

Design and Configuration of Furniture Using Internet-based Virtual Reality Techniques

Bernhard Jung¹ and Mathias Nousch²

Abstract. BEAVER is a program for a specialized computer aided design (CAD) task: The design of custom-built closets for slanted walls or ceilings. BEAVER is based on standard web technologies only, such as the Virtual Reality Modeling Language (VRML) and Java, which enable BEAVER to be run in a suitable internet browser. A key idea of BEAVER is that the user designs the closet by direct interaction with a visualized 3D model, e.g. when adding shelves or changing the size of the raw construction. In this way, an abstract model of the desired closet can be created and modified very easily and under permanent visual feedback. The knowledge-based system assists the user during this task and ensures that relevant design rules are obeyed. Upon completion of the closet's design, BEAVER automatically configures the abstract model with all fittings required for the assembly of the real closet. Thus, the abstract model is enhanced to a complete specification. This specification is used to generate a shopping list of all needed parts, such as boards and fittings. Also, a customized multimedia assembly manual can be generated on the fly, which guides the user when building the real closet. BEAVER not only demonstrates state-of-the-art capabilities of internet-based virtual reality tools but, also, introduces an easy-to-use, widely available method for customized furniture design: CAD for the rest of us, it's on the web!

1 INTRODUCTION

The immense growth of the World Wide Web (WWW) and the rapid development of related technologies for the presentation of multimedia content enables programmers to create even the most complex applications as integral parts of websites which makes these services reachable for a huge audience of internet users. The most important characteristic features of such applications are their time-, location- and platform-independent availability, (typically) free of charge usage, no cost for production and installation, and little effort for updating the underlying database. These advantages allow service providers to reach customers for whom the use of CD-ROM-based tools is either too large an expenditure or too time-consuming.

BEAVER's specialized application area of customized furniture design was motivated by a cooperation with a German company that produces furniture fittings (connecting fittings,



Figure 1. Virtual assembly with BEAVER. Shelves can be added, removed from, or shifted inside the 3D closet model. The program automatically calculates the required size and shape of the shelves, while they are moved to different positions.

furniture hinges etc.) for the do-it-yourself market. A large amount of their turnover is made in cases where available mass production furniture is suited only insufficiently to the needs of rooms with special characteristics. For example, the conversion of attics with slanted walls into living space calls for individual closets with exactly the roof's angle.

We chose this very specific, yet complex field of furniture design as testbed for the development of a VRML-based virtual construction tool for several reasons: First, the problem area is very narrow and well separable from others. Therefore, a small knowledge base, assisting the user in solving this task could easily be built. And second, the target group of the do-it-yourselfers (DIYer) is highly inexperienced both in designing and assembling of closets as well as in the use of computer aided design tools. Thus, a very intuitive user interface with a short learning phase is required. This makes the application area an excellent test case for verifying the proposed advantages of a web-based virtual design tool as compared to conventional CAD tools.

2 CONVENTIONAL DESIGN AIDS

The correct and complete design of furniture, when tried by DIYers is mostly an exception. The fitting producing company with which we cooperated saw the need, not only to sell the appropriate fittings but also to communicate their proper usage before the sale

¹ Faculty of Technology, AG WBS, University of Bielefeld, PO Box 100131, 33501 Bielefeld, Germany, jung@techfak.uni-bielefeld.de

² mathias@nousch.de



Figure 2. Virtual prototyping allows testing of the model's functions before its real assembly. With BEAVER, for example, the opening angle of the closet can be experienced.

(customers that buy the wrong products are unhappy customers, and potentially no customers at all the next time!).

The most common design aids in this field are printed brochures including a drilling sketch and an assembly instruction for each fitting. The information provided in these brochures may be enough for the assembly of closets from a given furniture construction kit. They are, however, inadequate for the design of completely new closets because they offer no guidance concerning the correct construction of closets as a whole.

The professional furniture manufacturing industry as well as carpenters and interior designers use specialized CAD programs. Such CAD tools are very powerful but also too expensive and too difficult to operate for being used by DIYers.

It was our goal to create a design tool, that would overcome these limitations.

3 CLOSET DESIGN BY VIRTUAL ASSEMBLY

BEAVER pursues the following strategy to solve the given design problem: The user designs an abstract model of the closet through the interactive assembly of virtual components. This is done through the direct manipulation of the visualized 3D model of the closet (see figure 1). The knowledge-based program assists the user during this task by ensuring that all construction rules are met, thus preventing the design of illegal, unbuildable closets. The abstract model is a correct, yet incomplete specification of the desired closet. At the same time, BEAVER translates this abstract specification into a precise one by choosing the appropriate type, number and position of all required fittings and hinges. Once the closet is fully designed, this precise specification is used to generate a shopping list of required parts as well as an individualized assembly manual which can both be used to buy and build the real closet.

BEAVER's multimedia user interface is split into a three-dimensional workroom and a two-dimensional menu. Both, the virtual model inside the three-dimensional workroom and the tabular construction parameters inside the program's menu are visible at any stage of construction. The parallel presentation of the

current scene within two different media types helps the inexperienced user in solving emerging ambiguities (cf. [6]).

The multimodality of the display is reflected in the multimodality of interaction: On the one side, the user can *directly* manipulate the virtual model of a component through interaction with its graphical representation. On the other side, the user can manipulate the component *indirectly* by altering the tabular form of its characteristic values. Such alterations are shown immediately inside the 3D-workroom and vice versa. The combination of the two modalities and their dependence upon each other permits a very easy, fast, and communicative introduction to the design process.

In BEAVER, different stages of the closet design process, such as selecting the closet type, setting its size parameters, or adding or modifying shelves, are presented in different frames. This step-by-step approach was chosen because not too much should be expected of an inexperienced user at one time.

One of the most important advantages of virtual prototyping is the possibility to visually experience and verify certain functions of the model before its real counterpart is built. In BEAVER, for example, the hinges are not only static fittings, but consist of two parts, a cross mounting plate and a contained hinge, that can be moved against each other with respect to the hinges' opening angle as shown in figure 2. This enables the user to experience the opening angle of the selected door type and to verify whether this is really the desired angle. In BEAVER, the door can be opened by clicking a button or by dragging the virtual handle of the door.

3.1 Parametric components instead of a static construction kit

The boards used for building a closet are hardly subject to any rules concerning their size or shape. Therefore, in theory, BEAVER has to manage a modular assembly kit of boards that is of infinite size. To prevent this, only specific, but parametric kinds of boards were modeled: Boards are six-sided blocks that can



Figure 3. The closet's size can be altered either by typing the absolute values, or by directly dragging the closet's walls. The visual and tabular presentations are updated simultaneously. Any shelves or doors that have already been added to the closet are reshaped and repositioned to fit the new closet geometry.

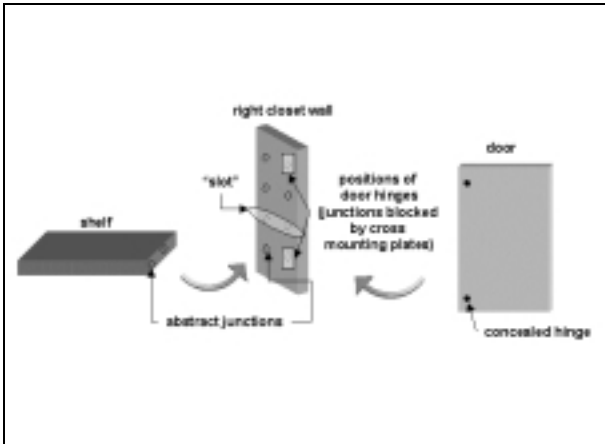


Figure 4. Virtual assembly of closet components is supported by a knowledge-based snapping mechanism. A horizontal pair of junctions forms a *slot* in which a shelf can be placed, if the slot is not already blocked. If the user moves a shelf to a new position, BEAVER automatically snaps it into the nearest free slot.

contain a maximum of two miters at opposing sides. The back or side boards of the closet may be seven-sided, which means they are slanted blocks. Apart from the miters or the slant, every edge is right-angled. The size of the boards however is dynamic and can be altered during the design process. Such alterations are triggered by the user; however, the required calculations are performed by BEAVER.

The user is only allowed to change the parameters of the entire closet, such as height or depth (see figure 3). It is possible to change the size parameters of the whole closet even if other components have already been added to the model, such as shelves or a door. BEAVER then recomputes a new, basic closet model and tries to place all components at their former positions, if possible. The program is also able to resolve conflicts that might occur during this phase (see below, Subsection 3.2).

3.2 Assembling virtual components at abstract junctions

One of the main problems when implementing virtual assembly is an appropriate model of the objects connection possibilities (see [3], [4]). This is somewhat easier if the real counterparts of the virtual components have junctions that can be defined precisely inside the model. Boards, however, have no concrete junctions at all. So we had to assume a fixed number of abstract junctions with a clear location on top of a board's surface. We decided to use the same grid of junctions that is commonly used in the furniture industry to prepare a board with drilling holes for its further use. However, there was still another problem: if certain fittings are attached to these junctions, the shape of the inlaying shelf should, in reality, be modified according to the specific shape of the fitting. Because such a manipulation of the shelf is permanent, no other fitting may be attached to the junction at that location later on. While this is very important for the real assembly task, it would be a great slowing-down, if this behavior were reenacted in our virtual model. At this point, we decided to differentiate between the abstract closet model that the user creates and the detailed closet description that is derived by the program: When the user sizes and

positions components, BEAVER only computes Boolean values of the junctions, i.e., whether they are blocked or still free. At this point of the design process, however, BEAVER is not yet concerned with the shelves' detailed shape enabling attachment of fittings.

The placement of shelves in the closet during direct manipulation of the 3D model is supported by a knowledge-based *snapping mechanism* (see [2]). To place the shelf in the closet, the user needs to bring the shelf and the side wall in overlap at their abstract junctions. This overlap does not need to be a complete and precise one, but is supported automatically when the distance between two suitable junctions narrows down below a certain threshold. The snapping mechanism serves as substitute for the lack of precision when positioning the components in the 3D model.

The realization of the snapping mechanism is based on the abstract junctions of the closet's side walls. For the assembly of shelves, two horizontal pairs of junctions at the interior surface of the closet's side walls are needed. We call this a *slot*. A hinge of the closet's door can block one junction of a slot, which prevents the whole slot from being occupied by a shelf, as can be seen in figure 4. During the design of the closet, the user can also shift the shelves in vertical direction. The program keeps track of already used or otherwise blocked slots. When the user drags a shelf with the mouse and drops it, the program interprets this as the command to add this shelf to the nearest empty slot. It then automatically selects the appropriate type and number of fittings for the given shelf-closet combination. If a shelf is dragged to a position where the closet is already slanting, it is obvious, that a normal shelf cannot be placed there. BEAVER recalculates the required shape and size of a shelf, whenever it is repositioned in the closet.

If the size of the closet is modified during the design process, the size and position of all its components must be recalculated. BEAVER uses several heuristics to solve conflicts that can emerge in such situations. For example, if the closet's size is reduced too much, some of its components must possibly be removed. Doors and their hinges are placed with the highest priority. If there is a conflict between a hinge and a shelf, where the latter should be placed in the same slot as the former, the shelf is removed from the model. If there are too many shelves because the closet has been shrunken, only the maximum number of allowed shelves for a certain closet height are retained. If there are no such conflicts, BEAVER tries to position every shelf into the same slot as it was before the change to the closet height was made.

3.3 Configuration, shopping list, and customized assembly instructions

Once the abstract model of the closet consisting of all boards, shelves, and doors has been designed the next step is to configure this abstract model with all fittings and hinges necessary for the assembly of the real closet. As opposed to the interactive design of the abstract closet model, the configuration process is performed automatically by BEAVER. BEAVER's configuration rules ensure that only fittings and hinges suited for this particular piece of furniture are selected. Furthermore – and somewhat unsurprisingly given the commercial interest of our industrial partner – BEAVER only considers products of one manufacturer and, at the request of our industrial partner, products from one product line only which simplifies the configuration process even more.



Figure 5. Fittings and hinges can be examined during the design phase if the model display type is set to transparency. In BEAVER, the user needs not care about which type of fitting he should use or where to place the fittings. BEAVER finds the optimal configuration of fittings for the specific closet designed.

BEAVER's configuration process not only calculates the appropriate kinds and number of fittings and hinges but also their spatial positioning within the closet's 3D model. Algorithmically, the configuration process is very simple. It exploits the fact that - after decade long efforts by the furniture industry of simplification and standardization - there are well-defined rules for equipping furniture with fittings and hinges (these rules are however not usually known by the end user and that's one of the points of a tool like BEAVER): Basically, the type and number of fittings and hinges is very similar for all closets. Closets with doors will need hinges for their attachment, where the number of hinges (two or three) for each door depends on its size. Similarly, closets with a base will need extra fittings and each shelf is attached with four fittings. Variations in the selection of fittings and hinges stem from differences in thickness of the boards and visibility considerations.

Configuration of the closet with fittings and hinges occurs in two phases. The first phase computes number, spatial layout, and product family of the connecting elements but not their concrete model numbers. This first phase is actually interleaved with the design process as e.g. the hinges necessary for the attachment of the doors may conflict with the positioning of the shelves. The positions of fittings and hinges within the closet can be inspected at any stage of the design in a special rendering mode where boards are visualized transparently (see figure 5).

Once the design task is complete, a shopping list of required parts for the real closet is generated. To achieve this, the configuration process enters its second phase. Given the number and product families of connecting elements computed before, exact model types are proposed that e.g. also account for the thickness of the boards used in the assembly. The shopping list contains the types, serial numbers, and number of packages of the fittings and hinges to be purchased. Each fitting is also described with a photograph to make it easier for the customer to find the specific package in the hardware store. Furthermore, each board is presented with a drawing of its shape - annotated with numeric size information - to remind the customer of the different shapes the board can have and to prevent wrong cuts.

Besides the shopping list, BEAVER also generates assembly instructions customized for the particular piece of furniture. In the

generated assembly instructions, the installation process of the fittings is described with local drilling sketches (see figure 6). Their global position, which the user could experience during the design process (see figure 5), is now described in textual form that refers to the size of the abstract drilling grid. Here the generated assembly manual supersedes conventional print-media instructions, as it not only provides information about *how* a fitting is fixed, but also *where* with respect to the complete assembly.

The assembly instructions describe difficult installations of fittings in step-by-step drawings. Also, alternative parameters of fittings, especially of hinges, that can be modified even after their installation can be mentioned. The combination of text and graphics helps the DIYer to check his imaginations against the real assembly task and so to prevent errors.

The shopping list and the assembly instructions are presented as "ordinary" web pages that can be bookmarked and revisited a long time after the closet's design. They can, of course, also be printed out on paper. After all, paper still is the more convenient medium for the DIYer when going to the hardware store or when building the real closet in his attic or workroom.

4 IMPLEMENTATION

We decided to rely on three different programming languages for implementing BEAVER on the World Wide Web.

4.1 Knowledge base and design logic: Java

Java is an excellent solution for the development of complex program structures and attractive graphical user interfaces. Java can be interpreted by any modern web browser and is fast enough for most applications. Further, the acceptance of Java applets is very high among internet users. Thus, we chose Java for controlling the design task and for realizing the two-dimensional user interface. The Java applet's classes are structured in a hierarchical order and mirror our top-down view of the design task:

The *applet(-class)* controls BEAVER's main menu and overall control flow of the program. It also establishes a communication

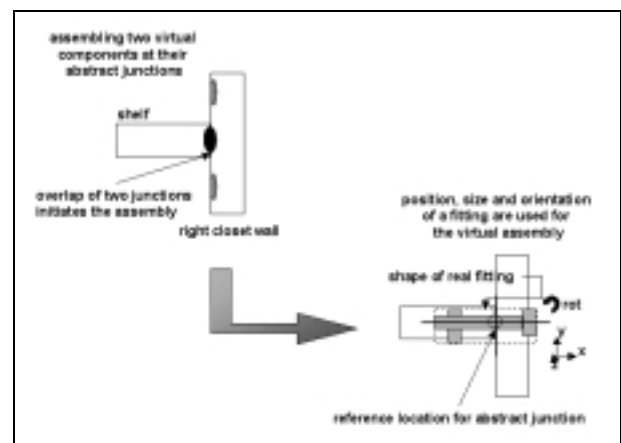


Figure 6. The fittings' detailed shapes are only used in the later stage of generating customized assembly instructions. For the preceding stage of virtual assembly, only basic data of fittings such as position, orientation and size are maintained.

link with the 3D-renderer and provides this link to the other classes.

The class *closet* is the largest class of our applet. It contains the structure of all virtual components, dependencies, and all design and configuration rules modeled in the program. The design and configuration rules are implemented in a straightforward procedural fashion. We favored the procedural approach over a declarative one because the underlying rules for design and configuration are very stable, for efficiency reasons, and to keep the size of the applet (and thus its download time) small. There is only one instantiation of this class, but an enhancement of the applet to allow more than one closet to be built is easily possible. *Closet* is in complete control over the assembly of one closet.

Board exclusively manages the presentation of a closet's components (walls, shelves, door) inside the 3D-workroom. It calculates the numerical shape descriptions of each component to build a virtual three-dimensional object. This class not only controls the appearance of the virtual objects concerning transparency, color, texture, etc. but also handles the user's manipulation of the virtual objects. For example if a component is dragged and released inside the 3D-world this movement is translated into an assembly command and sent to the *closet* class, which in turn decides what further actions to perform.

The class *fitting* is somewhat similar to *board*. It manages the 3D representation of the fitting. This class however does not need any methods for the surveillance of user actions performed on the virtual objects because there are none allowed for fittings. The only dynamic information about fittings maintained by this class are type, function, size, position and orientation. The detailed shape of the physical fitting is also modeled. The detailed shape information serves however only visual purposes and is of no importance for the virtual assembly. As mentioned above, this information is only used in the generation of the assembly instructions.

4.2 Visualization and virtual assembly: VRML

The Virtual Reality Modeling Language (VRML) allows for easy creation of interactive, three-dimensional environments on the WWW. In BEAVER, VRML is used essentially as a 3D renderer with the further possibility of creating „sensible“ virtual objects, whose manipulation inside the 3D workroom can be controlled by the applet. The *External Authoring Interface (EAI)* [5] is used to establish the communication link between the VRML browser and the applet that contains the complete design logic.

Our virtual world consists of three different sections: First, there is the main, static VRML file that defines the user's viewpoint and a viewmodel. The viewmodel allows the user to rotate and zoom the view of the designed closet. He cannot navigate freely in the world, because we wanted to prevent inexperienced users from losing sight of the object. The main VRML file also contains the lights and background for the scene.

Second, a navigation console (see figure 7) was developed exclusively for this application. It is realized as an external VRML-File, including a script which generates translation and rotation commands for the main world.

Third, there are the virtual components and fittings of the closet. These are created dynamically by the Java applet via the EAI. For these objects no scripts or routes exist inside the VRML world. All manipulations, even the smallest dragging movements,

result from commands that are generated by the applet after it receives input from the VRML sensors. We chose this way of interaction modeling to exclude all possibilities of differences between the internal model of the closet and its external, visual presentation.

4.3 Communication and dynamic page generation: JavaScript

With JavaScript, it is fairly easy to generate dynamic and interactive web pages on the client's side. It is very well suited for the creation of rich multimedia documents. In BEAVER, we use JavaScript to generate the shopping list of required parts and the assembly instructions.

This is done in the following way. The applet breaks down the entire internal representation of the closet into a list of variables and their specific values. Then a new URL on the server is called and the list of parameters is appended to the URL. The HTML file, containing the JavaScript functions reads these parameters and generates the shopping list and assembly instructions. The results are presented as „ordinary“ WebPages with text and images, which can be printed or saved to disk. A further advantage of placing the whole parameter list into the URL of the JavaScript page is the following: By just bookmarking this link, the user can save his current design and call the assembly instruction even a long time after the closet was designed. Thus, it is possible to temporarily save the design and continue working on it later.

An important aspect of using JavaScript, as opposed to e.g. signed applets or Perl scripts, is that it avoids the need for writing permission on the client's side or data storage on the server. As future extension, JavaScript'ing may also be used as interface for exporting 3D models of designed closets into CAD programs.

4.4 Notes on development

A fully functional prototype of BEAVER was developed in about



Figure 7. The navigation console is a part of the initial VRML world. In addition to the tabular presentation of the closet's size, a pictograph of a human figure can optionally be displayed in the virtual workroom, lending the user a visual impression of the closet's size.

three months time. This rather small development effort can be attributed to the usage of high-level programming environments such as Java and VRML. The development effort for the prototype also included design reviews by our industrial partners.

Besides providing the mere functionality, a next step was the decision about which products were to be included in BEAVER's database of fittings and hinges. Obviously, this decision was mainly at the side of our industrial partner. Accordingly, the development effort for this second phase was more on the coordination side than on the implementation itself. Interestingly, our industry partner preferred the inclusion of only a rather small number of products into BEAVER's configuration process.

5 CONCLUSIONS

We have described BEAVER, a web-based program for the interactive design of furniture. BEAVER's design methodology draws on multiple knowledge-based technologies to assist the Do-It-Yourselfer in the design process: A snapping mechanism supports the direct manipulation of the visualized 3D model. A configuration process completes the abstract closet model with all required fittings. BEAVER's knowledge of design rules further prevents the construction of unbuildable closets. Besides supporting the mere design of individual closets, BEAVER also generates a shopping list of the required parts and customized multimedia assembly instructions. BEAVER is unique in the combination of these features.

The web-capabilities and free of cost usage of our program were highly welcomed by our industry partner because of BEAVER's potential to enhance the sales of their – but not their competitors - products. Seen from this point, BEAVER is not only an easy-to-use and powerful aid for DIYers for the design and assembly of furniture, but also a somewhat visionary prototype of a new kind of software system for product presentation, promotion, and distribution, possible only on the internet.

BEAVER is specialized on the design of individual closets of non-standard shapes. Its design methodology could also be applied to other design tasks. Considering the interior design industry, other types of furniture could be assembled and brought together in web-based 3D presentation programs such as [1]. BEAVER demonstrates that it is feasible to create specialized CAD tools based on web-based technologies like VRML and Java.

References

- [1] J. Dauner, J. Landauer, E. Stimpfig: 3D Product Presentation Online: The Virtual Design Exhibition. *Proceedings of the VRML 98 Third Symposium on the Virtual Reality Modeling Language*, Monterey, CA, ACM press, 1998, p. 57-62
- [2] B. Jung, M. Hoffhenke, I. Wachsmuth: Virtual Assembly with Construction Kits. In *Proceedings of the 1998 ASME Design for Engineering Technical Conferences (DECT-DFM '98)*.
- [3] B. Jung, M. Latoschik, I. Wachsmuth: Knowledge-Based Assembly Simulation for Virtual Prototype Modeling. *IECON'98 – Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, Vol. 4, IEEE, 1998, 2152-2157.
- [4] S. Kopp. *Ein wissensbasierter Ansatz zur Modellierung von Verbindungen zur virtuellen Montage*. Diplomarbeit.

- Technische Fakultät, Universität Bielefeld, 1998.
- [5] C. Marrin and J. Couch. *Specification of the VRML-EAI Interface*, <http://www.vrml.org/WorkingGroups/vrml-eai/>. The WWW VRML Consortium – Online Resource. 1998.
- [6] W. Wahlster. *Text and Images. NSF-CEC Survey on Speech and NL Processing*, 1994