

A Knowledge Based Engineering Application to Support Weight Distribution Analysis for IRL Cars

L. Susca¹, F. Mandorli², C. Rizzi¹, and U. Cugini¹

¹*Dipartimento di Ingegneria Industriale, Universita' degli Studi di Parma*

²*Dipartimento di Meccanica, Universita' degli Studi di Ancona*

Key words: Knowledge based engineering, configuration system, weight distribution

Abstract: Knowledge based engineering is a technology whose aim is to provide tools to develop vertical applications which can integrate different types of knowledge representation within a common model and carry out model evaluation, modification and updating. This paper presents a knowledge based engineering application to support the design of Indy Racing League (IRL) cars. The IRL Championship is regulated by a very strict set of technical specifications, which limit the designer as far as car structure and sizes of the parts are concerned. For this reason it is very important to evaluate the effects of modifying main design parameters on the car configuration. The first part of the paper describes how Dallara's know-how on IRL car design has been re-organized with specific reference to the problem of weight distribution. This has led to the definition of a hierarchical IRL car model containing the fundamental parameters for weight computation. In the second part the use of examples highlights that different types of knowledge representation are necessary to compute weights distribution. The final part contains the description of the prototype implemented by using results from the previous phases. Comparing the results obtained with experimental data has validated the application.

1. INTRODUCTION

In mechanical engineering various design problems can be seen as configuration tasks, i.e., the overall structure and the design/manufacturing process of a family of products is already known, but, time after time, the designer has to select the right solution among possible alternatives in order to define the family member which meets specific requirements.

The designer has to quickly and accurately select the most desirable choices from hundreds or thousands options and instantly understand the implication on the final product of each selection in terms of cost, component incompatibility, mandatory requirements, performances, and manufacturing and delivery constraints.

In general, to evaluate several possible alternatives, the designer has to manage different information typologies, concerning many product aspects (e.g., machining processes, marketing perspectives, physical resources, properties of materials, engineering design). Thanks to personal expertise and knowledge about the problem the designer is able to evaluate the best solution intuitively; this solution then needs to be verified before it can become operative.

While developing a decision support system for a configuration problem (AIEDAM 1998), a key aspect is the definition of a suitable knowledge base containing all the different types (declarative, procedural, control) and forms (tables, geometry, functions, etc.) of the knowledge required, organized into a structured product model. This product model becomes the kernel of the system and the designer can inquire it in order to have feedback on feasible alternative solutions.

In this work we describe a system which supports a configuration problem related to the design of Indy Racing League (IRL) cars. The developed knowledge-based application is able to evaluate different configurations of the car and resulting distribution of weight, a key parameter to evaluate car performances.

The paper presents underlying philosophy and motivations that guided the development of the system; it also describes how Dallara's know-how on IRL car design has been re-organized with specific reference to the problem of weight distribution. This phase of the analysis has led to the definition of a hierarchical IRL car model where atomic components have been described in terms of a set of significant properties, and grouped to form hierarchical sub-assemblies.

Examples are also utilized to show how different types of knowledge representation are necessary to compute weight distribution. The final part is about the prototype implemented by using results from the previous phases. The prototype was developed using Selling Point by Concentra (Concentra

1995) and validated by comparing the results obtained with experimental data.

2. MOTIVATIONS AND OBJECTIVES

A new racing car championship was introduced in 1996: the Indy Racing League (IRL) championship is held on oval circuits in the US. Its aim is to allow teams with a limited budget to participate in the race. A very strict set of technical specifications was established (Indy, 1996), and the number of car producers has been limited to two engine producers, two tire suppliers, and two manufacturers for the chassis and the bodywork. Dallara, an Italian company involved in the production of racing cars, is one of the chassis and bodywork producers.

In order to beat competitors, designers have to define specific sub-goals that directly concern some product features and guarantee the achievements of the original aims. For example, in order to improve car performances in acceleration and deceleration the designer can work on brakes and engine power, or he/she can lighten the car. The two alternative solutions produce the same main result, but in very different ways and with different secondary consequences (e.g., changing the material to lighten a part requires verifying that new mechanical properties assure structural resistance under stress conditions). Moreover, other constraints must be verified, such as functionality of the part, compatibility between the new material and other ones used, relationships among parts, machining and so on.

In this paper we have focused the attention on the critical problem which deals with the impact of design decisions on the car weight distribution.

Traditionally design alternatives are evaluated by performing simulation on a scale model of the car. It is evident how expensive and time consuming this solution is in respect with car weight distribution analysis (all parts should be included in the model).

On the other hand, CAD systems are useful for detailed and realistic geometric representation of one or more parts, but they are not suitable for complex structures involving “fusion” of different knowledge contexts (geometric, functional, technological...). For these reasons, the designer usually evaluates different car configurations relying on his/her own experience and knowledge of car structure and the effects of modifying design parameters on all sub-assemblies.

In such a context our objective has been the development of a decision support system able to evaluate the weight distribution for different design

solutions of an IRL car. The system should support the designer and allows him configuring the car and evaluating consequent weight distribution.

The interaction between the system and the designer generates a loop (Figure 1), in which the operator provides alternative ideas (possible car configurations) and the computer evaluates car asset and weight distribution of the current solution. The output from the designer (values of independent parameters) is the input for the computer, which uses the data to configure the model. At the same time the output from the computer (car asset) is the input for the designer, who interprets the results and re-considers the initial hypothesis to move towards an optimum solution.

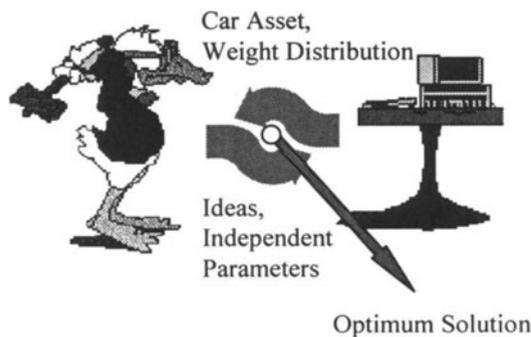


Figure 1. Computer-man interaction

To address such a problem, different types of knowledge need to be managed and related to each other. Knowledge based engineering technology can be used since it provides the tools to represent the designer's decision making process, integrate different types of knowledge within a common model and carry out model evaluation, modification and updating (Cornet 1991, Kariko-Buhwezi 1995, Di Lecce 1997, and Mandorli 1997).

A KBE shell is a development tool made of: a proprietary Object Oriented (OO) language to define the product model, a tracer to support the debugging activity, a set of drivers to communicate with external data base, and a set of libraries to integrate the model within other available tools for the user interface development (e.g. Visual Basic, Visual C++).

Libraries of basic objects are supplied with the shell in order to provide basic functionality to the language. These libraries contains objects that can be roughly grouped into the following categories: objects to handle series of clones, objects implementing methods to access external data bases, objects implementing calls to a geometric modeler and objects to handle input/output functionality.

The OO language allows the user to define models, which represent the product in terms of its basic components and their following aggregation in functional sub-systems down to the complete system. Each model component is defined by its name and a set of property/expression couples representing respectively the name of a property and the property value. The expressions can be any valid sentence of the language. These expressions are usually called "rules", but notice that in KBE context, the word "rule" does not have the traditional meaning used in Artificial Intelligence, but it is considered as a symbolic relationship among several properties of the parts).

These symbolic expressions (that can represent appropriated dimensioning, positioning, analyses relationships among properties of the model components) represent the model parameterization and define the model behavior when a property changes.

The shell provides the mechanisms, which update and check properties consistence. These mechanisms are called *Demand Driven Processing* and *Dependency Backtracking* (notice that a KBE does not have an inference engine). The Demand Driven Processing mechanism checks that the properties of a model are calculated only when they are explicitly required. The Dependency Backtracking mechanism automatically updates all the properties that depend on a modified one.

The kernel of a KAE application is then a hierarchical structure representing the product model in terms of its part and sub-parts defined by all their significant properties, related to each other by appropriated symbolic expressions. It is important to notice that, thanks to the flexibility provided by the KAE shell, a KAE application can represent the product related knowledge in all of its required forms (See Figure 2).

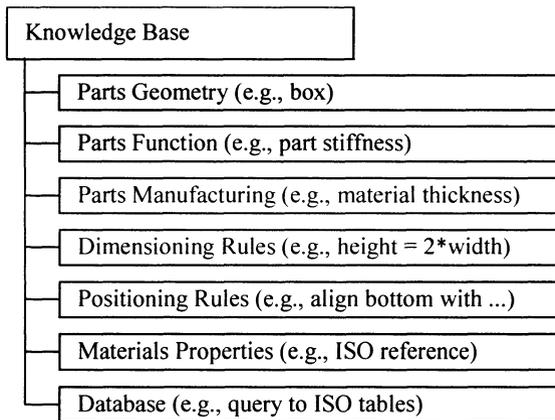


Figure 2. Different forms of Knowledge representation in a KAE application

Following sections contain a detailed description of the knowledge required to address the problem of weight distribution, and how it has been structured into the developed KAE-based model.

3. STRUCTURE OF THE IRL CAR MODEL

A hierarchical model of an IRL car, containing all types of data (shapes, materials, part positioning, standard components) which are relevant for the correct computation of weight distribution, has been defined.

The IRL car structure has been studied in order to identify sub-assemblies and parameters that are significant for the weight evaluation. This activity was carried out in close collaboration with Dallara engineers who provided their know-how on the problem.

The model hierarchy reflects the functional sub-assemblies, which constitute the car. The parameters needed to carry out the correct evaluation of weight distribution have been identified for each sub-assembly and related to each other.

Several sub-systems have been analyzed: the bodywork, the chassis, the front and rear wings, fuel, water, oil and exhaust systems, pedal, steering and wheel assembly, etc. (see Figs. 3-5).

One of the most important parts is the chassis. It is made of composite material and its shape is defined by the geometry of four bulkheads (Figure 4), dimensioned and positioned within ranges imposed by the league's technical specifications.

The shape of the four bulkheads also defines the main cross sections of the chassis: the first bulkhead (front bulkhead, Figure 4-1) is positioned as interface between the front part of the chassis and the front wing. The second bulkhead (pedal bulkhead, Figure 4-2) is positioned close to the pedals. The technical specifications place constraints on the minimal cross section in this area and the distance from the front bulkhead.

The third bulkhead (dashboard bulkhead, Figure 4-3) 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, Figure 4-4) is positioned behind the driver's seat.

The tank is located behind the seatback bulkhead, into the space between the seat and the engine. The back part of the chassis is constrained by the shape of the engine and there are no particular specifications for this part, as just two types of engine are allowed.

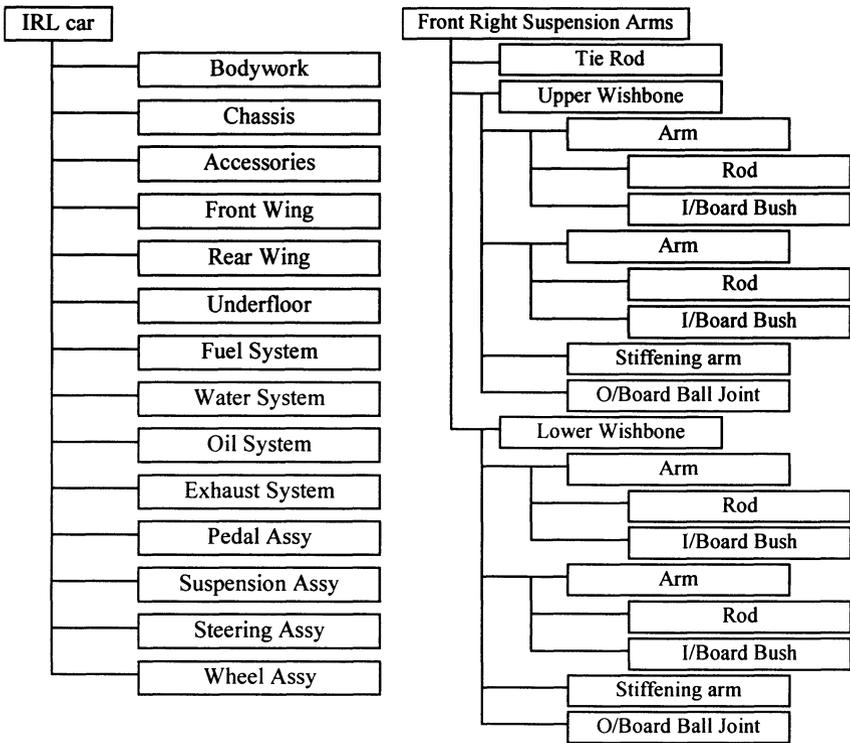


Figure 3. First level of car hierarchical structure and suspension arms assembly structure

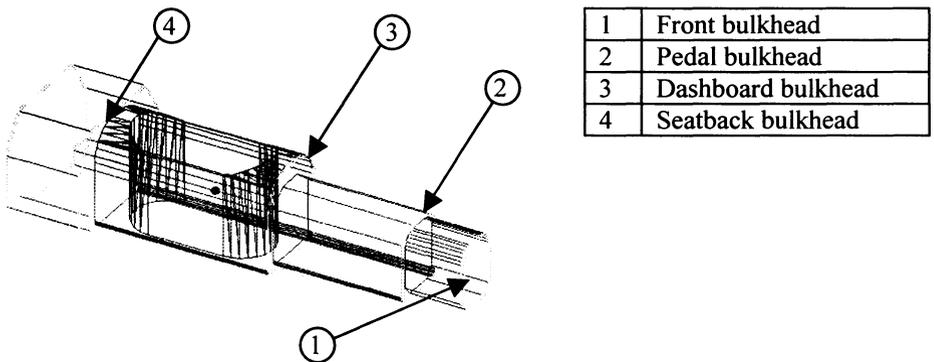


Figure 4. Chassis structure with bulkheads

However, there are specifications to constrain the shape of the underfloor, which must be flat, at least 1778 mm long, with a maximum

angle of 5° from the engine interface to the pedal bulkhead and 10° from the pedal bulkhead to the car nose.

The engine, the bellhousing and the gearbox make up the back part of the car. Those parts also provide interfaces to connect other sub-assemblies, such as the suspensions.

Suspensions have been carefully analyzed because different suspension configurations may generate different weight distribution and therefore different car attitudes.

A tie rod and a couple of wishbones make up the suspension arms. The base vertices of the wishbones are fixed with spherical joints respectively to the chassis (front suspensions) and to the bellhousing and the gearbox (rear suspensions); the opposite vertex is fixed to the hub carrier. The front suspension arms have an additional stiffening rod located between the wishbones.

Notice that the bases of the upper and lower wishbones of the front suspension arms may be different in size. However, the wishbone vertices need to be aligned because they define the wheels' vertical axis of rotation for the steering and the toe-in. The damper and the antirolling systems are directly connected to the suspension arms' assembly.

Figure 3 (tree on the right) shows the hierarchical structure of the sub-assembly representing the front right suspension arms.

4. WEIGHT AND CENTER OF MASS COMPUTATION

The weight of the different components that make up an assembly or sub-assembly can be computed in different ways: from data bases storing sizes of standard parts (e.g., spherical joints) provided by external suppliers; from data bases containing experimental results; from the computation of formulas containing references to the size, material, and position of other parts.

In the following we report, as an example, the weight computation of some of the components mentioned in the previous section.

Theoretically, the weight of the chassis is computed as follows:

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

where:

- P = computed weight of the chassis
- $P_{stiffening}$ = total weight of additional stiffening parts (retrieved from a data base)
- $d_{honeycomb}, d_C, d_{glue}$ = density of the honeycomb, carbon fiber and glue
- $S_{honeycomb}, S_{C1}, S_{C2}$ = thickness of the honeycomb and internal and external carbon fiber
- ds = infinitesimal surface element

In practice, the weight is computed by approximating the previous formula, i.e., it is calculated for each one of the four sections defined by the bulkheads and a sum is applied as follows:

$$P = \sum_{i=1}^4 P_i + P_{stiffening}$$

$$P_i = C_i \cdot [(d_{honeycomb} \cdot S_{hi}) + (d_c + S_{cli}) + (d_c \cdot S_{c2i}) + d_{glue}] \cdot Sup_i,$$

where:

- P = computed weight of the chassis
- P_i = weight of each section
- $d_{honeycomb}, d_C, d_{glue}$ = density of the honeycomb, carbon fiber and glue
- S_{hi}, S_{cli}, S_{c2i} = thickness of the honeycomb and internal and external carbon fiber for the i-th section
- Sup_i = area of the i-th section

The weight of the suspension components (wishbones and stiffening arms) can be computed from their volume and density. The volume depends upon the dimensions of the components; these are based on the dimensions and position of the spherical joint. The joints are selected from a database of standard parts, depending on the strength required (See Table 1).

As for the joints, the weight of other standard parts (dampers) can be retrieved from databases too. Regarding the wheels, the technical specifications limit their dimensions and therefore the resulting weight.

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

Code	D int	D ext	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

The car's center of mass is computed with respect to a reference coordinate system. The X-axis of this system coincides with the car's longitudinal axis and is oriented in the direction of its motion. The Z-axis is the vertical axis and its origin is located in the intersection between the longitudinal axis and the front wheel axis. The XY plane coincides with the underfloor, as the XZ plane defines the car symmetry plane.

The distribution of weight at the front and back can be computed from the car's center of mass as follows:

$$P_{front} = \frac{P \cdot d_{back}}{D}$$

$$P_{back} = \frac{P \cdot d_{front}}{D}$$

where:

P_{front} , P_{back} = load respectively on the front and back car axle

P = total car weight

d_{front} , d_{back} = center of mass distance from the front and back car axle respectively

D = car axle base

5. DEFINITION OF THE IRL CAR MODEL

Some fundamental functionality is required in order to implement an automatic application which supports the evaluation of different configurations and corresponding weight distribution. These include the possibility to:

- represent the hierarchical structure of the car in terms of parts, sub-parts and related properties (as required by the analysis performed);
- represent relationships among properties of the different parts and re-compute the whole model as a consequence of modified properties;
- link external data bases;
- generate geometric models of parts and sub-parts (to compute volumes, weights and centers of mass).

Selling Point by Concentra is a KAE development shell, which provides mentioned functionality. This tool has been used to implement the prototype.

The car model is defined as a tree where the root represents the whole car, nodes represent sub-assemblies and leaves represent atomic components. The root stores general parameters, which have an impact on the whole car configuration, and nodes and leaves store local parameters, which have an impact on specific sub-assemblies.

The components whose weight and center of mass need to be computed starting from their shape (e.g. wishbones and stiffening arms) are represented by leaves, which inherit mass properties from linked solid models. These leaves must also store parameters, which represent the appropriate positioning of the solid model.

Components whose weight is known and which contribute to the total car weight, with (e.g. engine) or without (e.g. liquids) a significant geometric representation, are represented by leaves, which are linked to external databases storing their weight values.

Figure 5 shows the geometric representation of the car in standard configuration.

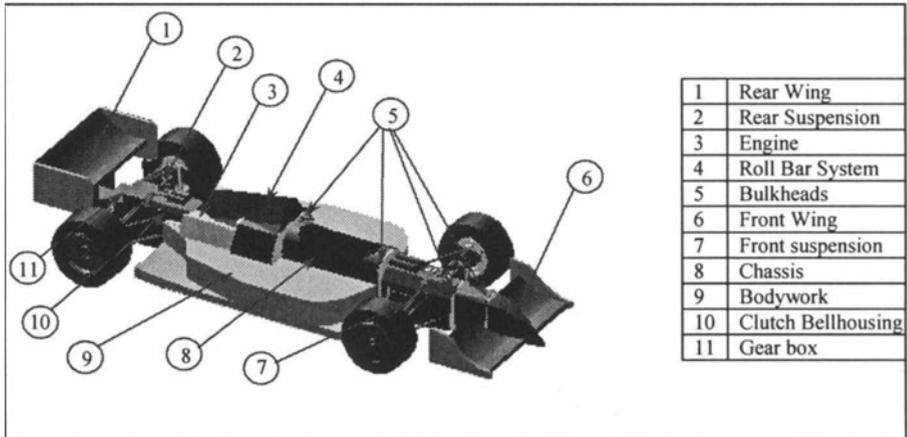


Figure 5. Geometric representation of the car in standard configuration

The user interacts with the system modifying general and local parameters and looking at the resulting configuration. Computed results are reported to the user in both graphic and numerical representation. The main general parameters, which the user can set, are shown in Table 2, together with an indication of the impact they have on the configuration of the car.

Table 2. General model parameters and related impact on model configuration

Parameter	Impact
front, back track	position and dimension of suspensions
rear wing height from reference plane (constrained by safety specifications)	vertical dimension of wing support
underfloor height and angle from the ground	front wing and suspension configuration
ballast position	weight distribution
Weight of the driver	weight distribution
Percentage of tank filling	weight distribution

Different configurations, related to different values of the front track, are shown in Figure 6. Notice that in the picture on the right, the ARB system center of mass (small dot in the center of the ARB system) is located behind the axle, whilst in the one on the left it is located on the axle.

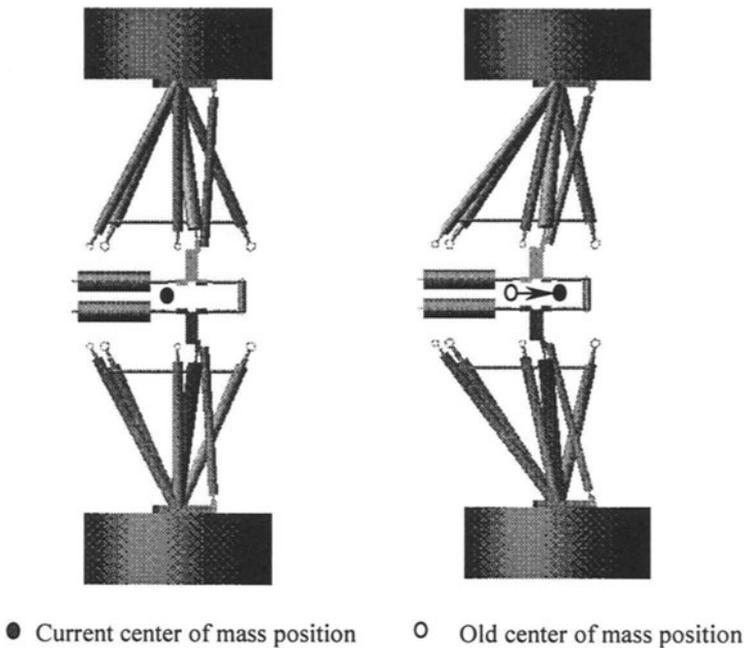


Figure 6. Top view of front suspensions in different configurations

The system can be upgraded by changing the declarative knowledge stored in the external database. Due to the nature of the configuration problem considered, an upgrading of procedural knowledge is not generally required but, if necessary, it can be done by changing the model structure.

6. EXPERIMENTS AND RESULTS

Front and back loads have been computed for different car configurations and different tank load situations, i.e., three conditions (93.5%, 39% and 0%, respectively 35, 15 and 0 gallons), with and without driver, and three wheel base configurations.

The experimental data related to the car base configuration (named wheel base 2998) without driver were available. These data (values in brackets in Table 3) were provided by the racing teams and compared with computed data (Table 3). Approximations are due to the fact that the geometry of some parts is not precise (e.g., the shape of the chassis) but was approximated.

Table 3. Computed loads for different car configurations

Tank Load %	Driver's Weight (Kg)	Front Load (%)		
		Wheel base 2998	Wheel base 3048	Wheel base 3098.8
93.5	0	41.39 (41.43)	40.85	40.33
39	0	40.90 (40.96)	40.4	39.92
0	0	40.41 (40.18)	39.91	39.43
93.5	68	43.03	42.44	41.89
39	68	42.67	42.12	41.60
0	68	42.37	41.81	41.29
Difference Full/Empty (%)		0.66	0.63	0.60

Table 3 shows how the car's center of mass moves to the back as fuel is consumed; this situation is reflected in a different performance of the car attitude during the race. The difference in the front load between the conditions of full and empty tank can be taken as a measure of the center of mass variability, and must be minimized.

The car's center of mass moves because the tank's center of mass is not located in the same X position of the car's total center of mass (Figure 7 and Table 4 show how the location of the center of mass of the tank/fuel system depends on the fuel load).

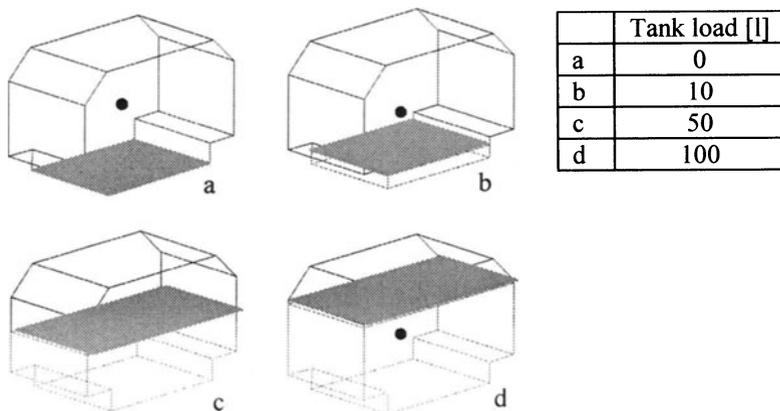


Figure 7. Location of the tank's center of mass for different load conditions

Table 4. Center of mass co-ordinates values

Tank Load		Center of mass location [mm]		
%	l	X	Y	Z
0	0	-1578,5	0	22,88
10	12,5	-1562,1	0	16,57
20	25,9	-1556,9	0	14,4
30	41,7	-1557,4	0	14
40	57,1	-1559,1	0	14,66
50	71,6	-1562,2	0	15,81
60	85,1	-1566,5	0	17,12
70	98,74	-1569,8	0	18,66
80	112,4	-1572,5	0	20,36
90	124,4	-1574,5	0	22
100	135,2	-1576	0	23,56

One possible solution of the problem is to move some parts (e.g. battery, electronic components, and radiator) to the front of the car so that the car's center of mass may coincide with the tank's center of mass.

Of course, functional components are not always easy to move, due to other considerations such as bounding volumes, functionality, etc.

The ballast is the easiest component to locate, however within constraints imposed by technical specifications. Computing the car weight in standard configuration (empty tank and no driver) and then subtracting it from the minimum allowed weight (734 Kg) sets the weight of the ballast. It is evident that the lighter the car is, the heavier the ballast must be, in proportion to the weight, which is necessary to move the center of mass.

A new car configuration has been worked out, by modifying many parameters in order to reduce weight, and the ballast has been located as near as possible to the front of the car.

In the new configuration, with a wheel base set at 3048 mm and a driver weighting 68 Kg, the front load distribution varies from 44.78% (full tank) to 44.47% (empty tank), thus the front load difference between the two conditions is 0.31%, i.e., about half of the value computed for the configurations previously presented.

Figure 8 shows the car's and tank/fuel centers of mass for both the standard and the new configurations. Notice how the two center of mass are closer to each other in the new car's configuration.

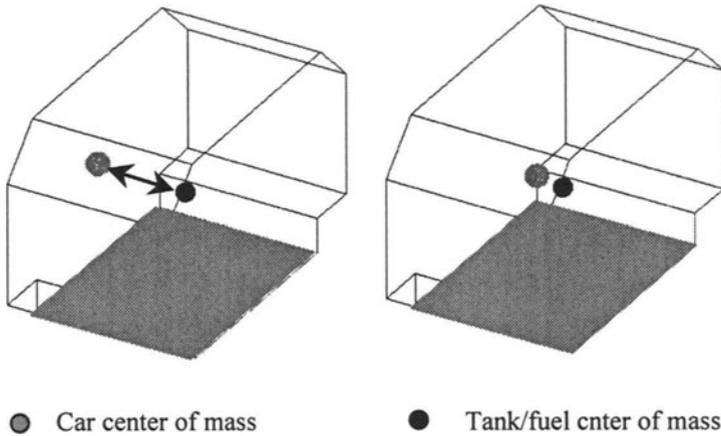


Figure 8. Car's center of mass compared to the tank's center of mass for two different car configurations

7. CONCLUSIONS

A prototype application for the evaluation of weight distribution in IRL cars has been studied and implemented within a KBE environment. The application is based on a hierarchical model containing different types of engineering knowledge.

Many kinds of data have been used to represent all information involved in weight computation: constant values (to represent input parameters), formulas (to represent evaluated parameters), SQL strings (to represent queries to external databases), geometric models (to represent shape of the components), and constraints (to represent relative positioning of components). Notice that all pieces of information are related to each other, in the sense that the procedure to compute one value may refer to other computed values (e.g., the dimensions of components, needed to compute weight and center of mass, may depend on properties of the material or applied loads).

The comparison of computed data with experimental ones provided by the racing teams validated the model. After the tuning activity, the difference between the computed values of total weight and center of mass and the experimental data is less than 0.57%.

The prototype has been used to study a new car configuration that allows reducing the changes in the car's center of mass as the fuel is consumed in the race.

The objective is not to produce detailed models of car sub-systems; however, the design parameters and related data provided can be used as an input to other specific software tools, like traditional CAD systems, which can perform this job. Moreover, it is possible to achieve kinematics analysis, FEM analysis, etc, starting from the geometric models of car sub-assemblies, generated automatically by the prototype. In this context, the possibility of re-using car prototype information in commercial CAD/CAE systems has been tested. In particular we have verified two different integrative solutions:

- the first one is about the automatic generation of 3D CAD model of car components;
- the second one concerns the possibility of exporting and re-using geometric elements produced by the prototype in a simulation environment; the objective has been to analyze dynamic and cinematic behavior of the car and of the front suspension assembly.

Future developments are as follows:

- extension of the model parameters for the computation of moment of inertia;
- implementation of a user-friendly interface for the final user (at present model parameters are set by the user interface of the development tool);
- porting on an equivalent development tool with higher capabilities for surface representation (at present the bodywork is approximated with a set of extrusions with constant cross sections).

AKNOWLEDGMENTS

We would like to thank staff from Dallara, and in particular Ing. Andrea Toso e Ing. Davide Seletti for their collaboration and for the information provided. A special thanks to all reviewers for their valuable suggestions.

REFERENCES

- Concentra Corporation (1995), Selling Point reference guide.
- Cornet, P., Elu, P., Dossier, G., (1991), Systemes de conception a base de connaissance: une application industrielle, Acte de MICAD, pp. 598-608.
- Di Lecce, B., Tagliabo', F., Mandorli, F., Accornero, B., Strata, A., (1997), A Knowledge-Based tool for automatic generation of wire harness design and schematics, 6th European Conference, Lightweight and Small Cars, the Answer to Future Needs, V: 2, pp. 987 - 998.

- Kariko-Buhwezi, B., Cugini, U., (1995), A Knowledge Based Computer Programming for the Automatic Design of a Gear Pump, Preprints of the First IFIP WG 5.2 Workshop, Knowledge Intensive CAD-1 (KIC '95), T. Tomiyama, M. Mantyla and S. Finger eds., 421-434.
- Indy Racing League (1996), Division Technical Specifications.
- Mandorli, F., (1997), Customized Product Design: Shearing Dies Test Case, 10th ADM Conference, Design Tools and Methods in Industrial Engineering, pp. 189 - 198.
- AIEDAM, (1998), Special Issue: Configuration Design, William P. Birmingham, Editor, V. 12, N. 4, ISSN 0890-0604.