

CAVALCADE: a System for Collaborative Prototyping

Patrice TORGUET[^], Olivier BALET[^], Enrico GOBBETTI[^],
Jean-Pierre JESSEL[♦], Jérôme DUCHON[^], Eric BOUVIER^{*}

[^]CS-SI, 13, Rue Villet, Z.I. du Palays
31029 Toulouse, France
Patrice.Torguet@c-s.fr

^{*}CRS4, VI Strada OVEST, Z.I. Macchiareddu
09010 UTA (CA - Italy)

[♦]IRIT, Paul Sabatier University, 118, Route de Narbonne
31062 Toulouse Cedex, FRANCE

Abstract

Prototype design and testing is an indispensable stage of any project development in many fields of activity, such as aeronautical, spatial, automotive industries or architecture. Scientists and engineers rely on prototyping for a visual confirmation and validation of both their ideas and concepts. Using computers for designing digital prototypes is not a new idea since CAD applications are nowadays widely used. In this paper we present how new advances in 3D interaction and real time visualization research domains lead to the development of a collaborative and really interactive system for virtual prototyping. This work is supported by the European Community through out the ESPRIT programme 4.

Key Words: Virtual Prototyping, Collaborative Working, Distributed Virtual Reality

1. Introduction

Decisions taken during the design phase of large scale engineering projects are often the most delicate ones, because of their possibly dramatic effect on final results, timings, and costs. Mock-ups are routinely used for applications such as testing equipment integration, accessibility and space requirements in domains ranging from aerospace and automotive manufacturing to architecture. Virtual prototyping enables designers to test and improve their design. The process is the same as when using physical mock-ups but virtual prototyping is more efficient, you can use it and with more opportunities for multi-site collaborations [1].

The visual capabilities of present CAD tools are much too limited to make interactive inspection and model manipulation possible. With such systems, it takes a fair amount of time and imagination to isolate design errors. Even more important, the lack of support for co-operative work makes information sharing among geographically remote people difficult. Thus, large engineering projects have often been accompanied by the development of in-house virtual prototyping tools [2], showing the need for a general-purpose collaborative virtual prototyping solution.

This document first presents the methodologies and technologies which enable collaborative virtual prototyping tools to be built. Then, CAVALCADE concepts and its architecture are described. We then present each software related workpackages. Finally, the conclusion deals with the actual implementation of CAVALCADE along with future uses of the software.

2. Methodologies and Technologies

Natural interaction with digital mock-ups is very important, especially for testing purposes: by providing a virtual reality interface to the model, it is possible to give direct answers to important questions such as "can a short driver reach the controls on the dash ? Can an oil filter come straight out or does it have to be wiggled past some other components ?" [3, 4].

In an attempt to overcome present CAD system interactivity and concurrent design limitations, large engineering projects have often been accompanied by the development of various kinds of specialised virtual prototyping tools [2]. Examples include the ISS VR Demonstrator used by Rolls-Royce to make an assessment of how easy it would be to build an engine and maintain it [5] and Boeing's high-performance engineering visualisation system used during the design of the 777 [6]. Moreover, the French Space Agency (CNES) and CS-SI have jointly launched the PROVIS [7, 8] research project in 1995 in order to develop software solutions for satellite designers to create, manipulate, and study their models using digital mock-ups. Meanwhile, CRS4 and CERN have jointly developed the i3d system for supporting the design of CERN's Large Hadron Collider [1, 9]. These efforts show the interest of interactive virtual prototyping for early testing of designs.

The need for large-scale collaborative design tools has been recently recognised in the USA by the Defence Advanced Research Project Agency, which, in June 1997, announced its support for the VELA project, the latter being a proof of concept for a globally distributed design of multimedia processor chips [10].

Recently, research and development efforts for building virtual prototyping systems have started independently from the need of a specific project. Ongoing research at the Fraunhofer Centre for Research in Computer Graphics is studying how to integrate existing tools to provide virtual prototyping capabilities to existing CAD systems [11]. Some recent CAD products add-ons such as PTC Pro/Fly-Through, Mechanical Dynamics' ADAMS, Matra Datavision's Megavision, Prosolvia Clarus Real-Time Link, and EDS Unigraphics Modelling module follow the same direction, providing real-time visualisation capabilities for engineering designs.

Division's dV/Mock-Up and dV/Reality and EAI's VisMockUp are among the few products allowing designers to work on a shared prototype. However, their 3D interaction model does not permit accurate and fast operations and their co-operative features are mainly limited to networked review session without any telecommunication facilities. Working Model 3D, a Windows/NT product by the American company Knowledge Revolution, which concentrates more on motion simulation than on real-time interaction, has been selected by readers of NASA Tech Brief as the most significant new product introduced for the engineering community in 1996, thus showing the strategic importance of virtual prototyping software to industrial companies.

Unfortunately, most of these solutions suffer from severe drawbacks concerning either 3D interaction or collaborative capabilities. The latter issue is for sure one of the less developed in the virtual prototyping software industry although most of the recent large projects have reached a world-wide dimension.

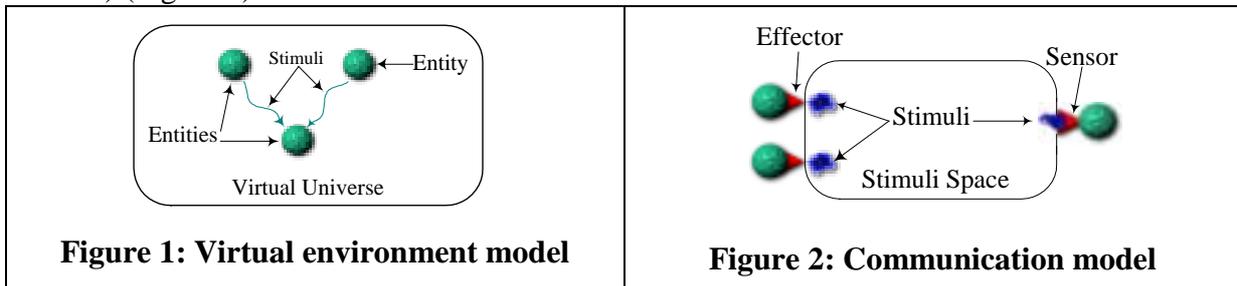
We think that it is now technologically possible to provide managers and designers with a powerful and intuitive software solution for collaborative virtual prototyping by integrating recent advanced visualisation techniques in several technological areas including distributed virtual reality, time-critical rendering techniques, 3D user interfaces, and projection-based visual displays.

CS-SI is currently leading the CAVACALDE Esprit project which aims at using all these techniques together for the development of a commercial product. This two-year-long project involves several end-user partners coming from different industrial domains to ensure the success of the final product: ESA/ESTEC (the European Space Agency), SEAT (the Spanish car builder), CSTB (the French scientific and technical centre for construction) and SNCF (the French national railway company). They provide a complete industrial setting for the specification and the tests of the product by being greatly involved in the project. The software application is currently being developed by CS-SI, IRIT and CRS4 while AIS (an Italian software/hardware development society) will enhance a hardware device for stereo viewing.

3. CAVALCADE concepts and its software architecture

Within CAVALCADE we use two main components in order to define a virtual environment: *entities* which encapsulate graphical objects and their behaviours; and *stimuli* which are the base of interaction and information exchange between entities. Both components are brought together in a *virtual universe* (Figure 1).

The units of communication, *stimuli* (phenomena or events perceptible by an entity), are exchanged through media called *stimuli spaces*. The stimuli spaces have been introduced in order to permit communications and interactions between many entities simultaneously. Each stimuli space is in fact a projection of the environment along a specific type of stimulus (3D graphics' space, multimedia space...). An entity receives perceptible stimuli (3D objects, near sounds...) through *sensors* and acts on its environment through *effectors* (producing new stimuli) (Figure 2).



We have introduced the entity paradigm in order to define in a generic way every kind of scene components. Entities can represent virtual objects, *avatars* (users' representation within the virtual environment) or 3D tools.

An entity is defined by its *behaviour*, a set of *attributes*, a set of *sensors* and a set of *effectors*. Both sensors and effectors enable an entity to communicate with its surrounding environment while its attributes define its internal state.

3.1. The software architecture

The CAVALCADE software architecture is made of a number of layered blocks. Each block being a piece of software which provides features to its upper blocks and uses the features provided by its lower blocks. The lower blocks are the network API (TCP/IP sockets in the current implementation) and the low level graphical API (currently OpenGL). Above these blocks are found general libraries like VERTIGO (a high level graphical toolkit based on the VRML97 standard built by CS-SI) as well as others developed within the CAVALCADE project: "VIPER" is a generic distributed platform, built by IRIT, composed of a library and a set of servers which enables shared virtual environments to be built. "Now" is a set of C++ classes, developed by CRS4, designed to render large CAD databases under "time critical" conditions.

The top-most layer, “Behave” (built by CS-SI), provides the behavioural core of the application, using all other blocks in order to build a 3D, collaborative high performance tool for virtual prototyping.

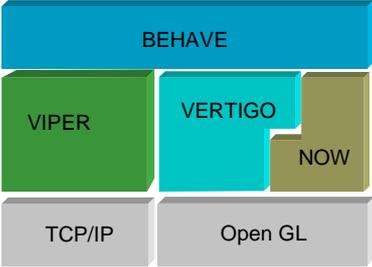


Figure 3: the CAVALCADE architecture.

4. Behave: Cavalcade entities

This layer aims at providing the behavioural components which enable to create CAVALCADE entity behaviours, thus defining the final application. In its basic version, CALVALCADE is functionally defined by the following entities : the Builder, the GUI, the Expert and several prototypes.

4.1. The Builder

The aim of this entity is to provide the user with high-level commands in order to build 3D prototypes. It manages the interaction process (multimodal dialogue) and the application’s specific functions (components assembly, documentation, etc.). In fact, it specifies the application’s functionality.

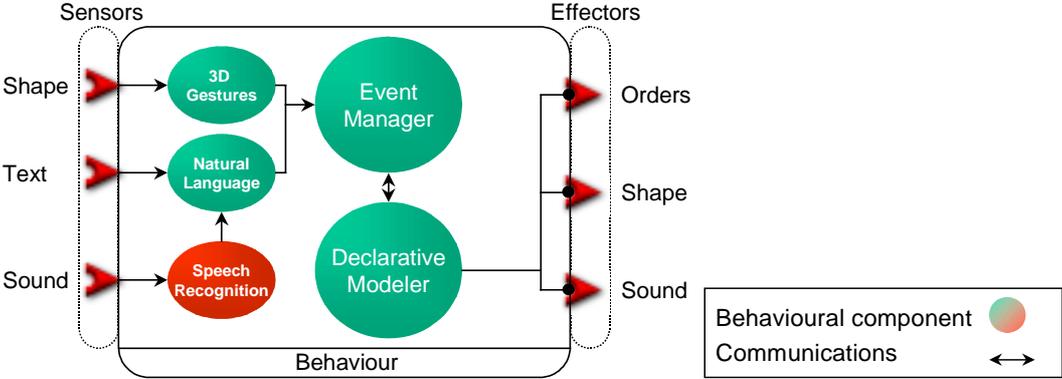


Figure 4: A simplified view of the Builder's behavioural components

Figure 4 presents the simplified model of this entity: Several behavioural components (input modules) analyse and control incoming stimuli such as gestures or sound in order to convert them into normalised events. These events are sent to the *Event Manager* which merges them into a single command which is then sent to the modeller.

Thanks to its multimodal event manager, CAVALCADE enables the user to combine several input devices (dataglove, speech recognition system, spaceball, flying mouse, mouse...) to build a single command. For instance, s/he can say to the system : “Put this antenna on this”, and in this case the second *this* stands for a support (previously selected with his or her flying mouse) which is retrieved in the dialog history thanks to its syntactic and semantic attributes (Figure 5).

Moreover, the multimodal event manager allows information coming from different users to be combined into a single command, thus addressing the co-operative dimension of our project.

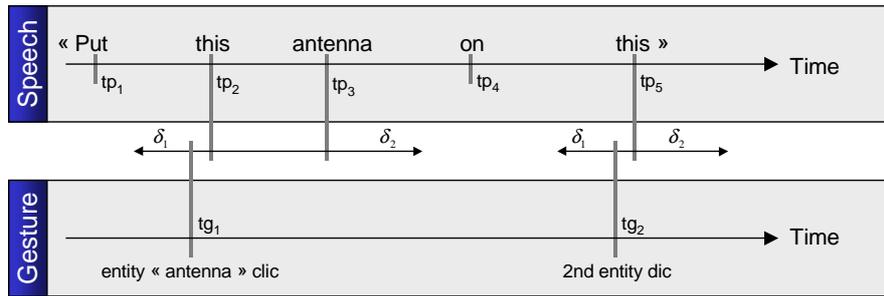


Figure 5: Event merging.

4.2. The GUI

The graphical user interface is, in fact, a simple frame where other entities add or remove their own widgets. For instance, an entity adds its widgets by sending stimuli to the GUI. Thereafter, this entity will receive stimuli through a specific sensor, each time its widgets are activated.

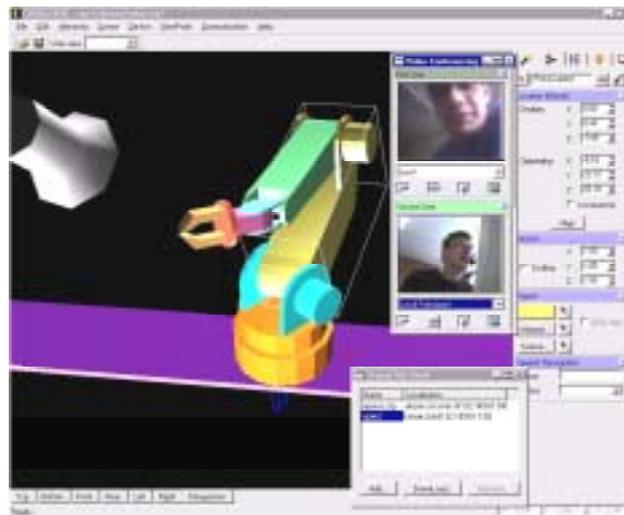


Figure 6: Cavalcade GUI

Two kinds of objects are present in the GUI (Figure 6): the 2D and the 3D widgets. The 2D GUI is a set of menus, dialog areas and buttons which surround the 3D GUI. The latter is made of the rendering window and a set of 3D manipulators enabling to interact with prototypes.

4.3. The Expert

It aims at checking for the coherence of a prototype: weight assessment, power consumption, construction integrity... This entity is to contain the expert's knowledge and the final industrial domain specificity. It will use several attributes (semantic attributes, etc.) of the prototype components in order to assess the validity of the prototype.

4.4. The prototypes

The prototypes will also be modelled according to the entity model. Thus, a 3D entity will embed several behavioural components such as “Dynamics” to simulate forces and torque applied to the entity or “Documentation Access” to enable the user to browse associated online documentation browsing. The prototype stores all its documentation with an URL pointing toward a user-defined HTML page. This page contains an HTML form which permits the edition of non 3D prototype properties and triggers access to external technical databases thanks to SQL request and ODBC¹ technology.

5. VIPER: Distributed Core

VIPER [12] is used within CAVALCADE to provide the necessary distributed architecture of such a collaborative application.

It is a generic Distributed Virtual Reality system, like MASSIVE [13], DIVE [14] or VLNET [15], and has been created in order to design any kind of application.

We will describe in this part of the document the distribution of a virtual environment using VIPER.

5.1. Distributing entities

VIPER entities exist in two flavours: simple entities which are managed by the site where they have been created and duplicated entities which exist on a set of sites (usually all the sites which manage a given collaborative session). Duplicated entities may define duplicated attributes which are synchronised by VIPER on a set of sites (i.e. when such an attribute is modified on one site, all other sites copies of this attributes are updated).

5.2. Entity direct interactions

Direct interaction (i.e. when an entity manipulates other entities or sends them messages) in a distributed context implies that interacting entities may happen to reside on different sites. We have decided to model those interactions by exchanging *orders* (a subclass of stimuli) between entities within the framework of a specific distributed stimuli space. This space, called the *order space*, sends orders to interested entity sites using point to point communications through a server. The server permits the use of fewer connections between sites as in the case of a site to site direct connection but also permits the sending of orders to a set of sites (in order to manage duplicated entities). In the current implementation we use TCP² but we plan to use IP³ multicasting for frequent and unimportant orders (e.g. position and orientation modification requests which are exchanged when an entity – for example the avatar – moves another one).

In order to send orders to the correct sites, the server translates the receiving entity ID to a site ID where the entity is stored. For duplicated entities, it sends the order to all sites in a given session (the session ID is also stored in orders). In the following figure, the server is hidden within the order space because its management is transparent to the programmer of the application. For example, there is absolutely no programming differences between the sending of an order to a local entity and the sending of the same order to a distant or duplicated one. The order space takes care of everything transparently.

¹ Open DataBase Connectivity is a standard application program interface for accessing databases.

² Transmission Control Protocol is the reliable data transport protocol of TCP/IP (the Internet protocols Suite).

³ Internet Protocol.

5.3. Transparent access to remote VRML files

The VRML space permits transparent download of VRML files from distant VRML servers (Figure 7).

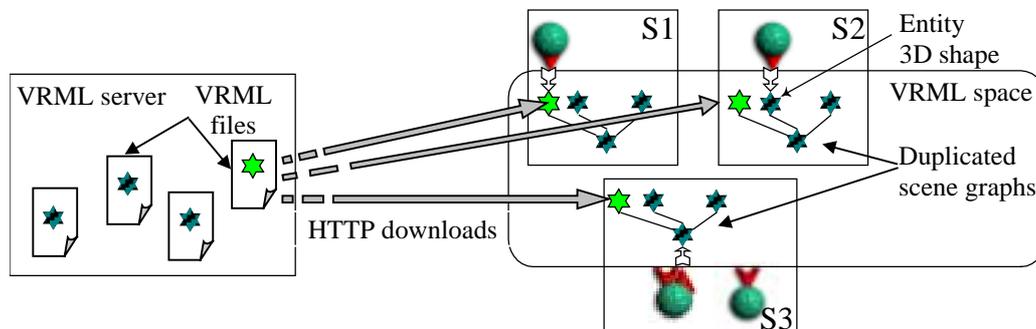


Figure 7: The VRML space and a VRML server

Those servers and the VIPER library use an HTTP-like protocol in order to permit the download of files as well as their upload when they need to be saved. Moreover, VIPER implements mechanisms enabling several users to work on the same VRML file (static and dynamic access rights).

5.4. Communication features

In order to implement the required communication features of a collaborative application such as CAVALCADE we have developed a stimuli space called the Multimedia space. This space enables users to communicate using both audio and video streams. Moreover, users may also exchange any kind of files through a shared file shelf.

The audio and video streams are exchanged using UDP⁴ socket communication. We are taking into account the non reliability of UDP with standard mechanisms like sequential numbering of network messages as well as data-specific parameters (a video frame number is encoded in potentially fragmented network messages so that the total frame might be discarded if any constituting message is lost).

Both audio and video stream features are available easily to the application developer: s/he only needs to add an audio or video effector so that an entity is able to send sound or images; similarly, adding an audio or video sensor to an entity enables it to receive sounds or images.

6. Now: Optimising Cavalcade for large CAD databases

The Now package is a set of C++ interfaces designed to render large CAD databases in “time critical” conditions. It provides automatic simplification algorithms for triangular meshes and level-of-detail management tools that are built on top of an efficient multiresolution structure.

6.1. Level of Details

Despite the continuous improvement in performance of CPUs and graphics’ accelerators, scenes exceeding a million of polygons cannot be handled directly at interactive speeds even on high-end machines. The traditional approach is to pre-compute a small number of independent level-of-detail (LOD) representing each object composing the scene, and to choose at run-time the best one based on a cost/benefit analysis [16, 17]. Even if LODs

⁴ User Datagram Protocol is the unreliable, connectionless data transport protocol of TCP/IP.

introduce a significant memory overhead, they are commonly used as they match easily with current “time critical” hardware and software architectures. Actually, as LODs are pre-computed, the polygons can be organised in the most efficient way (triangle strips, display list), exploiting raw graphics processing speed with retained-mode graphics. Moreover, LODs can be inserted directly into a scene graph architecture with minor changes in the rendering engine. Therefore, the LOD approach is particularly adapted to a system like CAVALCADE, where the scene is modelled as a set of entities organised in a VRML scene graph.

6.2. Geometry simplification

Users of virtual mock-up system do not want to have to deal with time critical issues when they conceive new prototypes. Therefore we have developed simplification algorithms to automatically create various representations of the cavalcade geometric entities.

As in [18], we implement mesh simplification schemes based on iterative vertex substitution in a generic framework of greedy algorithms for heuristic optimisation. The generic greedy algorithm is parameterised by a *binary oracle*, deciding which vertex substitutions are legal, and a *fairness predicate*, which assigns priorities to all vertex substitutions in the legal candidate sets. The algorithm proceeds in a series of greedy steps until the candidate set is empty. In each step, the best vertex substitution is removed from the candidate set, the mesh is modified accordingly, and new substitutions are re-evaluated for the vertices affected by the mesh change.

6.3. Results

The simplification algorithm has been implemented and tested on both Windows NT and IRIX machines. The results presented here have been measured on a Silicon Graphics Onyx with 2 MIPS R10000 194 MHz CPU, primary data cache size of 32 Kbytes, secondary unified instruction/data cache size of 1 Mbyte, main memory size of 1Gb. Similar results were obtained with a Pentium II processor at 300Mhz with 128 Mb RAM.

The comparison has been done using the publicly available *Metro* tool [19], which measures surface deviation. Table 1 presents the results obtained on the *bunny* model⁵. The maximum and average errors are similar to those obtained with more complex competing algorithms [20]. We also present in the next figure (Figure 8) pictures of the Fandisk model⁶ created using our simplification algorithm in several levels of details (20%, 10%, 5%, 2%, 1%).



Figure 8: Fandisk model in various LOD (20%, 10%, 5%, 2%, 1%)

The simplification time is linear to the number of triangles in the original model. We achieved a simplification rate of more than 1200 triangles per second on all the models we have tested (1286.8 for the Fandisk). This rate includes a time overhead that we introduced to build a multiresolution structure during processing. This structure is not required, but it provides

⁵ This model is available at the following URL: <http://www-graphics.stanford.edu/data/3Dscanrep/>

⁶ This model is available at the following URL: <http://www.research.microsoft.com/~hoppe>

CAVALCADE users with an interactive way to select the more pertinent LOD without changing the algorithm complexity.

Vertices	Triangles	Maximum error	Average error
17418 (50%)	34653	0.47433	0.01009
8079 (25%)	16006	0.83984	0.01631
3484 (10%)	6873	0.79855	0.02583
1742 (5%)	3428	1.04764	0.04387
697 (2%)	1361	1.02499	0.09544
349 (1%)	675	1.78891	0.18555
175 (0.5%)	337	2.41363	0.34466

Table 1- Geometric error measurements for the *bunny* model at various resolutions. Errors are % of the dataset bounding box diagonal.

7. Conclusion and exploitation plans

This document has presented CAVALCADE, a collaborative virtual prototyping tool which is currently being developed within an Esprit project. A pre-release version of the software is currently being assessed by end-users.

CAVALCADE will be used by end-users for several activities:

SEAT will internally exploit the tool for car conception during collaborative sessions, thus speeding-up the design process and lowering communication costs. Moreover, it will extend the use of CAVALCADE to its designer partners in order to maintain the coherence of each partner's work. CAVALCADE is expected to be the core software system of SEAT's new design concept (Concepto Modular de Diseño: Modular Design Concept) which specifies collaborative working procedures for its partners and sub-contractors.

For ESA, CAVALCADE could be integrated to the design process in order to increase the capability of quick and complete design iterations, speed-up documentation production and management which will be done in the early phase of the design. ESA will carry out internal exploitation for small satellites programs and scientific experiments.

CAVALCADE features are ideally suited to answer the needs of SNCF projects in train design, railways infrastructure implementation, and train station design. In particular, SNCF, as a train designer, would benefit from a more efficient and accurate prototyping tool, like CAVALCADE, during the collaborative design processes with its sub-contractors.

CSTB will investigate the use of a collaborative virtual prototyping tool like CAVALCADE in the design phase of construction projects. In this phase, different CSTB teams (architects, structural engineers, etc.) on different locations study the project from different points of view.

8. Acknowledgements

The CAVALCADE project is supported by the European Community through the ESPRIT programme 4 (project number 26285).

9. References

- [1] Balaguer J.F. *VRML for LHC Engineering*. Proceedings Eurographics Workshop on Virtual Environments, 1996.
- [2] Ellis G., *They're Not Making 'Em Like They Used To: Virtual Reality Saves Time and Money in Manufacturing and Construction*, Iris Universe, Summer 1996.
- [3] Potter C. D. *Digital Mock-up Tools Add Value to Assemblies*. Computer Graphics World, November 1996.
- [4] Balet O., *Toward Collaborative Construction and Design of Interactive Virtual Prototypes*, proceedings of Information Society Technologies conference, Vienna, 1998.
- [5] Greenfield D., *Virtual Prototyping at Rolls-Royce*. Cambridge, Mass., MIT Press, 1996.
- [6] Mc Neely W. *Boeing's High Performance Visualization Software: FlyThru*, CERN Computing Seminar, June 1996.
- [7] Balet O., Luga H., Duthen Y., Caubet R., *PROVIS : A Platform for Virtual Prototyping and Maintenance Tests*. Proceedings IEEE Computer Animation, June 1997.
- [8] Torguet P., Balet O., Caubet R., *A Software Architecture for Collaborative Virtual Prototyping*, *COMPUGRAPHICS'97*, Vilamoura, Algarve, Portugal, December 1997.
- [9] Gobbetti E., Balaguer, J. F., *I3D: An Interactive System for Exploring Annotated 3D Environments*. Reprinted in Scateni R (Ed) *Scientific Visualization'95*, Symposium Proceedings, World Scientific, 1995.
- [10] Goering R., Clarke P. *Web-Based Design Hoists New Sail*, EETimes, 16 June 1996.
- [11] Jasnoch U., Klement E., Kress H., Schiffner N., *Towards Collaborative Virtual Prototyping in a World Market*, Proceedings FAIM, Atlanta, 1996.
- [12] Torguet P., Rubio F., Gaildrat V., Caubet R. *Multi-user interactions in the context of concurrent virtual world modeling*, *Virtual Environments and Scientific Visualization'96*, Springer Computer Science, pages 121-136, 1996.
- [13] Greenhalgh, C.M., *Awareness Management in the MASSIVE Systems*, Distributed Systems Engineering, Vol. 5, No 3, September 1998, pp. 129-137, IOP Publishing.
- [14] Hagsand O., Lea R., Stenius M., *Using spatial techniques to decrease message passing in a distributed VE*, *VRML'97*, Monterey, CA, Feb, 1997.
- [15] Pandzic I., Capin T., Lee E., Magnenat Thalmann N., Thalmann D., *A flexible architecture for Virtual Humans in Networked Collaborative Virtual Environments*, *Eurographics'97*, Budapest, Hungary, 1997.
- [16] Funkhouser T. A., Sequin C. H., *Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments*. *Computer Graphics (SIGGRAPH'93 Proc.)*, 1993.
- [17] John Rohlf and James Helman. *IRIS performer: A high performance multiprocessing toolkit for real-Time 3D graphics*. *Computer Graphics (SIGGRAPH'94 Proc.)*, pages 381–395. ACM Press, July 1994.
- [18] Kobbelt L., Campagna S., Seidel H. P., *A general framework for mesh decimation*, *Graphics Interface Proc. (GI-98)*, pages 43–50, San Francisco, June 18–20 1998. Morgan Kaufmann Publishers.
- [29] C. Rocchini P. Cignoni and R. Scopigno. *Metro: measuring error on simplified surfaces*. Technical report B4-01-01-96, Institute I.E.I.-C.N.R., Pisa, Italy, Jan. 1996. <http://miles.cnuce.cnr.it/cg/metro.img.html>.
- [20] Cignoni P., Montani C., Scopigno R., *A comparison of mesh simplification algorithms*. *Computers and Graphics*, 22(1), pages 37–54, January 1998.