Acquiring, structuring shapes and annotating them with semantics to improve design activities

Jean-Claude LEON[1], Franca GIANNINI[2], Bianca FALCIDIENO[2],
Gino BRUNETTI[3], Holger GRAF[3], André STORK[3], Frédéric VEXO[4], Daniel THALMANN[4]

(1) : Laboratoire Sols, Solides, Structures
Domaine Universitaire, BP 51,
38041 GRENOBLE Cedex 9, France
+33 4 76 82 51 27/4 76 82 70 43
E-mail : Jean-Claude.Leon@inpg.fr

(2) : IMATI-CNR
via de Marini 6, Torre di Francia
16149 GENOVA, Italy
+39 010 6475667 / 010 6475660
E-mail : bianca.falcidieno@ge.imati.cnr.it

(3) : Fraunhofer IGD
Fraunhoferstrasse 5
D-64283 Darmstadt, Germany
+49 (61) 51 155 469/(61) 51 155 299
E-mail : andre.stork@igd.fhg.de

(4) : EPFL/IC/ISIM - VRlab
INJ 130 - Station 14
CH 1015 Lausanne, Switzerland
+41 (-21) -693-5216 / (-21) -693-5328
E-mail : Frederic.vexo@epfl.ch

Abstract: Shapes incorporate geometry as well as semantics and can be defined in 2D, 3D or higher dimensions. Geometric models can be structured in such a way that they can provide a relevant framework to insert semantic information or view dependent information about a product to form features. Combining such features with an immersive environment allows the user to interact more efficiently with an object. This framework will be illustrated in the context of free-form surface modelling for aesthetic design and in the context of engineering and simulation, both among the research themes of the network of excellence AIM@SHAPE. Virtual environments can also contribute to the efficiency of the design process through appropriate models of interaction between the objects contributing to a given simulation environment (a maintenance task, a surface deformation process for aesthetic design for example). To this end, virtual humans bring new capabilities to enhance simulations though they require new models of interaction to incorporate the appropriate semantics during an object manipulation. This third aspect is also part of the ongoing research actions of AIM@SHAPE and will be addressed to show how it is complementary to the previous one. As a result, this set of contributions demonstrates how various categories of semantic can enhance the design process within virtual environments.

Key words: shapes, semantics, aesthetics, simulation, design process, annotations, virtual humans.

1- Introduction

During the product development process, the geometry or the shape of the product evolves in accordance with the simulations [1][2][3][4] and other various assessments of the product that can be performed. These shape evolutions depend also on the view considered about the product and the corresponding skills aggregated to form a given view. During these phases, the product is tuned to its requirements and various mechanical and technical information can be attached to components, e.g. for its manufacture or its structural behaviour modelling, to enrich the semantic of the corresponding model. Thus, the corresponding components are annotated with semantic information to tightly associate their geometry to technical and mechanical data to improve the design process.
Further shape changes are required when product components or subsystems progress into the product development process and move from one view to another [5]. When the product development process evolves and the amount of information grows, the corresponding complexity is handled through appropriate information structures such as Product Data Management systems to structure its shape in accordance to product versions and configurations. There, technical information is the mean to bind together its individual components, which form a set of disconnected geometric partitions that cannot efficiently characterize the relationships among the components from the geometric point of view.
Though such a structuring reflects the needs when shapes are made up from a collection of disconnected geometric components, shape structuring also takes place when the shape of a component needs to be processed for adaptation from one view to another, or inside a view, to meet user's requirements. As an example of such shape structuring, processing a component for Finite Elements Analysis often requires idealization processes to transform subsets of the object shape from volumes to surfaces (shells) or lines (beams) [5][6].

Interacting with these various shapes along the product development process to generate or modify them is a well-known problem and it is particularly emphasized when the shapes addressed are complex in the sense that they are not formed with analytic surfaces, i.e. the user does not know precisely the nature of the surface she or he wants to specify. Immersive techniques are an efficient and well established mean to ease this interaction process and improve the efficiency of design phases.

During the product development process, product behaviour simulations or product assessments can require a model of a subset of the product environment. As an example, maintenance simulations and ergonomic studies require a model of virtual human to take into account some of the relationships between the product and the end-users or the product and operators taking part to the manufacturing process. Therefore, virtual human modelling and interaction modelling between virtual humans and a product, are of increasing interest to be able to incorporate efficiently these new simulations into the product development process. There, semantic information needed to model a subset of the environment and its interaction with the product requires also annotation and structuring principles.

Based on free-form shape modelling subjected to aesthetic constraints, immersive approaches to generate and manipulate the corresponding shapes as well as some annotation and structuring principles will be discussed. Illustrations of some of the problems raised by virtual human modelling as well as their interaction with the product are also incorporated in the present paper to highlight the new challenges brought by these new possibilities of product assessment and product behaviour simulation.

To this end, section 2 introduces the concept of semantic annotation and structuring applied to free-form shapes subjected to aesthetic constraints and their complementarities to virtual styling, Section 3 concentrates on virtual engineering and section 4 on virtual human modelling, interaction modelling between virtual humans and the product.

2 - Virtual Styling

Despite the innumerable methods and techniques research has produced to support product conceptualization, it still can be observed that industry mostly follows traditional methodologies [7]. The origin of the problem is that academia tends to introduce abstract models and highly specialized tools giving preference to automated solutions, rather than addressing highly interactive means, like that provided by virtual environments.

For computer-aided styling Catalano et al. [8] confirm that designers still like drawing in the traditional way and see the computer in the conceptualization phase as an obstacle to their creativity, also because of the lack of integration between sketching and corresponding 3D model. There is a need for an intuitive modelling process where interaction should be performed through a direct control over the 3D space in the same way a pencil dominates the 2D space, simulating the traditional method of stylists' work [9]. To improve the usability, current styling systems have to overcome the limitations imposed by the low level geometric manipulation they provide. Two complementary categories of approaches to meet this requirement are discussed.

The first ones are free-form feature modelling techniques to provide an abstraction from a pure low-level geometry manipulation to a semantics-based modelling of design intent, just as it is the case with regular form-features widespread in Computer-Aided Design (CAD).

The second one is the application of virtual reality techniques to provide an intuitive immersive modelling environment suitable for interaction techniques that are based on existing skills of stylists.

2.1 - Free-form features for semantic annotation

To create semantically enriched shapes during the design phase, it is important to provide tools and methods that, from one hand, efficiently support designers to obtain rapidly the desired form in an intuitive way and, from the other hand, contribute to a valid support for semantic annotations. Form features have been widely accepted in the mechanical engineering domain as key entities associating semantics to geometry. They allow to treat sets of geometric entities as a unique one. Such grouping is guided by common functionality or other important meaning, e.g. subjected to specific operations. Features can produce a structured organisation of the geometric model which is suitable for associating semantic interpretation and tagging. This is valid both for product design but also for all the other contexts in which the introduction of semantic data to the models can help intelligent decision making and automatic operations. Such an association, which is context dependent in the sense that a given shape can produce different actions according to the aim we are looking at it, is only possible if a formal taxonomy of features is provided together with the tools for their insertion in the model. Such an insertion that could be performed a priori, during the object creation phase according to a design by feature approach, or a posteriori through a recognition process. In our network, we are dealing with both approaches.

The feature taxonomy considered is an extension to the free-form domain of the well known mechanical ones [10][11]. In the free form domain, features may be broadly distinguished between those covering the overall structure of the object, those changing its genus and those affecting its skin. Specific sections and profiles belong to the first category; whereas through holes and handles modify the genus; finally the third category involves what we indicate as deformation features. They are characterised by the type of change on the shape, e.g. depression or protrusion, enforcing the visual aspect of a given perceived curve, resembling planar areas, etc.

In our approach, the definition of deformation features is based on a parametrical deformation tool which enables high level constrained manipulations of the surfaces [12]. Constraints monitoring the deformation process always include at least the specification of the extension of the feature over the surface, and elements, which could be portions of surfaces, curves and points, that constrain the surface resulting from deformation feature to pass through them. The deformation area is bounded by what we call a limiting line, possibly composed of several curves that bounds a closed area of the surface where the feature has to be inserted. The other elements are called respectively target surface, target line and target point. The deformation method is based on an extension of the Force Density Method (FDM) applied to a set of rigid bar networks coupled with the control vertices of the geometries to be deformed [13], i.e. 3D curves, surfaces, curves in the parametric space of a surface. It can be applied on traditional CAD surfaces, i.e. B-Spline/NURBS as well as on tessellated models. Thus, it is suitable both for traditional modelling approaches where NURBS parametric are preferred and for immersive virtual environments. This deformation process includes also the capability of mixing both types of representations and gives the possibility of importing existing models generated in different modelling environments. In fact, the bar networks can either be a coupling with the control polyhedron of B-Spline/NURBS surfaces/curves, or to the nodes of meshes/polylines [14][15]. In these networks, each bar is seen as a spring with a null initial length and a given stiffness. To preserve the static equilibrium state of the bars, external forces proportional to the bar length and the associated stiffness are applied at the endpoints of each bar. The specification of the target elements provide parametric and geometric point constraints which specify new positions of some of the network elements. In fact, all the various constraints to be processed are translated in point constraints. Thus, the problem is reduced to the definition of the new set of external forces on the bar network needed to deform it according to the given points constraints. In order to select one solution among all the possible ones, an objective function is added to set of geometric constraints and a criterion reflecting this objective function has to be chosen, such as the minimisation of the variation of the external forces or the minimisation of the bar length variations. Using the geometric coupling, the new positions of control polyhedron vertices are obtained through the new positions of the bar network nodes, thus inducing the surface deformation (see Fig. 1).

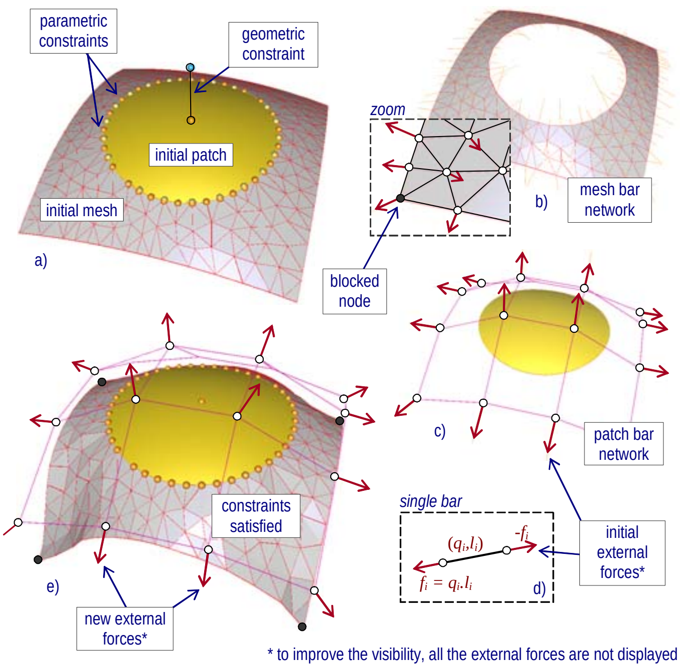


Figure 1: The surface deformation process, parametric constraints prescribe continuity conditions across patches, whereas geometric constraints correspond to target positions in 3D space to be reached by the deformed surface.

The objective function represents an effective shape behaviour and tuning parameter. Choosing a specific function expressing a specific surface behaviour is in fact possible to obtain more or less stretched or round shapes, possibly resembling the original one to which the deformation is applied. It is even possible to apply different functions over sub domains of the deformation area, thus prescribing different behaviours whose variation can be continuously monitored [16], see Figure 2.

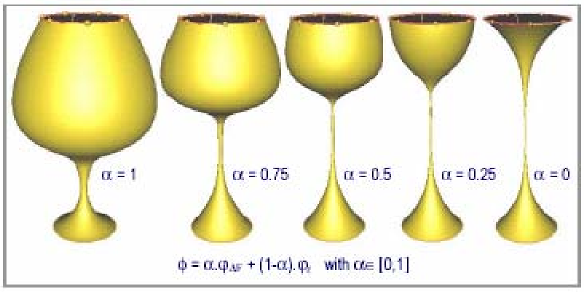


Figure 2: Example of continuous shape variation between two extreme shapes (α=1 and α=0) associated to two different minimisation functions.

The developed tools aim also at reducing as much as possible the insertion of new trimmed surfaces, since it is difficult, time consuming and not intuitive to handle them when sequential modifications occur, which is very common at the early design phases when not all the choices are made. In addition, the use of VR environments is another justification for setting more intuitive shape modification approaches. Moreover, avoiding trimming operations is a way to maintain the original surface semantics. Therefore, methods for inserting tangent plane discontinuities, i.e. sharp edges, have been implemented (see Fig. 3) as well as methods for inserting planar-like areas to express functional specifications over a free-form object (see Fig. 4). For further details on this see [17] and [18].

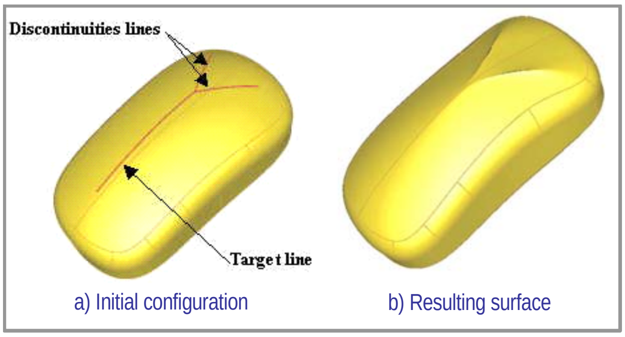


Figure 3: Example of tangent plane discontinuity insertion performed in a single deformation step.

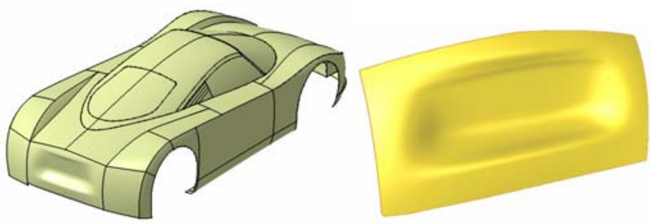


Figure 4: Insertion of an free-form feature containing a planar area defining the location of number plate (courtesy of S.t.S.).

This section demonstrated how semantic annotations, through free-form feature generation and manipulation, can be performed during the design process to handle aesthetic constraints. Such an approach can also fit into the context of virtual styling to provide high level interaction functions using immersive VR environments as described in the following section.

2.2 - Virtual Styling for shape acquisition

Current computer-aided styling systems are in the view of stylists and industrial designers still more an obstacle to creativity than a support in the conceptualization of new designs [7][8]. Our virtual modelling environment in turn proposes a human-machine interface (HMI) that builds on the well trained skills of designers for sketching free-form curves and surfaces [9].

For styling, the output device should provide high quality rendering and immersion. We use a semi-immersive virtual table with a diagonal of 1.7 meters, which allows creating e.g. parts of a car body in scale by appropriate hand gestures and arm movements. The user wears a pair of tracked glasses to render the scene according to her/his view point (see Fig. 5). For head tracking and wireless input devices, optical (infrared) tracking is applied, as it provides the required high accuracy and low latency. Various tracked objects are used as input devices. Besides the glasses, a wireless pen (3 buttons) is used as main input device. The PIP (Personal Interaction Panel) is a transparent plexiglas panel on which the application menu is projected. Another important input device is the navigator axis. It allows the user to literarily hold the virtual object in her or his hand to navigate it with 3 degrees of freedom.

Figure 5: Set-up for Virtual Styling.

As it is the case when sketching shapes on paper, the virtual styling process is also based on sketching contour lines in a virtual space. For this, we developed several stroke input methods (e.g. oversketching and tape drawing) allowing stylists to apply well trained sketching techniques.

The approach to stroke input processing is straightforward. While moving the pen device in space the tracker continuously delivers position events. These positions are used as sample points for approximating a NURBS curve.

When using a pen as an input device it is appropriate to simply redraw parts of curves to refine them. This technique is called oversketching (see Fig. 6), as a sketched curve is usually defined by a bundle of oversketched strokes. To apply oversketching in our immersive modelling environment it is possible to define arbitrary cutting planes that are projected on the tabletop of the virtual table and to then draw on the tabletop like on a sketching board.

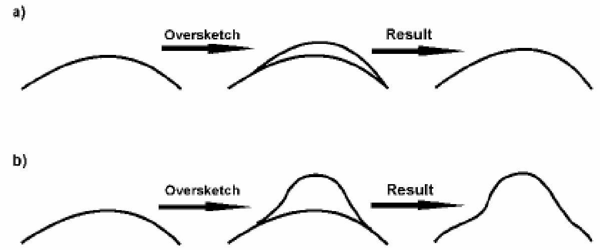


Figure 6: Oversketching with scaling factor 0.5 (a) and 1.0 (b).

Oversketching uses discrete curve points [19]. After selecting the curve $C_d=(x_0,\ldots,x_n)$ to be modified, the user sketches a new curve $C_o=(y_0,\ldots,y_m)$ near the part of the destination curve she/he wants to change. The algorithm then creates the resulting curve $C_r=(r_0,\ldots,r_o)$ by substituting the oversketched part of C_d with C_o. The segment (x_s to x_e) to be substituted with C_o is determined by finding the minimum distance between x_j and y_0 and y_m (j=0,..,n). The resulting modified curve that is then approximated by a NURBS containing the points $C_r=(x_0,\ldots,x_{s-1},y_0,\ldots,y_m,x_{e+1},\ldots,x_n)$. To further control the effect of oversketching the user can define a scaling factor that weights the influence of C_o on C_d (see Fig. 6).

Another one-stroke curve modelling method that has been adapted from a traditional styling method is 3D tape drawing. Tape drawing is a wide-spread modelling technique in industrial design that is used to define characteristic lines. Instead of using pencils, a special adhesive tape is used, which the stylist unrolls with one hand (typically the right hand), sliding over the tape with the other hand applying gentle pressure to fix it on the drawing. In this way smooth 2-dimensional curves are designed, where the first hand defines the tangent line with respect to the curve point currently fixed by the other hand as illustrated in Fig. 7.

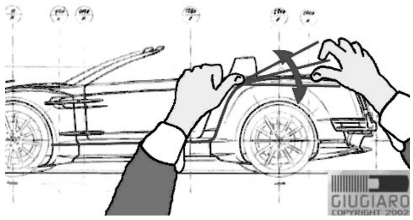


Figure 7: Traditional 2D tape drawing.

The 3D devices applied for virtual tape drawing are shown in Fig. 5. Those devices are a "tape finger" device (left hand), which is tracked for the position of the hand and finger, respectively, defining the curve points, and a pen device (right hand) corresponding to the roll of tape. When starting the virtual tape drawing, the method is initialized with the current position of the finger P_{f1}, which also defines the starting point P_1 of the curve, and the current position of the pen P_{p1}.

The iterative definition of the sample points P_i of the curve are determined as follows: Given the last point P_i, two vectors are defined, the relative displacement of the finger $\overrightarrow{P_iP_{fi+1}}$ and the vector defined by $\overrightarrow{P_iP_{pi+1}}$. Now the current finger position is projected onto the trajectory of $\overrightarrow{P_iP_{pi+1}}$ (Fig. 8).

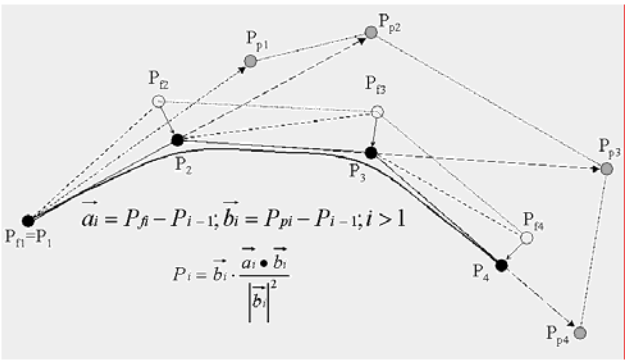


Figure 8: Calculation of the taped 3D curve with points P_i derived from the current and last finger and pen positions.

Any subsequent point is defined such that the vector between the new P_{i+1} and the pen position defines a tangent line on this point P_{i+1}, as it is the case in the traditional 2D tape drawing. Finger displacements in depth are corrected avoiding unwanted oscillations in that direction. The orientation, which in contrast to 2D tape drawing is now a 3D vector, is defined by the leading pen (taking the place of the tape roll) exactly as it is the case of the original tape drawing process. Hence, it is a mean of adding degrees of freedom to tape drawing, benefiting from the already well developed skills of stylists.
Once the contour lines of a model have been sketched they are used to create surfaces. The most used techniques applied by stylists in user tests have been to interpolate a set of curves with a skinning function and the stroke-based extrusion of a selected curve. Both operations work on the NURBS representation of the curves. The resulting NURBS surface is tessellated for visualization. An example styling session is shown in Fig. 9.

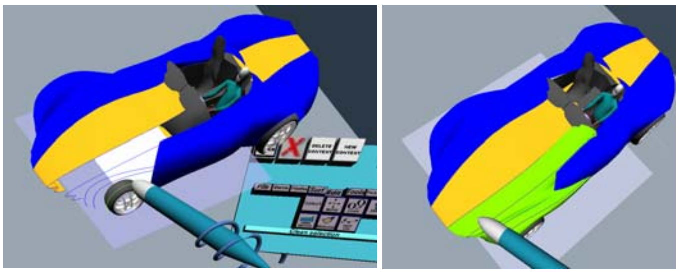


Figure 9: Virtual styling session example: the contour lines are selected to interpolate them with a skinning operation.

It should be noted that direct methods we implemented to create free-form surfaces (e.g., coons patches) have not been accepted in user tests with stylists, due to the difficulty to control the 6 degrees of freedom with free-hand surface sketching.

2.2- Complementarities of the free-form features and the virtual styling concept

The previous sections highlight the complementarities between shape annotation and structuring through the feature concept applied to free-form objects and the acquisition (i.e. creation) capabilities brought by virtual styling approaches.
The concept of fully free form features acts as a mean to structure a shape and capture some aspects of the designer's intent since the deformation operations reduce the number of elementary operations performed by the designer to better match her or his intent. Similarly, the object shaping is performed without changing the decomposition of the surface into patches unless it is meaningful from the shape point of view, i.e. insertion of sharp edges. Therefore, the parameters defining a fully free-form feature better reflect the semantics of the shaping process than the geometric constraints set by the current computer-aided styling systems, which are still extensively based on surface decomposition techniques. Thus, these systems are not able to incorporate the designer's intent during shaping operations.
Using the concept of fully free form features, the resulting annotations of shapes still needs to be acquired from the designer's input. Indeed, VR environments efficiently contribute through metaphors to establish appropriate links between the VR devices and the feature data structures to preserve as much as possible the semantics of the user's input. Globally, the complementarities of fully free-form features and immersive environments demonstrate the design improvement they can bring.

3 - Virtual Engineering

Further illustration of the above aspects can be highlighted for another view of the product development process related to simulation.
Current research within integrated VR simulation environments focuses on a seamless integration of data and information sources for post-processing and visualization, rather than addressing the chain from computer-aided design to engineering (CAD-CAE) as a whole. The aim of CAD/CAE/VR integration in turn is to realize an open platform providing an integrated front-end to engineers facing massive numerical analysis problems, allowing seamless interactions with traditional engineering applications in immersive environments.
The expected benefit from performing simulation pre- and post-processing steps within immersive virtual environments is, for instance, to support the intuitiveness of engineers during the optimization process. Optimization today is still far from being automated and relies on the experience and intuition of the engineer. As intuition here is related to estimate the validity of a simulation model and the correctness of a simulation result with respect to the real object, an interaction with a virtual object being closer to its physical counterpart provides some obvious advantages. As in the case of modelling, also for simulation proposes VR allows the development of more natural 3D interaction techniques to also increase the intuitiveness of the problem treatment in terms of model shape and simulation mesh, for instance. Another argument is that virtual environments provide an improved medium for interpretation of simulation results, reducing the number of miss-interpretations typical for presentations in form of numerical values or a series of 2D plots.
Simulation tasks are characterized by a series of shape modification and acquisitions of different views of a object. Semantic annotation of the different views is a basis for realizing the addressed integration. On the one side they are a mean to provide a more abstract and intuitive interaction with the geometry of the different views on the object. On the other side semantic annotation is a vehicle to keep track of the interrelationships between the shape features of the different views.

3.1 - Small feature identification

Small feature identification deals with the suppression of design details that are too small to affect the analysis results, but impose severe constraints, for instance, on a finite element meshing procedure. Since mesh generation algorithms are guided by the boundary representation of a model, the mesh connectivity has to follow the face and edge boundaries in the model. Hence, if the model local feature size is small compared to the desired mesh element size, it adversely effects the mesh, resulting in reduced mesh quality and larger number of mesh elements than necessary.
An example of detail suppression is shown in Fig. 10, where a fillet narrower than the desired mesh size is removed. Other functionality which has been realised is the semi-automatic detection of detail features such as narrow regions, sliver edges, complex connect operations, removal of through holes.

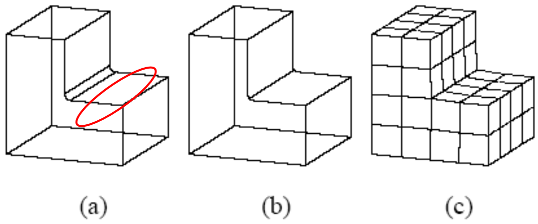


Figure 10: Semi-automatic face collapsing of an L-shape.

3.2 - Interactive mesh refinement and adaptive solving

Leading an analysis task can be quite cost intensive. Doing simulation refinements during an analysis to get more precise results is one of the main tasks during the analysis stage. However, the engineers are not interested to lead a re-analysis of the overall domain containing the refined areas which result in long simulation runs if models get large (e.g. >500.000 elements). She or he might be much more interested in a refined analysis on a sub-domain where critical behaviour of the design is expected. Therefore also interactive refinement strategies have been realized in which the engineer can fully make use of 3D interactive mesh refinements (Fig. 11).
A direct coupling to the simulation engine, which is responsible for the recalculation of the results, offers an adaptive solving step in which the area of interest is directly re-calculated after the refinements have been done. Other functionalities that have been realized comprise a coupling of sub-domain derivation and a sub-model analysis.

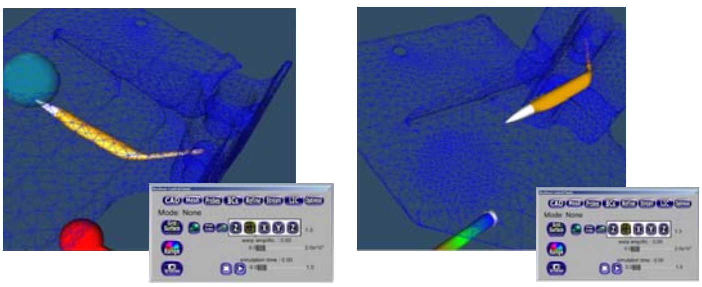


Figure 11: 3D interactive mesh refinement by selection of the area of interest in 3D.

3.3 - Integrated optimization environment

As an extension to the described integration of the virtual engineering chain, work has been done to integrate parametric optimization modules into the system. Extended by these modules our working environment features embedded decision making tools to support the interactive consideration of variants as well as a systematic, simulation-based optimization by specifying the target criteria (reverse engineering).
The decision making process usually is a trade off between a set of different target criteria. The optimal set of criteria is usually not objectively measurable and the optimal result in a given situation is subjective to the decision maker. One of the key components is here usually the engineer’s experience and intuition that can be supported through advanced and realistic graphical representations.
As an example application that uses an optimization module we present the optimization of car bodies through virtual crash experiments. The functional characteristics of automobiles are in general computed using a set of different simulations. For instance, to identify the crash characteristics of a car body the front wall intrusion may be simulated as well as the width of the doors after the crash, and the effects on passengers inside and outside the car. The car body is optimized with respect to these parameters.
We have successfully integrated the forward engineering cycle for a small virtual crash experiment into our environment. The engineer’s aim in this example is to find a good compromise for different material parameters. The engineer is optimally supported during the decision making process as the application runs interactively. Testing several combinations of parameters is just a matter of moving a set of sliders - the complex interplay of tools in the background is completely hidden to the user. While the user interactively changes the parameters for the material thickness the resulting animation adapts to the new simulation results. The animation finally is presented to the user in a immersive view (Fig. 12).

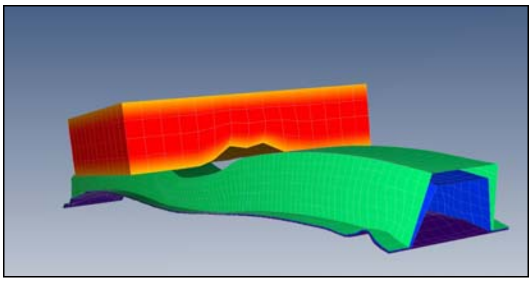


Figure 12: 3D Animation of box beam intrusion during an optimization session.

The presented example only works for the mere academic box beam example. But, together with partners from the computational simulation domain, new modules are developed that will support the interactive consideration of variants as well as the reverse engineering of more complex models. The above examples however illustrate through engineering simulations how parameters related to shapes (here shapes can be understood as finite element meshes) can be acquired, updated to interact efficiently with shapes and their underlying semantics like the mechanical behavior of a structure in the present case.

4 - Simulation of manipulation of virtual objects by virtual humans

The simulation of the manipulation of virtual objects by virtual humans is increasing due to the industrial request of the development of virtual prototyping. There are many potential applications, such as industry processes planning and optimisation, virtual prototyping, design reviews, assistance to assembly and commissioning, assistance to production planning, assistance to maintenance and inspection planning, assistance to revamp planning, quality control, training, training for maintenance and inspection, documentation of procedures, marketing, concept presentation, mock-up design alternatives, and product configuration catalogues.
The above list shows the wide variety of product development stages or product life-cycle stages where virtual environments of the product are required to achieve appropriate simulations. Among the components of such environments, virtual humans form complex data structures where semantics takes an important role to annotate and structure shapes, i.e. virtual humans in this case.
In this area we could mention the EU-STAR project which is an European project covering various aspects of augmented reality applied to enhance animations concerning virtual actors manipulating objects for the purpose of training operability of industrial equipments (see Fig. 13). For that we need to develop realist virtual world where the simulation could be done.

Figure 13: Virtual Factory worker interacting with a virtual machine.

Moreover the virtual environment should be populated by virtual humans that will simulate the interaction with the objects which are composing the scene. Nevertheless, the creation of this virtual prototype is highly expensive. We will show in the next sub-section how the use of semantic information could help the creation of virtual worlds.

4.1 - Semantic modelling of Virtual Environments

The creation of Virtual Environments is a challenging problem requiring diverse areas of expertise, which may range from networks to psychology. Developing VE systems is a very expensive task in terms of time, financial and human resources. This limits the creation of novel simulations and makes it difficult to improve the existing ones. Oliveira and al. [20] describe some of the main problems faced when developing interactive virtual environments: the non-extensibility, limited interoperability, poor scalability, monolithic architecture, etc.

Research in this area usually focuses on defining development frameworks with reusable and pluggable components. E.g. Alexandre [21] proposes a framework to build dynamically extensible networked environments and puts special emphasis on reusable components. Similar research includes the Java Adaptive Dynamic Environment (JADE) [20] and the NPSNET-V framework [22]. Both of them specify the notion of a plug-in architecture [23] made of dynamically loadable modules organized into a hierarchy of module containers.
Therefore, all this systems or framework are still centred on Geometric Scene graph which is the traditional data structure for the Virtual Environments (VEs). This is due to the fact that originally, the developed technologies around the VEs were related to the graphic rendering which need of course of 3D representation of the objects forming the VEs.
Nevertheless, virtual entities such as virtual humans, or any object "living" in a VE have various kinds of associated knowledge. For instance, when we consider a virtual human at a higher level, we don't see it just as a deformable 3D mesh whose animation is controlled by an internal hierarchy of joints and segments (animation skeleton). Instead, we consider it as a "living" creature with some degree of autonomy and a particular role to play in the environment. Virtual characters can modify their behaviour depending on the experience acquired through contact with human users and virtual entities.

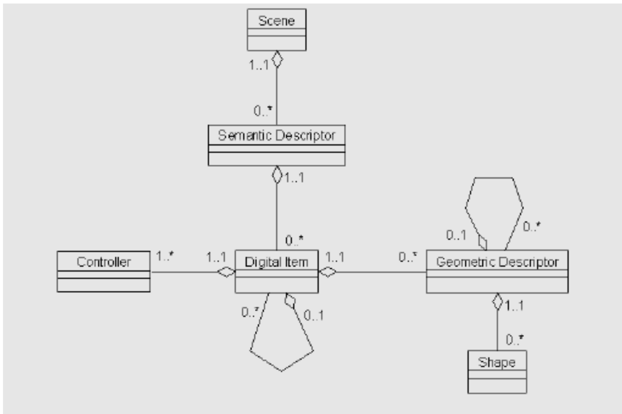


Figure 14: Semantic Virtual Environment Structure.

Our main hypothesis is that a higher-level semantic representation of VEs can enhance the reusability and adaptation capabilities of the virtual entities participating in a VE. The semantics based representation (Fig. 14) we propose builds upon the concept of scene graph, with the difference that it does not focus on visual or spatial relationships between entities, but on higher level semantics. Our approach is based on the semantics of the virtual entities and not on their geometry, as is the case in most of the VE applications. The semantic model we propose provides a way to specify alternative geometric representations for each entity and a range of functionalities. This is the same difference that could be foreseen in CAD between functionality and Shape. Given that, our work is this paper is focused on the simulation of manipulation of virtual objects by virtual humans. In the next paragraph will present how it is possible to use semantic information to model virtual humans.

5.2 - Semantics for virtual humans modelling

Virtual Humans, as graphical representations of human beings have a large variety of applications. Within inhabited VEs, Virtual Humans (VHs) are a key technology that can provide virtual presenters, virtual guides, virtual actors, and be used to show how humans behave in various situations.

Our main contribution, in this paper, focuses on proposing a semantics-based method for organizing the various types of data that constitute a Virtual Human. The knowledge related to the synthesis, animation and functionalities of VHs is formally specified in the form of ontology. The following schema presents the Ontology for the Virtual Humans (Fig. 15).

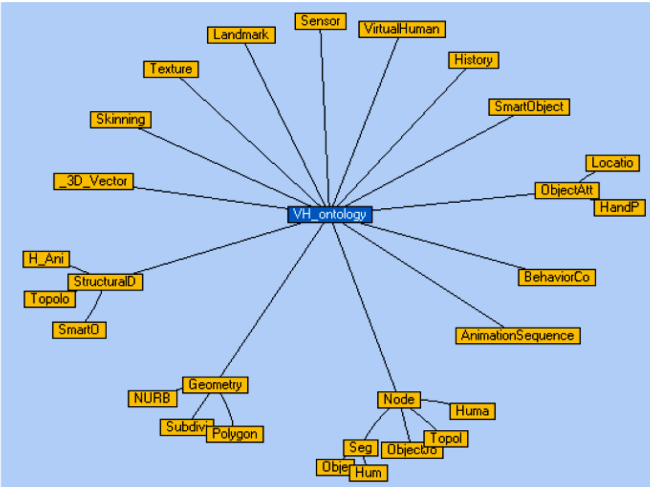
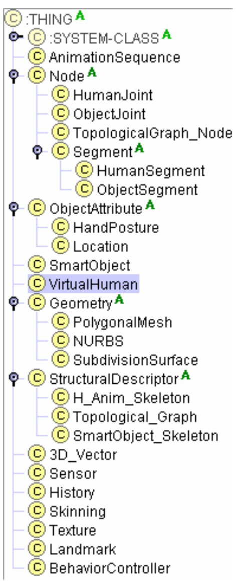


Figure 15: Main components of an Ontology for Virtual Humans.

The Ontology for Virtual Humans aims at organizing the knowledge and data of three main research topics and applications involving graphical representations of humans:

- Human body modelling and analysis: morphological analysis, measuring similarity, model editing/reconstruction,
- Animation of virtual humans: autonomous or pre-set animation of VH,
- Interaction of virtual humans with virtual objects: virtual -smart- objects that contain the semantic information indicating how interactions between virtual humans and objects are to be carried out.



Ontology development is a continuous process and as such the results we present here are work in progress. Nevertheless, the current ontology provides a good starting point towards the creation of a more versatile and reusable representation of Virtual Humans. An ontology representation of a VH must be closely linked to the associated graphical one. It is particularly required to be able to go from the graphical representation to the ontology -semantic- one: with the analysis of the 3D graphical representation in order to query the 3D models for semantic information. It is also required to be able to go from the ontology description to the graphical representation: with the integration of the semantic descriptors in the modelling and animation process, which means that we need to construct the graphical representation of a virtual human from the semantic descriptors.

Fig. 15 presents a simplified diagram of the main components of the ontology. The main classes define the geometry of the VH, which can be represented as a polygonal mesh, NURBS, etc. The Structural Descriptor class (abbreviated as StructuralD in the simplified diagram) allows for deriving a variety of descriptors such as: nodes for topological graphs, animation skeletons (H-ANIM compliant, standardized hierarchical structure for humanoid animation), or animation skeleton for smart objects (objects which can be manipulated by a VH). The ontology considers that a VH can have associated information about its -modelling- history, landmarks, sensors used for behavioural animation algorithms, animation sequences (e.g. keyframes), smart objects and other accessories.

In the next sub section, we will present how we add semantic information of manipulated object to create Smart Objects.

5.3 - Semantic Smart Object: manipulation of virtual objects

As we have shown in the previous subsections there, a need for model interactions between an object and a virtual human appears in most applications of computer animation and simulation.

Such applications encompass several domains, as for example: virtual autonomous agents living and working in virtual environments, human factors analysis, training, education, virtual prototyping, and simulation-based design. Commonly, simulation systems perform agent-object interactions for specific tasks. Such an approach is simple and direct, but most of the time, the core of the system needs to be updated whenever one needs to consider another class of objects.

Smart Objects are an interesting way to model general agent-object interactions based on objects containing interaction information of various kinds: intrinsic object properties, information on how to interact with it, object behaviours, and also expected agent behaviours. The smart object approach, introduced by Kallmann and Thalmann [24] extends the idea of having a database of interaction information. For each object modelled, we include the functionality of its moving parts and detailed commands describing each desired interaction, by means of a dedicated script language. A feature modelling approach [10] is used to include all desired information in objects.

In essence, smart objects provide not only the geometric information necessary for drawing them on the screen, but also semantic information useful for manipulation purposes. We have built a framework for real-time animation of virtual human -- object manipulation sequences. This framework provides smart objects capabilities and is composed of the following components:

- A design tool that incorporates the definition of semantic information in the process of object design,
- An XML-based specification for virtual objects, including appearance, animation and interaction aspects,
- An extended scene-graph structure that enables storage and query of semantic information at run-time,

- An event-based mechanism for controlling and coordinating animation of objects and virtual humans,
- Scripting functionality for higher-level integration of all of the animation components.

Figure 16: Virtual human interacting with smart objects in a kitchen..

Attributes are the primary means of specifying information on how a virtual human manipulates its environment. They convey various kinds of information – e.g. important places on or around the object (e.g. where and how to position the hands of the virtual character in order to grasp it), animation sequences (e.g. a door opening) and general, non-geometric information associated with the object (e.g. weight or material properties). The semantic information in the smart object is used by the virtual characters to perform actions on/with the object, e.g. grasping, moving it, and operating it (e.g. a machine or an elevator). Fig. 16 shows an example within a kitchen scenario.
The animation of virtual humans is handled by “actions”. Actions provide a higher level view of animation tasks. E.g., the look action requires a vector as a parameter and keeps the virtual human looking at this position while it is active. For a human, multiple actions can be in execution simultaneously, and the resulting animations are mixed in real-time.
Among the more complicated actions we have available are walk and reach. Just like the look action, the walk action takes a vector as a parameter, which is used as the target of the walk. The reach action takes a hand posture and a matrix as parameters. The hand of the virtual human is brought to the position and orientation specified by the matrix by using inverse kinematics. Once the hand is at the target, it assumes the given posture to simulate grasping.
Scripts and events are used for flexible high-level control and coordination of animation elements. Consider the example in Figure 9 from a training application, where a virtual human needs to manipulate a machine. In this particular case, the action to be performed is turning a wheel for adjustment. The sequence of movements that the human should make and the changes in the state of the machinery in response is described by the following script:

```
Human.WalkTo(Wheel.FrontPosition)
WaitUntilEvent(WalkReached(Human))
Human.Reach(Wheel.LeftHand)
Wheel.StartAnim(Wheel.Turn)
Repeat
  Event = WaitAndReceiveEvent()
  If
event.Is(AttributeChanged(Wheel.LeftHand)
    Human.NewReachTarget(Wheel.LeftHand)
Until Event.Is(AnimFinished(Wheel.Turn)
Wheel.Turned = True
```

5.4 – Virtual objects and virtual humans as contributors to product design

The previous subsections have highlighted how semantics can interact with geometry or shapes. It also illustrates how product components can be subjected to new views of the product development process when the product evaluation incorporate the interactions with the end-user(s), i.e. here a virtual human(s), or with a series of operators contributing to the manufacturing process, i.e. virtual humans here again.
Interactions with these virtual humans give a new insight to the semantic information that should be part of the object designed. Currently, functional information is addressed from an intrinsic point of view of standalone mechanism whereas most of the products designed incorporate interactions with human beings at the manufacturing stage or at the usage stage. It is therefore a new set of data that should be considered to be able to simulate interactions between the product and virtual users or virtual operators. Similarly, the requirements for virtual humans that should take part to simulations is a new issue for the integration of new categories of simulations in the product development process.
Since the design activity is addressed from a process point of view, the above topics raise the questions of how and when the description of a product can be efficiently enriched with the semantics related to component, sub-system or product interaction with virtual humans. Addressing these issues is a key to improving product design while taking into account early in the product development process some of the interactions between the product and human beings.
In addition, the above subsections show some issues of virtual environment modelling since the corresponding simulations take part to the design process and need to be appropriately integrated to be efficient, i.e. provide results on time with respect to the product development process, and financially accessible, i.e. the cost of setting up a virtual environment, updating it must be consistent with respect to the savings obtained after the simulation process.

6 - Conclusion

In this paper we give a description of some of the current mechanisms set up to annotate, structure and acquire shapes taking part to the product development process. Specific stages of the product development process have been addressed in accordance with some of the current topics developed in the scope of the AIM@SHAPE network of excellence.
The given illustrations shows that semantic annotations mechanisms take place at various levels ranging from the modelling phases to the interaction phases between the product and virtual humans or between the product and the designer.
Further work is ongoing among the partners to analyze, refine and develop the annotation, structuring and acquisition processes forming the main research activities of the network partners. To this end, knowledge engineering is a common approach adopted among the partners to identify structure and manipulate semantics through ontologies.

7 - Acknowledgments

The work has been carried out within the AIM@SHAPE (Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects) European network of excellence, contract n°506766 (http://www.aimatshape.net/). This network incorporates 14 European partners among which there are INPG, in Grenoble (France), CNR IMATI-GE, in Genova (Italy), IGD, in Darmstadt (Germany), EPFL, in Lausanne (Switzerland). The INPG and IMATI authors want to thank Jean-Philippe PERNOT and Vincent CHEUTET, both PhD students in co-tutelle, for their contribution to the work performed on fully free-form surface features. The EPFL authors want to thank also Mario Gutierrez and Tolga Abaci for their valuable help.

8- References

[1] MG-IT, (1999), 'Multi-views and « multi-representations » design framework applied to a preliminary design phase', P. Chedmail et al. editors, Integrated Design and Manufacturing in Mechanical Engineering, Kluwer, pp. 553-560.

[2] Bidarra, R. and Bronsvoort, W.F., (2000), 'Semantic feature modelling', Computer-Aided Design, 32(3), pp. 201-225.

[3] Giannini, F., De Martino, T. and Falcidieno, B., (1997), 'Context-dependent recognition and parametrization of shape features', IFIP WG.5.2 Workshop on Geometric Modelling in CAD, Airlie, Virginia, U.S.A., May 18-23.

[4] Bronsvoort, W. F., Noort, A.., van den Berg, J. and Hoek, G. F. M., (2003), 'Product development with multiple-view feature modelling', in Soennen, R. and Olling, G. J. (eds.), Feature Based Product Life-Cycle Modelling, Kluwer, pp. 57-76.

[5] Léon, J.-C., Fine, L., (2005), 'A new approach to the Preparation of models for F.E. analyses', Int. Journal of Comp. Appl., 23(1, 2, 3), pp. 166-184.

[6] Armstrong, C.G., Monaghan, D.J,. Price, M.A., Ou, H., Lamont, J., (2002), 'Integrating CAE Concepts with CAD Geometry', Engineering Computational Technology, Saxe-Coburg Publications, pp. 75-104.

[7] Rusák, Z., Horváth, I., Kuczogi, Gy., Vergeest, J. S. M., Jansson, J., (2000), 'Discrete domain modeling for shape conceptualization', 4th Int. Conf. of Eng. Design and Automation - EDA 2000, Orlando, Florida, USA, July 30-August 2.

[8] Catalano, C., Falcidieno, B., Giannini, F. and Monti, M., (2002), 'A Survey of Computer-Aided Modeling Tools for Aesthetic Design', J. Comput. Inf. Sci. Eng., 2(1), pp. 11-20.

[9] Fleisch, T., Brunetti, G., Santos, P., Stork, A., (2004), 'Stroke-Input Methods for Immersive Styling Environments'. Int. Conf. Shape Modelling Int., Genova, Italy, June, pp. 275-283.

[10] Shah, J.J., Mantyla, M., (1995), Parametric and feature-based CAD/CAM, Wiley-Interscience Publication, John Wiley Sons, Inc.

[11] Fontana, M., Giannini, F., Meirana, M., (1999), 'A Free Form Feature Taxonomy', Computer Graphics Forum, 18(3), pp 107-118.

[12] Pernot, J.P., Falcidieno, B., Guillet, S., Léon, J.C., (2004), 'Aesthetic design of shape using Fully Free-Form Deformation Features', Int. Conf. TMCE04, Lausanne, Switzerland, 15-18th April.

[13] Schek, H.J., (1974), 'The force density method for form finding and computation of general networks', Comp. Meth. in App. Mech. and Eng., 3(2), pp115-134.

[14] Guillet, S., Léon, J.C., (1998), 'Parametrically deformed free-form surfaces as part of a variational problem', Computer-Aided Design, 30(8), pp 621-630.

[15] Pernot, J.P., (2004), 'Fully Free Form Deformation Features for Aesthetic and Engineering Designs', PhD Thesis, INPG Grenoble (France) and Univ. Genova (Italy), October 14th.

[16] Pernot, J-P., Léon, J.-C., Falcidieno, B., Giannini, F., (2005), 'Multi-minimisations for shape control of fully free form features (δF4)', Int. Journal ASME, Comp. In System Engineering (to be published).

[17] Cheutet, V., Pernot, J.P., Léon, J. C., Falcidieno, B., Giannini, F., (2004), 'Fully Free-Form Deformation Features (δ-F4) incorporating Discontinuities', Int. Conf. ASME DETC DAC'04, Salt Lake City, USA.

[18] Cheutet, V., Falcidieno, B., Giannini, F., Pernot, J.P., Léon, J.C., (2005), 'Extension of the fully free-form deformation features to planar functional areas', Int. Conf. CAID05, Delft, June.

[19] Fleisch, T., Rechel, F., Santos, P., Stork, A.., (2004), 'Constraint Stroke-Based Oversketching for 3D Curves', In Eurographics Workshop on Sketch-Based Interfaces and Modelling 2004, Aire-la-Ville.

[20] Oliveira, M., Crowcroft, J. and Slater, M., (2003), 'An innovative design approach to build virtual environment systems', Proc. of the EUROGRAPHICS workshop on Virtual environments 2003, Zurich, Switzerland, pp. 143-151.

[21] Alexandre, T., (2003), 'Using design patterns to build dynamically extensible collaborative virtual environments', Proc. of the 2nd international conference on Principles and practice of programming in Java, Computer Science Press, Inc., Kilkenny City, Ireland, pp. 21-23.

[22] Kapolka, A.., McGregor, D. and Capps, M., (2002), 'A Unified Component Framework for Dynamically Extensible Virtual Environments', Proc. of the 4th international conference on Collaborative Virtual Environments, Bonn, Germany, pp. 64-71.

[23] Douglass, B., (2002), 'Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems', Add.-Wesley, September 2002.

[24] Kallmann, M. and Thalmann, D., (1999), 'Direct 3d interaction with smart objects', In VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology, ACM Press, pp. 124–130.