After analyzing modern approaches to the design process, we propose a five steps model, fitted to take the maximum benefit from available top-down conceptual design tools. Our model distinguishes two different tasks embedded in what is usually known as *embodiment design*. This division allows to separate the management of conceptual information, typical of the embodiment design phase, from the task of transferring and integrating this conceptual data into a commercial CAD tool. This strategy ensures that the conceptual information is kept aside from the particularities imposed by a specific CAD system. With this objective, our approach introduces conceptual design through top-down methodology and integrates it with final geometry. In this way, subsidiary industries can improve their design process, by means of the elaboration of product-oriented modeling guidelines, or “best modeling practices”, instead of CAD-oriented modeling guidelines.

The approach has been validated by testing the conceptual design tools of two commercial high-end CAD systems at use in many subsidiary automotive industries. Bottom-up and top-down strategies are been considered to support this initial design stage. Top-down is preferred because it is better fitted to supporting Concurrent Engineering. Nevertheless, top-down methodology is not easy to implement, especially in subsidiary industries, because CAD systems are not easy to use in the field of design intents, especially if incompatible requirements appear that must be satisfied because of different customers constraints. Efficiency-oriented strategies must be developed and design modeling guidelines, where good practices are explained, are required.

Current CAD systems have very different flavors in the way design intents can be introduced through fundamental structures. Hence, a common modeling guideline and fundamental structure is out of the question. The aim must be to get "transparent" methodologies, i.e. methodologies completely independent from the application being used. In the same time, those methodologies must exploit the particularities of every model, to get the most reusable en re-designable models.

Using a *native* CAD and exchanging information between the native and the *imposed* CAD is also very difficult. Although it has not been described in this paper, since it was not the aim of this work, we also suffered limitations in data exchange during the test. We can therefore emphasize that the current limitation of simultaneous use of different CAD systems in subsidiary industries is due to the absence of a good neutral format that really supports parametric models.

7 References

Anderl, R. and R., Mendgen, 1998, Analyzing and optimizing constraint-structures in complex parametric CAD models, Geometric Constraint Solving and Applications, Springer-Verlag, Germany, 1998.

Avatech Solutions: DesignQA.
http://www.avatechsolutions.com/products/software/avatech/prescientqa/designqa.asp

Bigelow D., 1995, Thinking Pro/ENGINEER: Mastering Design Methodology, Onword Press, 1995. ISBN: 1566900654.

BMW A.G., 2002, Guidelines to using Pro/ENGINEER intra BMW and subcontractors, http://zulieferer.bmw.de/en/tec/ca/d/e_proe/regelwerk/richtlinien/index.html, accessed 01-2002.

Bronsvoort, W.F., R., Bizarra, M., Domen, W., van Holland and K.J., de Kraker, 1997, Multiple-view feature modelling and conversion, Geometric Modelling: Theory and Practice – The State of the Art, Strasser, W., Klein, R. and Rau, R. (eds), Springer, Berlin, pp 159-174.

Brunetti, G. and B., Golob, 2000, A feature-based approach towards an integrated product model including conceptual design, Computer Aided Design, Vol. 32, pp. 877-887.

Cleetus, K.J., 1992, Definition of Concurrent Engineering, Concurrent Engineering Research Center Technical Reports, CERC TR TM 92-005.

Contero, M., P., Company, C., Vila and N., Aleixos, 2002, Product Data Quality and Collaborative Engineering, IEEE Computer Graphics and Applications, special issue in Graphics in Advanced Computer-Aided Design, May-June 2002, ISSN 0272-1716, pp.32-42.

Cross, N., 2000, Engineering Design Methods. Strategies for Product Design. John Wiley & Sons, England, ISBN: 0-471-87250-4.

Gott, B., 1999, UG/Wave and Microsoft's DNS: Aligning design engineering computer technology with the product development process. Cambashi Ltd, 1999, Registered in England and Wales No. 1360554.

Van Holland, W. and W.F., Bronsvoort, 2000, Assembly features in modeling and planning, Robotics and Computer Integrated Manufacturing 16(4): 277-294.

de Kraker, K.J., M., Dohmen and W.F., Bronsvoort, 1997, Maintaining multiple views in feature modeling, Solid Modeling '97, Fourth Symposium on Solid Modeling and Applications, 14-16 May, Atlanta, GA, USA, Hoffmann, C.M. and Bronsvoort, W.F. (eds), ACM Press, New York, pp 123-130.

Mosca, F., C., Di Martino and N., Aleixos, 2001, Complex CAD project management by the means of designing criteria control tools. Deployment of a vehicle gearbox archetype with the aid of WAVE by UNIGRAPHICS, XII ADM International Conference, Rimini (Italy), September 5th-7th, 2001.

Pahl, G. and W. Beitz, 1996, Engineering Design: A systematic Approach. Springer-Verlag, London, ISBN: 3-540-19917-9.

Parametric Technology: ModelCHECK.http://www.ptc.com/products/proe/modelcheck.html

Pawar, K.S., K., Thoben and R., Oelhmann, 1995, The Development of a Conceptual Model and a Knowledge Platform for Concurrent Engineering, CE95 Conference pp.: 487-494.

Prasad, B., 1996, Concurrent engineering fundamentals. Integrated product and process organization. Vol I & II, Prentice Hall.

Pro/ENGINEER Guides, 2000, Top Down Design, PTC Inc., ISBN: n.a.

Shah, J.J. and M. Mäntylä, 1995, Parametric and feature based CAD/CAM concepts, techniques and applications. John Wiley and Sons, New York, ISBN 0-471-00214-3.

Shum, S.S.P., Lau, W.S., Yuen, M.M.F., Yu, K.M., 2001, Solid reconstruction from orthographic views using 2-stage extrusion. Computer Aided Design, Vol. 33, pp. 91-102.

Trascendata: CAD/IQ. http://www.transcendata.com/products_cadiq.html

Ullman, D.G., 2002, Toward the Ideal Mechanical Engineering Design Support System. Research in Engineering Design, http://www.cadsociety.org/wholepage_summit.html, Vol 13, pp55-64.

VDA Recommendation 4955/2: Scope and Quality of CAD/CAM Data. VDA e.V., Frankfurt, Germany, 1999.

VDI Design Handbook 2221: Systematic Approach to the Design of Technical Systems and Products. (transl. By K. Wallace). Düsseldorf: VDI-Verlag 1987.

Vila, C., M., Contero and P., Company, 2000, Extended Modelling, a Tool for Cooperative Design, 6th International Conference on Concurrent Enterprising, Toulouse, 28-30 June 2000, ISBN 0-9519759-9-4, pp 171-179.

Whitney, D.E., R., Mantripragada, J.D., Adams and S.J., Rhee, 1999, Designing assemblies, Research in Engineering Design, Vol. 11, pp. 229-253.

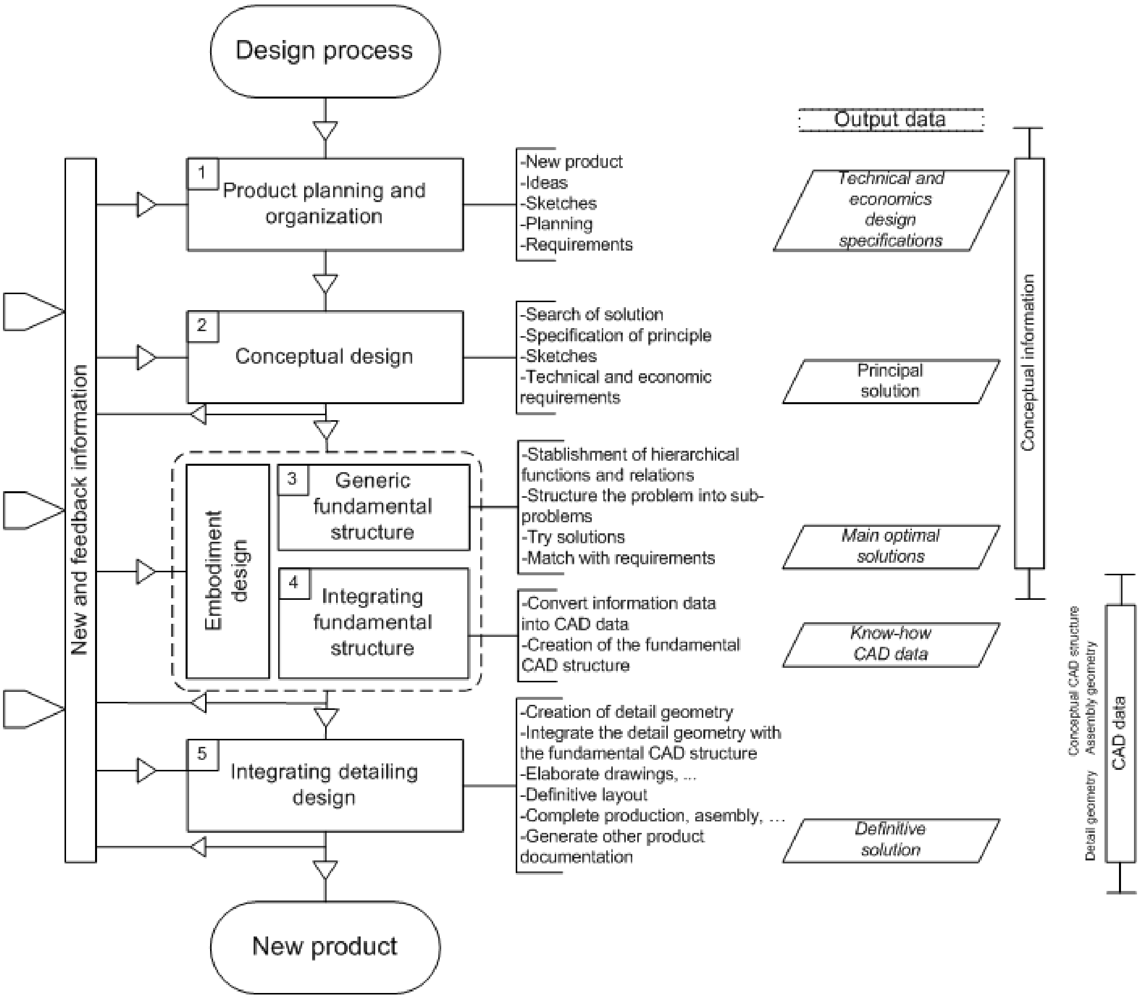

Design process
1
Product planning and organization
-New product
-Ideas
-Sketches
-Planning
-Requirements
2
Conceptual design
-Search of solution
-Specification of principle
-Sketches
-Technical and economic requirements
Embodiment design
3
Generic fundamental structure
-Stablishment of hierarchical functions and relations
-Structure the problem into sub-problems
-Try solutions
-Match with requirements
4
Integrating fundamental structure
-Convert information data into CAD data
-Creation of the fundamental CAD structure
5
Integrating detailing design
-Creation of detail geometry
-Integrate the detail geometry with the fundamental CAD structure
-Elaborate drawings, ...
-Definitive layout
-Complete production, asembly, …
-Generate other product documentation
New product
New and feedback information
Output data
Technical and economics design specifications
Principal solution
Main optimal solutions
Know-how CAD data
Definitive solution
Conceptual information
CAD data
Conceptual CAD structure
Assembly geometry
Detail geometry

Figure 1. Steps of the proposed approach to the design process

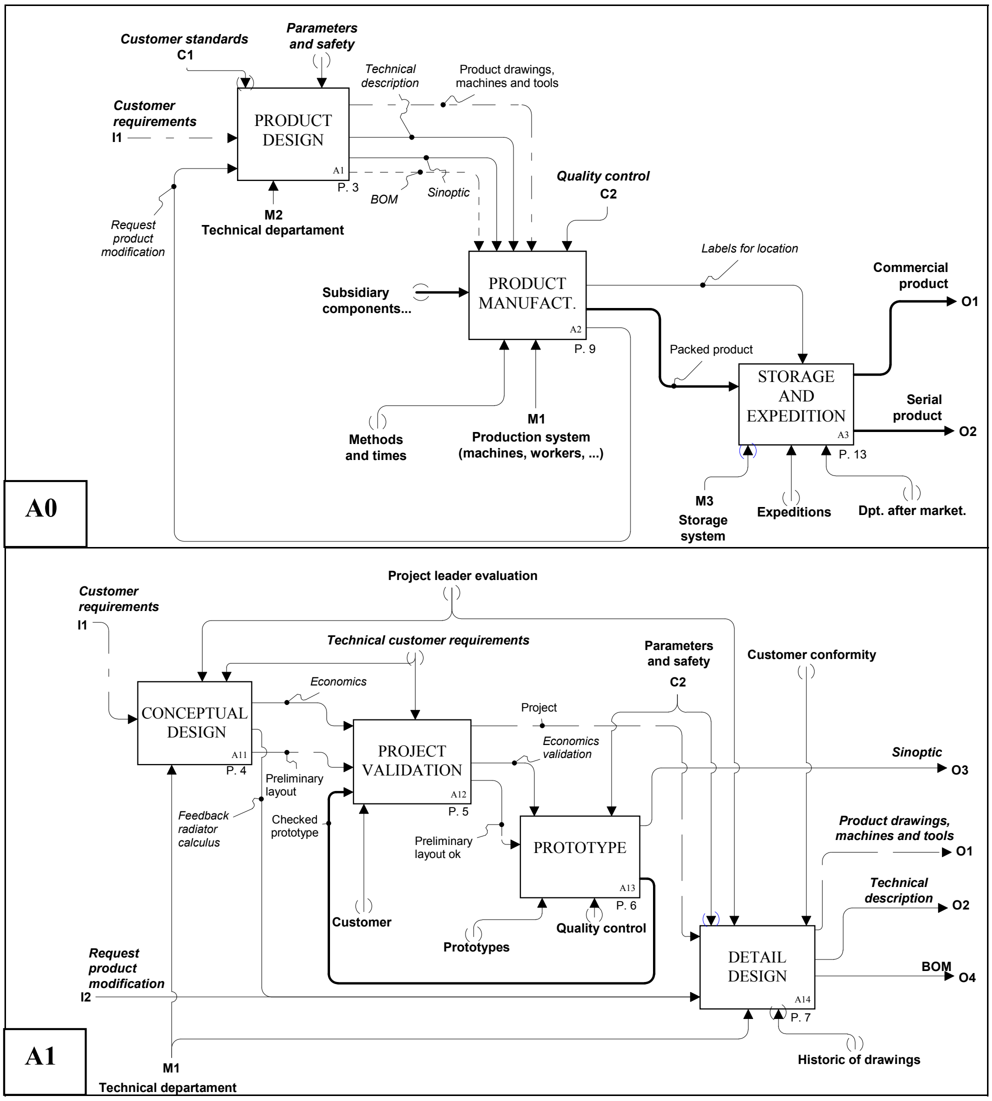


Figure 2. Idef0 example model for an automotive product design process

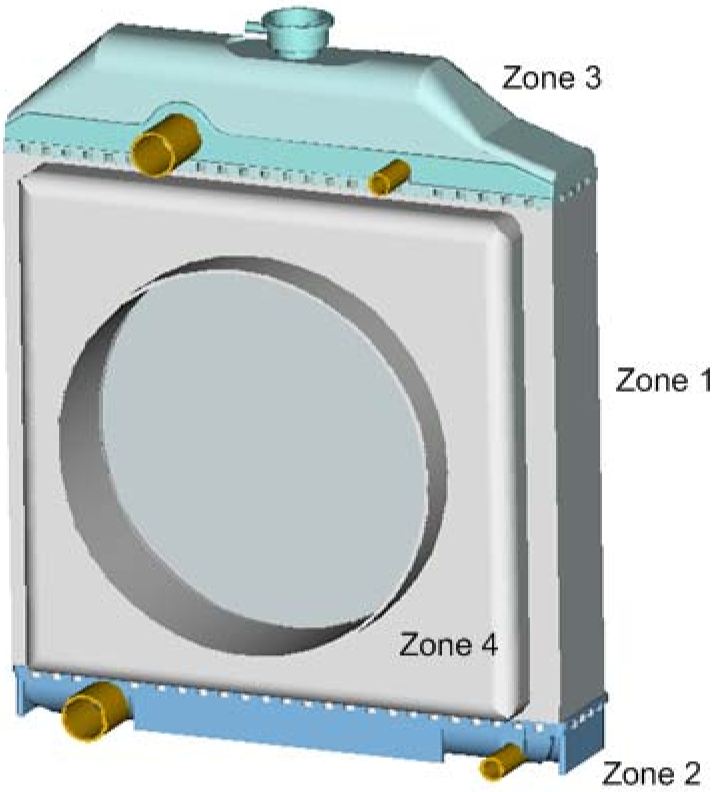


Figure 3. Split up of the final radiator geometry

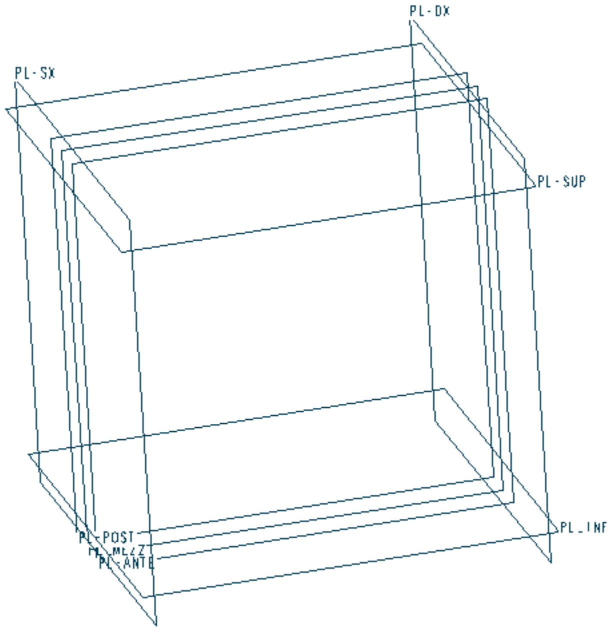


Figure 4. Highest level radiator skeleton

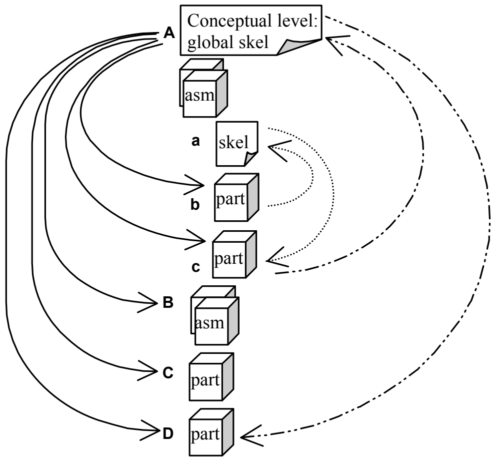


Figure 5. Transference of conceptual data in Top-Down design approach with Pro/Engineer

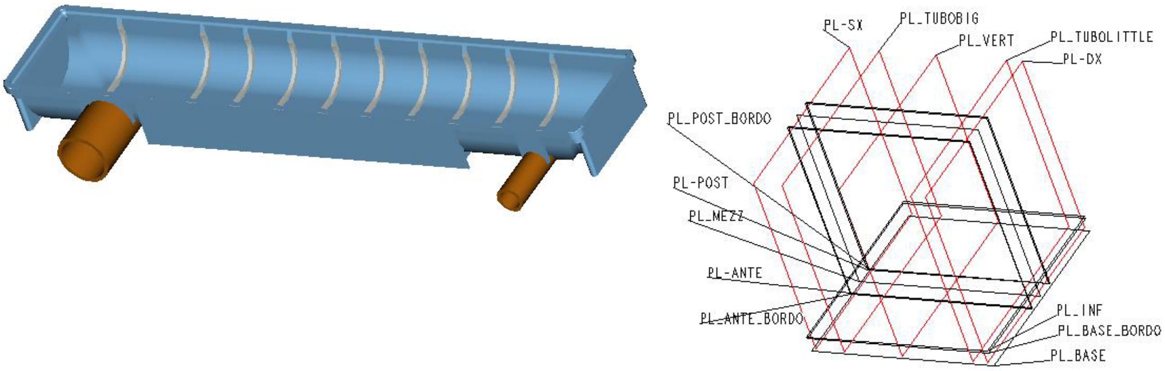
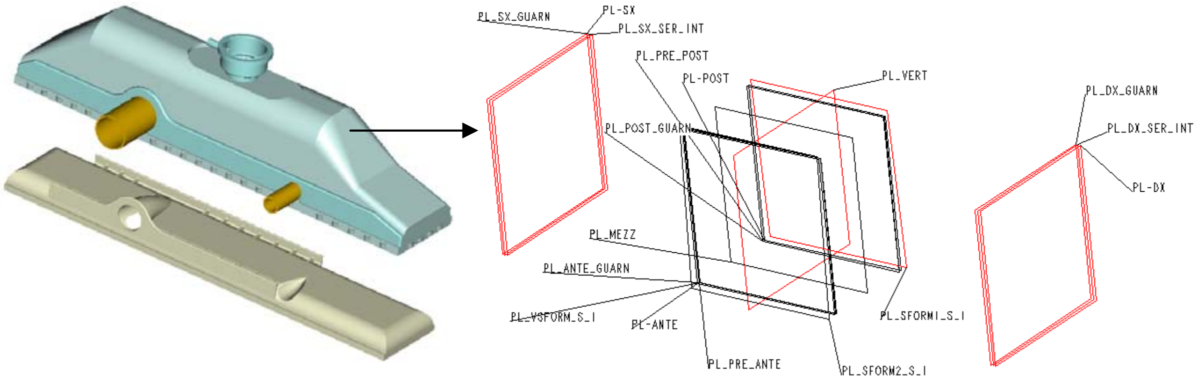


Figure 6. Water output dump (a) and its skeleton (b)



(a) (b)

Figure 7. Entry dump sub-assembly (a) and the skeleton for the upper component (b)

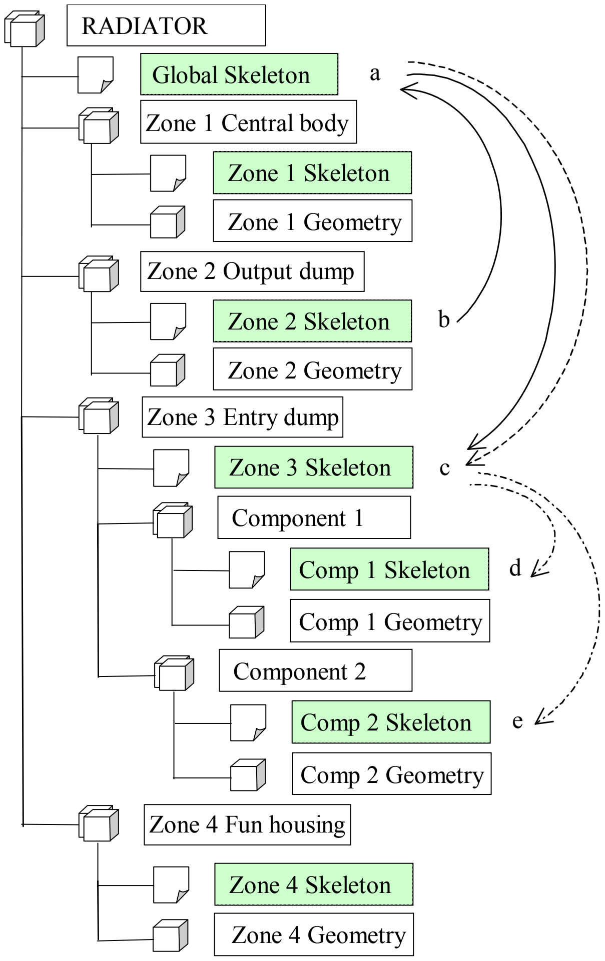


Figure 8. Transmision of fundamental data through the model tree

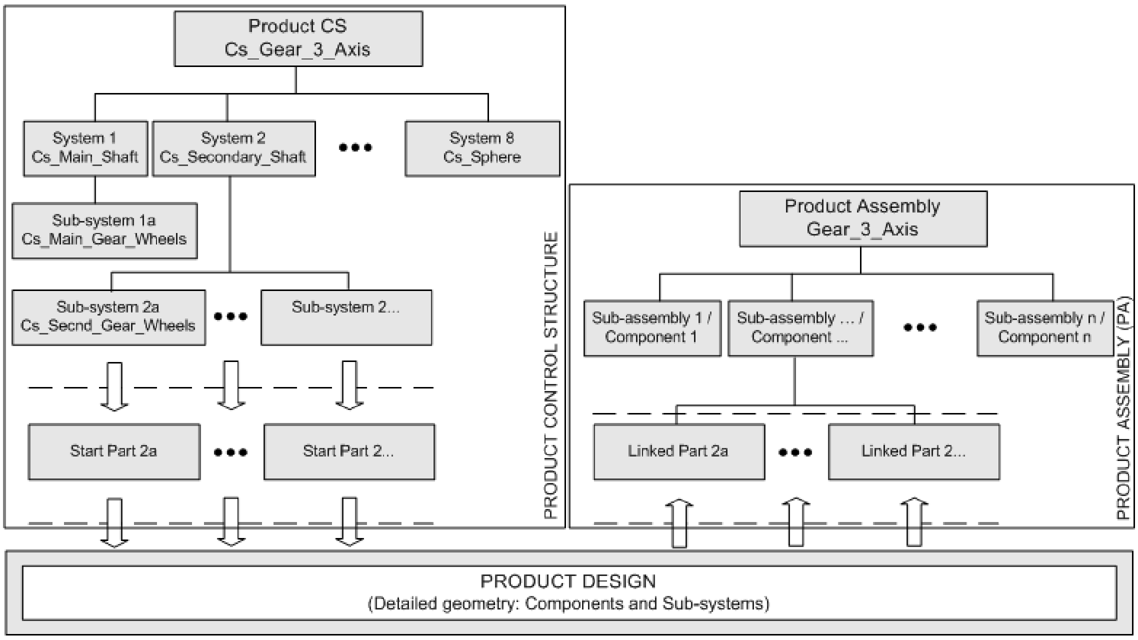


Figure 9. Transference of conceptual data in Top-Down design approach with Unigraphics

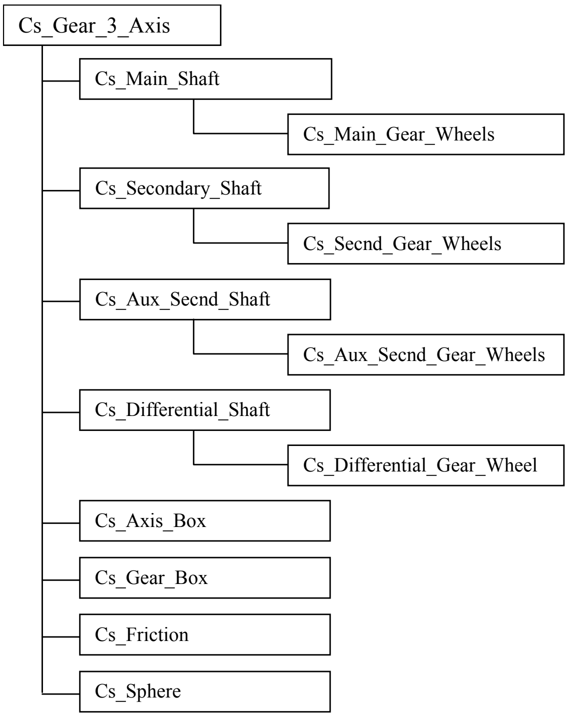
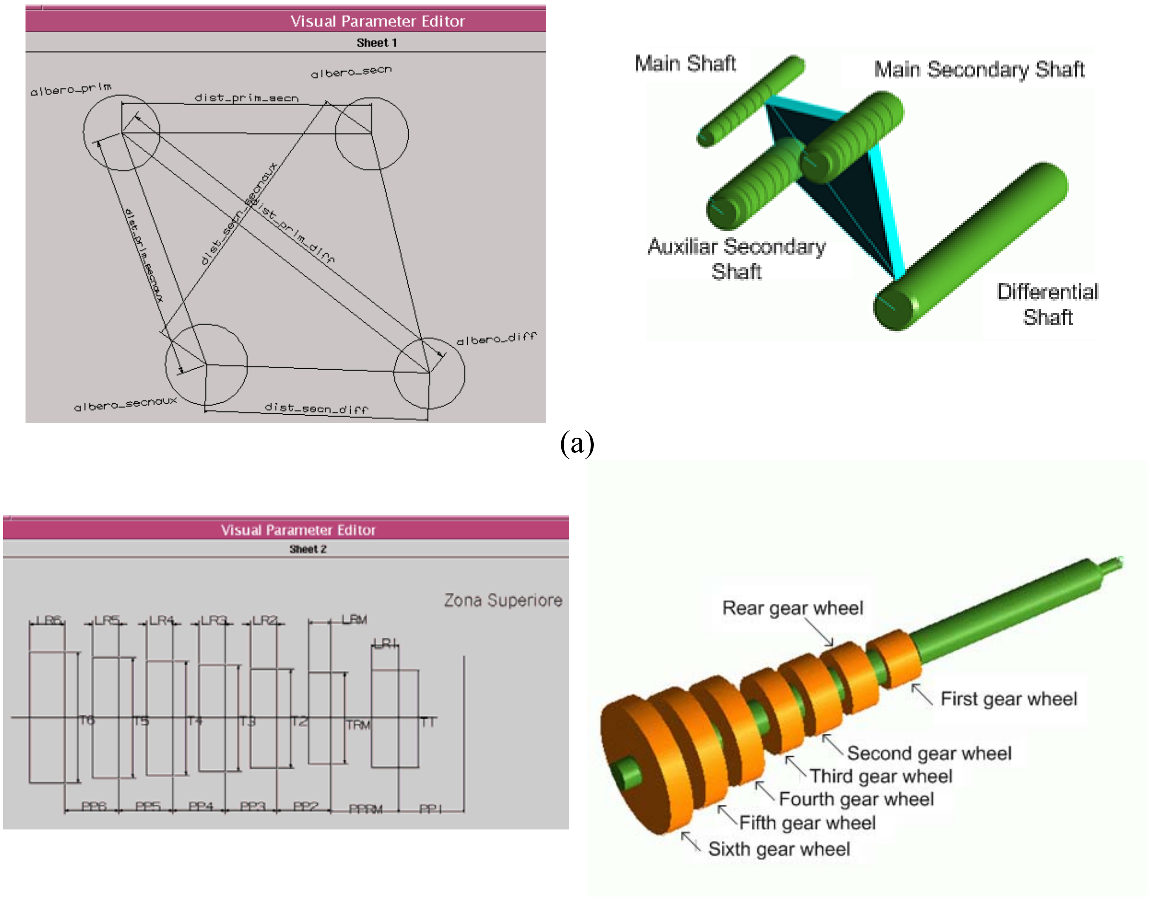


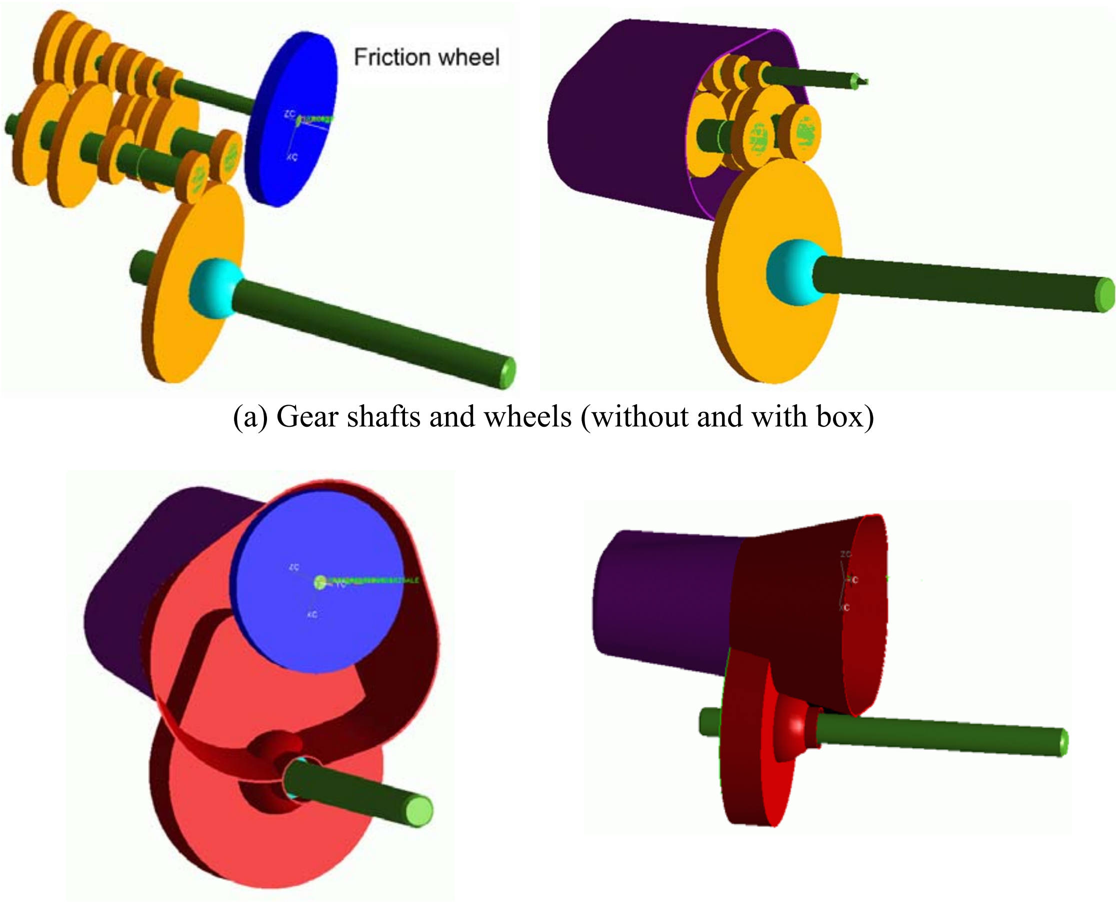
Figure 10. Organization of the entire product control structure



(a)

(b)

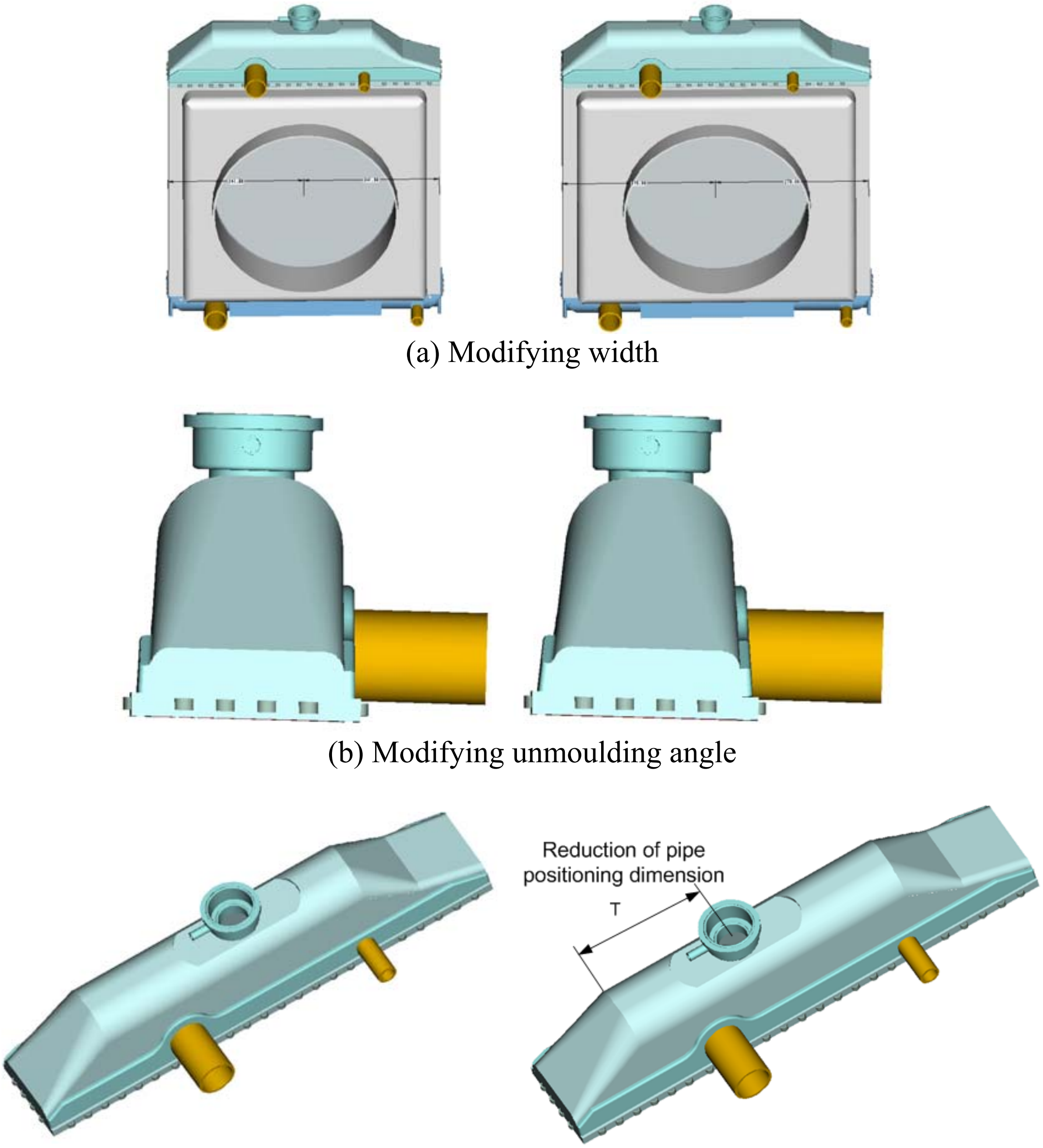
Figure 11. Main parameters in the visual editor and Control structure for gear shafts triangulation (a) and for the main gear shaft wheels (b)



(a) Gear shafts and wheels (without and with box)

(b) Two views of the complete gearbox

Figure 12. Control structure for shafts and box and cover



(a) Modifying width

(b) Modifying unmoulding angle

(c) Modifying pipe positioning

Figure 13. Modifying the radiator skeleton, and its influence in the detailed geometry

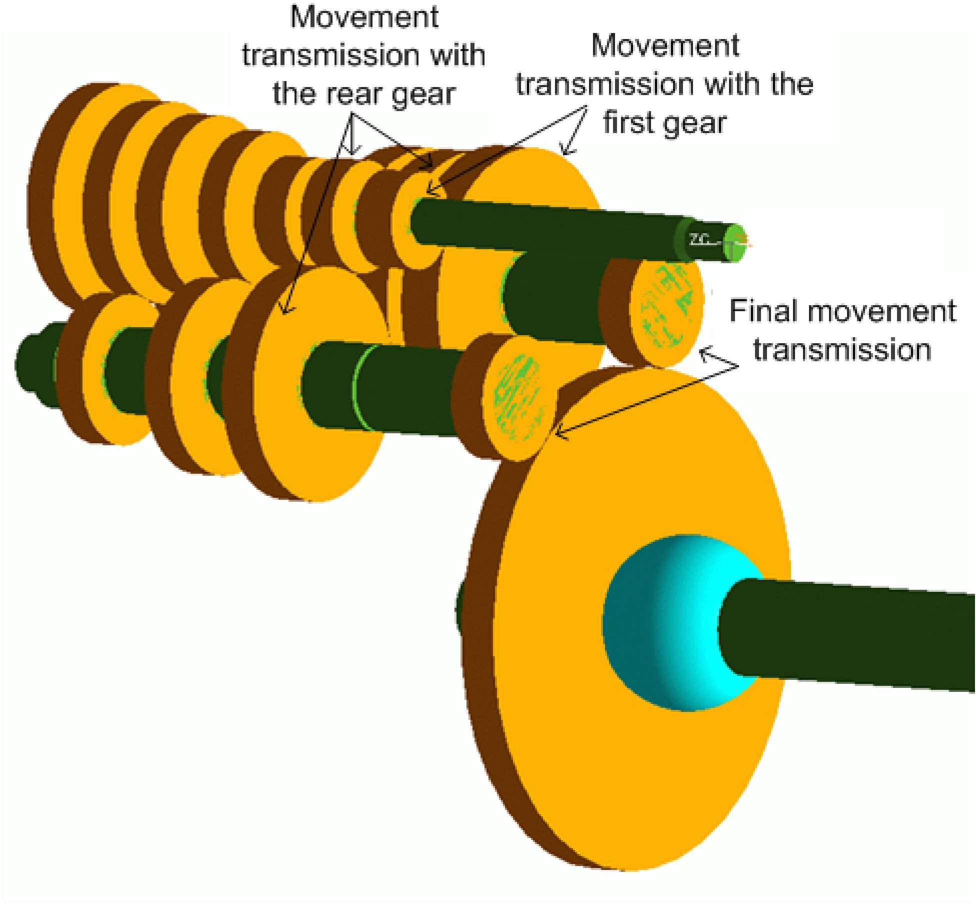

Movement transmission with the rear gear
Movement transmission with the first gear
ZC
Final movement transmission

Figure 14. Initial geometry of the car engine gearbox

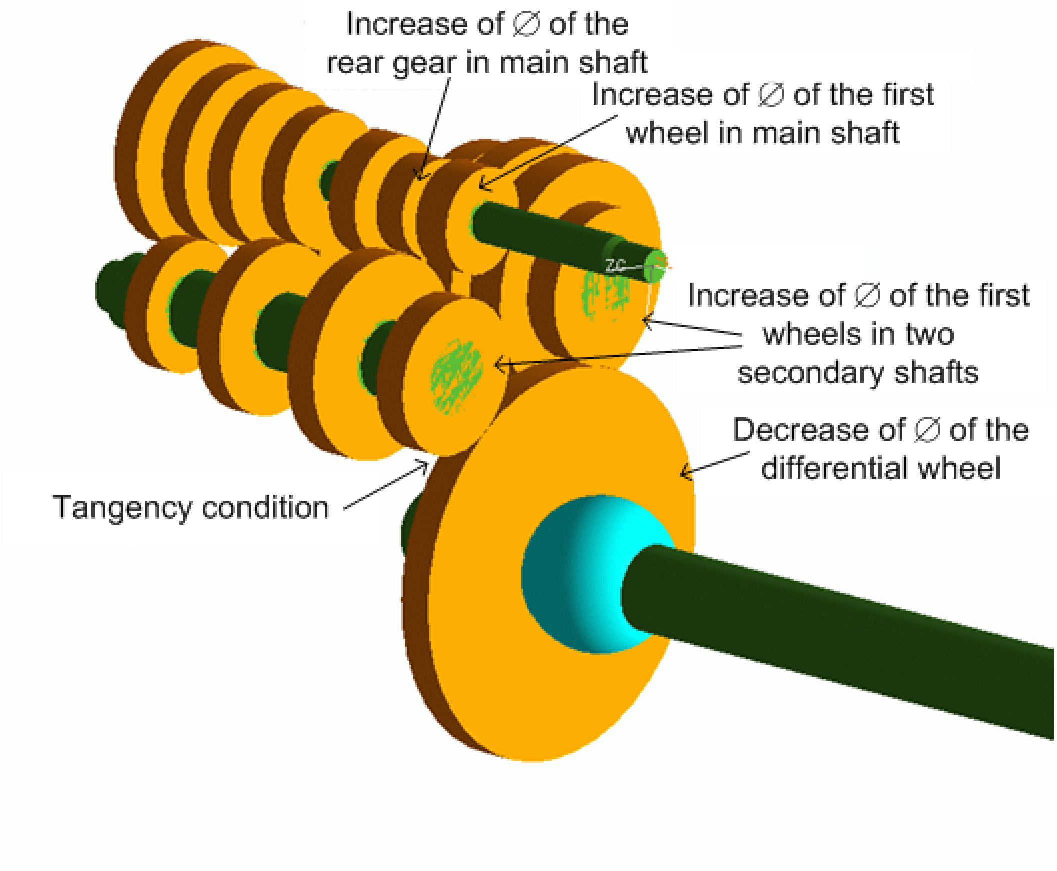
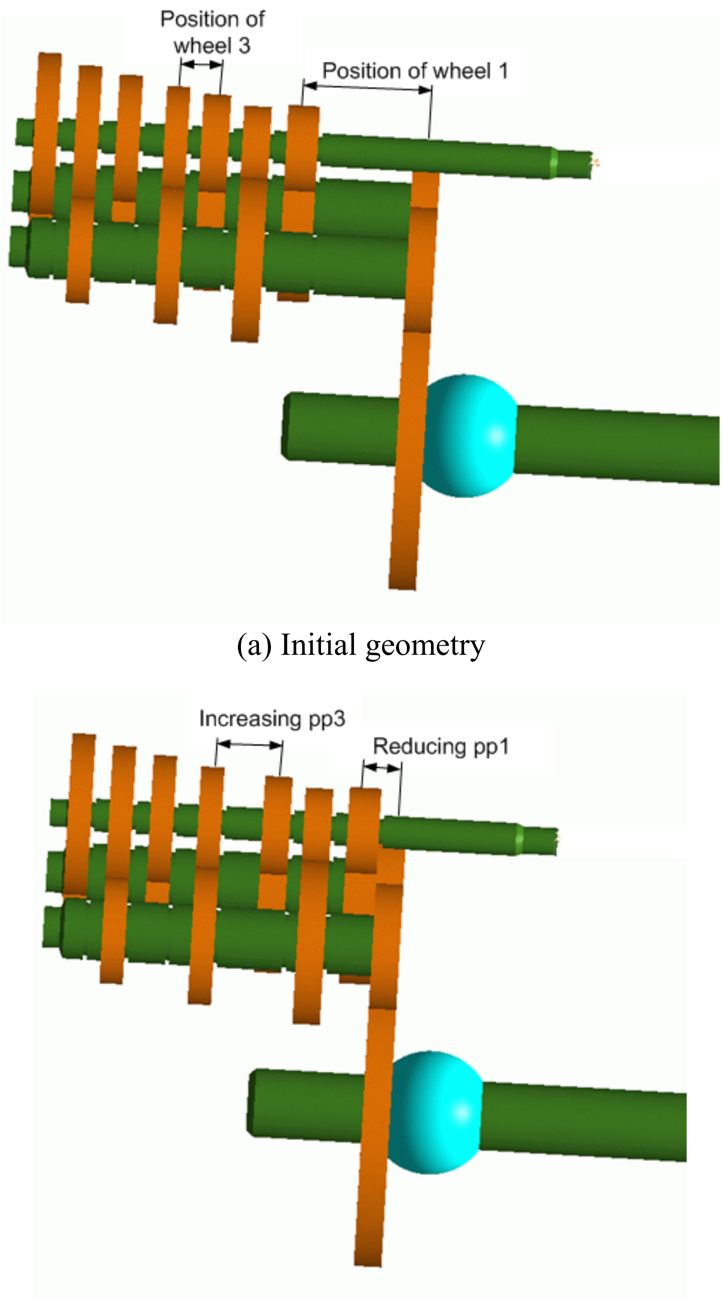

Increase of ∅ of the rear gear in main shaft
Increase of ∅ of the first wheel in main shaft
ZC
Increase of ∅ of the first wheels in two secondary shafts
Decrease of ∅ of the differential wheel
Tangency condition

Figure 15. Updating the shafts control structure after the modification of τ values



(a) Initial geometry

(b) Geometry after modifying positioning

Figure 16. Updating the shafts control structure after the modification of wheel location values

Table 1. Funtional requirements for the movement transmission

Parameters	Significance	Relations
Gear shaft parameters	Manage their location in the three dimensional space	3D coordinates and orientations
Fundamental motion parameters	Manage the movement transmission	$\boldsymbol{\tau}$: motion transmission between the main shaft gear wheels, and main and auxiliary secondary shaft gear wheels. $\boldsymbol{\tau}_{\mathbf{fin}}$: output transmission between the end gear wheels and the differential gear wheel
	Manage the positioning of gear wheels	$\mathbf{pp}_{\mathbf{i}}$ (i=1..6, for 6 gears): manage the gear wheel position in the shafts $\mathbf{pp}_{\mathbf{rm}}$: reverse gear $\mathbf{pp}_{\mathbf{fin}}$: manage final gear wheel and differential gear wheel
	Manage the size of gear wheels	$\mathbf{lr}_{\mathbf{ri}}$ (i=1..6, for 6 gears): manage the length of the gear wheels $\mathbf{lr}_{\mathbf{rm}}$: reverse gear $\mathbf{lr}_{\mathbf{fin}}$: final gear wheel $\mathbf{lr}_{\mathbf{corona}}$: differential gear wheel
	Manage the diameters of the main shaft wheels	$\mathbf{X}_{\mathbf{i}}$ (i=1..6, for 6 gears): manage the ∅ of the main shaft gear wheels, relating them to the motion transmission $\boldsymbol{\tau}$ $\mathbf{X}_{\mathbf{rm}}$: reverse gear $\mathbf{X}_{\mathbf{fin}}$: final gear wheel