

HOM3R: A 3D Viewer for Complex Hierarchical Product Models

González-Toledo, Daniel¹, Cuevas-Rodríguez María¹ Garre, Carlos¹, Molina-Tanco, Luis¹, Reyes-Lecuona, Arcadio¹

Universidad de Málaga

E.T.S.I. Telecomunicación, Departamento de Tecnología Electrónica, Grupo Diana

Campus Universitario de Teatinos s/n, 29071 Málaga, Spain

email:{dgonzalez, mariacuevas, cgarre, lmtanco, areyes}@uma.es

www: <http://www.diana.uma.es>

Abstract

We present HOM3R, a novel 3D viewer designed to manage complex industrial product models. The viewer includes a JavaScript API to interface with existing or new browser-based applications. We extend state-of-the-art interaction techniques and introduce a novel navigation metaphor to navigate around complex products using constrained trajectories. To address the challenge of discovering and accessing the parts of the complex object, which are not visible, a set of occlusion management techniques have been implemented. The viewer presents other useful features such as hierarchical part selection and linking of information to 3D geometry. A user-centred evaluation of the tool has been carried out and is described in the paper.

Keywords: Human-centered computing~Virtual reality. Human-centered computing~Web-based interaction. Human-centered computing~Interaction techniques.

1. Introduction

Many products developed or used by different industries can be described as an assembly of a set of components or parts. A model of such a product may consist of a set of 3D objects representing the product parts and the information needed to assemble

the parts into the final product. The model may also contain additional information relevant to the product context; this information may be linked to specific parts of the product or to specific areas in the product geometry. Some application scenarios may require the availability of a 3D viewer for this type of product models. These scenarios include, among others, design aid, maintenance support or training. The requirements of a 3D product viewer depend on the specific application needs and on the specific characteristics of the product model. Most products, with the sole exception of the simplest ones, present some type of hierarchical organization of their parts. A 3D viewer of complex hierarchical product models presents a number of challenges:

- Selection and manipulation of product parts should be available at different levels within the hierarchy. One may want to manipulate or select the whole product, a single part or even a small area within one part. The viewer should provide some way to work at different levels within the product hierarchy.
- Some parts may not be visible or accessible for selection and manipulation. This is typically the case of internal parts covered by container parts or hulls. The viewer should implement occlusion management techniques to allow access and discovery of all product parts.
- Some information of the product may be linked to a given part or even to specific points or areas

on the 3D geometry, requiring some mechanism to link the information with its correct place on the geometry.

- Navigation around the 3D model of the product may depend on the geometrical characteristics of the product. Generic navigation metaphors may not be suitable for particular geometries, such as elongated objects, where the three main axes are not equally relevant.

This paper presents the design and implementation of HOM3R (a Hierarchical prOduct Model 3d vieweR), with the following goals:

- Providing solutions for the aforementioned challenges (hierarchical selection, navigation, occlusion management and information linking).
- Optimizing navigation for visualization of elongated objects with a certain axial symmetry. Navigation should adapt to the shape of the object to be appropriate also for objects with different degrees of sphericity or without a privileged axis, without the need to switch between different navigation modes.
- Allowing the integration of the viewer within any existing or new web application. This implies building the viewer as a module compatible with most web browsers and providing a flexible API for easy inter-face through JavaScript.
- Allowing 3D interaction with the object using regular desktop devices like keyboard and mouse
- Following a modular architecture with easy expandability, where modules can communicate through a centralized message server.

The viewer is currently being used in the context of the EU funded project Use-it-Wisely. The work developed in this context was described in [GCG_15]. The 3D viewer shown in [GCG_15] has been replaced by HOM3R, presented in this paper. HOM3R is implemented as a WebGL script which is decoupled from the web application and can be integrated in other applications. The 3D models can now be downloaded (with geometry, hierarchical and assembly information) from any external URL.

HOM3R also offers completely new functionality, such as 3D labelling and multi-level selection.

The rest of the paper is organized as follows. Section 2 presents a brief summary of the state-of-the-art in 3D viewing techniques and technologies that are related to the present work. In Section 3, the main features of HOM3R are presented. Section 4 presents its modular architecture, with emphasis on its interface to external web applications. Section 5 presents the results of an evaluation of the tool that was carried out at the end of the Project and Section 6 concludes with a discussion including future work plans.

2. Related work

The challenges presented in the Introduction for occlusion management, navigation, selection and linking information to 3D geometry have been extensively addressed in the fields of 3D interaction [BKLP04] and visualization [ALB11][VG05]. Typical approaches for occlusion management include the use of transparency [Bur11], cutaways [LACS07] and exploded views [TKS10] [LACS08]. HOM3R extends these techniques to interactive web-based visualization, with a focus on providing easy access to all product parts for selection while trying to keep the main structure of the whole product. We provide transparency adapted to viewpoint, which resembles the work of [RH04], but instead of rendering fixed transparent volumes, we chose the significant areas of the product to be rendered transparent to help to understand the global product structure. Our 3D viewer implements interactive exploded views following the symmetric and focused layouts of [KTS09], and blocking relations are based on product assembly information rather than being computed automatically, as in [LACS08]. Although HOM3R does not implement cutaway views, we provide mechanisms to hide and isolate at part-level rather than cutting arbitrary planes or volumes.

Techniques for linking information to 3D geometry are typically based on attaching text labels to the 3D models [AHS05]. We implemented the hedgehog labelling technique for external labels described in [TKGS14], allowing for 3DoF movement of labels, but rather than automating all DoFs to avoid

stacking, we allow the user to move manually the label in the X/Y plane to customize the layout.

There exist many approaches to navigation around 3D objects. Many 3D viewers provide generic navigation metaphors, not taking into account the shapes of the objects, which may imply varying distance to the object while navigating, looking away from the object or having inaccessible viewpoints [Han97]. To alleviate these issues, some viewers provide a set of different navigation modes that can be switched manually [Ado16]. We follow an exocentric navigation approach that adapts both to the shape of elongated or almost spherical objects (or without privileged axis) without the need to switch between modes. Our approach follows the line of [KKS_05], but instead of adapting camera movements to convex features of the object, we approximate its global shape with an ellipsoid built around the bounding box ex-tents, thus obtaining a smoother navigation.

There is a large amount of previous work on the design and implementation of 3D viewing tools. The characteristics of each 3D viewer differ mainly based on the goal and context of each application. There exist many viewers for generic 3D objects, which are not product models in the sense that there is no product information accompanying the 3D geometry [Ble16][Ske16]. Some of these viewers provide some of the interaction techniques implemented in our 3D viewer [Nwn15], but they lack hierarchical selection and manipulation and cannot interface with external web applications.

CAD software for engineering often includes powerful 3D viewers for models of engineering products. However, the focus is put in the creation and edition of products or in tasks such as simulation or generation of engineering lay-outs, rather than visualization to access the information of the product model and interfacing with external applications. One of the most remarkable examples is SolidWorks [Sol16], which provides many of the interaction techniques implemented in our 3D viewer, including transparency (as a rendering characteristic of the parts instead of a technique for automatic handling of occlusion management), exploded views or 3D labelling (without automatic placement depending on viewpoint, which makes them more appropriate for

the generation of 2D layouts than for interactive visualization).

A 3D viewer with goals similar to ours is TeamPlatform Online 3D Viewer [Tea16]. This viewer works with complex hierarchical product models and implements many occlusion management techniques, including exploded and cutaway views and transparency, providing an API for interfacing with external applications. However, transparency is not adapted to the viewpoint and exploded views are always symmetric, without the possibility of exploding independent parts or groups of parts. Although it allows for multi-level selection over a textual tree structure, in practice, this selection serves only for toggling between hide/show for each part. Labelling is only possible in the form of vector graphics drawn over a 2D screenshot of the 3D model. Navigation follows spherical coordinates around a selectable centre and cannot be adapted to the particular shape of long objects.

3. Supported features

This section presents the interaction strategies implemented in HOM3R to address the challenges presented in Section 1: (1) hierarchical selection, (2) navigation around the product, (3) occlusion management, (4) 3D labelling. These interaction techniques will allow the user to discover and access every part of the complex product to have a global view of the product, understand the spatial relationship between the different parts and obtain relevant information to analyse in detail a specific part.

Each of the elements of the geometry being rendered has an associated visual state machine, representing its visualization state (See Figure 1). The mechanisms for selection and occlusion management, which will be described in this section, are in charge of updating the state of every element.

3.1. Hierarchical selection

Selection is a first step for other manipulation tasks, such as focusing on a specific part, visualizing a selected part without occlusions or extracting certain information regarding the chosen part, as the following sections will describe. The implemented

selection algorithm allows for the selection of a product part at any hierarchy level, which results in the automatic selection of all children nodes of the current selection.

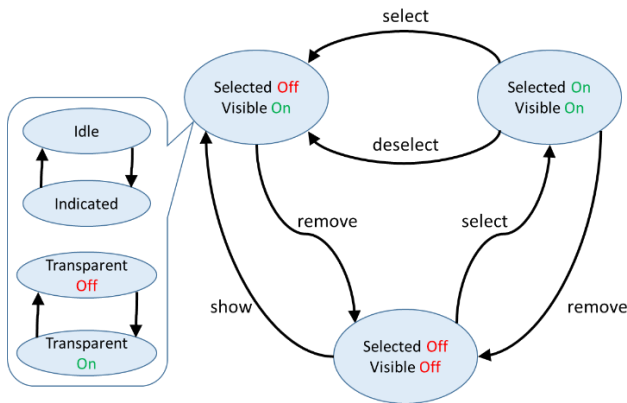


Figure 1: State machine representing the visualization state of one geometry element

The selection algorithm is based on the product model supported by HOM3R, which is shown in Figure 2 using UML. This product model, which is based on [Rey13], consists of a tree structure, where each node of the tree represents a part or an area within one part. Geometry information is linked only to the areas so that selection over a higher-level node implies selection of the geometry of all children areas. Having the 3D transformation data for each area allows to reuse the same 3D mesh files for geometries that are repeated frequently within the product, such as screws. Each node of the tree has associated data, including assembly information (used in the exploded views, as will be explained in Section 3.3) or other data relevant for each application.

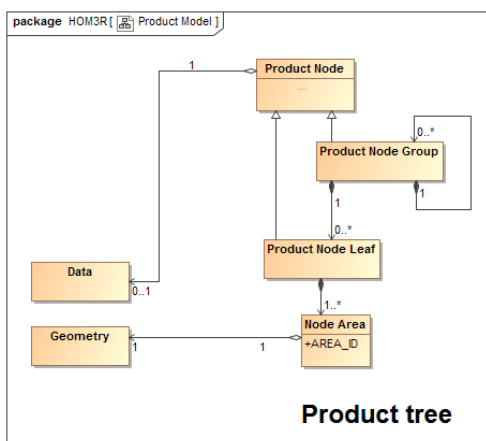


Figure 2: Simplified product model class diagram

Selection over the product tree is based on mouse ray-casting. This 3D interaction feature allows the user to highlight a specific node (part or area) or a set of nodes (by a mouse click).

While ray-casting is used for selection from the built-in user interface of HOM3R, the HOM3R API interface (described in detail in section 4.2) allows also for part selection at any level of the hierarchy, using any colour to highlight selected parts or areas. This allows the applications to setup colour codes to transmit information to the user. Figure 3 shows an example of use of this feature.

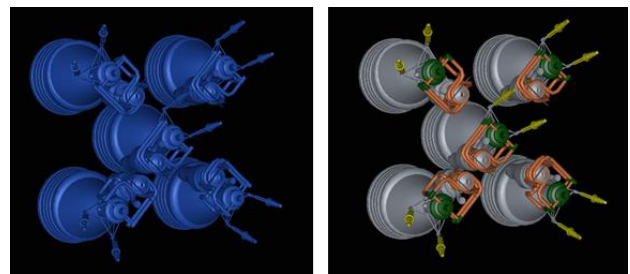


Figure 3: Hierarchical selection. Left: selection at one product node level (rocket engines). Right: selection at area level. In this case, different colours could indicate different operation temperatures

3.2. Navigation

The algorithms implemented to navigate around the product allow the user to manipulate the virtual camera, using the mouse, with a restricted set of trajectories. Currently, the tool supports two different strategies for camera movements: cylindrical and ellipsoidal.

The cylindrical navigation system has 3DoF, allowing the user to (a) translate the point of view along a path parallel to the symmetry axis, (b) orbit the camera around the product and (c) zoom in and out with respect to the object, as shown in Figure 4. The ellipsoidal navigation system also has 3DoF (See Figure 5). In this case, the camera translation movements follow a path marked by the ellipses resulting from horizontal cuts in the ellipsoid (a), with the camera heading towards the product. Furthermore, as in the cylindrical navigation, the ellipsoidal navigation allows the camera to circularly orbit around the axis (b) and zoom in and out with respect to the object (c).

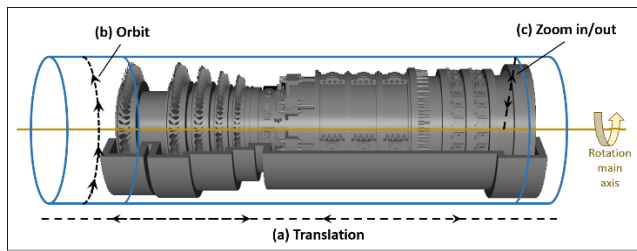


Figure 4: Degrees of freedom in the cylindrical navigation system

The presented navigation systems are especially suitable for elongated products with a certain axial symmetry, e.g., a turbine rotor or a rocket. In this way, the shape of the navigation trajectory is adjusted to the bounding box of the product. Spherical coordinates are a particular case of the ellipsoidal coordinates, which means that this navigation system is also suitable for products without a privileged axis.

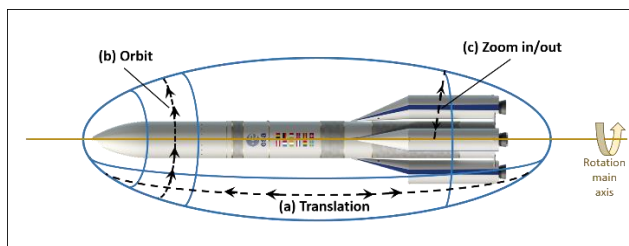


Figure 5: Degrees of freedom in the ellipsoidal navigation system. Photo: ESA–David Ducros, 2016

Additionally, HOM3R offers pan navigation, which allows the user to drag the virtual camera in a 2D plane, moving the centre of the cylindrical or ellipsoidal coordinate system.

Another notable feature supported by the 3D viewer is smooth navigation to minimise user surprise and maximise user understanding of changes to the 3D scene. The HOM3R interface is user centred; the supported features have been designed with the main purpose of improving usability. In this way, the tool implements a navigation algorithm with gradual transitions, where smooth camera movements are implemented to provide the user with a feedback of his/her actions and, therefore, make him/her aware of what's happening at every moment, avoiding abrupt transitions, which can cause the user to feel lost. This smooth mechanism is used when the camera moves to a predefined position, when switching between navigation strategies and when focusing on a specific part.

3.3. Occlusion management

To manage occlusions between different parts of a complex product, HOM3R implements four different techniques: parts removal, part isolation, adaptive transparency and exploded views.

The parts removal mechanism allows users to select certain parts of the model to be removed from visualization, as shown in Figure 6. The removal of selected parts allows visualization of other parts in which the user might be interested which were occluded by the removed parts. While this feature allows for manual control of part occlusions, it requires the user to know in advance which parts are occluding the parts of interest.

With the isolation feature, the user selects a part and all other parts of the product model (occluding or not) are automatically removed and the point of view is automatically changed so that the selected part is centred. The isolation feature is useful to focus the attention of the user on a single part for a specific task, but at the cost of losing (temporarily) all spatial relations between the isolated part and the rest of the product model. To overcome the limitations of parts removal and isolation, the mechanisms of adaptive transparency and exploded view have been implemented.

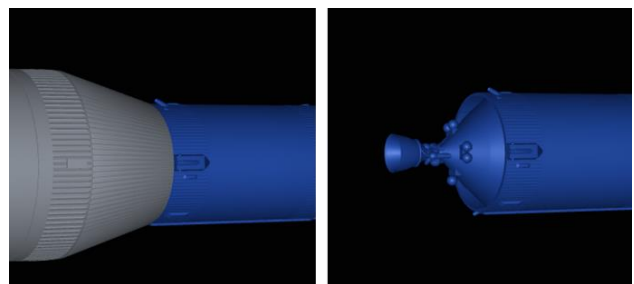


Figure 6: Example of parts removal. Left: the selected part, in blue, is not fully visible. Right: the user has manually removed the occluding parts in order to visualize the selection completely

The adaptive transparency algorithm has been implemented to discover a selected part that is totally or partially occluded by other parts from a specific viewpoint by making transparent a set of areas of the occluding parts, as shown in Figure 7. This feature offers the possibility to visualize a hidden part without losing the spatial relationship between the different parts of the product. The mechanism uses ray-casting in order to remove automatically the

parts which are occluding, depending on the view point at any time. Occlusions are detected by shooting a set of rays through the scene, from the camera to selected points in the bounding-box of the occluded object.

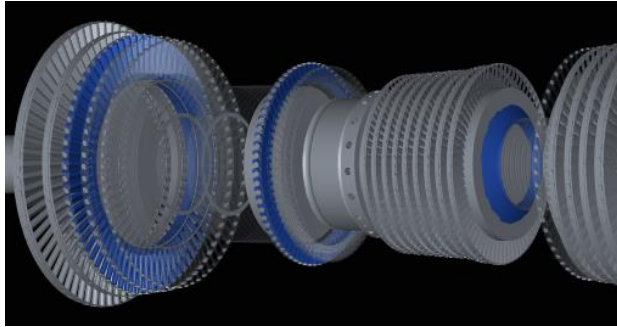


Figure 7: Example of the adaptive transparency feature. The selected parts are marked in blue and all areas occluding the selection are rendered with partial transparency

The exploded view feature is based on an explosion algorithm that shows the main parts of the product separated by a specific distance, following the assembly information to preserve spatial relations within the product. An example is shown in Figure 8. The exploded view can be symmetric (separating every component of the object) or focused (separating just a selected part). With this feature, the user can discover and access most parts of the product without the need to make transparent or remove other parts.

Additionally, the tool offers the possibility to remove a selected part to access a hidden part.

In the same way as in the navigation features, these occlusion techniques follow the principle of gradual transition to provide smooth feedback to the user. In this way, the explosion will be carried out with smooth movements of the 3D models, and the deletion and transparency of parts are made with a gradual fade out.

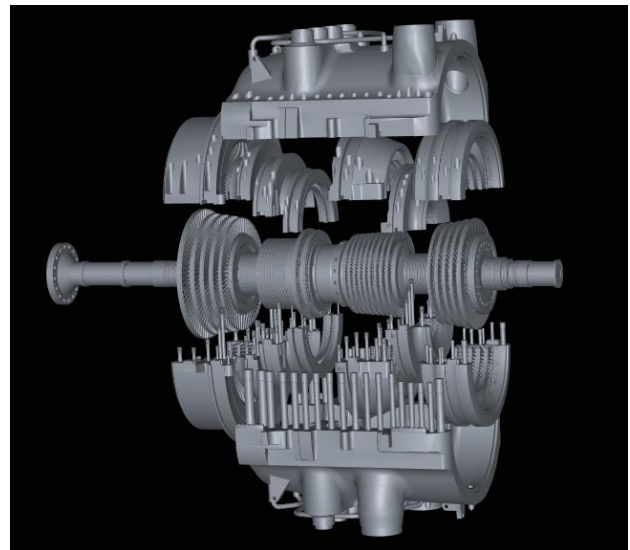


Figure 8: Example of a symmetric exploded view layout

3.4. 3D labelling

Labels provide information regarding the description or identification of a specific product part or any additional information present in the product model that is linked to the 3D geometry (see Figure 9). Labels behave as 2D objects that are placed in the 3D environment and are connected to the product by a leader-line, which always faces towards the camera. A physics-based algorithm [TKGS14] has been implemented to maximize visibility of the labels while changing the viewpoint. The algorithm imposes soft constraints in order to avoid leader-line crossing, to avoid occlusion between labels or between labels and the product, and to keep labels inside the viewport. Labels can be automatically positioned in a centred position on the surface of the parts or located at any specific point or area of one part. Once placed, labels can be manually moved in screen space to customize the view.

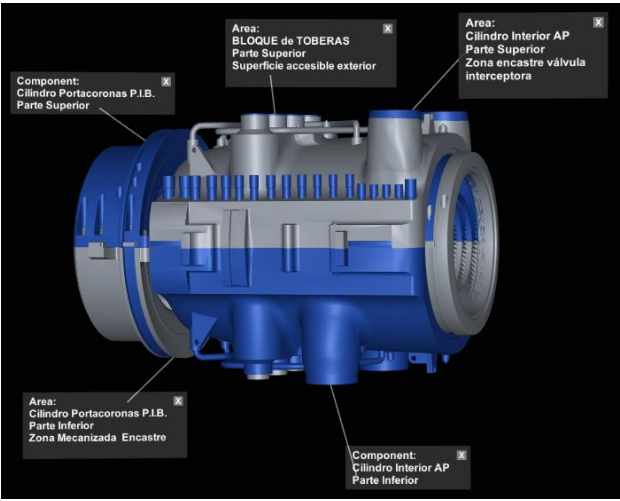


Figure 9: Example of the 3D labelling feature. Labels move while navigating, so that the layout maximizes visibility of the labels from any viewpoint

4. Architecture and interfaces

HOM3R is designed to work in a web browser. Its architecture intends to simplify its integration into any web-based application while ensuring compatibility with most web browsers. To achieve both goals, HOM3R is divided into two subsystems based on the different natures of their purposes: an API module and a 3D module (See Figure 10). The 3D module provides the interactive 3D rendering of product geometry, manages user interaction and presents visual information, received from the web application, over the 3D geometry. This module was developed using the platform Unity3D in C# and cross-compiled by Unity3D into JavaScript that uses the WebGL rendering API.

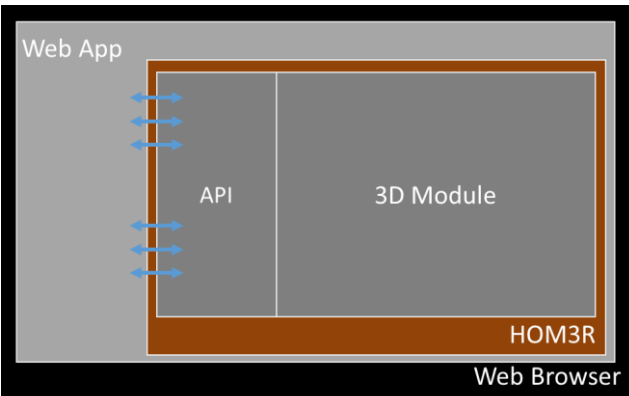


Figure 10: High-level architecture block diagram of HOM3R

The API module provides a JavaScript programming interface between the application and the embeddable 3D module. More details follow in the remainder of this section.

4.1. Architecture of the 3D Module

Section 3 described the core functions of HOM3R, aimed at facilitating the visualization and navigation of complex hierarchical product models. Additionally HOM3R offers a set of supporting functions, such as product model loading via URLs, GUI event management or callback registration for communication with the rest of the web application. The logic view of the architecture (Figure 11) reflects both the core and the supporting functions of HOM3R.

The 3D module is organized into several submodules around a central core. The mission of this central core is to receive and distribute notifications of all events happening in the system or received from the web application, delegating execution to the rest of submodules.

The main submodules include those with supporting functions, such as the Mouse/Keyboard Manager, the Product Loader, or the I/O module, which manages the interface with the rest of the web application; and those responsible for the core functionality of the viewer, such as the manager of the hierarchical selection (Section 3.1), the Focus and Navigation managers (Section 3.2), the Occlusion managers (Remove, Transparency, Explosion: Section 3.3), and the 3D labelling manager (Section 3.4).

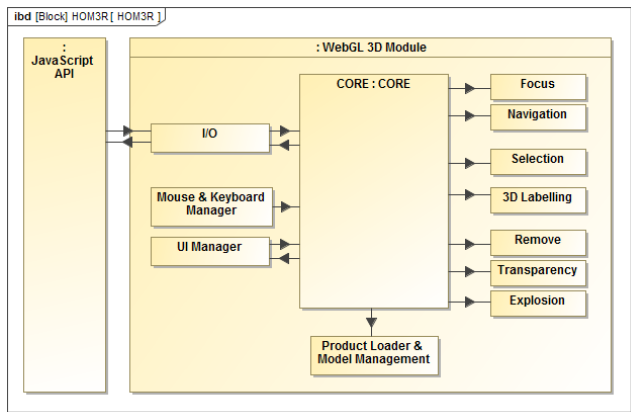


Figure 11: Functional block diagram of HOM3R

Figure 12 shows how the core module receives events coming from the I/O interface and user

interface and makes calls to the rest of the submodules. Events arrive first to the core, are classified and transformed by the core into calls to the rest of the submodules, in order to react to the event in an appropriate manner.

The relation between the core and the submodules that generate events is facilitated by a design that follows the Command Pattern [GHJ95], in such way that the core is only in charge of receiving commands and delegating their execution. This design makes this module, and the system in general, easily configurable and scalable.

4.2. API

This section presents the API module that HOM3R provides to exchange messages and data with a web application. This API consists of a set of JavaScript files whose goals are to facilitate the integration and the communication of HOM3R with the rest of the web application.

Communication from the web application to HOM3R is carried out via different methods that are defined in the interface. These methods allow the

application to, for instance, load the product model from a specific URL, select or deselect a part of the product model, add/remove a 3D label with information over a part of the product model or activate/deactivate any of the occlusion management mechanisms.

Communication from HOM3R to the web application is managed via a subscription mechanism consisting of a single callback function. This mechanism was chosen to make integration with any web application as simple as possible. Therefore, the web application developer has the freedom to determine the name and how to implement this function. However, its definition should accept two input parameters with the pattern *callbackFunction (message, value)*.

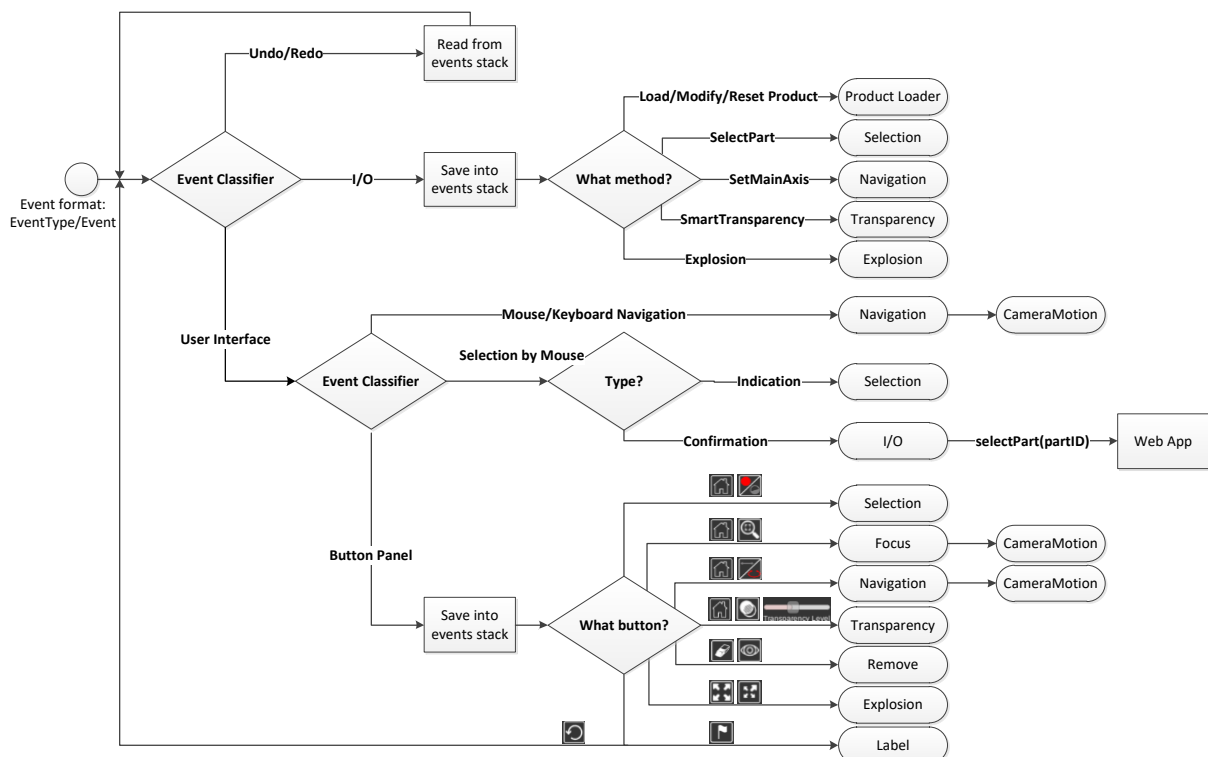


Figure 12: Core Activity diagram. After receiving an event, this is classified and executed with the help of the different submodules. The diagram only reflects the main actions in each case

Through these parameters, HOM3R will inform the application of every event occurring during the user interaction with the 3D module.

The type of messages and possible values, as well as other details of the API, are described in detail on-line [HAPI17].

Figure 13 describes an example of initialization and loading of a product model scenario. In this scenario, while HOM3R is been loaded and embedded into the web application, the application should register its callback function within the API. HOM3R will use this callback to indicate when it is ready and waiting for commands. After that, the application should ask HOM3R to load the product model, including geometry and attached data. HOM3R will inform the application with the result after loading.

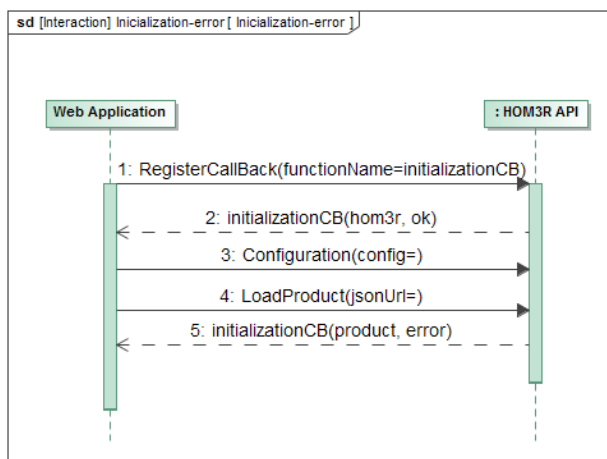


Figure 13: Sequence diagram of an initialization and load of models scenario with errors

As shown in the figure, the application can send configuration parameters at any time.

In addition, Figure 14 shows an example of an interaction scenario, where the application requests loading the product model and reacts to the user interaction notified by HOM3R. Please refer to the online documentation for a full understanding of the interface methods.

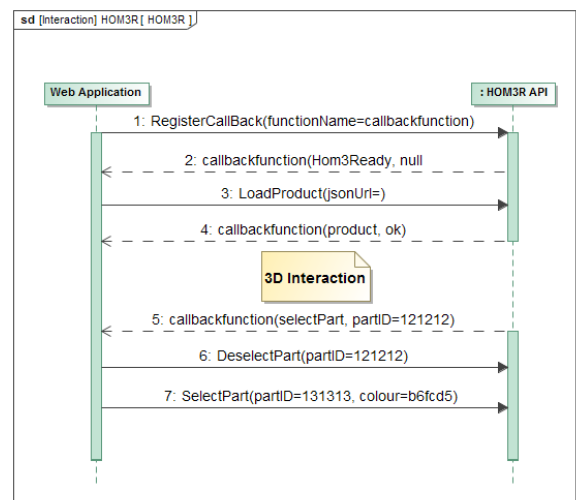


Figure 14: Sequence diagram showing an example of the interface inter-action. The web application registers a callback function and asks HOM3R to start the product model load. During interaction, HOM3R uses the callback to inform the web application that a part has been selected by the user and the web application requests HOM3R to unselect that part and select a new one

5. Evaluation

HOM3R has been used in a web-based application developed in the context of the EU funded project Use-it-Wisely. This 3D interactive application is used to manage the results of the inspections carried out in a turbine of a power plant. The end user of the application was Tecnatom, a company involved in the project [Tec17]. Tecnatom is an engineering service company responsible, among other tasks, of performing inspections of turbines in fossil and nuclear power plants.

The Use-it-Wisely application integrates HOM3R to allow users to interact with a 3D model of the turbine, to navigate around the model, access and discover specific turbine parts, and to visualize information linked to these parts. At the end of the project, an evaluation with a group of experts was carried out. The main goal was to evaluate the usability and the acceptance of the tool by potential end users.

Ten participants were selected from eight different departments of Tecnatom (from engineering to business development), including three senior engineers, four department directors, and three

engineers directly involved in the turbine inspection process. These engineers participated in an evaluation session where they tried the tool.

All participants in this session showed great involvement, with fruitful comments. The feedback received was very positive, emphasizing the usefulness of the application and how intuitive and friendly the tool was. Participants agreed that the tool can be very useful in collaborative environments, to share knowledge among different users. Also, they think that the viewer can add a great value to training applications.

Regarding the features of HOM3R, participants highlighted the intuitive and easy navigation metaphor. Constrained navigation, which ensures that the product is always inside the user field of view, was appreciated when mouse was the only input device. Some participants remarked that, with this kind of navigation they never had the feeling of being spatially lost. The occlusion management techniques for access and discovery of hidden parts, such as the exploded view or the adaptive transparency, were found to be very useful. The Participants remarked the fact that these techniques allow them to visualize any turbine part without losing the spatial relations between parts. Successful feedback was also received regarding the usability of the HOM3R interface. Participants found the interface buttons and user interaction very intuitive and they affirmed that it is easy to get familiar with the tool without the need of a long learning curve.

6. Conclusions

This paper presents HOM3R, a novel 3D viewer for complex hierarchical product models.

HOM3R is now a stable piece of software, ready to be used in experiments aimed at evaluating the hypotheses that guided its development: user-centered interaction mechanisms that deal with occlusion management, navigation and presentation of information linked to 3D geometry.

The current version of HOM3R, including full documentation of the API for integration in web applications, can be freely downloaded at <http://proyectos.diana.uma.es/hom3r/>. Although the current version is fully functional, HOM3R is in

continuous development and new features are expected to be released soon. Currently, 3D meshes are encapsulated in Unity asset bundles, which implies the need to use Unity (the Personal edition is sufficient) to generate the bundle for a new set of 3D meshes. We include an editor script for generating the bundles in Unity with only one click, but our near-future work plans include the possibility of directly loading 3D meshes from standard file formats such as OBJ, to avoid the need to use Unity or any other proprietary application to change or edit geometry.

As future work we will look at improving the navigation system, in order to add a first-person free camera navigation and to compare it with the constrained navigation system presented in this paper, with the goal of facilitating experiments that study the advantages of each navigation metaphor. We are also considering adding other occlusion management techniques such as cutaway views and point-of-view-dependent exploded views.

7. Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement No. 609027 (Project Use-it-wisely). We would like to thank Tecnatom S.A. and the Human Factors Research Group of University of Nottingham for their collaboration.

References

- [Ado16] Adobe Systems Inc., *Displaying 3D Models in PDFs*. [https:// helpx.adobe.com/acrobat/using/ displaying-3d-models-pdfs.html](https://helpx.adobe.com/acrobat/using/displaying-3d-models-pdfs.html). Retrieved September 2016.
- [AHS05] Ali K., Hartmann K. and Strothotte T, *Label Layout for Interactive 3D Illustrations*. Journal of WSCG, 13(1), 1-8. doi:10.1016/j.cag.2010.05.002
- [ALB11] Agrawala M., Li W. and Berthouzoz F, *Design Principles for Visual*

- Communication. Communications of the ACM, 54(4), 60.doi: 10.1145/1924421.1924439
- [BKLP04] Bowman D., Kruijff E., Laviola J., and Populyrev I., *3D User Interfaces, Theory and Practice*. Addison-Wesley, 2004.
- [Ble16] Blender: Home of the Blender Project. <https://www.blender.org>. Retrieved September 2016.
- [Bur11] Burns M., *Efficient and Comprehensible Visualization of Complex 3D Scenes*. PhD Thesis, Department of Computer Science of Princeton University, 2011.
- [GHJ95] Gamma, E., Helm, R., Johnson, R. and Vlissides J.: *Design patterns: elements of reusable object-oriented software*. Addison Wesley, 1995.
- [Bur11] Burns M.: *Efficient and Comprehensible Visualization of Complex 3D Scenes*. PhD Thesis, Department of Computer Science of Princeton University, 2011.
- [GCG_15] Gonzalez-Toledo D., Cuevas-Rodríguez M., Garre C., Molina-Tanco L. and Reyes-Lecuona A., *A Tool for Collaborative Decision Making on Service Information Linked to 3D Geometry of Complex Hierarchical Products*. In Proc. EuroVR, 2015.
- [Han97] Hand C., *A Survey of 3D Interaction Techniques*. Computer Graphics Forum, 16(5), pp. 269-281.doi: 10.1111/1467-8659.0019
- [HAPI17] hom3r API Documentation. (2017). 1st ed. Available at: http://proyectos.diana.uma.es/hom3r/img/how_integrate_hom3r/hom3rAPI_english.pdf [Accessed 31 Mar. 2017]
- [KKS_05] Khan A., Komalo B., Stam J., Fitz-Maurice G. and Kurtenbach G., *HoverCam, Interactive 3D Navigation for Proximal Object Inspection*. In Proc. Symposium on Interactive 3D Graphics and Games, 1(212), 73-80, 2005.doi:10.1145/1053427.1053439
- [KTS09] Kalkofen D., Tatzgern M. and Schmalstieg D.: *Explosion diagrams in augmented reality*. In Proc. 2009 IEEE Virtual Reality Conference (pp. 71–78).doi:10.1109/VR.2009.4811001
- [LACS07] Li W., Ritter L., Agrawala M., Cur-Less B. and Salesin D., *Interactive Cutaway Illustrations of Complex 3D Models*. ACM Transactions on Graphics, 26(3), 31.doi:10.1145/1276377.1276416
- [LACS08] Li W., Agrawala M., Curless B. and Salesin D.: *Automated Generation of Interactive 3D Exploded View Diagrams*. ACM Transactions on Graphics, 27(3), 1.doi:10.1145/1360612.1360700
- [Nwn15] Biesiadowski A. and Borges A., *NWN Java 3D Model Viewer*. At The NeverWinter Vault: <https://neverwintervault.org/project/nwn1/other/nwn-java-3d-model-viewer>. Retrieved September 2016.
- [Rey13] Rey J., *Modeling with VSEE, Definition of the Guidelines and Exploitation of the Models*. YGT Final Report, Virtual Spacecraft Design Project. European Space Agency, 2013.
- [RH04] Ropinski T. and Hinrichs K.: *Real-Time Rendering of 3D Magic Lenses having Arbitrary Convex Shapes*. In Proc. 12th International Conference in Central Europe on Computer Graphics, Visualization, 2004.
- [Ske16] Sketchfab: Your 3D on the Web. <https://sketchfab.com>. Retrieved September 2016.

- [Sol16] Dassault Systems: SolidWorks Official Site. <http://www.solidworks.com>. Retrieved September 2016.
- [Tea16] 3D Systems Inc.: TeamPlatform. <http://www.teamplatform.com>. Retrieved September 2016.
- [Tec17] Tecnatom S.A. [Online]. Available: <http://www.tecnatom.es/es/>. [Accessed: 31-Mar-2017].
- [TKGS14] Tatzgern M., Kalkofen D., Grasset R. and Schmalstieg D.: *Hedgehog Labeling, View Management Techniques for External Labels in 3D Space*. In Proc. IEEE Virtual Reality, 27-32.doi:10.1109/VR.2014.6802046
- [TKS10] Tatzgern M., Kalkofen D. and Schmalstieg D.: *Compact Explosion Diagrams*. In Proc. 8th International Symposium on Non-Photorealistic Animation and Rendering, p.17, 2010.doi:10.1145/1809939.1809942
- [VG05] Viola I. and Gröller M.: *Smart Visibility in Visualization*. In Proc. 1st Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging, 209-216, 2005