

# A DIYD (DO IT YOURSELF DESIGN) SYSTEM FOR VEHICLE DESIGN BASED ON 3D VISUALIZATION AND ONTOLOGIES

Lambros Makris, Nikolaos Karatzoulis, Dimitrios Tzovaras  
*Informatics & Telematics Institute, 1<sup>st</sup> km Thermi-Panorama, Thessaloniki, Greece*

Evangelos Bekiaris  
*Hellenic Institute of Transport, 6<sup>th</sup> km Charilaou - Thermi, Thessaloniki, Greece*

Keywords: DIYD system, Configurator, 3D Visualization, Ontologies.

Abstract: The customization level of vehicles is growing in order to deal with increasing user needs. Web browsers are becoming the focal point of vehicle customization, forming personalized market places where users can select and preview various setups. However the state of the art for the completion of the transaction is still very much characterized by a face-to-face sales situation. Direct sales over the internet, without sales person contacts, are still a small segment of the market, of only a few percent, for European manufacturers. This paper presents an Intelligent DIY e-commerce system for vehicle design, based on 3D Visualization and Ontologies that aims at enabling a suitable representation of products with the most realistic possible visualization outcome in order to help prospective customers in their decision. The platform, designed for the vehicle sector, includes all the practicable electronic commerce variants and its on-line product configuration process is controlled by an ontology that was created using the OWL Web Ontology Language.

## 1 INTRODUCTION

Automotive enterprises are becoming more customer-centric to meet today's challenging market demands. This calls for restructured B2C relations and related new technologies. The automotive industry has furthermore become highly networked and requires improved communication on products and components in relation to its B2B relations.

Information and communications technology (ICT) can be used to support business and design activities. ICT does not change the fundamental goals of any organisation but makes it possible to optimize and coordinate design, manufacturing and marketing. In the automotive industry, ICT can:

- Improve design procedures;
- Allow optimization of design, manufacturing and marketing;
- Fine tune manufacturing processes;
- Provide the best product or service to current and potential customers; and
- Respond rapidly to customers' needs.

It has been envisioned that e-commerce and mass customization will emerge as a primary style of manufacturing in the coming decade and beyond. The integration of design, manufacturing, and logistics over the Internet will be the trend for the factory of the future. Effective supply chain management for mass customization will enhance profitability through a synergy of increasing customer-perceived value while reducing the costs of design, production and distribution. Companies successfully adapting to this new style of manufacturing will be able to reduce reliance on the traditional marketing channels, to gain more market shares globally, and to achieve high-efficiency product realization.

This technology can enhance the established strengths of nowadays' industries in global manufacturing. It will benefit a wide variety of industries such as electronics, machinery, appliances, and logistics.

## 2 DIYD SYSTEMS AND THE AUTOMOTIVE BUSINESS

DIYD (Do It Yourself Design) systems enable companies to extend their markets anywhere, anytime via the Internet. BMW, for example, sells six out of ten cars to order. Although the order-to-delivery time is very long, up to two months, much longer than that for regular cars, customers are prepared to wait (The Economist 2004). However current systems usually only allow for a simple selection of options and a visual presentation of the result in the format of data sheets, tables and photographs of the vehicle exteriors and/or interior. The configuration system is not driven by customer requirements. The user has to interpret to what extent the various technical features will satisfy his/her functional requirements and needs.

In order to support the user in his/her choices, information should be presented in the most comprehensible way. This can be accomplished by offering the user detailed access to information using reconfigurable electronic catalogues and presenting the resulting configuration using 3D virtual prototypes.

The use of 3D virtual prototypes in a virtual environment can enhance visualisation and perspective viewing of the designed car. Unlike 2D graphics, users can interact via the web browser to navigate around an object and to move and rotate it. This type of Virtual Reality is much more flexible than a static image and allows for an apparently infinite number of different views on the vehicle. It is characterized by the use of 3D computer models presented on a 2D computer screen using 2D interaction devices like a mouse. The use of such interactive visuals has already undergone two cycles of hype in the internet business. But it was not successful due to overloaded solutions and bandwidth problems of the internet. Recently the technologies behind have gained momentum in the European automotive industry in the field of Digital Mock-ups (Döllner & Kellner 2000), which are used very successfully in the development process employing again the internet for both intranet and B2B communication. Thus the time has come to transfer this success to field of mass customization and DIY design.

But Virtual Environments go beyond. Immersive systems allow for a dynamic stereoscopic view on the vehicle exterior and interior and add intuitive 3D interaction by tracking technology to the user interface. These sophisticated VR solutions are currently used in vehicle design and development.

Internal projects in the automotive industry showed that VR technology was still too expensive to be used in customer communication and vehicle configuration around 5 years ago. This was mainly due to the use of expensive hardware. In the meantime inexpensive VR systems on commodity hardware basis have been developed in Europe, e.g. within the VIEW project (Wilson & D’Cruz 2006). These results can be further exploited to develop appropriate VR systems for mass customization and visualisation.

## 3 THE CATER SYSTEM

As already stated, current systems usually only allow for a simple selections and 2D visual presentation. We present an intelligent and user-friendly e-commerce solution, namely CATER, by adopting additional technologies such as a configuration engine supported by ontologies, advanced search mechanisms, and 3D visualisation in a virtual reality environment. The focus of the system is on the vehicle industrial sector; however the intention is that the system will be suitable for suppliers, and wholesalers, from other sectors, such as furnishing, clothing etc.

In our use scenario a customer is connected to the CATER system using a traditional web browser. He searches in the 3D object database, by example, to find particular components that are of interest to him. The system, using an ontology, prevents him from selecting components which are incompatible. At the same time the user can pose particular constraints, such as maximum cost, which are honoured by the system. He can then use a VR interface to connect the components together and form a design that suits him. The final selection can then be saved or forwarded to the factory for realization. Figure 1 presents the basic modules of the CATER system architecture.

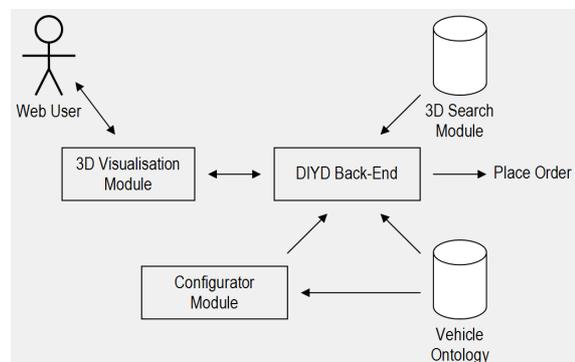


Figure 1: The main modules of the DIYD system.

### 3.1 Intelligent Configurator Module and Ontology

The Intelligent Configurator module in CATER is a web based application that allows the user to assemble vehicles based on the available vehicle parts that are being stored in the systems repository maintained by the vehicle manufacturer. Figure 2 displays the Units of the Configurator module.

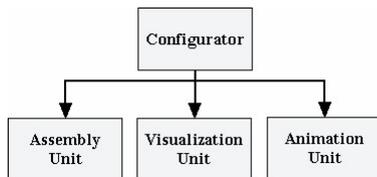


Figure 2: The Units of the Configurator Module.

The Assembly Unit allows the user to insert individual 3D objects to the scene that can consist of a fully functional vehicle. The user can compose the desired vehicle according to his/her needs by selecting the vehicle's parts. The vehicle part and the texture selection processes are being controlled by the system Ontology (Kompatsiaris et al 2005, Tsampoulatidis et al 2004, Mezaris et al 2003). The main functionalities of the Assembly Unit are the following: (i) Insertion of 3D object parts, (ii) Selection of texture and (iii) Assembly process based on dynamic constraints (e.g. weight).

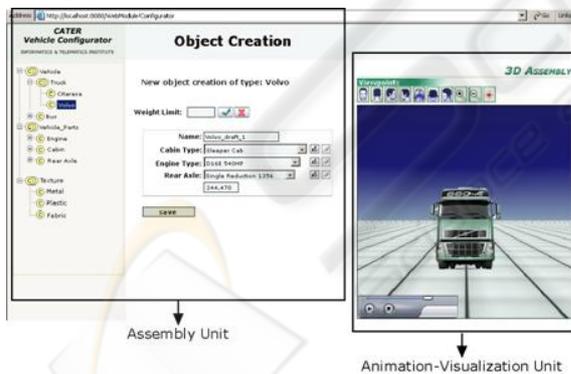


Figure 3: The web interface of the Configurator.

Once the user has selected the preferred vehicle parts then the selected parts are loaded to the 3D scene and the user is allowed to modify the colouring scheme of each selected part by activating the textures menu option (Figure 4). The allowed colouring scheme is defined in the CATER ontology.

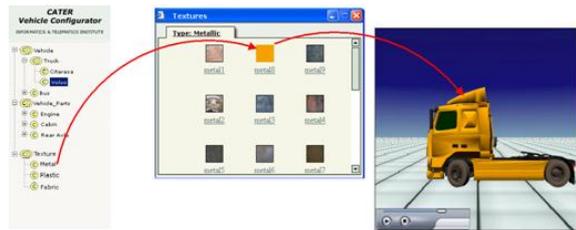


Figure 4: Modification of the colouring scheme of the inserted 3D parts based on the CATER ontology.

The purpose of the Visualization Unit is to record and store the 3D object assembly steps in real-time. The assembly sequence is being stored in the 3D animation repository for future reproduction. The Visualization Unit allows the user to select various viewpoints in order to preview the assembly process from various viewing angles (Figure 5).

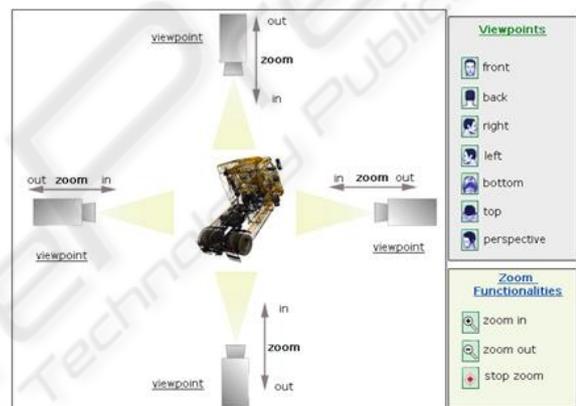


Figure 5: The viewpoints supported by the Visualization Unit.

The Animation Unit allows the reproduction of the vehicle parts assembly processes that are stored in the animation system repository. In the Animation Unit the user can control the viewpoints and the playback of the loaded vehicle assembly process. Animations can be prepared by the manufacturer to enhance the user experience or to highlight particular features of the vehicle. Animations can also be useful in situations where the assembly order is important for the customer to understand how a particular set of parts work together. The web interface of the Configurator Module is depicted in Figure 3.

### 3.2 3D Visualisation Module

Virtual Reality (VR) interfaces can provide the most realistic presentation of a configuration for end users. It combines high quality visualization with the correct perception of depth and scale, which enables

a feeling for the roominess of the interior of a car. Additionally, a highly intuitive interface allows easy manipulations of models, assemblies and parts. Combined with simulations (e.g. physics or a man-model), the models can be evaluated in terms of packaging (fit of the components) of ergonomics.

The 3D Visualization Module of the CATER system is realized on the Visualization and Animation Units. The structure of the individual 3D scenes supported by the 3D Visualization Module can be seen in Figure 6.

The 3D scene contains the viewpoints, the lighting of the 3D world, the background, the 3D objects and the object functionalities (interactions) that are created dynamically according to the ontology specifications. The user interaction with the 3D scene is achieved by the use of predefined VRML Protos. For every 3D object that is inserted in the 3D scene an animation representing its assembly process is dynamically generated.

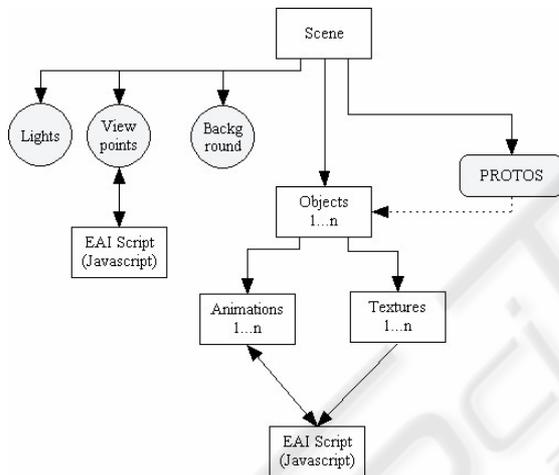


Figure 6: The 3D scene structure of the 3D Visualization Module.

The playback functionalities of the assembly process are controlled by a panel (Figure 7) that was developed using several VRML sensors (TouchSensor & PlaneSensor).

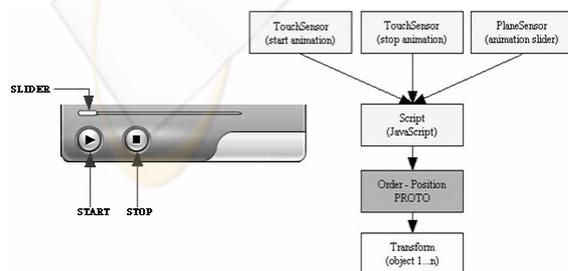


Figure 7: The assembly process control.

### 3.3 3D Geometry Search Module

The 3D geometry search module utilizes novel algorithms for low-level feature extraction from 3D objects, based on geometric characteristics. The algorithms are robust to affine transformations (rotation, translation, scaling) and are applied to 3D objects regardless of their degeneracies, formats and levels of detail. This results in a more compact representation of the objects, which uniquely characterizes them. The 3D search module can be used whenever the user wants to provide a specific part of a vehicle as a query and retrieve similar objects from the repository.

Every 3D object is described with a rotation, scaling and translation invariant descriptor vector, which is formed according to the Spherical Trace Transform (STT) (Zarpalas et al 2007). Initially, every object is translated and scaled so that all objects are expressed in the same coordinate system. To achieve the latter, a local coordinate system has been defined centred to the centre of mass of the object and scaled so that the object fits to the unit sphere. Then, the object's binary volumetric function is computed

$$f(x) = \begin{cases} 1 & \text{when } x \text{ lies within the 3D model's volume,} \\ 0 & \text{otherwise.} \end{cases}$$

and the STT (Figure 8) is performed as follows:

- A set of radius segments is defined. Every radius segment  $\Lambda_i$  is formed by the intersection of a radius  $(\eta_i, \rho_j)$  with the object.
- A set of spheres, concentric to the unit sphere is defined.
- A set of plane segments is defined for every sphere  $S_r$ . Every plane segment  $\Pi_i^r$  is formed by the intersection of a plane tangent to the sphere at point  $P_i$  with the object.

The points  $P_i$  and radii  $r_i$  are uniformly distributed on the sphere's surface exploiting the icosahedric-based tessellation.

Every  $\Lambda_i$  segment is treated as a one-dimensional signal and descriptors based on classical 1D Discrete Fourier Transform and an integration transform are computed. Every plane segment  $\Pi_i^r$  is treated as a 2D signal and descriptors based on the Krawtchouk, the Zernike and the Hu moments, the polar wavelet transform and the 3D Radial Integration Transform (Daras et al 2006) are computed. Then, the spherical Fourier Transform is applied separately on every extracted descriptor, so as the final descriptors are invariant under rotation and, thus, appropriate for 3D object matching.

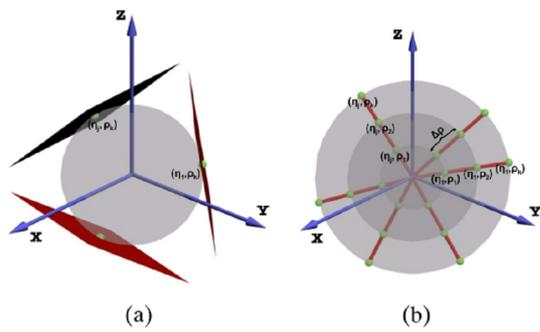


Figure 8: The Spherical Trace Transform.

The matching procedure is based on a mixture of the weighted Minkowski L1 distance and the normalized distance.

$$L1_n^i = \sqrt{\sum_{k=1}^S |W_{nk}D^Q(k) - W_{Ck}D^{A^i}(k)|},$$

Equation 1: Weighted Minkowski L1 distance, where  $D^T(k)$  is the k-th element of object.t T descriptor vector and  $W_i, T$  is the assigned weight

The computation of the weights for every single descriptor is based on the statistical behaviour of every descriptor for every class (e.g. mean value, standard deviation, etc). Two different methods for weight assignment have been proposed. The method has been tested on the Princeton Shape Benchmark. Figure 9 depicts the results in terms of precision and recall diagrams.

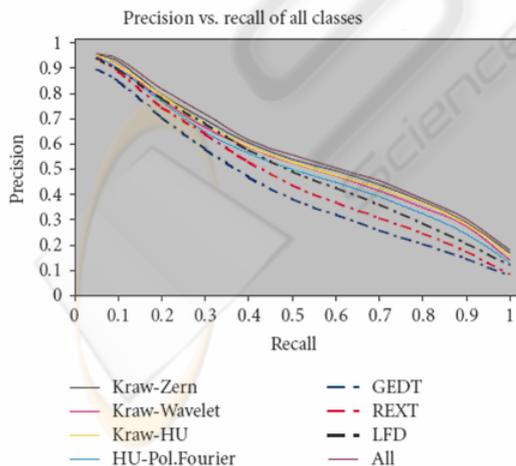


Figure 9: Efficiency of the STT using combination of different descriptors (Krawtchouk, Zernike, Hu etc).

### 3.4 Configurator Implementation

The Configurator Module is implemented using the Java programming language. The system runs on Apache Jakarta Tomcat as a Java Servlet and it is based on the Jena framework, which is a Java framework for building Semantic Web applications.

The ontology was created using the OWL Web Ontology Language, and the Protégé OWL-Plugin. The persistent store of the ontology is achieved using the persistence subsystem of Jena, while the 3D visualization was developed using the VRML standard and External Authoring Interface (EAI) mechanisms.

The OWL-DL profile which was used in order to create the ontology, is based on Description Logics. Description Logics are a decidable fragment of First Order Logic2 and are therefore amenable to automated reasoning. It is therefore possible to automatically compute the classification hierarchy and check for inconsistencies in an ontology that conforms to OWL-DL (Horridge et al 2004).

#### 3.4.1 Specification of Classes

The classes in the Ontology are interpreted as sets that contain individuals. They are described using formal descriptions that state precisely the requirements for membership of the class. For example, the class “Vehicle” contains all the individuals that are of type Vehicle in the CATER domain. The taxonomy of the classes is being achieved using the superclass-subclass model hierarchy.

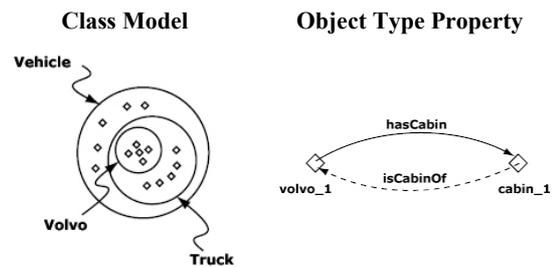


Figure 10: Example of the class hierarchy of the class “Vehicle” and an example of an Object Type Property for the individual “volvo\_1”.

#### 3.4.2 Specification of Properties

There are two types of properties supported by our ontology a) Data Type Properties and b) Object Type Properties. These OWL Properties represent relationships between two individuals.

In OWL, properties are used to create restrictions. In our ontology the restrictions were

used to restrict the individuals that belong to a class. We used the universal quantifier  $\forall$  restrictions to constrain the relationships along a given property to individuals that are members of a specific class. For example, the universal restriction  $\forall$  hasCabin cabin\_1 describes the individuals all of whose hasCabin relationships are members of the class Cabin.

Table 1: List of the property restrictions applied to the example class Volvo.

Class: Volvo
NECESSARILY
Truck
hasCabinEngineType
hasEngineType
hasRearAxle
accessType write
INHERITED
Root
owl:Thing
Vehicle

Cardinality restrictions were used to define the order in which the individual object parts should appear during the 3D assembly process (i.e. real-time animation). The cardinality restrictions provided the way to describe the class of individuals that have at least, at most or exactly a specified number of relationships with other individuals or datatype values.

The hasValue restrictions, denoted by the symbol  $\exists$ , were used to describe the set of individuals that have at least one relationship along a specified property to a specific individual. For example, when we wanted to predefine the dimensions of an individual object part we used a hasValue restriction (dimensions  $\exists$  "40-50-80").

### 3.4.3 RDQL – Data Oriented Query Model for the CATER Ontology

The CATER ontology uses an implementation of the RDQL query language for querying RDF models using the Jena API. RDQL provides a data-oriented query model so that there is a more declarative approach to complement the fine-grained, procedural Jena API.

RDQL queries only the information held in the models; there is no inference being done. The RDQL system receives the description of what the application requests, in the form of a query, and returns that information, in the form of a set of bindings.

RDQL is an implementation of the SquishQL RDF query language, which itself is derived from

rdqDB. This class of query languages regards RDF as triple data, without schema or ontology information unless explicitly included in the RDF source.

RDF provides a graph with directed edges - the nodes are resources or literals. RDQL provides a way of specifying a graph pattern that is matched against the graph to yield a set of matches. It returns a list of bindings - each binding is a set of name-value pairs for the values of the variables. All variables are bound (there is no disjunction in the query).

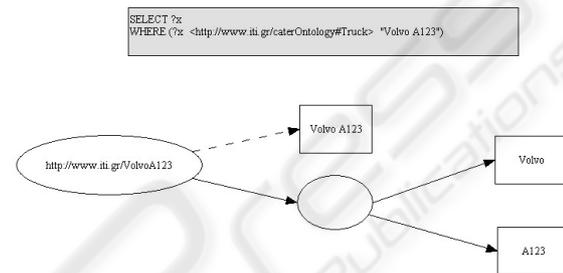


Figure 11: Graphic representation of an example RDQL query used for the CATER ontology.

## 3.5 The e-shopping Platform in Practice

The use of the CATER platform “brings” advantages for both suppliers and buyers regarding (i) the cutback of transaction costs, (ii) the use of automated supply procedures, (iii) economy of scale, (iv) wide access on both local and international markets, (v) dynamic real-time price mechanisms/modules and (vi) the use of compatible/expandable technologies.

The requirements of the described CATER platform for vehicle products that together with the Intelligent Configurator Module and the 3D Visualisation Module comprises the advanced 3D Shop system are:

- search and present all the available products, based on multi-criteria search engines
- group products into multilevel categories (set by the e-shop administrator)
- make offers/ sales and promote them
- update both the product catalogue and all items’ availability (set by the e-shop administrator)
- create/use shop baskets (by the end buyers)
- provide several convenient pay/receive methods
- provide a secure e-payment credit card transaction (with the use of HTTPS and SSL protocols).

However, the efficiency and overall quality of an e-commerce service depends “heavily” on its automatic connection with the existing ERP (Enterprise Resource Planning) system for the catalogue, prices, stock and product update. In order to integrate all the available ERP data with the e-shop database, a powerful staging mechanism is developed and securely transfers all necessary data. This staging process uses a smart “track changes” algorithm, to enhance the update speed.

There are two staging processes, Real Time Staging and Off Line Staging (that uses an automated batch process). The characteristics of the two staging “methods” are compared in the following table.

Table 2: Staging Procedures comparison.

	<b>Real Time Staging</b>	<b>Off Line Staging</b>
<b>Data Update</b>	(+) All data are updated at all times	(-) All data are updated at specifically defined time periods
<b>Infra-structure</b>	(-) Reliable, high speed, technical infrastructure is necessary, available on a 24x7x365 basis	(+) Not so advanced technical infrastructure is necessary
<b>Security</b>	(-) The system can be secure but certain “protective” actions must be taken	(+) Security is obvious

The previous table shows that a real time staging procedure should be followed only if the nature of the commodity traded imposes the constant database update. In our case an every day off line procedure is chosen for both security and convenience reasons.

Yet, if we try to deduct a general case example we must notice that each company’s and product’s needs, concerning the use of an e-market, are different; therefore the connectivity solutions (between an e-shop and an ERP) provided vary depending on: (i) the ERP used (it can be a widely used international ERP such as SAP, Oracle Applications, etc. or it can be a custom made system that fits to specific needs), (ii) the transaction volume and the form of the data transferred, (iii) the importance of the information transferred (regarding time, safety etc. aspects), (iv) the use of unilateral or bilateral communication and (v) whether it is an on-line or a batch transfer of data.

E-commerce services offered through a B2C (business to consumer) or B2B (business to

business) system, provide the necessary infrastructure for real time e-business and an added value package of services that guarantee faster and more efficient buy and sell transactions, access to a broadened database of buyers/suppliers and business opportunities through the development of new partnerships.

## 4 CONCLUSIONS

In conclusion, in this paper we presented an interactive and user-friendly e commerce solution for the vehicle sector, but appropriate for other sectors as well. Volvo Technology Corporation (VTEC) has been the end-user responsible for using and testing the CATER platform, so a number of its vehicles were integrated in the platform for evaluation and testing purposes.

Finally, the main contribution is that our approach adopts additional technologies such as a configuration engine supported by ontologies, advanced search mechanisms, and 3D visualisation in a virtual reality environment aiming at enabling a suitable representation of products in order to achieve the most realistic possible visualization and simulate an up to close shopping procedure.

## ACKNOWLEDGEMENTS

We wish to acknowledge the CATER project Consortium for their valuable contributions to this work. The CATER project is partially funded by the EC.

## REFERENCES

- The Economist, 2004. Research Tools Survey: Fighting Back.
- D. Döllner and G. Kellner, 2000. Digital Mock-up in Rapid Prototyping in Automotive Product Development. *Journal of Integrated Design and Process Science* 4 (1), p. 56-66.
- J. Wilson and M. D’Cruz, 2006. Virtual and interactive environments for work of the future. *International Journal of Human-Computer Studies* 64 (3), p. 158-169.
- I. Kompatsiaris, V. Mezaris, and M. G. Strintzis, 2005. Multimedia content indexing and retrieval using an object ontology, *Multimedia Content and the Semantic Web: Methods, Standards and Tools*, G. Stamou and S. Kollias (Eds), ISBN 0-470-85753-6, Wiley, p. 339-371.

- I. Tsampoulatidis, G. Nikolakis, D. Tzouvaras and M. G. Strintzis, 2004. Ontology Based Interactive Graphic Environment for Product Presentation, in Proc. CGI 2004, p.644 - 647, Heraklion, Crete, Greece.
- V. Mezaris, I. Kompatsiaris, M. G. Strintzis, 2003. An Ontology Approach to Object-Based Image Retrieval, Proc. IEEE International Conference on Image Processing (ICIP 2003), Barcelona, Spain, vol. II, p. 511-514.
- D. Zarpalas, P. Daras, A. Axenopoulos, D. Tzouvaras, and M. G. Strintzis, 2007. 3D Model Search and Retrieval Using the Spherical Trace Transform. EURASIP Journal on Advances in Signal Processing, vol. 2007, Article ID 23912
- P. Daras, D. Zarpalas, D. Tzouvaras, and M. G. Strintzis, 2006. Efficient 3-D model search and retrieval using generalized 3-D radon transforms. IEEE Transactions on Multimedia, 8(1), p. 101-114.
- Apache Jakarta. Available at: <http://jakarta.apache.org/>.
- Jena, A Semantic Web Framework for Java. Available at: <http://jena.sourceforge.net/>.
- The OWL Web Ontology Language. Available at: <http://www.w3.org/TR/owl-features/>.
- Protégé OWL Plugin. Available at: <http://protege.stanford.edu/plugins/owl/>.
- Web 3D Consortium, VRML Standard. Available at: <http://www.web3d.org/x3d/vrml/>.
- M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe, 2004. A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools, Edition 1.0, The University of Manchester.

