MODIFICATIONS DE FORMES COMPLEXES

COMPLEX SHAPE MODIFICATIONS

Patrick BOSINCO*, Gérard DURAND*,
Jacques GOUSSARD*, André LIEUTIER*,
Alain MASSABO*

**MATRA DATAVISION, Direction Scientifique*
53, avenue de l'Europe
13082 Aix en Provence Cedex 02
a-massabo@aix.matra-dtv.fr

RÉSUMÉ

La Conception par Objectif (optimisation) permet d'aider les utilisateurs à résoudre des problèmes ayant un grand nombre de paramètres. La modification de formes peut être traitée de cette façon. Réalisée dans le cadre d'un projet Brite-Euram (FIORES), MATRA DATAVISION présente une solution pour la branche directe de l'optimisation, sous forme de composants CAS.CADE. Les résultats laissent entrevoir des gains conséquents dans les processus de conception de formes complexes.

ABSTRACT

Generally speaking, Engineering in Reverse (EiR) is a process for generating a model from its properties. As there is often no biuniqueness between a model and its properties, an initializing model is required to start refinement in an EiR loop. It helps the user to solve complex problems having a high number of control parameters. One of its implementations could be used to control the shape of a model via target properties (Target Driven Design). The **modification of shapes** is an important issue of computer-aided geometry whether to fit local/global aesthetic requirements, or engineering constraints. A modification operation has to satisfy certain criteria while keeping, or even improving, the original shape quality **independently of its underlying representation.** This functionality is a good candidate for EiR. The algorithm that permits the elementary action of the forward branch of a EiR process is required. Such an algorithm could be implemented as a software component. The purpose of this contribution is to show how MATRA DATAVISION addresses this issue. Since the work is related to a Brite-Euram project (FIORES), a brief overview of it will precede the presentation of a MATRA DATAVISION CAS.CADE component. The resulting functionality is, by far, a step ahead from what is available today on the market. A significant decrease in the final style tuning time can be expected.

I. INTRODUCTION

Generally speaking, Engineering in Reverse (*EiR*) is a process for building or generating a model from its properties. Up to now, it has been successfully used in domains where properties can be quantifiable, such as the optimization of mechanical parts under high levels of stress, and is becoming more and more broadly and efficiently implemented in CAD/CAM/CAE systems.

As there is often no biuniqueness between a model and its properties, an initializing model is required to start refinement in an EiR loop. At MATRA DATAVISION, we call this approach **Target-Driven Design** (*TDD*). It helps the user to solve complex problems having a high number of control parameters. One of its implementations could be used to control the shape of a model via target properties.

The **modification of shapes** is an important issue of computer-aided geometry whether to fit local/global aesthetic requirements (in styling), or engineering constraints, or to anticipate mechanical behavior such as spring effects. In general, CAD shapes are piece-wise defined (*Brep*[1]). Each piece is connected to the others with a given order of continuity. A modification operation has to satisfy certain criteria while keeping, or even improving, the original shape quality **independently of its underlying representation.**

Considering the number of its control parameters, this functionality is a good candidate for TDD. However, an algorithm that permits the elementary action of the forward branch of a TDD process is required (see figure below). Such an algorithm could be implemented as a software component.

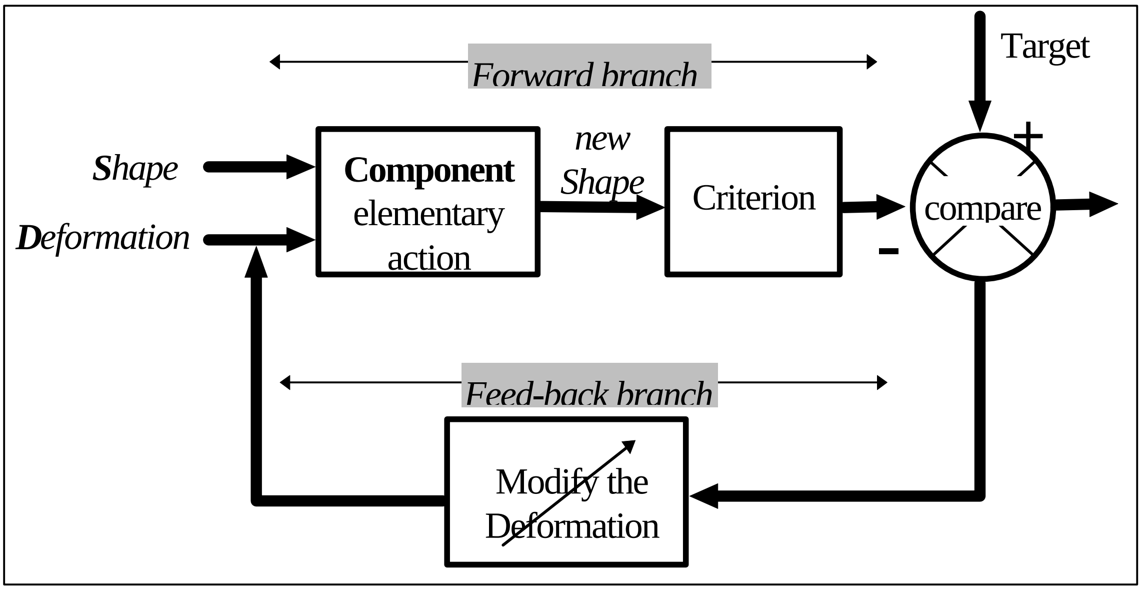

Fig.1: Engineering in Reverse

Conception par objectif

The purpose of this contribution is to show how MATRA DATAVISION addresses this issue. The component allows the modification of an **n-sided area of piece-wise defined shapes** while keeping **Gk-continuity** constraints along the boundaries of the domain as well as inside it (see figure below).

[1] **B***oundary* **rep***resentation.*

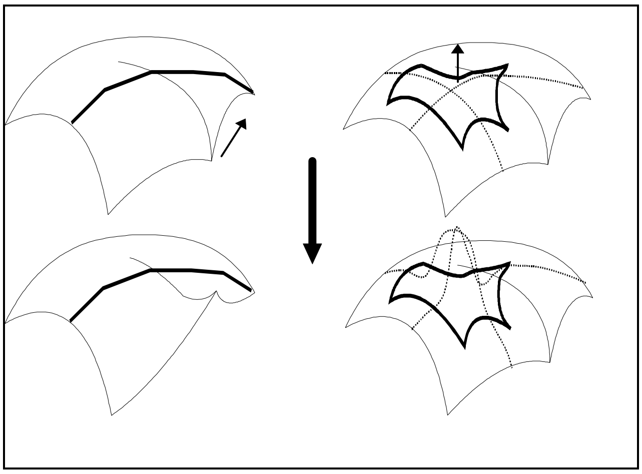

Fig.2: Two deformation cases
Deux cas de déformation

The following items will constitute the presentation:

- explanation of why the component is designed to facilitate its use in a TDD process,
- general description of the underlying mathematics,
- component software architecture,
- different cases illustrating the possibilities of the component,
- conclusion.

Since the work is related to a Brite-Euram project called FIORES[2], a brief overview of this project will precede the presentation of a MATRA DATAVISION CAS.CADE[3] component.

II. FIORES

II.1. Background

In a very competitive world wide market, styling is the key factor for the success of a product. In addition, the process of styling is becoming more complex, since new materials provide a greater freedom to the designer's creativity. As a consequence, companies have to be successful in improving efficiency and quality of their styling processes. While the use of computer-aided technologies in the area of mechanical product design is quite well established, the gap in the computer-aided process chain due to the styling phase is a severe problem: styling of complex-shaped products (like car hoods, consumer appliances, toys, packaging, etc.) is still a domain of hand drawing and model building. The main reason for this is that the system handling does not fit the way stylists work. Moreover, there is a lack of objective formal criteria for evaluating aesthetic shape properties, which leads to drawbacks in the design process chain, such as:

- expensive manual work on "scale one" models before the final style decision,
- bad communication between aesthetic and engineering design.

The necessary translation of the styling model into the CAD format for further processing leads to discrepancies, loss of time and forbids concurrent engineering, making any feedback from downstream process phases very expensive.

[2] **F***ormulation and* **I***ntegration of an* **O***ptimized* **R***everse* **E***ngineering* **S***tyling workflow* http://rkk.mv.uni-kl.de/FIORES.

[3] It is a MATRA DATAVISION product: complete set of Object Oriented C++ classes for 3D modeling and scientific application developments.

II.2. Goal

The goal of the FIORES project is to provide an innovative solution to the above problems. On one hand, a new user interface combined with a new set of tools for checking the shape quality will increase the stylist's acceptance of computer-aided tools. On the other hand, the user will have the possibility to specify the target properties of the product shape in such an objective way that the communication will be improved and techniques for engineering in reverse (EiR) can be used to build the proper model.
By redesigning the definition process, FIORES will constitute a breakthrough in CAD/ CAM treatment of complex-shaped products involving an innovative information technology to support all engineering functions for the product life cycle.

II.3. Strategy

FIORES has started in January 1997 as a Brite-EuRam project (BE96-3579) and has a duration of 3 years. It will realize its goals according to the following strategy:

- analysis of the application areas for computer-aided styling, by collecting information about used and desired tools, methods and information flows,
- formalization of aesthetic properties of free form shapes, by also using feature-like concepts as they are known from mechanical engineering,
- development of new theoretical methods, concepts and algorithms for design and quality control in the process of engineering aesthetic shapes in reverse,
- implementation of a software prototype, based on a modular, object-based approach,
- validation and measure of the prototype on real industrial parts within the companies.

II.4. Benefits

The benefits of the project will be equally distributed among all involved groups, but the greatest impact will be achieved in the application industry:

- a new workflow in the production process of complex-shaped objects will avoid expensive optimization loops, with the parallel benefit of a higher aesthetic product quality,
- new functionality for aesthetic design will help the design departments of the industrial partners and of other enterprises to be more efficient in regard to time and quality,
- a new modular system architecture will enable smaller companies to get access to **C**omputer **A**ided **A**esthetic **D**esign (**CAAD**) which, up to now, has mainly been available for large and powerful companies. Thus, these small companies can be more easily integrated into extensive workflow in cooperation with large companies.

II.5. Consortium

The FIORES consortium was assembled with the aim of combining both complementary and common skills in the areas of fundamental research as well as in industrial application and CAD/CAM system supply as a basis for a tight cooperation. It consists of 12 partners from 6 European countries:

The **Universität Kaiserslautern**, Germany, acts as Coordinating partner and contributes with experience in the field of CAD/CAM technology and user interface implementation. Additional research institutes are **IMA-CNR** (Istituto per la Matematica Applicata del Consiglio Nazionale delle Ricerche, Genova, Italy) mainly focusing on Feature-based Modeling and **CIMNE** (Centro Internationel de Métodos Numéricos, Barcelona, Spain) who are experts in Geometric Modeling.

MATRA DATAVISION, Paris, France, acts as the main CAD system developer and vendor in the project, which is especially skilled in complex shape design tools. **SAMTECH**, Liège, Belgium, contributes with multi-purpose optimization software, while the small Swedish company **UDK Utveckling**, develops special mathematical software.

Some kind of bridge to the application companies is an other Swedish company, **FORMTECH**, also located in Göteborg, which is a design and styling studio also writing application software. Pure application companies in the consortium are **BMW**, München, Germany, **PININFARINA**, Torino, Italy, and **SAAB Automobile**, Trollhättan, Sweden, with their styling or car body design departments. For capturing the non-automotive styling sector, the small Spanish companies **EIGER** and **TAURUS**, both located in Barcelona, Spain, and both designing consumer appliances joined the consortium.

III. THE COMPONENT

III.1. Objectives

As presented above, one of the main part where time is spent in the "styling process", is the final tuning of the shape. In this phase, local and global slight modifications of shapes take place. Our aim is, therefore, to provide a tool that can be used either in a direct mode (modify the shape and look at the result) or in an EiR one (define the target and get the "closest" model).

III.1.1. Addressed problems

- The modification of a Shape area, independently of its underlying representation,
- Under quality and continuity constraints (i.e. keeping or even improving the original shape quality),
- Such that it can be used either in a direct mode or in a EiR one,
- Therefore, with a minimal set of parameters,
- The resulting Shape being directly usable in a general CAD/CAM process.

III.1.2. Input data

- Shape: CAS.CADE definition (might be a Non Manifold Topology) in general STEP compliant,
- Domain: any set of n-sided G0-continuous 3D loops,
- View of modification: mapping plane (cf. III.3.),
- Constraints: {points, curves}⊗{G-1 (rip), G0, G1, G0+G1, G1+G2, G0+G1+G2},
- Handles (used to control the deformation):
  {points, curves}⊗{G-1 (released handle), G0, G1, G0+G1, G1+G2, G0+G1+G2},
- Accuracy,
- Quality check modes or customized methods: {planar sections with/without curvature, isophotes lines, shadow lines, Gauss curvature maps}.

III.1.3. Output data

- Exact: in CAS.CADE, procedural representation may be used,
- Approximated by NURBS[4], for exchanging with other CAD/CAM systems,
- Tessellated for real time use,
- Quality check results: {planar section(s) with/without curvature, isophotes lines, shadow lines, Gauss curvature maps}.

III.2. Designed for TDD and Direct Design

The component of the forward branch of the EiR process (cf. figure 1) can be embedded in a direct approach mode. In this case, it is activated by the user himself in an interactive action. In addition, when embedded in an EiR process, the component must offer various levels of activation such as initialization, exit, feedback computation. Its implementation satisfies the following criteria:

- modularity, a hierarchy of components constitutes the final one,
- reusability, as mentioned it can be used either embedded in an interactive action or in an EiR forward branch. Some of its "*internal components*" can be reused in other contexts such as, real time checks for other interactive actions or filling a n-sided hole,
- uncoupling, for allowing evolution of its semantic with minimizing the code modification of the using environment.

[4] **N***on* **U***niform* **R***ational* **B**-**S***pline*

III.3. Underlying mathematics

The basic idea is to add a vector **D**eformation function to a given Shape. For easiness, 2D drawings will be used, they correspond to sections of 3D entities.

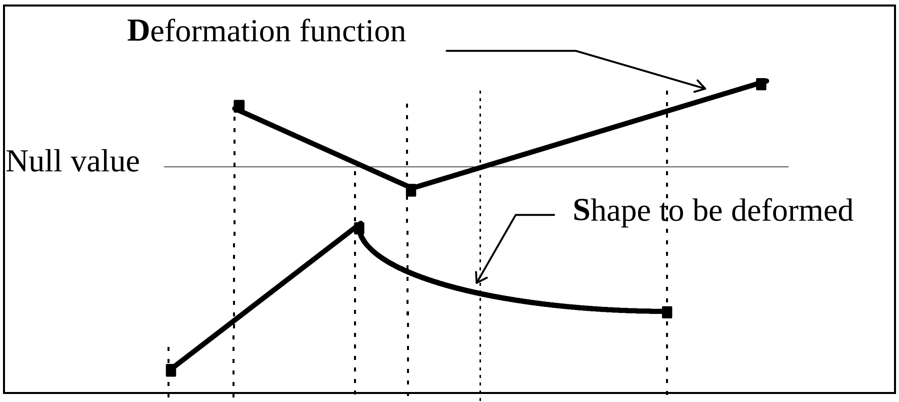

Fig.3: Data for shape deformation

Les Données de la déformation de forme

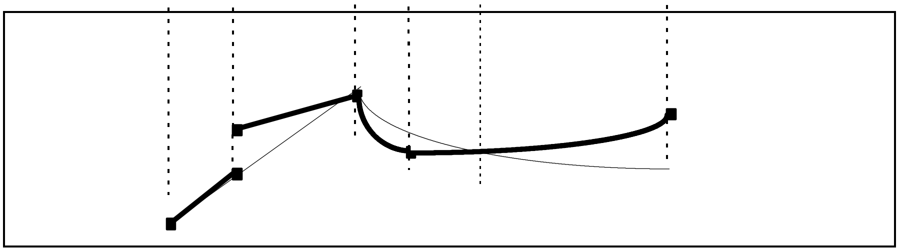

Fig.4: Expected result

Le résultat attendu

The mapping **D**eformation → **S**hape is given as input to an internal component. In general, it is a planar mapping, perpendicular to the view of modification. The resulting Shape **S**_result is:

S_result = **S**hape + **D**eformation(Mapping(**S**hape))

III.3.1. Deformation computation

The method to compute the **D**eformation function is an extension of the one presented in [MAS.96]. For each set of input, the **D**eformation function is a solution of a PDE minimizing an energy like criterion. Constraints and Handles constitute "boundary conditions". The **D**eformation is null outside the domain.

Since the exact solution is not a rational one, an approximation step of the modified part is necessary for transferring data to non CAS.CADE based systems. The approximation is based on NURBS and can keep the Gk continuity (in the current implementation k is up to 2) with its surrounding geometry.

III.3.2. Quality checks

Two modes of quality checks are implemented:

- those that have to be expressed by explicit functions, such as Gauss curvature map,
- those that have to be expressed by implicit functions i.e.: *as function(Shape)=0* such as planar sections, isophotes lines, shadow lines...

No complex problems to be solved, algorithms are tuned for real time feed-back.

III.4. Component software architecture

In what follows, we call *resource* a piece of code that implements a given component interface.

The global software architecture reflects (see below) what has been described above, i.e. a hierarchy of three components, each of them having specific purposes:

- FdF0 applies a deformation to a given shape, knowing two resources, the *mapper*, that specifies how the **D**eformation maps on the **S**hape,

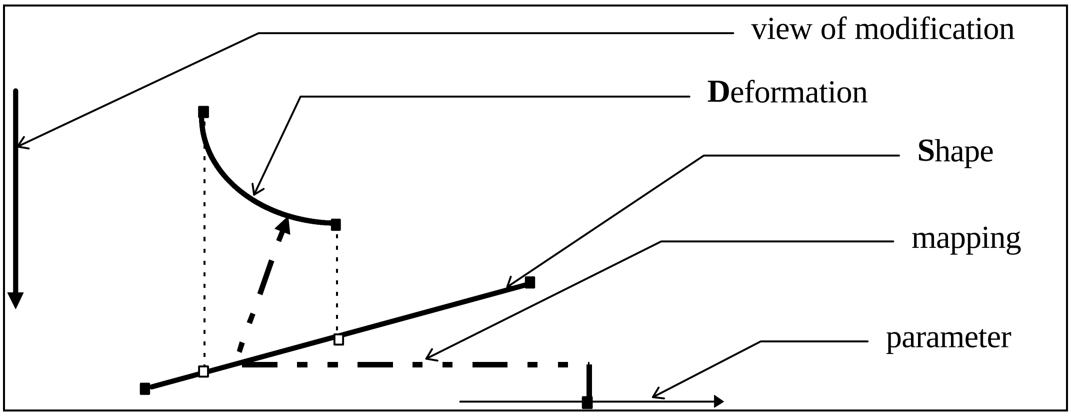

Fig.5: mapper

the *localizer*, that helps to determine which part of the **S**hape has to be modified in case of ambiguity,

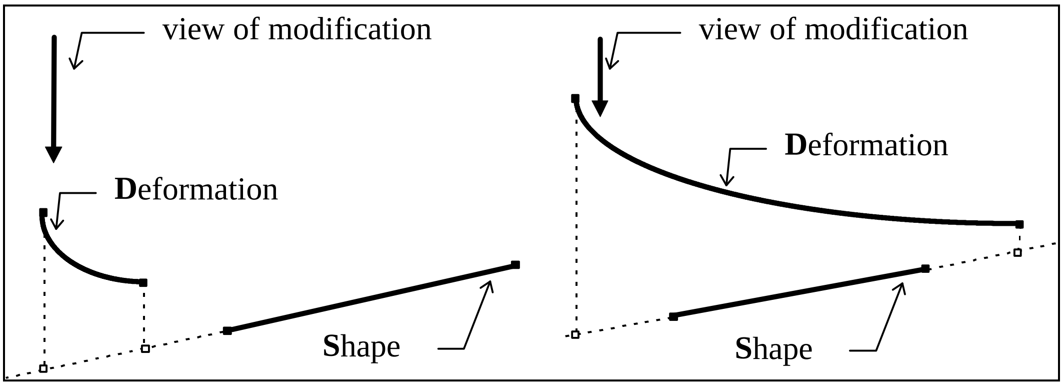

Fig.6: Ambiguity cases

Cas d'ambiguïté

and provides various types of representations of the modified **S**hape (i.e. exact, approximated by NURBS, or tessellated),

- FdF1 computes and provides the **D**eformation from the Domain, the two resources, the Constraints and Handles (which are special constraints cases). It uses a CAS.CADE component called "*Plate*" where the main mathematical part is done,
- FdF2 computes and provides the two resources from the **S**hape, the Domain and the View of Modification. It also computes and provides the real time checks (such as planar sections, isophotes, shadow lines, Gauss curvature maps) from real time FdF1 output. It uses an other CAS.CADE component called "*RealTimeCheck*" that, from some other resources, is able to compute any kind of real time information specified by its input resources (up to some extent) on a given shape.

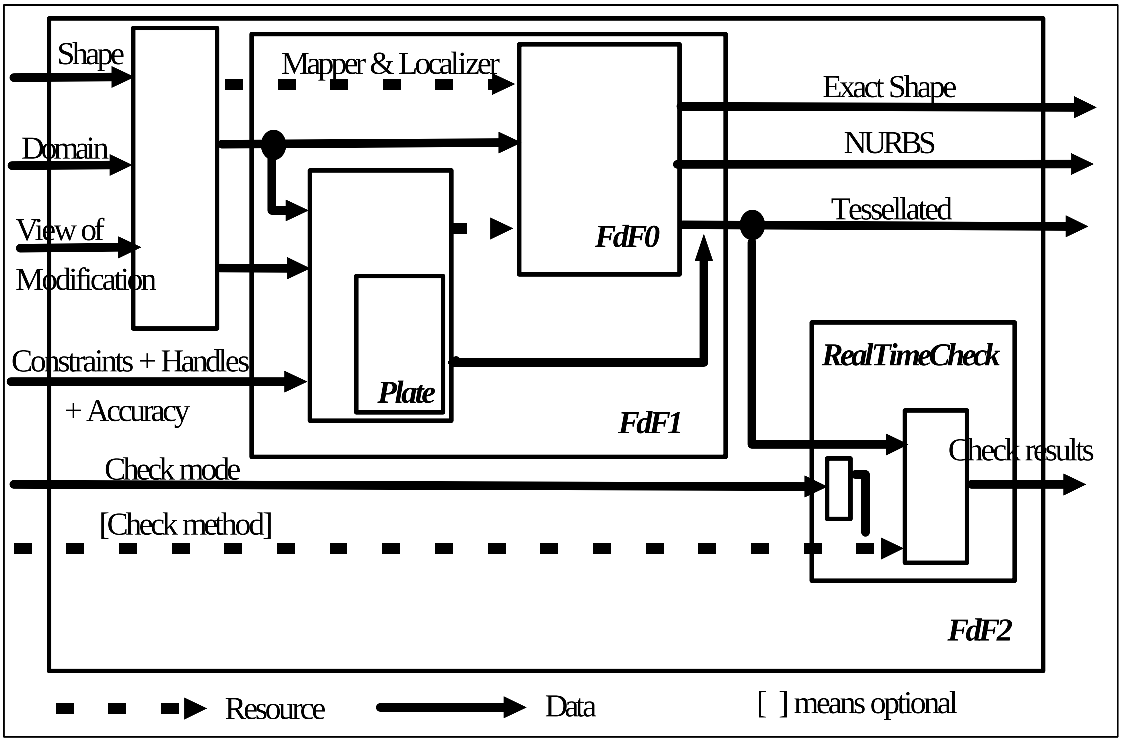

Fig.7: Component software architecture

Architecture logicielle du composant

III.5. Results

In all cases, real time checks can be displayed and modified.
Below, an example of a rim part global modification. The constraint is on the upper boundary (G0+G1+G2 fixed). The handle is the lower curve (translated).

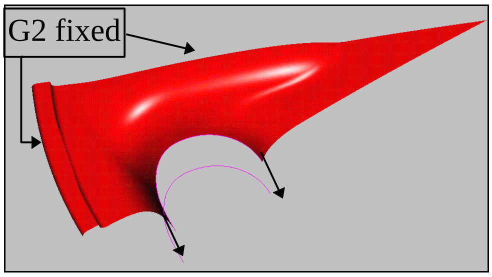

Fig.8: Rim part before deformation
Partie de jante avant déformation

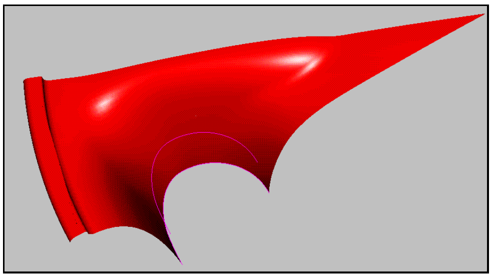

Fig.9: Rim part after deformation
Partie de jante après déformation

An other example of global deformation: the front hood of a sport car. Two cases:

1. Figure10 and the result in figure 12, handles are the front curve and the boundary of the air intake. The constraints on the back part and the wheel arch boundaries (G0+G1+G2 fixed).
2. Figure 11 and the result in figure 13, handle is the front curve. The constraints are on the back part and the wheel arch boundaries (G0+G1+G2 fixed) and on the boundary of the air intake (G0 fixed).

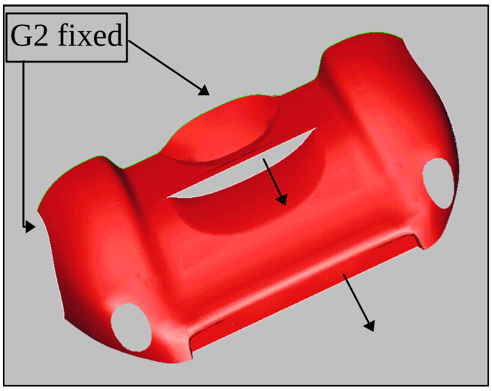

Fig.10: Car front hood, case 1
Capot avant cas 1

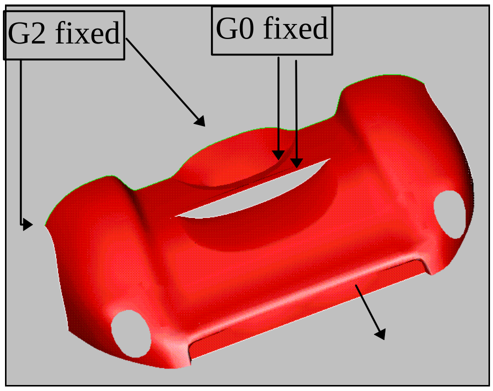

Fig.11: Car front hood, case 2
Capot avant cas 2

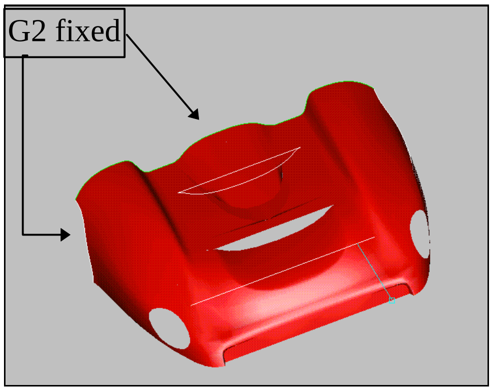

Fig.12: Car front hood, case 1
Capot avant cas 1

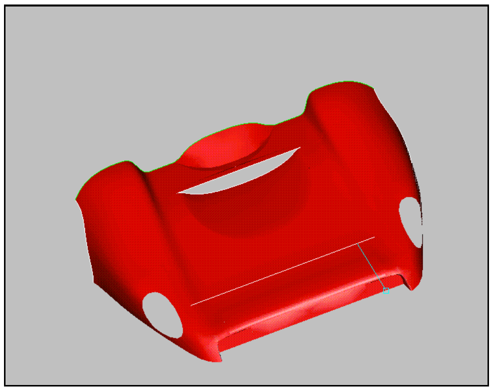

Fig.13: Car front hood, case 2
Capot avant cas 2

A last example is a "local" deformation: the front hood of a tourism car. The constraints are G0+G1+G2 along the boundary of the area to be deformed, excepted on its front part where a rip is expected. Real time isophotes are displayed.

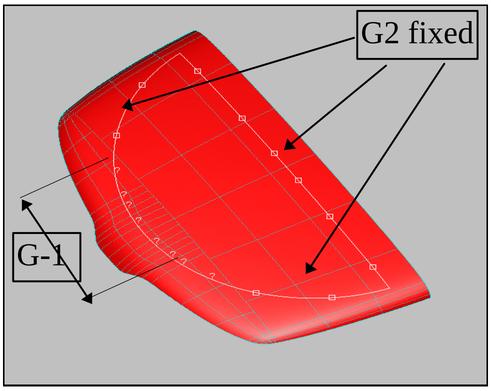

Fig.14: Car front hood, case 3
Capot avant cas 3

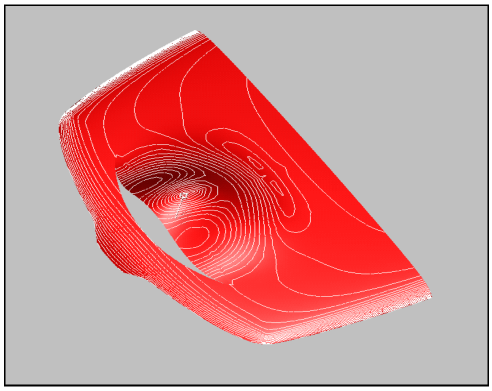

Fig.15: Car front hood, case 3
Capot avant cas 3

A live demonstration of the components , partially plugged in STYLER[5] (i.e. only some of the component possibilities are implemented) is preceding the conclusion.

IV. CONCLUSION

The resulting functionality is, by far, a step ahead from what is available today on the market. A significant decrease in the final style tuning time can be expected.

References

[MAS.96] G. DURAND, A. LIEUTIER, A. MASSABO. *"An aesthetic preserving algorithm for data exchange between accuracy incompatible modelers"*, IDMME'96 Proceedings, N°1, 1996, pp. 465-473.

[5] STYLER is a MATRA DATAVISION product: mainly used for surface design.