VRML authoring in the context of industrial applications

C. Elcacho, A. Schäfer, R. Dörner, V. Luckas
Fraunhofer Institute for Computer Graphics
Rundeturmstr. 6
D-64283 Darmstadt
Germany
Phone +49-6151-155-638
Fax +49-6151-155-139
Email: {elcacho, aschaefe, doerner, luckas}@igd.fhg.de

Abstract

VRML is playing a more and more important part in the context of industrial Intranet applications for the 3D visualization of product and production data. New industrial areas of application are opening up and create new requirements for VRML authoring tools. In this paper we want to show the use of VRML in the context of industrial applications based on several examples. Furthermore we will deduce additional authoring requirements that must be met by VRML authoring tools in order to suit the discussed industrial application examples and suggest a solution concept. We show the realization of this approach on the basis of a new set of VRML authoring tools and make a point of how the special authoring concepts described can be used to solve the general authoring problem.

Keywords

VRML, 3D animation, production planning, product visualization, VRML authoring

1 INTRODUCTION

The increasing use of VRML for 3D visualization, animation and interaction in WWW-based applications from many different areas of interest creates demand for new specialized VRML authoring tools.

With VRML authoring we mean the support given to users when editing a VRML scene, such that they can incorporate all features of VRML relevant in the specific application context into the scene they are generating. These VRML features include the following general aspects of VRML authoring:

Composing scenes
- opening/importing VRML scene descriptions;
- opening/importing VRML objects;
- positioning of VRML objects inside the scene.

Modifying objects
- defining properties (color, reflectivity, material properties);
- defining behavior (interaction, hyperlinks, reaction of an object to an interaction (animation, audio, video, etc.));
- defining keyframe animations (low-level);
- defining complex patterns of behavior (e.g. Java programming).

Editing scene parameters
- illumination, camera, views.

The 3D modelling of single scene objects does not immediately belong to the tasks a special purpose VRML authoring tool is expected to perform and is not further considered in this context. For 3D modelling a variety of geometric modelling tools are available on the market (Lemay, 1996).

Requirements that result from special purpose applications and go beyond the above mentioned general aspects of VRML authoring are of high interest and will be considered here.

Aspects of suitable authoring tools for special applications are:
- optimized support for the specific application requirements;
- a high degree of automation in the generation of the desired application;
- support of the functionality that is required beyond VRML (e.g. database access, embedding in HTML context).

Special VRML applications that require specially designed authoring tools and support can be found, for instance, in industrial application scenarios (Ressler, 1997).

This paper is structured as follows: section 2 describes special industrial application scenarios and deduces requirements for application authoring tools. In section 3 we show that VRML can be used in meeting the requirements deduced in section 2. Section 4 gives an overview of current VRML authoring concepts and tools and shows deficiencies; section 5 describes the concepts that are the basis of our implementation. In section 6 we describe the tools we have developed. In section 7 a conclusion and a view on further work planned in this context is presented.

2 APPLICATION BACKGROUND

2.1 Application scenarios

In an industrial application context three-dimensional representations are used in many areas: in production management (e.g. process visualization), in development and engineering (e.g. construction), or in sales and maintenance (e.g. visualization of assembly). Three-dimensional data is used for many purposes reaching from planning, surveillance, optimization, to support of communication. Moreover, 3D data is used for internal and external presentations, marketing and public relations. In industrial teaching and training applications 3D data can also be used. We want to illustrate the spectrum of the industrial application context by describing some application scenarios.

One application scenario is product information. A consistent and intuitive depiction of the product data and an integrative access to it is important for a complex company with world wide distributed and differently organized branches. A suitable 3D representation of the product information is an exploded view (see Fig. 1) which, when equipped with hyperlinks, can serve as a navigation metaphor for additional product data, binary (e.g. CAD data) as well as textual (e.g. bills of material).

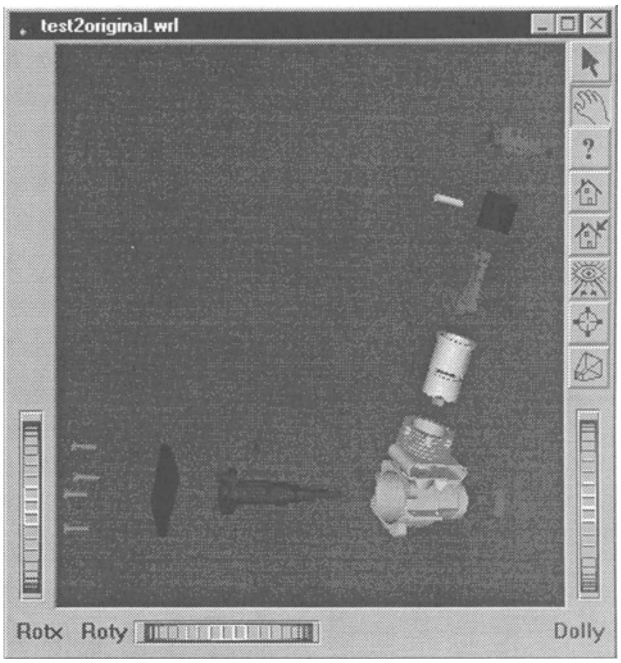


Figure 1: Example of an exploded view.

Another industrial application scenario are logistic simulations often used as a tool for planning or optimization purposes or for decision making (see Figure 2). An animated 3D visualization of simulation data improves realism and makes the interpretation of the simulation results possible in an intuitive way. Thus, conclusions taken from the simulation can be communicated more easily.

Product schooling performed for practising of operations and workflow in assembly and maintenance is another scenario. Here a depiction of assembly or disassembly in the form of 3D animations is suitable. Similar animations can also be given to customers serving as directions of use.

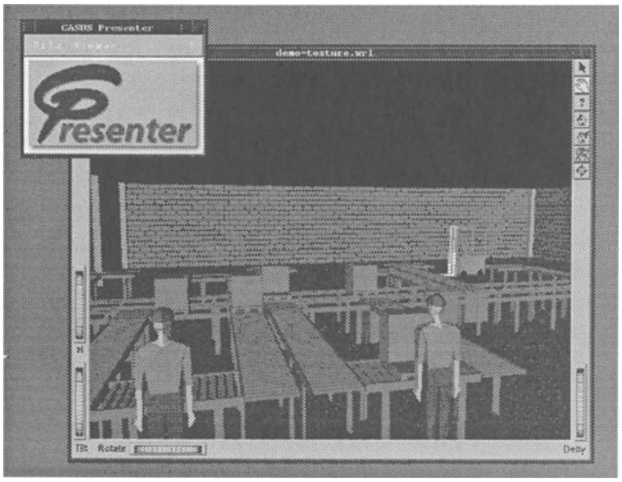


Figure 2: VRML Visualization of a Logistics Simulation.

2.2 Requirements

The broad spectrum of industrial applications implies many requirements to an authoring tool for 3D content creation. The support of integration in hypermedia structures, insertion of annotations and definition of interaction are important demands for which authoring has to provide solutions. As described in the second sample scenario, simulation or sensor data should serve as an input for behavior specification of the 3D representation (e.g. changes in sensor data should trigger animations in process visualization). Besides, the data formats used in industry (especially CAD data) have to be supported by the authoring tools.

In an industrial context a cost effective, fast and automated creation of 3D visualizations and animations is demanded. The creation process should fit well into the existing workflow. Different working forms, especially co-operative work via networks, should be supported by the authoring tool. Also different user groups with varying knowledge of 3D representations and computer usage should be considered.

3 POTENTIAL OF USING VRML

The requirements that arise in an industrial context such as 3D-visualisation of static and dynamic data, the use of interaction to access additional textual or non-textual information as well as the general need for global access to the information material can be solved in an Intranet context by using WWW technologies and an adequate 3D data format such as VRML. Among the available 3D data formats VRML offers several advantages, making it the primary choice over other alternatives.

These advantages include general **functional advantages** such as the ability to represent both static and dynamic 3D data, 3D animations, interactivity, level of detail, sound and video. Moreover VRML offers the advantage of being a **publicly available standardized format** of world wide acceptance for data exchange, that is supported at least as an export data format by a variety of modelling and authoring tools including especially CAD systems that are widely used in industry (VRML, 1997). For some applications this allows the direct translation of legacy data to VRML for special purpose applications.

Current formats used for 3D-animations such as OFF (Rost, 1989) generally do not support interaction or sound and are restricted to a very narrow area of application. VRML is designed for use over large area networks world wide and offers a compact data-format that is **space efficient**. This is also an advantage when storing a large number of VRML models in a database.

Moreover, a variety of VRML viewers are available on most platforms, most of them free of charge, such that any VRML application that is compliant to the specification can be run on any system and as such is **platform independent** (SDSC, 1995), (Schäfer, 1997).

The new features of VRML as opposed to traditional 3D formats enable the generation of innovative applications and allow the 3D graphical solution of problems that would not have been possible with other formats to date.

The following section gives an overview of the state of the art in VRML authoring, describes current authoring concepts and where they are not sufficient for the generation of special purpose industrial applications. In chapter 5 we suggest authoring concepts that are suitable for the efficient and automated generation of special industrial applications.

4 STATE OF THE ART

VRML authoring has many similarities to authoring of 3D animations for film and advertising, with some additional requirements. Therefore, many of the tools used for VRML authoring are similar to those used in 3D animation. These include

- 3D geometry modelers
- material and texture editors
- scene editors and 3D animation systems
- converters and export filters
- ASCII file editors.

In contrast to 3D animation, real-time 3D graphics and VRML in particular have several additional requirements. Scene complexity is limited by the necessity for real-time presentation on today's hardware, and must therefore be given special attention. Additionally, VRML supports user interaction with the scene, which is not possible in 3D animation. Furthermore, the programmability of VRML in Java and other languages requires additional programming tools such as compilers and debuggers. Because of these different requirements, additional tools for VRML authoring are used:

- tools for the reduction of scene complexity (esp. polygon reduction)
- Java programming tools
- special VRML editors and authoring tools.

Typically, not one tool is used for the whole process of creating VRML content, but different tools are combined to produce the desired results. A common approach is to use 3D geometry modelers to create the 3D objects (including material properties and textures), and a VRML authoring tool for scene composition, interaction and animation. Examples for these authoring tools are Cosmo Worlds by Silicon Graphics, Inc., V-Realm Builder by Ligos Technology, or TrueSpace3 by Caligari (SDSC, 1995). Very often however, even today large parts of the scene composition, animation and definition of interaction are still done by hand, i.e. with an ASCII file editor. Manual work is also required when defining programmed behavior for VRML, e.g. in Java.

Despite the increasing availability of authoring systems and supporting tools, VRML content creation is still a very labor intensive and time consuming process. The manual tasks such as editing of the VRML scene file and especially Java programming are often unavoidable even with today's tools, and require special knowledge, such as programming skills. They are therefore often not done by the same people creating the 3D models, which makes VRML content creation even more expensive.

5 DEDICATED AUTHORING CONCEPTS

In this context we want to concentrate on the demands of authoring in an industrial context. The authoring concepts we propose do not address aspects of human computer interaction such as graphical user interface design of the authoring environment or suitable interaction metaphors. These issues are general concepts that are applicable to any graphical interaction tool and are discussed abundantly in the literature (Preece, 1994).

5.1 The automation of VRML content creation

In industrial application scenarios large amounts of data are constantly generated and processed; even larger amounts of non-VRML legacy data usually exist. For easy handling of these large amounts of information, we propose to provide, as far as possible, automatic processing of the data and automated generation of VRML application content.

5.2 The concept of animation elements

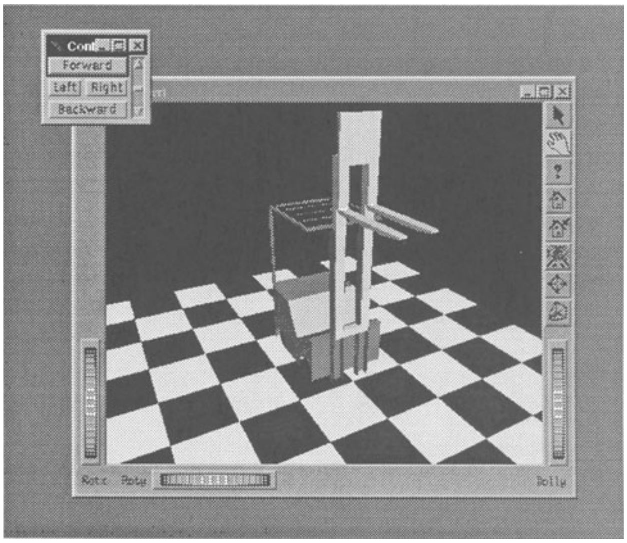


Figure 3: Animation element "Forklift" in VRML.

To achieve an automated generation of content, we suggest to build a library of reusable elements that can be combined for different applications. The combination of the elements is done within a special scene editor. Using predefined elements and combining them has the advantage of speeding up the process of content creation and avoiding the need for programming and VRML knowledge by the content author.

The key concept is the definition and use of animation elements (Luckas, 1997) (Dörner, 1997). An animation element not only has a geometry but also offers functionality regarding its behavior. For instance, an animation element "worker" has special methods, e.g. "move". This object-specific method implements the behavior in a parameterizable way; the "move" method, for instance, needs parameters such as start point, end point, and velocity. According to the parameters, the behavior of each part of the worker's body is determined, adapting the step width and making the worker walk or run. Instead of parameterized behavior it is also possible to use canned animations, e.g. to perform stream visualizations. For instance, when animating a bottle filling machine processing thousands of bottles per minute, it is not necessary to animate each bottle separately.

Animation elements can be collected and reused. An animation element library differs from catalogues of 3D elements (such as the ones from Viewpoint DataLabs) or Clipart collections as its elements offer behavior functionality. When concentrating on a certain application area, e.g. machine elements, a large part of often used objects can be covered.

According to our proposed concept the authoring process consists of two levels. Level one deals with the creation and modification of animation elements, level two with building up the 3D scene using these elements (see Figure 4).

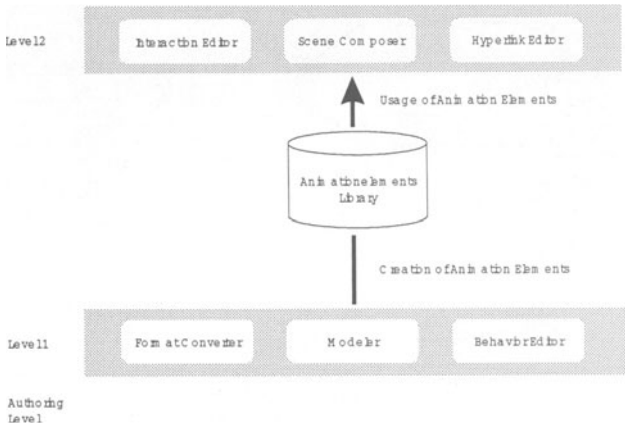


Figure 4: Authoring levels.

Generating animation elements

In a first level the animation elements are defined. This step is only necessary if the required animation element does not already exist in the predefined animation element library. Creating new elements is done by special authoring tools that support the association of VRML geometry models with behavior and behavior attributes. This is done by programming, e.g. with a special purpose language or in a visual style (Pausch, 1995), (Nakagawa, 1997).

The generation of new animation elements is generally considered as an external or low-level editing step, since it typically involves programming a new functional component for the animation element library, which is not a common task for application development but instead takes place before application content development starts.

Utilizing animation elements (scene composition)

At level two, the objects for the scene are selected from the animation element library. For this, the library offers multimedia database functionality. In the process, the user is informed about the methods offered (e.g. by short animations). If one element or functionality is not available, an existing element is modified or a new one is created from scratch.

In a next step the objects are placed and oriented. The object specific methods are used to define its behavior. This can be done by a user as well as by simulation or sensor data, i.e. simulation or sensor events are mapped to method invocations of the corresponding element. The mapping can be performed by the user or automatically by a translation module (see Figure 5).

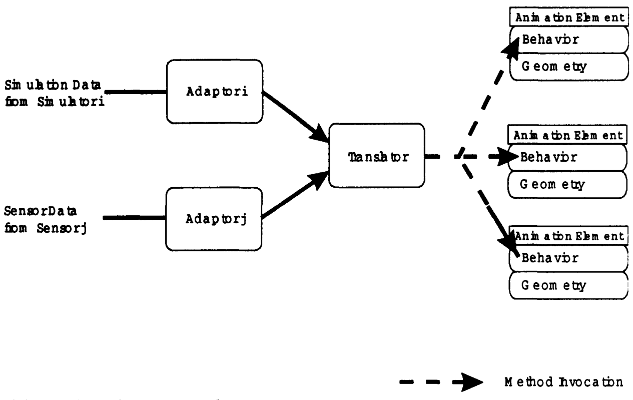


Figure 5: Linking simulators and sensors.

The scene parameters (camera parameters, illumination, background etc.) can be set or modified. Then annotations or a hyperlink structure may be inserted. Furthermore, the interaction can be defined. The interaction definition is also supported by the animation elements as the methods of the animation elements are still available when the user interacts with the scene. This is another advantage of using VRML because the animation elements can be used as is in the final scene.

Technical concepts

Animation elements can be thought of as classes in the sense of object orientation (Lafon, 1990), (Beeson, 1997). The class instances provide methods as well as attributes. For example an animation element could have sound attributes or posses state parameters.

This animation element concept can be implemented in VRML in an elegant way as VRML offers scripting nodes for including programs. The geometry of an animation element and some of its attributes can be realized in VRML, the methods and the other part of the attributes can be realized as Java classes (see Figure 6).

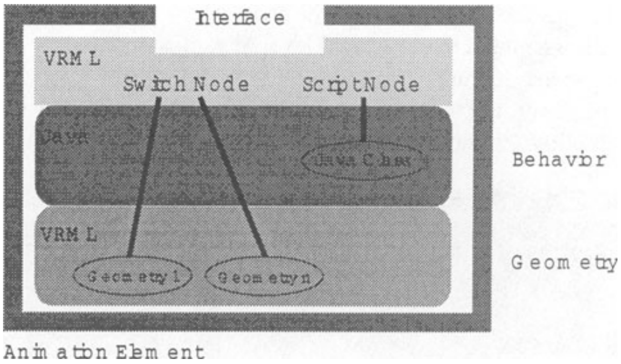


Figure 6: Implementation concept of an animation element.

6 REALISATION

In the context of the Demonstration Center for Simulation in Production and Logistic and the ESPRIT WIRE project, sponsored by the European Commission, a set of authoring tools has been developed according to the concepts presented above. In the projects several industry companies, such as Gillette, Krones, Audi and Zanussi are involved, providing feedback and evaluating the concepts in real industrial applications.

6.1 VRML authoring for product models

In the context of the 3D visualization of product data in VRML the following VRML authoring functionality has been implemented in our authoring environment. It is specially designed as a VRML scene authoring tool that incorporates additional behavior editing and animation generation functionality.

The VRML authoring component

This component (see Figure 7) supports the loading and modification of VRML scenes as well as the loading and manipulation of single VRML objects. The objects are imported and can be manipulated, positioned inside the scene, grouped together to form new more complex objects.

The VRML structure editor

This component allows to define scene hierarchies. The hierarchies are useful to represent the grouping of parts into modular part groups. These modular part groups represent the different levels in assembly, disassembly, and explosion.

Consider as an example a VRML model of a car. A disassembly of level one, for instance, yields the disassembly of all parts of the car body, i.e. the front doors, the back doors, the lid of the boot, etc. In a second level of explosion the parts in each of these part groups are shown. I.e. for the front doors the window-glass, the door locks etc. are disassembled in the second level of explosion.

Generally a scene hierarchy is assumed by default; if the assumed scene hierarchy is however not correctly modelled in the provided VRML file, the structure editor allows to modify and adjust it, in a graphically interactive way.

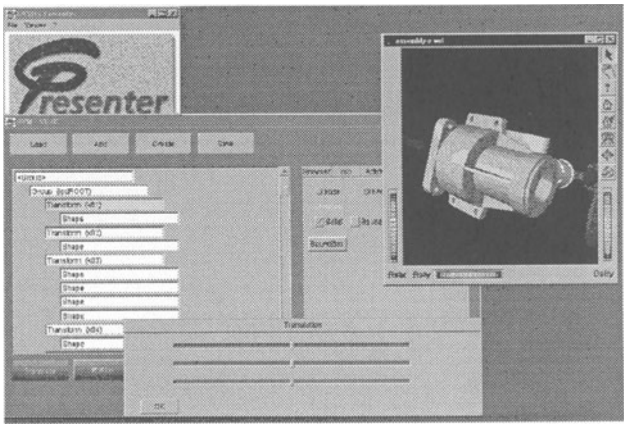


Figure 7: Special purpose authoring tool for assembly.

The VRML behavior editor

The VRML behavior editor allows to associate special predefined behavior components and attributes to the VRML models. The behavior components describe the movement the model performs when assembled or disassembled. The behavior attributes comprise additional information that can be displayed when viewing the VRML model in the application context. They can be used, for instance, to refer to the tools that are required for performing the manual assembly or disassembly of a part. Another aspect is the modification of the sequence in which parts are to be disassembled or assembled. This modification is necessary to adjust the model in case the default sequence is not suitable.

The VRML animation component

Based on the sequence of parts, the hierarchy, and on additional information that is either automatically extracted from the available material or added in the behavior editor, the 3D exploded view of the model is automatically computed. The results are shown in a graphically interactive way, such that the author can immediately verify the results of the explosion generation visually. In case the underlying default values are not adequate, they can be adjusted in the behavior editor such that the next automated computation of the 3D exploded view is semantically correct.

6.2 VRML authoring for process visualization

The visualization of processes is supported by the authoring tools described in section 5. We use the already existing CASUS system (Luckas, 1997), (Schäfer, 1997) that allows connection to arbitrary event-oriented simulators and sensor data. The connection is done in two steps: first the simulator or sensor specific format is converted to a normalized format, second the normalized data is mapped to method invocations of animation elements in the animation system CASUS Anim. From CASUS Anim, VRML output is generated. When trying to generate VRML output from an existing animation system, a typical problem is encountered. The easiest solution would have been to implement a converter that would convert from one of the supported output formats to VRML. This is not an optimal solution because conversion invariably implies loss of information. For instance the procedural animation possibilities of VRML are neglected and the conversion takes place on the basis of key framing only. Therefore, it was decided to let the VRML scene generation take place inside CASUS Anim using a VRML/Java representations of the elements. This is done by creating grouping nodes that integrate the VRML part of the elements and a scene description. The animation elements are regarded as sub trees in the scene graph, see Figure 8.

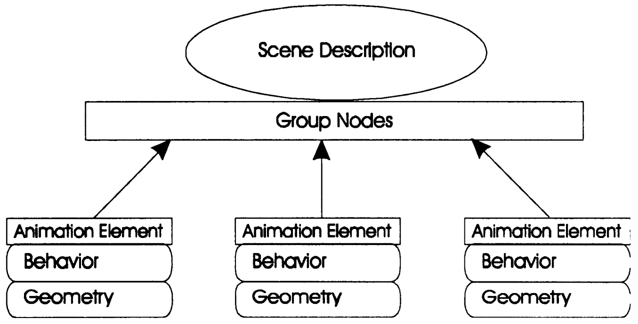
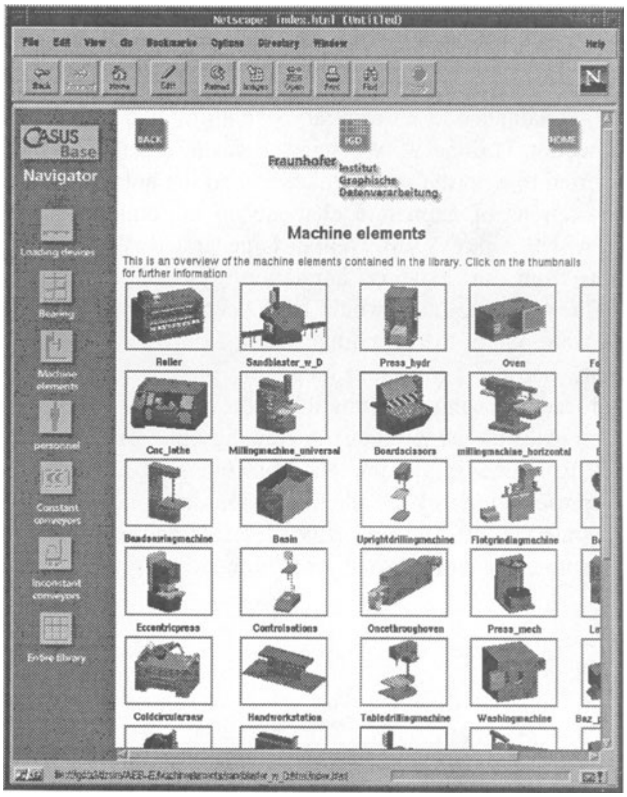


Figure 8: Building a scene with animation elements.

6.3 Scene composing component

The animation elements are presented in a web-based library (see Figure 9) that can be browsed and searched. After choosing the suitable animation element it can be fetched into the scene by a drag-and-drop mechanism.



This is an overview of the machine elements contained in the library. Click on the thumbnails for further information

Figure 9: An animation element library.

Then it can be placed, scaled and oriented similar to the Clipart functionality offered by many drawing and painting programs. In the case of process visualization based on simulations the scene composition can be integrated in the workflow easily as the editing of the 3D model and the simulation model can be performed in parallel.

6.4 Interaction metaphor component

Our VRML authoring tool provides different kinds of functionality for the VRML models it generates. This functionality or behavior must be made selectable in an intuitive way by the user who views the final application. For this purpose it is desirable to associate the VRML models with an intuitive **interaction control component** that allows to select the behavior context. This behavior context is, for instance, the assembly or disassembly display or the explosion animation ('assemble', 'disassemble', 'explode'). Furthermore the user may wish to view the "natural" object behavior. Returning to the example of a car, the natural behavior of a car might be assumed as 'drive' when activated. The interaction metaphor must thus indicate the state of the VRML model as 'natural'.

An additional issue that must be considered in the automatic creation of the VRML interaction control component is the possibility to define an **interaction** of the VRML scene itself **with an external document** such as an HTML-page. In the example of the car, this external document can, for instance, represent the bill of materials of the car, or a special part group of the car such as the motor. Considering the motor block, clicking on a part of the motor block in the corresponding VRML file will open a text-view of the bill of materials and highlight the corresponding part name in the list.

Vice versa clicking on a part name in the list will bring the corresponding VRML part of the VRML file into focus on the screen.

The interaction metaphor creation component supports the automatic creation of these interaction metaphors.

6.5 Technical realization of the VRML authoring tools

Technically the VRML authoring tools are implemented using Java and the External Authoring Interface (EAI) of VRML. The external authoring interface is part of the VRML standardization process and likely to be included into the next release of the VRML specification. Currently it is supported by Cosmo Player (Cosmo Software, 1997), WorldView (Intervista 1997), and CASUS Presenter (Schäfer, 1997). The advantage of the Java implementation is clearly the platform independent availability of the software. It can be used in conjunction with any Java-enabled VRML Browser supporting the EAI.

7 CONCLUSION AND FUTURE WORK

In this paper we have presented VRML authoring concepts and tools that are suitable for use in industrial applications. Application scenarios from a product and a production process point of view and relevant authoring concepts have been presented. It has been shown that VRML is successfully used in an industrial application context.

As a publicly available open standard, VRML offers a low-cost solution for 3D graphics. The fact that most 3D graphics modelers and tools by now support VRML in one way or another, highlights its being a de-facto standard of worldwide acceptance.

Our concepts emphasize the cost efficiency for special VRML applications because they allow the reuse of animation elements and speed up the authoring process in a significant way due to automation.

Checking the general criteria for VRML authoring listed in the introduction, it becomes clear that our concepts are not restricted to industrial application areas. The concepts are especially suitable for animation authoring of complex scenes. It can however be shown that in some special applications, requirements arise that cannot be met efficiently using a general authoring approach. For such applications as, for instance, the automated computation of exploded 3D views or disassembly and assembly animations there is a need to support additional specific functionality.

Future work will include exploration of new application scenarios such as teaching and training in an industrial context. On the technical side, future activities will focus on the integration of speech, security (watermarking of VRML objects), authorized access to VRML objects, an interface to agent systems and the use of an assistant concept for the user interface metaphor.

8 REFERENCES

Beeson, C. (1997) An Object-Oriented Approach To VRML Development, Proceedings of VRML'97, Monterey.

Cosmo Software (1997) Cosmo Player, a VRML2.0 Browser by SGI, http://cosmosoftware.com/.

Dörner, R., Luckas, V., and Spierling, U. (1997) Ubiquitous Animation: An Element-Based Concept to Make 3D Animations Commonplace, in *Visual Proceedings of Siggraph'97*, Los Angeles.

Intervista Software (1997) WorldView, a VRML 2.0 Browser by Intervista, http://www.intervista.com.

Lemay, L., Murdock, K., and Couch, J. (1996) 3D Graphics and VRML2.0, Sams Net Publishers.

Luckas, V., and Broll, T. (1997) CASUS - An Object-Oriented Three-Dimensional Animation System for Event-Oriented Simulators, in *Proceedings of Computer Animation'97* (Ed. N. Magnenat-Thalmann, D. Thalmann), Geneva.

Lafon, J.C., and Mahieddine, M. (1990) An object-oriented approach for modeling animated entities, in *Proceedings of Computer Animation'90* (Ed. N. Magnenat-Thalmann, D. Thalmann), Springer Verlag.

Nakagawa, S., and Ishida, H. (1997) Visual Behavior Programming with Automatic Script Code Generation, in *Visual Proceedings of Siggraph'97*, Los Angeles.

Pausch, R. et al. (1995) Alice - A Rapid Prototyping System For 3D Graphics, IEEE Computer Graphics And Applications, 15:3.

Preece, J. (1994) Human-Computer Interaction, Addison-Wesley.

Ressler, S., Wang, Q., Bodarky, S., Sheppard, C., and Seidmann, G. (1997) Using VRML to Access Manufacturing Data, VRML97 Second Symposium on the Virtual Reality Modeling Language, Monterey, Feb. 24-26.

Rost, J. (1989) OFF: a 3D Object File Format, DEC Workstations Systems Engineering.

Schäfer, A., Müller, W., Luckas, V. (1997) A Java-based VRML2.0-Browser: CASUS-Presenter, in *Poster Proceedings 6th International WWW-Conference*, Santa Clara.

SDSC (1995) The VRML-Repository List of VRML Browsers And Authoring Tools: http://www.sdsc.edu/vrml.

VRML (1997) The Virtual Reality Modeling Language, VRML97 Specification, ISO/IEC DIS 14772-1, April 4, 1997, http://www.vrml.org/Specifications/.

9 BIOGRAPHY

Colette Elcacho studied computer science and business administration at the Darmstadt University of Technology. Since she received her degree as Diplom-Informatiker in 1995 she works as staff researcher at the Fraunhofer Institute for Computer Graphics. Her major research interests are 3D animation and VRML authoring used in industrial applications.

Arno Schäfer studied computer science and law at the Darmstadt University of Technology. Since he received his degree as Diplom-Informatiker in 1996 he works as staff researcher at the Fraunhofer Institute for Computer Graphics. His major research interests are 3D animation and 3D technology emphasizing the use of Java and VRML.

Ralf Dörner studied computer science and pedagogics with focus on adult education at the Darmstadt University of Technology where he received his degree as Diplom-Informatiker in 1996. Since 1992 he worked as research assistant at the Darmstadt University of Technology and the Fraunhofer Institute for Computer Graphics. Since 1996 he has been staff researcher at the Fraunhofer IGD. His major research interests are innovative paradigms for 3D animation and their use in industrial and educational applications.

Volker Luckas studied Computer Science and Mathematics with focus on optimization algorithms at the Darmstadt University of Technology where he received his degree as Diplom-Informatiker in 1995. Since 1991 he worked as research assistant at the Fraunhofer Institute for Computer Graphics. Since 1995 he has been staff researcher at the Fraunhofer IGD. His major research interests are object-oriented, element-based 3D animation systems and their application in industrial scenarios.