Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2000), *14*, 235–249. Printed in the USA.

Racing car design using knowledge aided engineering

LORENZO SUSCA,[1] FERRUCCIO MANDORLI,[2] CATERINA RIZZI,[1] AND UMBERTO CUGINI[1]
[1]Dipartimento di Ingegneria Industriale, Università degli Studi di Parma, Italy
[2]Dipartimento di Meccanica, Università degli Studi di Ancona, Italy

(RECEIVED July 15, 1999; ACCEPTED March 15, 2000)

Abstract

The evolution of computer aided design (CAD) systems and related technologies has promoted the development of software for the automatic configuration of mechanical systems. This occurred with the introduction of knowledge aided engineering (KAE) systems that enable computers to support the designer during the decision-making process. This paper presents a knowledge-based application that allows the designer to automatically compute and evaluate mass properties of racing cars. The system is constituted by two main components: the computing core, which determines the car model, and the graphic user interface, because of which the system may be used also by nonprogrammers. The computing core creates the model of the car based on a tree structure, which contains all car subsystems (e.g., suspension and chassis). Different part–subpart relationships define the tree model and link an object (e.g., suspension) to its components (e.g., wishbones and wheel). The definition of independent parameters (including design variables) and relationships definition allows the model to configure itself by evaluating all properties related to dimension, position, mass, etc. The graphic user interface allows the end user to interact with the car model by editing independent design parameters. It visualizes the main outputs of the model, which consist in numeric data (mass, center of mass of both the car and its subsystems) and graphic elements (car and subsystems 3D representation).

Keywords: Knowledge-based Systems; Design Process; Racing Cars

1. INTRODUCTION

Concurrent Engineering methodology, process analysis and modeling have shown the importance of tools that allow designers to evaluate different aspects involved with the product definition from the beginning (Vernadat, 1996). Problems tied to manufacturing, assembling, time and costs, and aesthetic aspects, often become evident when it is too late and the costs of design correction and change are high. Three-dimensional (3D)-computer aided design (CAD) systems, simulation tools, and virtual prototyping technologies allow to foresee these faults in the design stage. In the past 10 years, these software tools have been studied widely and nowadays a large number of new systems are available on the market.

Different technologies can be used to support the different stages of the production process: CAD for design, CAM for manufacturing, CAPP for process planning, etc. Someone refers to the set of these technologies as CAx: Computer Aided for x.

This premise may lead one to believe that, because of CAx, the whole production process can be supported and controlled following the Concurrent Engineering paradigm. Unfortunately, in many cases this assumption is not completely true. This is due to the fact that CAx systems are general-purpose systems, that is, they are not tailored to support the production process of a specific product or component. As an example, think about the different problems related to the design and manufacturing of a gear pump, and a hydraulic cylinder.

Only in the past few years, due the ever increasing competition in the field of software tool development for engineering applications, CAx vendors seem to have taken into consideration more aspects related to system customization and new tools that allow the user to realize his/her own product-dependent application are now appearing on the market. We have focused our interest on the class of software tools named knowledge aided engineering (KAE) development shells. Within the design process an application developed by using a KAE shell can be placed before traditional CAD systems. In particular, vertical applications realized using these tools can support experts during the decision-making process when the evaluation of alternative solu-

---

Reprint requests: Lorenzo Susca, Dipartimento di Ingegneria Industriale, Università degli Studi di Parma, Parco Area delle Scienze, 181/A, I-43100 Parma, Italy. E-mail: [susca, rizzi, cugini]@ied.unipr.ir

tions can provide a kernel for the integration of different technologies. They therefore represent a basic step for any following development of the product design (Colombo & Cugini, 1992).

In this way, knowledge-based design process means acquisition and reorganization of information and rules that are part of the experts' personal skills and allow them to analyze and solve problems. This information is determinant for correct preliminary design, and the generation of solid foundation and a common reference to specific design tasks. Another objective of KAE systems is the creation of a knowledge base that may be shared among various activities avoiding misinterpretations. Different works have shown the importance of the development of ontologies that allow only intended meanings to be captured in the knowledge base (Guarino, 1998; Salustri, 1998; Soinien et al., 1998).

Communication and knowledge sharing are becoming the main evolution trends in an industrial scenario where an integrated design environment is needed. In this paper, we present a decision-support system for Indy Racing League (IRL) cars with regard to the computation of mass properties. The system was developed in collaboration with Dallara, an Italian company that produces cars for F3, IRL, and other leagues.

The first part of the paper focuses on decision-making process for IRL car design and the fundamental role of mass distribution with respect to car performance and component dimensioning.

The second part contains the car structure analysis and the definition of the knowledge required to address the problem of mass distribution and define a suitable car model. The model structure, where atomic components are represented by a set of significant properties, and grouped to form hierarchical subassemblies, will be described in detail.

The third part deals with practical aspects of the development of the computing core, which determines the car model, and the graphic user interface, thanks to which the system may be even used by nonprogrammers. Some experiments and results achieved with the implemented prototype will be also described.

The last part is about the integration of the prototype with traditional software tools that allows the designer to reuse the knowledge to perform specific tasks, like the semiautomatic generation of a 3D model of car components and the simulation of the dynamic behavior of the car or its main subassemblies.

2. DECISION-MAKING PROCESS FOR IRL CAR DESIGN

The Indy Racing League was founded in 1996 and is held on oval circuits in United States. The purpose of the league is to organize a highly spectacular championship with low costs for the teams. Therefore, the number of producers is limited to two engine producers, two tire suppliers, and two producers for the chassis and bodywork (Dallara and Gforce). Detailed technical specifications concerning car design, construction, material, and testing were established (Indy, 1996). IRL Specifications regulate structure and sizes of car components (e.g., *all car must incorporate a system of structural support bulkheads within the main chassis structure*, and *the minimum wheelbase shall be ninety-six inch*) as well as weights (e.g., *the overall weight of the car, including lubricants . . . , shall be a minimum of 1550 pounds*). As a result, IRL car designers have to face different problems in pursuing of a high-ratio $reliability \cdot performance/costs$ within technical specifications constraints.

During the design process, car components are sized to optimize the weight/strength balance with respect to the overall constraints imposed by the technical specifications. Car mass distribution is then derived from the dimensioning of the car subassemblies. However, mass distribution is a fundamental parameter that must be taken into account during design because of its contribution to car performance. The designer tries to improve car performance starting from the idea of a new car configuration. In this stage, the designer is guided by the perspective of achieving some particular advantages that s/he can foresee relying on personal expertise. Possible alternative solutions are evaluated to define the strategy of the following steps. A detailed analysis of the selected solution allows the designer to evaluate his/her original idea, giving a feedback of results on the hypothesis.

To complete the cycle, the designer applies his/her knowledge and skill: main steps of this job consist in foreseeing and verifying results. The development of ideas and design conceptualization involve creativity, expertise (Cross, 1998) and ability, while evaluating and verifying alternative design solutions are a traditional engineering problem which implies routine jobs and manual computation, and often entails a great waste of time.

Traditional CAD systems are not useful in this phase of the work, when the design is still evolving and a general perspective is needed. More appropriate tools are KAE systems (Kariko-Buhwezi & Cugini, 1995; Mandorli, 1997; Moulianitis, 1999; Ognjanovic, 1996), which exempt the designer from boring tasks by carrying out as much as possible automatic computation and leaving him/her the possibility to improve the product.

KAE systems include a development shell providing object-oriented languages as software tool to define the knowledge base representing the product models. The language features enable encapsulation of different types of knowledge within the model (technological and functional properties as well as shape aspects and dimensioning rules).

3. DEFINITION OF THE KNOWLEDGE BASE

The first step in building the prototype is the definition of the knowledge base (i.e., design variables, material properties, dimensioning rules, features and functionality of components, data) involved in the computation of mass properties.

The computation of mass distribution in a car requires the identification of the parameters and relationships that determine shape, geometry, and position of any part of the car. In this stage, any decision related to the relevance of a part is important for the final result.

Because of their contribution to mass computation and being strictly regulated by IRL specifications, some parts of the car have been studied in detail, while other parts did not require such an accurate analysis. An example of the former case is the suspension system (front or rear): if the designer modifies any suspension variable, the car set-up and performances change, and it is necessary to verify if the new configuration respects the technical specifications. An example of the latter case is the engine, which is produced by a different company and, even if heavy, can be considered a constant of the problem. In other words, a deeper analysis of the engine does not improve the computation of car mass distribution.

These examples show how car analysis, from mass point of view, refers to car subsystems such as the engine and the suspension system. In fact, to reduce the whole problem to several simpler ones, the car has been divided in main subassemblies, which can be studied in detail first, then related to each other.

The result of this process is shown in Fig. 1 in the form of a hierarchical structure representing the car with its main subsystems. Each car subassembly has been identified in relation with the role it plays in the whole system. Both company know-how on the problem and engineers' experience gave a fundamental contribution to this step and allowed the definition of the parameters and relationships needed to calculate mass distribution of car parts. In the following sections, we consider the example of two car subassemblies (chassis and suspension), which play an important role in determining car mass distribution.

3.1. Suspension System

Front and rear suspension systems support the car floor at a given distance from the ground. Their configuration also defines wheel axis. For these reasons, car mass distribution depends on suspension subsystems, for mass value and center of mass location. For example, front-wheel axis displacement involves a change of front suspension structure and mass. A new suspension configuration also generates redistribution of loads on front and rear wheels, with a consequent change in car attitudes. Figure 2 shows a tree representation of the suspension assembly with its main components, each one characterized by its function; these are: suspension arms (which link the chassis with the upright), wheels, antirolling bar (ARB), and damper system (both of which drive wheels vertical movements).

Different parameters concerning geometry, shape and material, define components. Some parameters are independent, while others depend on geometry and functional constraints. For example, each tubular rod, which makes up wishbones, tie rods and push rods, presents material density and cross-section dimensions as independent parameters; while rod length depends on the distance between the points to be linked.

Main independent parameters of the suspension assembly concern position of the wheel axis, chassis/wishbones linkages, chassis/damper and ARB linkages, cross sections of rod elements, and material density of all parts.

Once these values are defined, mass and center of mass coordinates can be easily evaluated, as reported in Eqs. (1) and

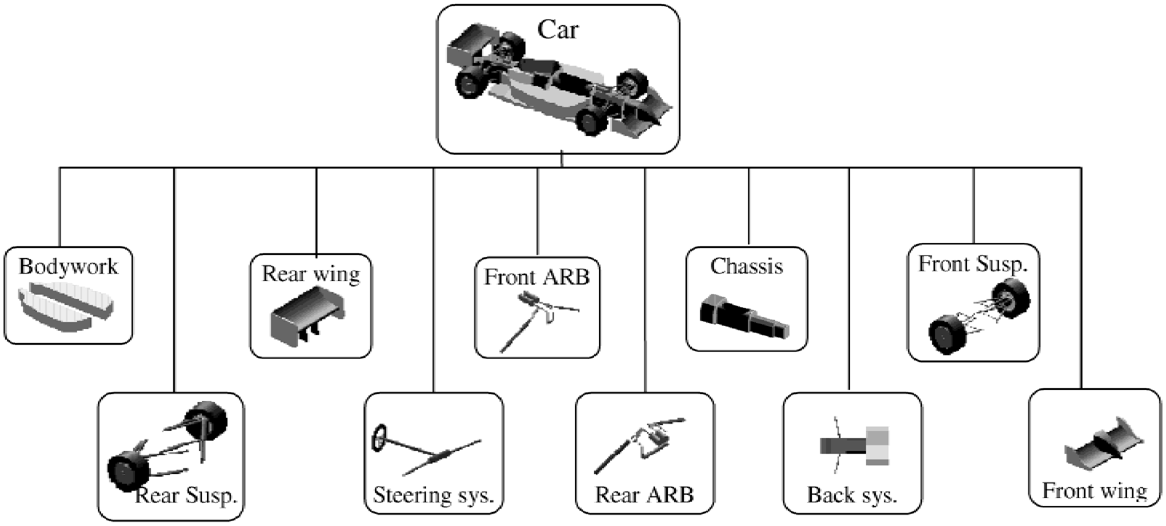


Fig. 1. Car tree structure (with main subassemblies).

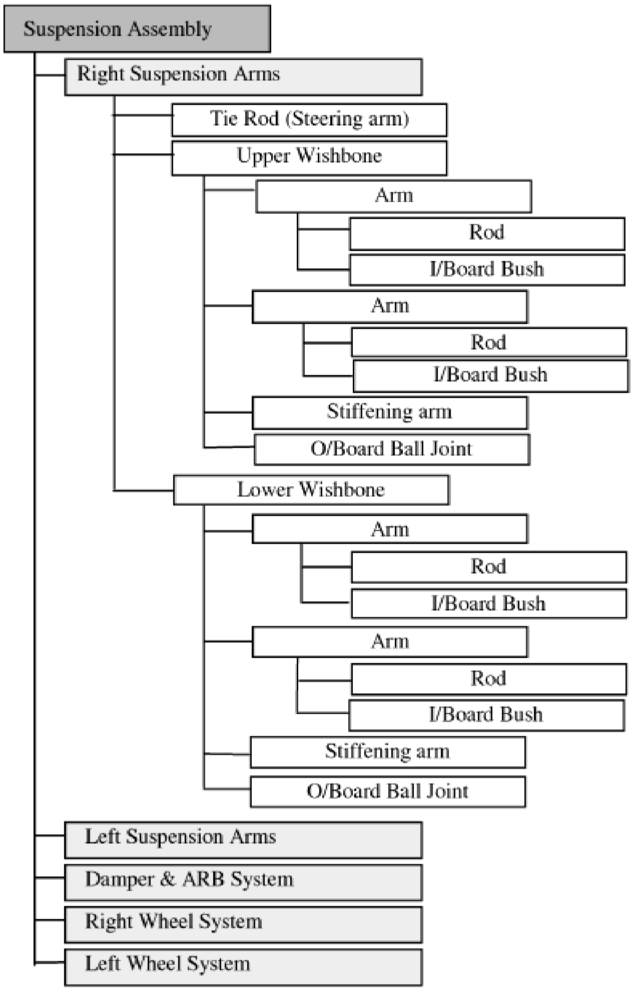


Fig. 2. Suspension tree representation.

(3). Equation (2) evaluates single-part mass and depends upon the real shape (and dimensions) of the component.

$$P_{\text{tot}} = \sum_{i=1}^{N} P_i, \tag{1}$$

where

P_{tot} = total mass;

$$P_i = V_i(\varphi_i) \cdot d_i, \ i\text{th part mass;} \tag{2}$$

V_i = $i$nth part volume, shape (φ) dependent;

d_i = material density; and

N = actual number of components.

$$\vec{G}_{\text{tot}} = \frac{\sum_{i=1}^{N} P_i \cdot \vec{G}_i}{\sum_{i=1}^{N} P_i}, \tag{3}$$

where

$\vec{G}_{\text{tot}}$ = center of suspension assembly mass; and

$\vec{G}_i$ = center of mass of the $i$nth part.

3.2. The chassis

Another important component of the car is the chassis, which represents a reference for other subsystem positioning. The chassis is made from composite material, which improves lightness and stiffness of the structure. Three different layers make up composite material: internal and external skins of carbon fiber and a central honeycomb structure. The thickness of the three layers determines the chassis mass, and the thickness values are constrained by the IRL specifications within predefined range (e.g., *minimum thickness honeycomb core of chassis from . . . to . . . should be 0.750 inches*).

IRL technical specifications give detailed information on chassis design (shape and sizes), taking into account that the system has to pass a crash test that evaluates the chassis's behavior during racing accidents. Driver's safety must be guaranteed in all conditions.

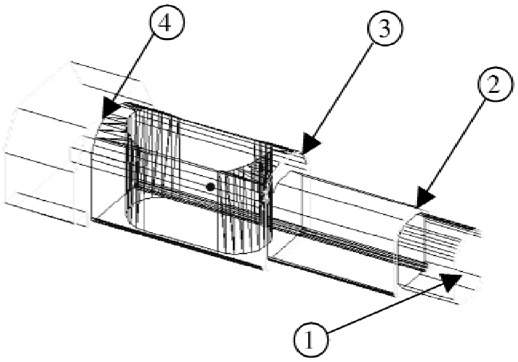
Technical specifications impose the presence within the chassis of structural support bulkheads. The system shall consist of a minimum of four primary bulkheads. In our car configuration, there are four stiffening bulkheads whose shape defines the main cross sections of the chassis. The first bulkhead (front bulkhead, Fig. 3-1) is positioned as interface between the front part of the chassis and the front wing. The second bulkhead (pedal bulkhead, Fig. 3-2) is positioned close to the pedals. The third bulkhead (dashboard bulkhead, Fig. 3-3) is positioned close to the dashboard area. The dashboard bulkhead has a central cut for the driver's legs. The dimensions of this cut are fixed by the league's specifications. This bulkhead also provides support for the steering column. The last bulkhead (seatback bulkhead, Fig. 3-4) is positioned behind the driver's seat.

The technical specifications place constraints on shape, sizes, and distance between them. For example, the first bulkhead must have *a flat floor with a minimum width of six inches*, the pedal bulkhead *must be located a minimum of twelve inches from the rear face of the front bulkhead to the front face of the pedal bulkheads*, and so on.

The cross section at the end of the chassis is positioned at the engine interface; the engine, produced by two different companies (Nissan and Oldsmobile), defines this section profile.

The definition of the chassis' shape and dimensions allows theoretical computing of the subsystem mass (and center of mass), as shown in Eq. (4):

$$P = \int_S [(d_{\text{honeycomb}} \cdot s_{\text{honeycomb}}) + (d_{C1} \cdot s_{C1}) + (d_{C2} \cdot s_{C2}) + d_{\text{glue}}] \cdot ds + P_{\text{stiffening}} \tag{4}$$



1	Front bulkhead
2	Pedal bulkhead
3	Dashboard bulkhead
4	Seatback bulkhead

Fig. 3. Chassis structure with bulkheads.

and in Eq. 5:

$$\vec{G} = \frac{\int_S [(d_{\text{honeycomb}} \cdot s_{\text{honeycomb}}) + (d_{C1} \cdot s_{C1}) + (d_{C2} \cdot s_{C2}) + d_{\text{glue}}] \cdot \vec{g} \cdot ds + P_{\text{stiffening}} \cdot \vec{G}_{\text{stiffening}}}{P}, \tag{5}$$

where

P = computed chassis mass;

$P_{\text{stiffening}}$ = total mass of additional stiffening parts (retrieved from a database);

$d_{\text{honeycomb}}, d_C, d_{\text{glue}}$ = density of the honeycomb, carbon fiber and glue;

$s_{\text{honeycomb}}, s_{C1}, s_{C2}$ = thickness of the honeycomb, and internal and external skins;

ds = infinitesimal surface element;

S = chassis surface;

$\vec{g}$ = center of mass of infinitesimal surface element;

$\vec{G}_{\text{stiffening}}$ = center of mass of stiffening elements; and

$\vec{G}$ = center of mass of the chassis.

3.3. Other subsystems

The analysis of other car components follows the method adopted for the chassis and the suspensions. The mass of the bodywork system and wing assemblies, mainly made of carbon fiber, depends on the surface area and thickness of different skins and can be calculated as in Eq. (4).

Some parts, such as the engine or gear system, come from different companies, together with known mass properties.

Finally, fluid systems (oil, fuel, and water) present difficulties for mass computing, because of the presence of non-solid elements. In particular cases (e.g., in static conditions), mass properties can be evaluated with the introduction of suppositions that simplify the computation (e.g., considering liquids as solids when computing center of mass).

In all cases, the handbook of car components is useful for a comparison between results coming from theoretical calculations and experience.

4. COMPUTING CORE DEVELOPMENT

The analysis of the car and of its main subsystems allowed the definition of the knowledge base needed to calculate mass distribution of the system. The knowledge base was captured and formalized using a suitable software tool (Selling Point), to build a structure that makes up the *Computing Core*. The computing core allows the automation of the calculation process and generates a model of the car (Fig. 4), which simulates its main mass properties.

The next section introduces the tool supporting the computing core and the reasons that justify this choice. Then the car prototype will be presented.

4.1. Selling Point: A development shell for product configuration applications

The development of the car prototype requires some fundamental functionality to enable the automatic configuration of the model, that is,

1. represent the complex structure as a composition of simpler parts, sorted hierarchically in a tree model (as required by car study, section 3, Fig. 1);

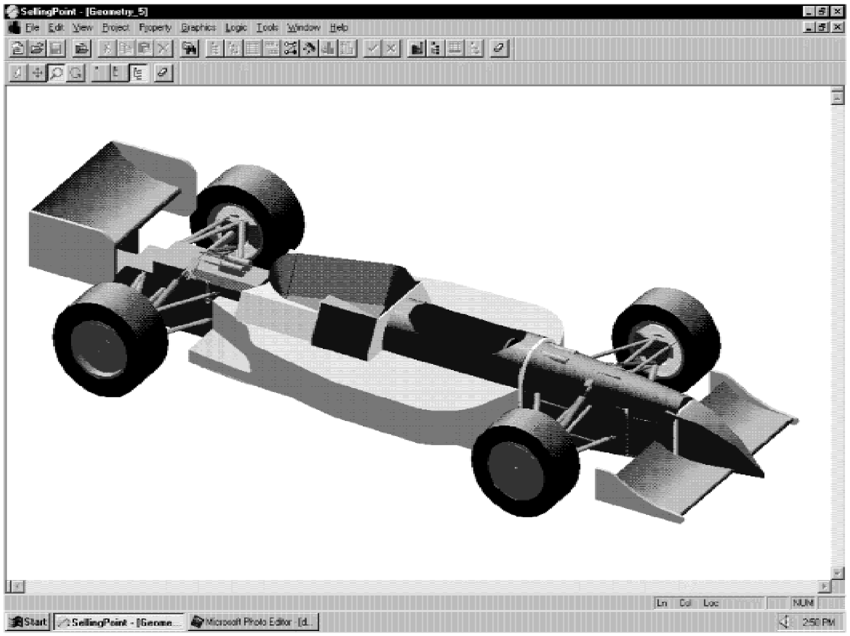


Fig. 4. Representation of the car prototype (selling point environment).

2. make use of graphic parametric primitives (for car components representation) with known geometry and mass properties;
3. link external database (e.g., book of car part masses); and
4. generate models that are able to configure themselves automatically, once the tree structure and independent parameters are defined.

Selling Point (SP) by Concentra Corporation (www.oracle.com/applications/sellingpoint) met these requirements and allowed the development of the prototype. The car model is described using SP language (*GSL = Generative Specification Language*), an object-oriented language that supports the description of objects listed in a user library (Concentra, 1995).

SP includes standard libraries (e.g., geometric library for graphic primitives) whose items can be used by a GSL programmer. All defined objects (member of user or standard libraries) can be reused to compose different assemblies; internal linkages, represented by relationships among the properties of different objects (= nodes), define the resulting tree structure.

For this reason, the development of the prototype begins with the definition of basic elements and ends with the description of the tree root, which in this case represent the entire car model.

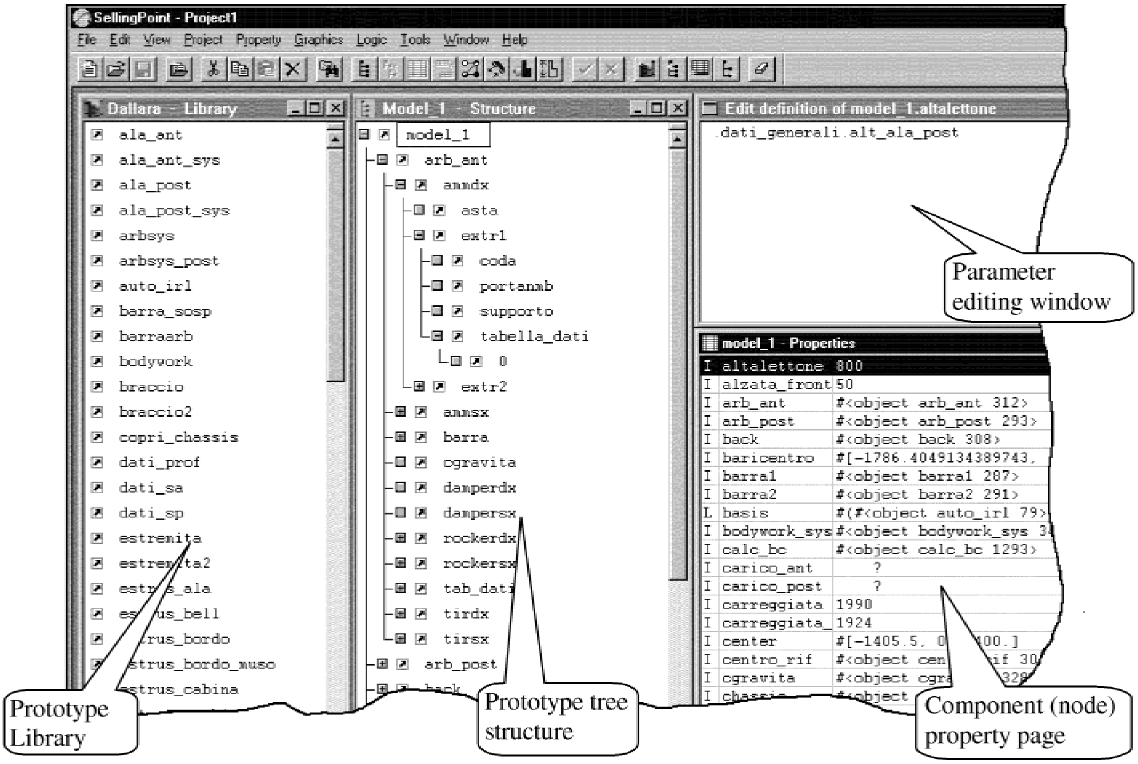
Once the tree structure has been defined, SP generates a parametric model in a standard configuration based on given default values of independent parameters. The end user interacts with the model editing the properties of different components; the model recomputes all dependent parameters and configures itself automatically.

Figure 5 shows an example of an SP working environment, with a component property page, the parameter editing window, the car library, and the car tree representation. The next paragraph describes the main features of the car prototype and its applications.

4.2. The car prototype

The car prototype represents the computing core of the system; its development involved the formalization of the knowledge captured during the study of the car. All data, either coming from experience or provided by the company, were saved in a database that represents the system data source. The system database contains all data retrieved from Dallara's handbook and other information needed to carry out the dimensioning of some parts. For example, ball-joint dimensioning requires a comparison between the calculated values of design variables (e.g., linkage screw diameter) and dimensions of part available (e.g., hole diameters of ball joints actually produced). As the end user modifies the linkage screw diameter, the system uses the new value as an input for a table (Table 1) of the database including all properties (code, type, dimensions, and mass) of ball joints available. As a result, the system provides a new ball-joint whose hole diameter best approximates the given value.

There are also tables including all dimensional parameters of a particular system (e.g., chassis, antirolling bar). In these cases, alternative configurations of the same compo-



.dati_generali.alt_ala_post

I	altalettone	800
I	alzata_front	50
I	arb_ant	#<object arb_ant 312>
I	arb_post	#<object arb_post 293>
I	back	#<object back 308>
I	baricentro	#[-1786.4049134389743,
I	barra1	#<object barra1 287>
I	barra2	#<object barra2 291>
L	basis	#(#<object auto_irl 79>
I	bodywork_sys	#<object bodywork_sys 3
I	calc_bc	#<object calc_bc 1293>
I	carico_ant	?
I	carico_post	?
I	carreggiata	1990
I	carreggiata_	1924
I	center	#[-1405.5, 0 400.]
I	centro_rif	#<object cen if 30
I	cgravita	#<object cgra 328
I	chassis	#<object

Fig. 5. Snapshot of selling point working environment.

nent may be saved and this component can be identified synthetically by a configuration code; in particular, this option is useful when teams need different configurations for some assemblies or components. The prototype can be easily adjusted to alternative solutions by inserting in the database different sets of values for the same group of independent parameters (related to a desired component or assembly). Each set corresponds to a specific configuration and can be retrieved directly by modifying the prototype parameter related to the table identification code within the database.

Table 1. *Dimensions of standard ball joints (values are in mm and g)*

Code	Hole Diam	Ext Diam	Radius	Thickness	Weight
3	4.826	15.875	11.0998	8.3058	14.075
4	6.35	15.875	11.0998	8.3058	14.074
5	7.9375	17.4625	11.0998	8.0518	15.89
6	9.525	20.6375	12.7	10.3124	27.24
7	11.1125	23.8125	14.2748	11.2268	36.32
8	12.7	25.4	15.875	12.827	45.4
9	14.2875	28.575	17.2212	13.6144	61.29
10	15.875	30.1625	19.05	14.4018	72.64
12	19.05	34.925	22.225	16.002	108.96
14	22.225	41.275	22.225	19.177	158.9
16	25.4	53.975	34.925	25.527	440.38

Once the system database was organized, relationships related to function, dimensions, and position were identified and reprocessed in accord with model hypothesis.

The use of graphic primitives and sweeping blocks, whose mass and geometric properties are automatically evaluated by SP, allows the description of car components and determines mass computing. For example, theoretical Eqs. (1) and (3) for mass analysis of the suspension assembly (Fig. 6) become:

$$P_{\text{tot}} = \sum_{i=1}^{M} P_i \cdot C_i, \tag{6}$$

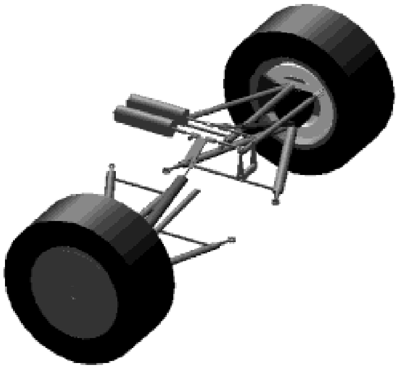


Fig. 6. Front suspension assembly.

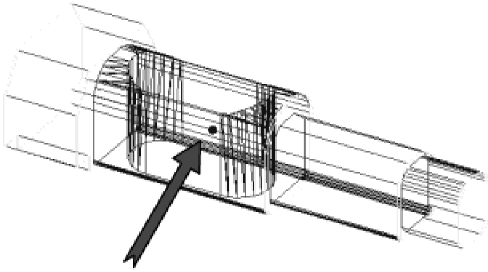


Fig. 7. Chassis system with center of mass.

where

P_{tot} = total mass;

$$P_i = V_i(\phi_i) \cdot d_i = i\text{nth part mass calculated}; \tag{7}$$

V_i = $i$nth part volume[1];

ϕ_i = shape of the $i$nth part;

d_i = material density;

C_i = corrective coefficient of the $i$nth part; and

M = number of components represented.

Equation (7) evaluates the single part mass and depends on the shape ϕ (and dimensions) of the component represented in the model.

$$\vec{G}_{\text{tot}} = \frac{\sum_{i=1}^{M} P_i \cdot \vec{G}_i}{\sum_{i=1}^{M} P_i}, \tag{8}$$

where

$\vec{G}_{\text{tot}}$ = center of suspension assembly mass;

$\vec{G}_i$ = center of mass of the $i$nth part.

At the same time, theoretical Eqs. (3) and (5) for the evaluation of mass and center of mass of the chassis (represented in Fig. 7) become:

$$P = \sum_{i=1}^{4} P_i + P_{\text{stiffening}}; \tag{9}$$

$$\vec{G} = \frac{\sum_{i=1}^{4} P_i \cdot \vec{G}_i + P_{\text{stiffening}} \cdot \vec{G}_{\text{stiffening}}}{P}, \tag{10}$$

where

P = computed chassis mass;

$$P_i = C_i \cdot [(d_{\text{honeycomb}} \cdot S_{\text{hi}}) + (d_c \cdot S_{c1i}) + (d_c \cdot S_{c2i}) + d_{\text{glue}}] \cdot \text{Sup}_i; \tag{11}$$

P_i = mass of one of the 4 sections delimited by subsequent bulkheads;

$d_{\text{honeycomb}}, d_C, d_{\text{glue}}$ = density of the honeycomb, carbon fiber, and glue;

S_{hi}, S_{c1i}, S_{c2i} = thickness of the honeycomb, internal and external skins for the $i$nth section;

Sup_i = area of the $i$nth section[2];

$\vec{G}$ = center of mass of the chassis;

$\vec{G}_i$ = center of mass of the $i$nth section[2]; and

$\vec{G}_{\text{stiffening}}$ = center of mass of stiffening parts.

NB: integrals of Eqs. (4) and (5) have been replaced with discrete summations (9) and (10), because of the representation of the real chassis surface through different sweeping blocks (Fig. 7).

The mass properties of all components represented in the prototype depend on the approximations introduced using solid primitives; corrective coefficients (C_i) have been used to balance model results and data coming from experience.

In some cases (e.g., engine, bellhousing, gear system) properties related to position only are represented graphically; mass was evaluated statistically as the sum of the single component masses retrieved from the system database.

The last class of car components includes parts without a graphic representation (e.g., liquids, air jack system, electric system, and driver); in these cases, mass and center of mass are constant and come from the system database. The end user can access the model by editing independent parameters. Table 2 reports some of the parameters available and shows their impact on car asset.

As the model recomputes all dependent parameters, a new configuration of the car is generated. The user can retrieve all results concerned, consisting of mass properties of the car and main components and in part dimensions evaluated through design rules. The following section will introduce the user interface of the prototype.

5. PROTOTYPE USER INTERFACE

Applications developed through SP have a user interface that is strictly oriented to the programming environment.

---

[1] Automatically evaluated by selling point.

[2] Automatically evaluated by selling point.

Table 2. *Configuration parameters of front suspension assembly*

Parameter Code	Meaning, Impact
Axis	Distance between front wheel axis and chassis/engine interface
Link1	Dist. 1st lower wishbone linkage/engine interface
Link2	Dist. 1st upper wishbone linkage/engine interface
Link3	Dist. 2nd lower wishbone linkage/engine interface
Link4	Dist. 2nd upper wishbone linkage/engine interface
L_steer	Steering arm length
D_Rim	Rim diameter
D_brake	Brake disc diameter
D_upright	Upright reference diameter
W_wheel	Wheel width

To interact with the prototype, the end user should know its structure, as well as GSL programming language.

For example, s/he must know the name used within the program for the parameters to be modified to study a new car configuration. Therefore, a graphic user interface was implemented using Visual Basic. The user interface has the look and feel that is typical of all applications developed in Windows NT or 95 environment. Figure 8 shows the main window of the user interface. The end user can interact with the prototype at different levels of the car structure, simulating in this way the designer's activity during the car study.

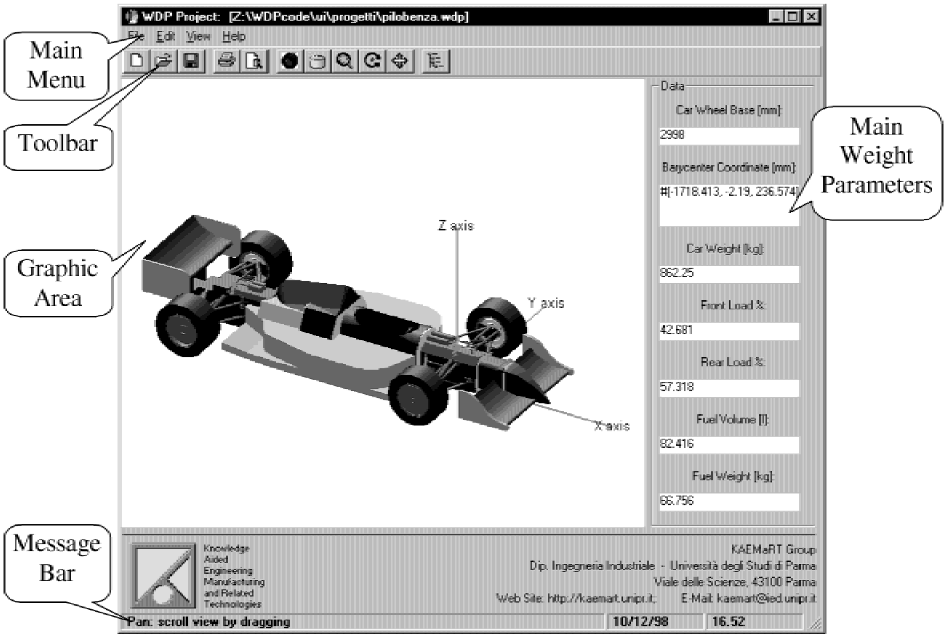
S/he can access the car model from a general point of view, as well as each car subassembly and related detailed information. To this end, a browser is available, named *Model Tree View*, which allows the user to navigate within the hierarchical structure of the car. Figure 9 shows the model tree view and the interaction with the bodywork system.

The car parameters have been functionally organized in independent and dependent (calculated results) parameters. The main window (representing the entire car model and the root of the tree structure in Fig. 9) and those of each car subsystem (representing intermediary nodes) provide dynamic graphic representations of specific assemblies and collections of data concerning mass properties calculated by the computing core.

The designer can modify these data indirectly by editing the value of the independent parameters of each component, which are collected into related windows (the leaves in the model tree view). The user can access subsystem component windows directly from the subassembly window or from the model tree view. Different pictures of the component help the user evaluate the consequences of his/her intervention, as shown in Fig. 10. When the user selects a parameter, its meaning is highlighted in the graphic representation of the current car part. After each parameter modification, the system automatically updates the mass data and the graphic representation of the car.

The designer can generate and save different car configurations to compare and evaluate possible alternatives or retrieve previous projects.

The organization of the different information provided to the user within the hierarchical interface structure, based on car one, is reported in Table 3; examples of typical data available allow an easier comprehension of the interface system.



WDP Project: [Z:\WDPcode\ui\progetti\pilobenza.wdp]

Data

Car Wheel Base [mm]: 2998

Barycenter Coordinate [mm]: #[-1718.413, -2.19, 236.574]

Car Weight [kg]: 862.25

Front Load %: 42.681

Rear Load %: 57.318

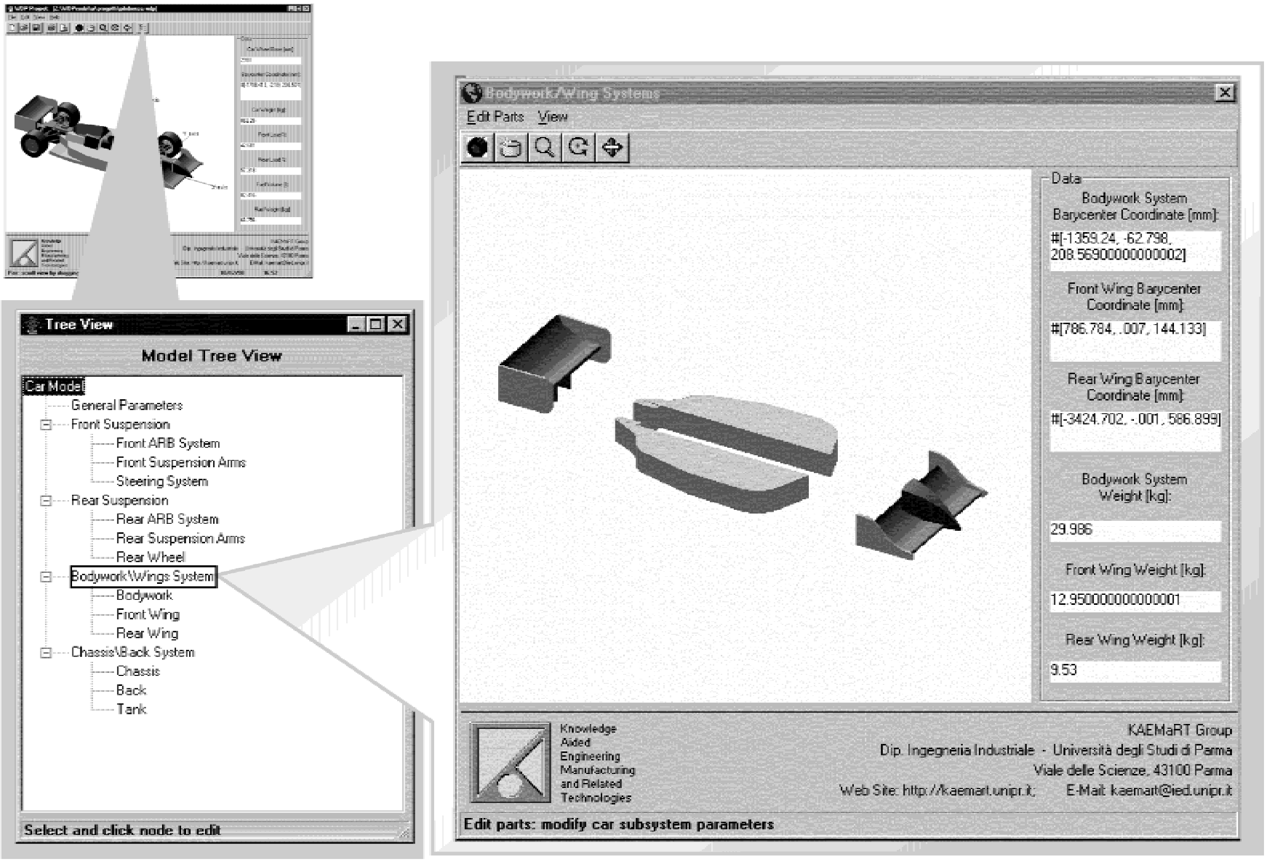
Fuel Volume [l]: 82.416

Fuel Weight [kg]: 66.756

KAEMaRT Group
Dip. Ingegneria Industriale - Università degli Studi di Parma
Viale delle Scienze, 43100 Parma
Web Site: http://kaemart.unipr.it; E-Mail: kaemart@ied.unipr.it

Pan: scroll view by dragging | 10/12/98 | 16.52

Fig. 8. Snapshot of the user interface (main window).



Bodywork System Barycenter Coordinate [mm]: #[-1359.24, -62.798, 208.56900000000002]

Front Wing Barycenter Coordinate [mm]: #[786.784, .007, 144.133]

Rear Wing Barycenter Coordinate [mm]: #[-3424.702, -.001, 586.899]

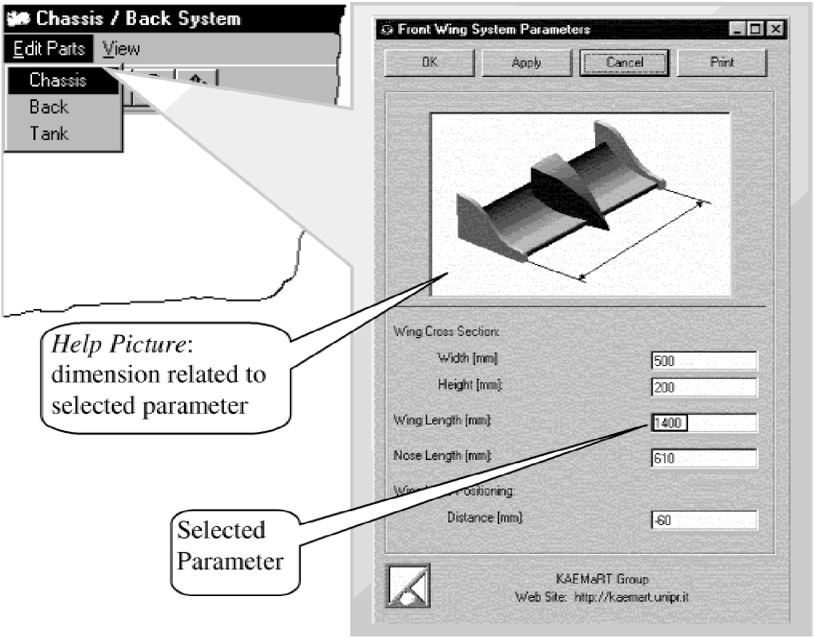
Bodywork System Weight [kg]: 29.986

Front Wing Weight [kg]: 12.950000000000001

Rear Wing Weight [kg]: 9.53

KAEMaRT Group
Dip. Ingegneria Industriale - Università degli Studi di Parma
Viale delle Scienze, 43100 Parma
Web Site: http://kaemart.unipr.it; E-Mail: kaemart@ied.unipr.it

Fig. 9. Model tree view and bodywork system with wings.



Wing Cross Section:

Width [mm]: 500

Height [mm]: 200

Wing Length [mm]: 1400

Nose Length [mm]: 610

Win[illegible] Positioning:

Distance [mm]: -60

KAEMaRT Group
Web Site: http://kaemart.unipr.it

Fig. 10. Front wing parameters window.

Table 3. *Organization of information within the interface*

Level	Example	Related Information Available
Root	Entire car (main window)	Dynamic graphical representation, mass and center of mass of the car
Intermediary node	Rear suspension assembly	Dynamic graphical representation, assembly mass and C. of M.
Leave	Wheel subassembly	Help pictures, independent parameters of the subassembly

6. EXPERIMENTS AND RESULTS

The tool developed has been tested comparing the results calculated by the prototype with the information retrieved from the company measurements. Front and back loads were computed for different car configurations and different tank load situations, that is, three conditions (93.5, 39, and 0%, respectively 35, 15, and 0 gallons), with and without driver, and three wheel-base configurations. These data computed were compared with those provided by the racing teams.

Table 4 shows data calculated for the car standard configuration (wheel base: 2998 mm) without driver. Also reported in Table 4 is the comparison for what concerns the front load rate.

Computed mass and center of mass coincide with a small approximation to the same data supplied by the factory. Approximations are due to the fact that the geometry of some parts is not precise (e.g., the shape of the chassis).

The analysis of results has emphasized a problem due to the variation of static load distribution between front and rear wheels during the race (aerodynamic supplementary loads are not considered). In particular, because of the different position of mass centers of the whole car and the tank/fuel system, the fuel burning induces a backward displacement of the car mass center.

To limit this undesired effect as much as possible, a new configuration of the prototype was developed considering a wheel base set at 3048 mm and a driver weighing 68 kg. The ideal condition is achieved when the two points lay on the same vertical axis. In this way, the x coordinate (see car reference system in Fig. 8), of center of car mass no longer depends on fuel load; at the same time, the problem concerning the change in wheel load is solved. Assuming that the tank cannot be moved, because of the technical specifications, the center of car mass has been moved toward the center of the tank/fuel system mass. This result could be achieved by moving forward some parts, which are not fixed to a specific position by the technical specifications or by functional constraints. Table 5 reports the results calculated by the prototype for the new configuration. The improvement of the car asset may be evaluated comparing the values of the front-load rates (from empty to full load of fuel) obtained for the standard (1.25%, see Table 4, front-load rate measured) and the proposed new configuration (0.59%).

Figure 11 shows the car's and tank/fuel centers of mass both the standard and the new configurations. Notice how the two centers of mass are closer to each other in the new configuration.

7. KNOWLEDGE SHARING

The use of automatic configuration systems during preliminary design allows the designer to manage all significant knowledge concerning the product and its life cycle, from a general point of view. The possibility of reusing and sharing this information during successive design stages supports the process. Different specific tasks can go on simultaneously and in a coordinated way because of the shared knowledge base that allows communication among various areas. The increased level of integration resulting from this methodology supports the development of more

Table 4. *Main results provided for a standard configuration, without driver*

Fuel load rate →	Empty (0%)	Medium (39%)	Full (93%)
Data calculated:			
Mass (kg)	734.5	781.5	841.9
Center of mass coordinates (mm)	(−1786.3, −2.5, 238.9)	(−1770.8, −2.5, 230.6)	(−1757.2, −2.2, 237.7)
Front load (kg)	296.9	319.9	348.4
Rear load (kg)	437.6	461.6	493.5
Comparison:			
Front load rate (computed)	40.41%	40.93%	41.38%
Front load rate (measured)	40.18%	40.96%	41.43%
Difference	0.23%	0.03%	−0.05%

Table 5. *Main results provided by the prototype for the new configuration*

Fuel Load Rate →	Empty (0%)	Medium (39%)	Full (93%)
Mass (kg)	735,3	782,3	842,7
Center of mass coordinates (mm)	(−1702.3, −3.9, 237.9)	(−1692.0, −3.6, 229.6)	(−1684.3, −3.3, 236.1)
Front load (kg)	312,6	335,2	363,2
Rear load (kg)	422,7	447,1	479,5
Front load rate	42,51%	42,85%	43,1%

controlled products from idea to production. The objective can be identified with the understanding that product quality can be increased by improving the design and manufacturing process.

In the context of the work presented here, the possibility of reusing car prototype information in commercial CAD/CAE systems was tested. In particular, we have verified two different integrative solutions:

- the first one is about the automatic generation of a 3D CAD model of a car component (front suspension wishbone); the software used also permits semiautomatic generation of technical drawings;
- the second one concerns the possibility of exporting and re-using geometric elements produced by the prototype in a simulation environment; the objective is to analyze dynamic and kinematic behavior of the car and of the front suspension assembly.

Benefits come from the fact that, once an approximate car configuration has been defined, designers can carry on working with the detailed definition of car parts or with the analysis of car behavior having some feedback on initial hypothesis. Different activities can be developed at the same time, starting from the robust common base provided by the car prototype.

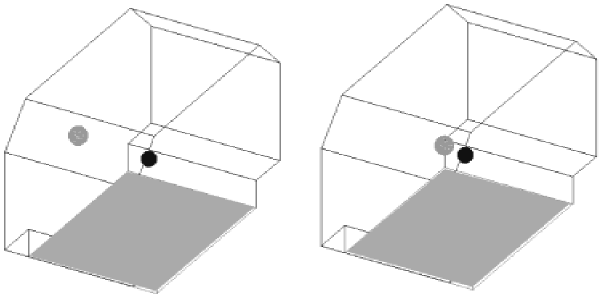


Fig. 11. Tank with center of mass of the car (gray) and of tank/fuel system (black).

7.1. Automatic generation a 3D model of the wishbone

The knowledge-based system described here supports experts during the decision-making process, and allows high-level problem managing, in relation with the preliminary car design phase. In next stages, the decisions adopted have a significant effect on the generation of detailed parts design that requires specific tools and methods.

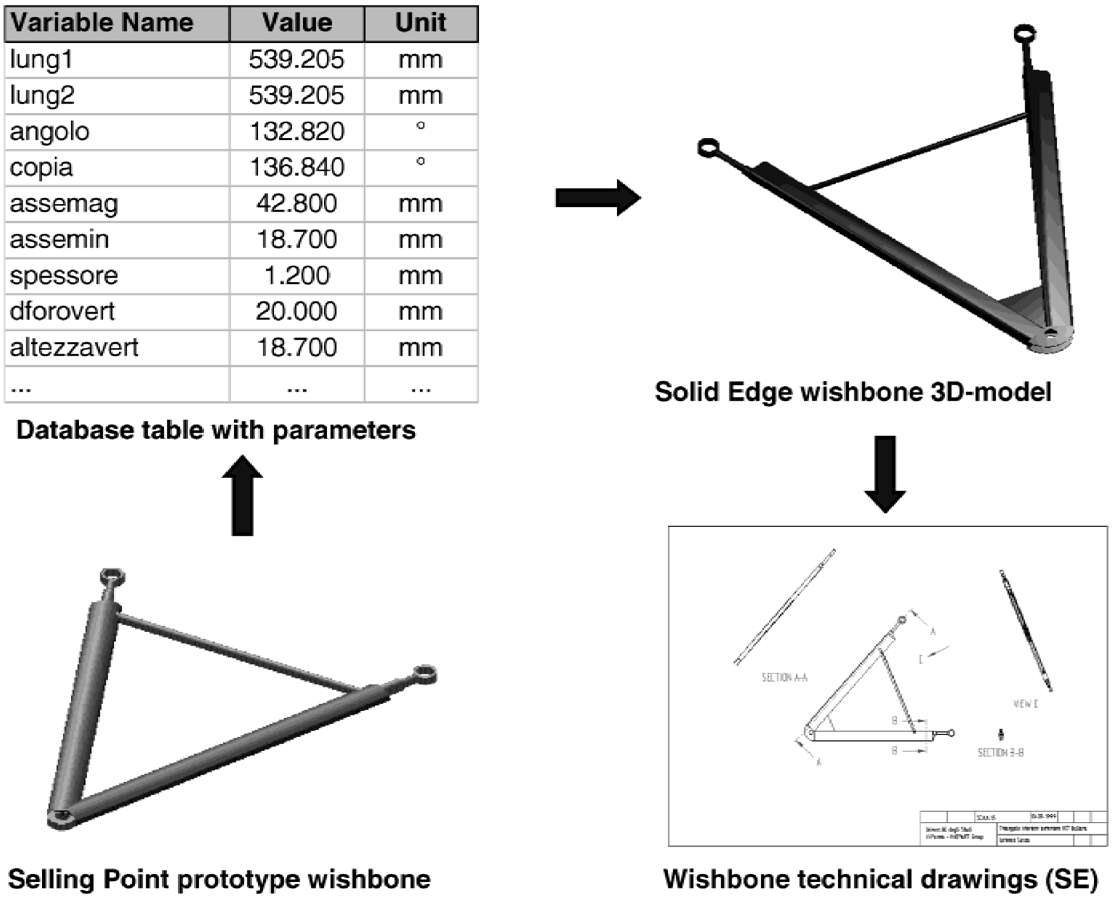
The distance between the two activities can be reduced by creating a *bridge* that allows the prototype to control automatically the generation of 3D-CAD models. This took place by linking model parameters to data and dimensions that the car model evaluates when configuring itself.

A model of the front suspension wishbone is presented as an example. The prototype updates the dimensioning of this part, testing various rules for the chassis dimensions and shape, front wheel dimensions and position, ARB system configuration, and technical specifications constraints. The end user determines the wishbone dimensioning by editing cross sections of rods (as rods make up the wishbone) and modifying the configuration of the chassis and the front suspension assembly. The system once again calculates the dimensions automatically and records them in a database that drives the 3D model generation. The tool used for the wishbone solid modeling is Solid Edge (www.solidedge.com), a commercial 3D-CAD system that supports integration with the car prototype. This modeling environment also allows the generation of technical drawings of the parts or assemblies represented (Fig. 12).

After the preparatory stage, dedicated to the creation of the part model and the *bridge* between the two systems, the prototype automatically updates the model of the part every time the designer modifies the car configuration.

7.2. Dynamic simulation

Car mass distribution and asset affect performance during races: for this reason it can be useful to compare different car configurations through dynamic and kinematic simulation. Geometric elements generated by the system can be exported into a simulation environment thanks to which the



Variable Name	Value	Unit
lung1	539.205	mm
lung2	539.205	mm
angolo	132.820	°
copia	136.840	°
assemag	42.800	mm
assemin	18.700	mm
spessore	1.200	mm
dforovert	20.000	mm
altezzavert	18.700	mm
...	...	...

Fig. 12. Automatic generation of wishbone 3D-model and technical drawings.

car's dynamic behavior can be verified. The tool that supports the integration of the prototype for the simulation is Working Model (www.workingmodel.com); information provided by different tests acts as feedback for the designer's first hypothesis. The prototype development cycle (Fig. 13) is based on a typical design procedure: the analysis of results performed by the designer determines changes to the set of independent parameters that define the prototype configuration. When performance, foreseen through simulation, coincides with expectations, an accepted model of the car is send to output.

Outcomes from other activities (such as wind tunnel tests) affect the configuration of the prototype and impact on the dynamic simulation results: for these reasons they should be considered as input of the loop. It is worth emphasizing that different aspects and problems concerning car performances are dependent and, for this reason, it is therefore necessary to ensure data exchange among concurrent activities.

The experience presented is about the analysis of front-suspension behavior; results provided were used to compare performances and different attitudes of the car while changing properties and configuration of the subsystem upon which the study was focused. Figure 14 shows the suspension model (with approximate chassis and rear wheels) during a simulation test, with a diagram of the chassis velocity.

8. CONCLUSIONS

A prototype application for the evaluation of IRL car mass distribution has been studied and implemented within a KAE environment.

The tool presented was used to simulate mass properties of the car in relation to a set of independent parameters that define the car asset.

We will summarize the results of our work from two different points of view: one related to the skill and the approach required when developing a KAE application, the other about the developed system itself.

Knowledge acquisition was the most critical part of the work right from the start. During this phase, the experience of Dallara experts was fundamental to define the system requirements and the car model structure. The prototype de-

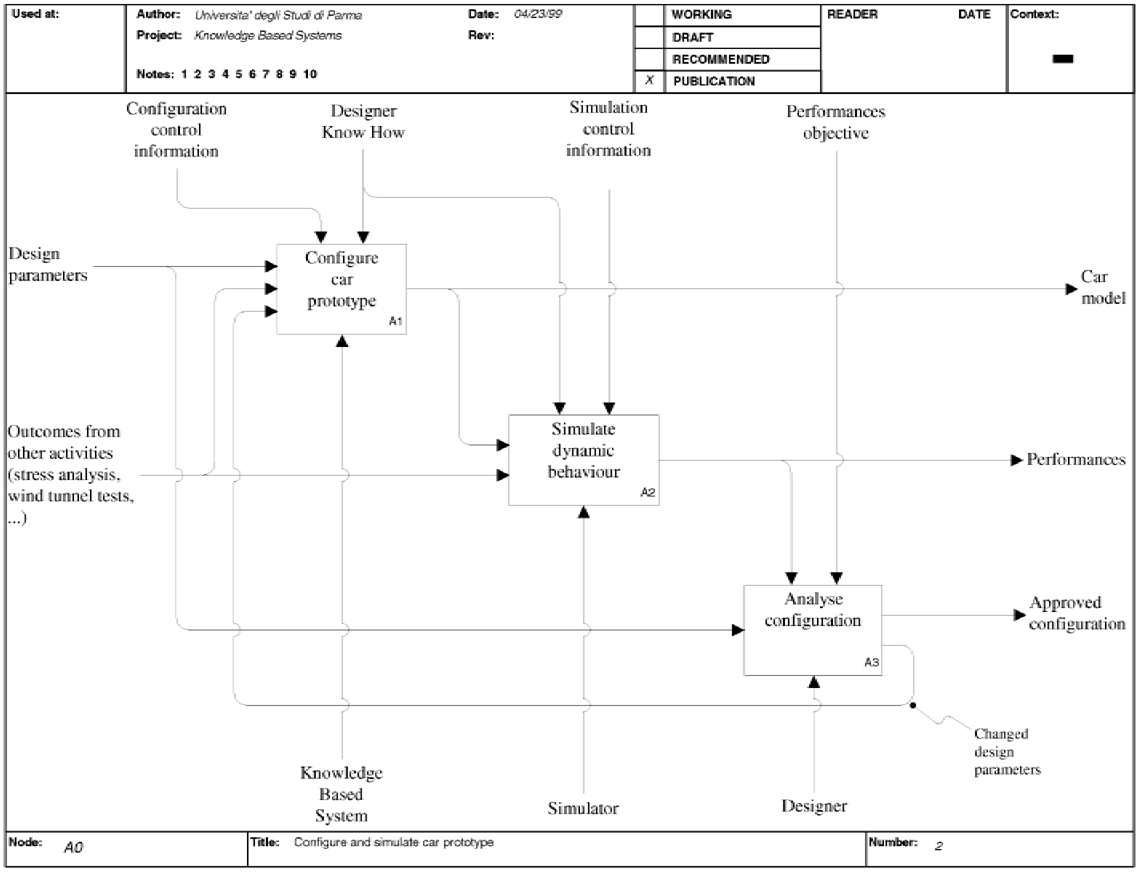


Fig. 13. Car development loop (IDEF0).

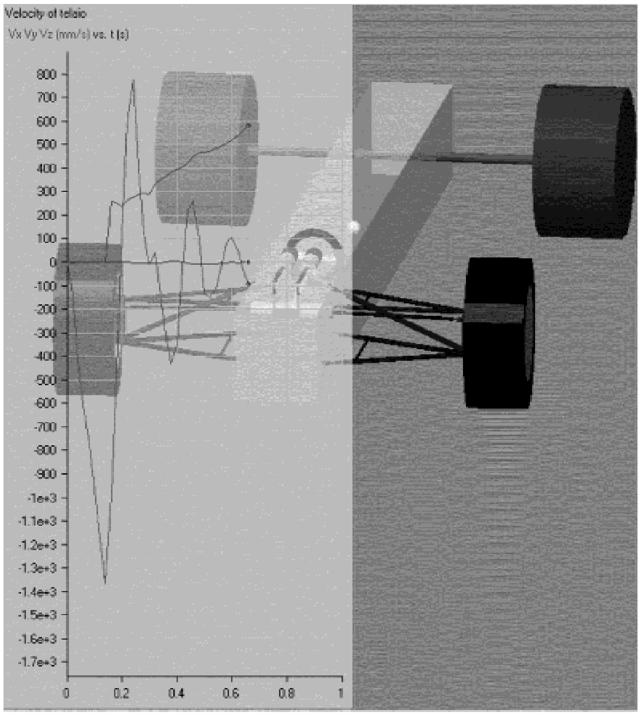


Fig. 14. Snapshot of a simulation test.

veloper had the same engineering background as the problem experts; however, a short training period was necessary before starting the system implementation. The fact that the actors had the same background guaranteed successful development, as it was possible to avoid the misunderstandings that often take place when experts of different domains (typically engineering and computer science) have to find a common language to transfer knowledge in the computerized system.

Talking about the prototype itself, we had to introduce some acceptable simplification for the generation of complex shapes. On the other hand, the flexibility provided by the language facilitated adjustments of the application on the basis of experimental results available. After this stage, the difference between the computed values of total weight and mass center and the experimental data is less than 0.57%. We also emphasize that the definition of external databases containing experimental data and standard components will facilitate future system updating.

Finally, the possibility to export geometrical and functional data in standard formats allows the system to share data with more traditional design/analysis support systems, and therefore to be fully integrated in the car development process.

REFERENCES

Colombo, G., & Cugini, U. (1992). Knowledge aided design: Requirements, systems and applications. *Revue internationale de CFAO et d'Infographie 7(3)*, 293–309.

Concentra Corporation (1995). Selling Point Reference Guide.

Cross, N. (1998). Expertise in engineering design. *Research in Engineering Design 10*, 141–149.

Guarino, N. (1998). Formal Ontology and Information Systems. *Proc. FOIS'98* (www.ladseb.pd.cur.it/infor/ontology/papers.html).

Indy Racing League (1996). Division Technical Specification.

Kariko-Buhwezi, B., & Cugini, U. (1995). A knowledge-based computer programming for the automatic design of a gear pump. *Proc. First IFIP WG 5.2 Workshop, Knowledge Intensive CAD-1 (KIC '95)*, 421–434.

Mandorli, F. (1997). Customized product design: Shearing dies test case. *Proc. 10 ADM Conf., Design Tools and Methods in Industrial Eng.*, 189–198.

Moulianitis, V.C., Dentsoras, A.J., & Aspragathos, N.A. (1999). A knowledge-based system for the conceptual design of grippers for handling fabrics. *AIEDAM 13(1)*, 13–27.

Ognjanovic, M. (1996). Decisions in gear train transmission design. *Research in Engineering Design 8*, 178–187.

Salustri, P. (1998). Ontological commitments in knowledge-based design software. In *Knowledge Intensive Computer Aided Design* (Finger, S., Tomiyama, T., & Mantyla, M., Eds.), pp. 41–73. Kluwer Academic Publishers, London, UK.

Soininen, T., et al. (1998). Towards general ontology of configuration. *AIEDAM 12(4)*, 357–373.

Vernadat, F.B. (1996). *Enterprise Modeling And Integration—Principles and Applications*. Chapman and Hall, London, UK.

Lorenzo Susca is currently a fellow at the University of Parma, Faculty of Engineering. He received a degree in Mechanical Engineering from the University of Parma in 1998. His current research interests include: solid modelling and simulation, knowledge aided engineering, and user interfaces for industrial applications.

Ferruccio Mandorli is currently Researcher at the University of Ancona, Faculty of Engineering. He received a degree in Computer Science from the University of Milan in 1990. He was guest researcher at University of Tokyo in 1992. He received his Ph.D. in Engineering of Industrial Production from the University of Parma in 1997. His research interests include: CAD systems, KAD applications, knowledge aided engineering tools, feature-based methodologies, and rapid prototyping.

Caterina Rizzi is currently an Associate Professor of Computer Aided Design at the University of Parma, Faculty of Engineering. She received a degree in Physics from the University of Milan in 1985. She has participated in several European projects funded by UE. She has been working in the area of CAD, CAM, and CIM since 1985 with particular interest in the following areas: CAD and KAD systems, modelling and behavior simulation of nonrigid materials, robots and industrial processes simulation, man–machine interaction in industrial applications, user-interface development tools, animation, and virtual reality in factory automation.

Umberto Cugini is currently a Full Professor of Technical Drawing at the University of Parma, Faculty of Engineering. He received a degree in Mechanical Engineering from Politecnico di Milano in 1966 and a Masters degree in management and Business Administration (University Bocconi, Milano). He has participated in several European projects funded by UE. His current research interests are as follows: intelligent design, system based on modelling techniques, knowledge engineering systems, robots and industrial processes simulation, modelling and simulation of nonrigid materials, simultaneous engineering, hypermedia techniques for CAD/CAM applications, man–machine interaction in industrial applications, animation and virtual reality in factory automation.