

## SNAPSHOTS AND BOOKMARKS AS A GRAPHICAL DESIGN HISTORY

Jouke C. Verlinden, Takeo Igarashi and Joris S.M. Vergeest

*Design history, visualisation, sketch-based modelling.*

### 1. Introduction

With the advent of innovative modelling techniques, the need for reflection and comparison is increasing. For example, sketch-based 3D modelling described in [Igarashi et al. 1999],[Qin et al. 2000] and [Do 2001] all support gestures, in which each modelling step radically alters geometry. In these applications, there is a large problem of losing track of the previous states. Most commercial CAD and 3D modelling packages allow the user to inspect and replay previous actions (undo-redo). However, such modelling actions are less traceable in gestural input. Here, the states of the model play a more prominent role than the transformations between them. A graphically oriented design history that displays the evolution of the model could better support this natural dialogue. However, no prior work has been found in research or practice that demonstrates this method.

We argue that the design history is not just a progressive record of the modelling process, but also contains a vast number of design alternatives. These alternatives should be kept in context, enabling the designer to switch between them, without having to switch between design projects.

This paper presents a simple technique to support this concept, allowing a graphical display of the model states and the opportunity to bookmark and annotate various stages of the design.

### Related research

In theory, the act of modelling can be represented in a transition network, in which each action connects two model states. As this network can be large and contains cycles, reduction schemes should be applied to present the network to the designer. The most obvious schemes either filter the design actions or the resulting states. Although richer displays could be employed, the information should not be overwhelming.

Unfortunately, not much has been documented on capturing and displaying the evolution of models. Two research topics show some similarities, namely design rationale and CSG trees. Design rationale research focuses on the documentation of the design decisions during modelling, although most work has been done in retrospective methods. The Electronic Design Notebook demonstrated a more integrated system [Leifer 1991]. Some automatic design rationale analysers have been developed, for example [Myers et al. 1999] employ activity patterns to relate the user dialogue with the Microstation CAD package to design activities.

In constructive solid geometry (CSG), a CSG tree represents the collection of primitive operations and geometric elements of a particular model [Mäntylä 1988]. Although this graph presents a visual anatomy of the model, it typically focuses on the structure of the current model, not on its evolution.

## Existing history support

A survey of commercial applications was done to characterize existing support for design history. The results can be found in Table 1.

**Table 1. Survey of commercial design packages.**

Application name	Application type	Undo capabilities	History display	Version management
Dreamweaver	Web design	Action History, preview	Separate Panel (list of action names), interactive feedback	None
Photoshop	Drawing	Action History, preview	Separate Panel (list of action names), interactive feedback	“snapshots”
SolidWorks	CAD	Multiple level undo	Popup menu (list of action names)	None
AutoCAD	CAD	Multiple level undo	Popup menu (list of action names)	
MicroStation	CAD	Multiple level undo	Menu item (only last action is mentioned)	None
Cinema 4D	3D modelling/ rendering	non-linear construction history	Separate panel (tree of action names and parameters), interactive feedback.	None
Maya	3D modelling/ rendering	Multiple level undo	Script panel (textual list of commands) – not directly linked to undo function.	None

From this survey, we conclude that 2D drawing and web design applications have more advanced features for managing previous states and alternatives than CAD and 3D Modelling packages. The only exception is Maxxon’s Cinema 4D, which offers similar facilities as Photoshop and Dreamweaver. In all cases, the action history is presented as a collection of actions rather than model states. This makes sense, as the actions are typically complex; often, a repeated execution of actions is required, and macros are easily constructed by selecting parts of the action history. Both Cinema4D, Dreamweaver and Photoshop provide interactive feedback while scrolling through the action list, displaying the resulting state in the drawing canvas.

Regarding the storage of alternatives, not much dedicated support exists. In most packages, alternate versions can be stored in layers. The only application that offers additional facilities to manage alternate versions is Photoshop, in which so-called “snapshots” can be made. These are displayed in a separate area of the history panel, each with a small picture (thumbnail) of the drawing. When Photoshop’s Non-linear History option is activated, all modelling steps are recorded in the action history, even when switching between snapshots. However, this introduces an extra level of complexity, as the action history has no apparent relationship with the effects on the drawing’s state.

## 2. Snapshots and Bookmarks

In sketching and other natural input modalities, actions are less obvious in the gestures and difficult to identify. Conversely, we explored the visualization of the evolving stages of the model itself. Two modelling concepts are introduced: snapshots and bookmarks.

- Snapshots correspond to the recorded states of the model of one modelling session. The definition of a model state is closely related to the modelling task or domain at hand. For example, in animation, the camera view and lighting model play an important role and changes should be captured, whereas these settings are of secondary interest for product designers. Typically, snapshots contain a timestamp, a small bitmap image (“thumbnail”) of the rendered model, the corresponding 3D model, and some information on the action that was executed.

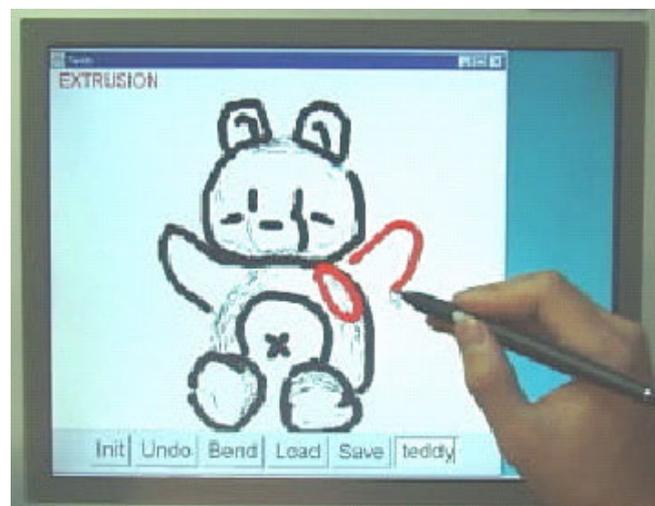
- Bookmarks are snapshots that represent important steps in the design history. The user creates these as a deliberate act, for a number of purposes. For example, to identify key stages in the design, or to store design alternatives in a shared context. Bookmarks contain the same information as snapshots, possibly extended by user-added annotations.

These concepts differ from Photoshop's notion of snapshots. The reason for introducing a different vocabulary is that Bookmarks represent more closely the function of user-based revisiting than Snapshots.

A variety of interaction techniques have been implemented to employ snapshots and bookmarks during modelling; the most important are presented in the following sections.

### 3. Implementation

A Java implementation was made based on Teddy. The complete prototype can be downloaded from the web at [Verlinden 2002]. Teddy is a powerful yet simple sketch-based 3D modeller proposed by Takeo Igarashi [Igarashi et al. 1999], Figure 1 shows Teddy on a pen-based system. It supports creation and editing functions, including creation, bending, painting, extrusion, and cutting. The basic method to construct 3D polygonal surfaces from a 2D silhouette is by inflating the region surrounded by the silhouette making wide areas fat, and narrow areas thin. This enables fast and simple construction of organic shapes.

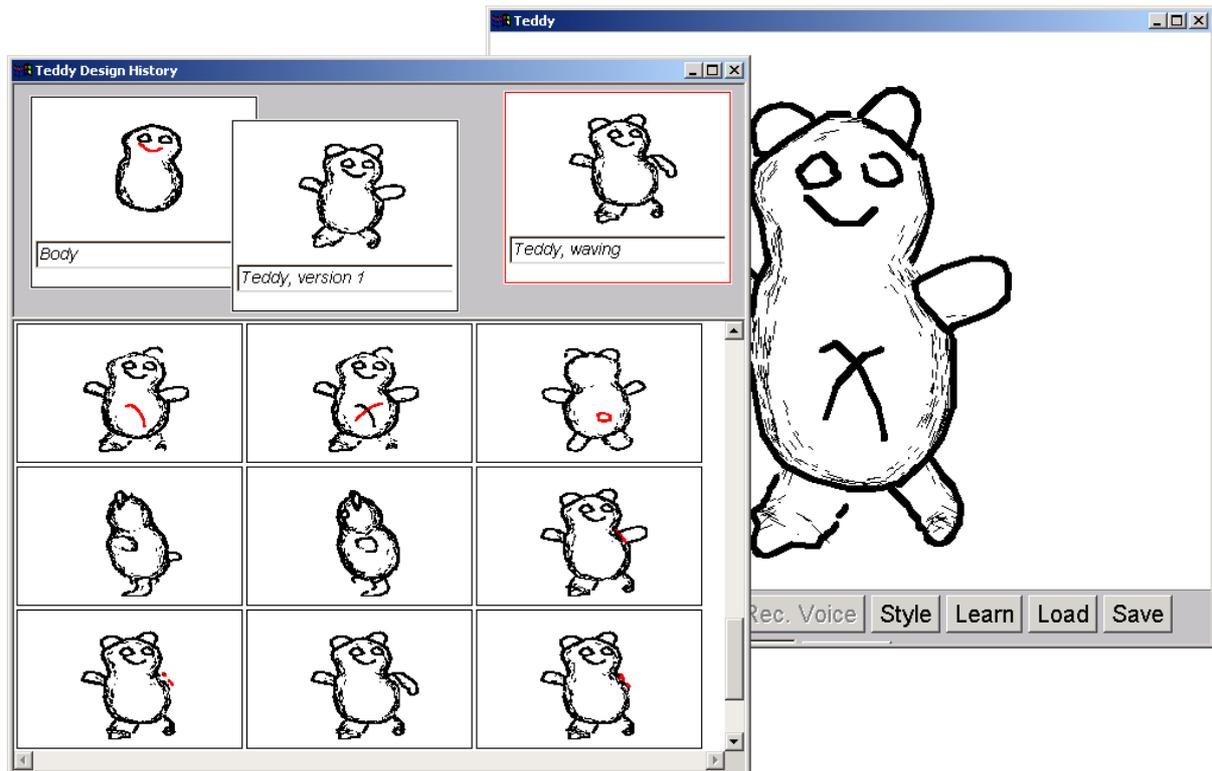


**Figure 1. The Teddy Modelling system.**

A separate window (the "history" window, see Figure 2) is created to display snapshots and bookmarks. Snapshots are added automatically when the model is altered in the main window. These are presented at the bottom of the window as a storyboard, each thumbnail representing one modelling state. The order is chronological, i.e. new snapshots are appended left to right, top-to-bottom. When necessary, a vertical scrollbar will appear when not all snapshots can be displayed at once. The view automatically scrolls to the latest addition. When the user double-clicks on one of the thumbnails, the selected state is reloaded in Teddy's main window. In contrast to commercial solutions, the snapshots show a cumulative result of the model states, which are not removed when a previous state is activated in the main window.

Bookmarks are presented at the top half of the same window. They can be reshuffled by dragging and dropping within the bookmark area. As with snapshots, double-clicking will reload the bookmarked model into the main window. Unnecessary bookmarks can be removed by selecting them with a single mouse click and subsequently pressing the "Del" key. Annotations can be added to bookmarks, currently implemented as text fields below the thumbnail image; the user clicks on the text to select or edit it with keyboard and mouse. In the future, the annotating possibilities will be extended by supporting simple freehand drawing on the thumbnail image. Currently, a bookmark is inserted when a user revokes a previous state by using double clicking on a snapshot. The assumption is that when

particular snapshots are used to navigate through the design history, these are probably important milestones for the designer.



**Figure 2. The implemented design history.**

In the implementation, the snapshots are stored in a linked list in a chronological sequence. A link to the preceding snapshot is kept as well, thus creating a hierarchical representation of the modelling process in which a collection of stages can originate from one single state. This is the case when the user revokes an earlier state. The snapshot data is saved in a GML formatted file (Graph Markup Language [Himsholt 1997]), enabling a number of existing visualization tools to display the design process. We selected the VGJ package [McCreary 1998] for this purpose, see Figure 3 for an example screenshot, which shows part of the history trace of a modelling session, including action name and timestamp. This capturing feature was particularly useful during evaluation (see the following Chapter).

#### **4. Discussion**

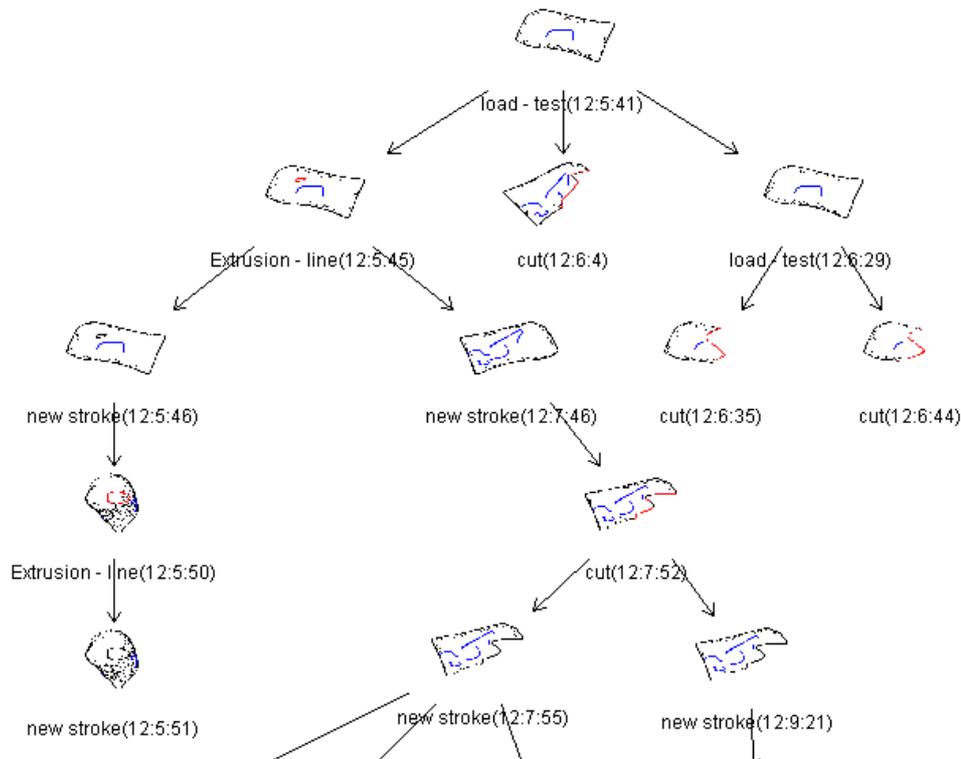
This prototype was preceded by a simpler implementation. History steps were displayed in a window as a textual list. A preview image was displayed when a step was selected, similar to the history panel in Photoshop. The actions mentioned as text made little sense in longer sessions; many shape modification operations were repeatedly activated (e.g. extrusions, 3D line drawing, bending), but each time with drastically different effects. This led to the notion of a storyboard and using bookmarks to identify the most important steps. The usability improved a lot by its new implementation.

A number of exploratory experiments were performed, in which industrial design engineering students had to model some existing shapes and had to create a new design as a final assignment. Preliminary results show positive effects in the amount of time required to finish these assignments (compared to the absence of a history window). However, a number of important issues were found:

- Although Teddy is a simple modeller, it took the subjects a considerable amount of time to get used to its modelling paradigm. They had prior experience with other CAD packages, and as

Teddy uses a completely different shape modification operation, new strategies had to be devised. This made it difficult to test the usability of the history features.

- Snapshots were immediately understood when few modelling steps were done. However, the bookmarks were less obvious from start – some introduction was required to demonstrate their use and functionality.
- The history window did give a good overview of the past steps, but in large sessions, the subjects had to scroll often to revisit older states. The storyboard visualization can be optimised in many ways.
- Related to the previous issue, is the complexity in managing screen estate, having to use both history and main windows concurrently. A more integrated configuration of interface elements would improve this.
- As the thumbnail images are copies of the complete main window, they often contain a lot of white space. A cropping algorithm could be used to maximise the legibility of the thumbnails.



**Figure 3. Visualization of the history trace.**

As was said in Section 2, the relevance of modelling states is domain and application-dependent. The prototype only focused on geometry, while the camera angle plays an important role in the act of extruding. In this case, it seems useful to keep separate histories for the model and its environment. The environment could even be split in a number of settings that are independent of each other (e.g. camera views and rendering backgrounds). However, this adds extra complexity and possibly clutter to the display and interaction with the modeller.

## 5. Conclusions and Future work

In this paper, a new interaction technique was introduced to display a design history. A platform-independent prototype was implemented which is available on the web. It is integrated in a sketch-based modelling package that could potentially benefit from such an extension. Exploratory evaluations showed that the use of snapshots in a storyboard visualisation was straightforward. Some important improvements could be made in the integration of the windows and display of the snapshots.

Notwithstanding these issues, we state that this technique can be useful for new modelling paradigms, including sketch-based and multimodal design systems. Existing CAD systems can easily be extended with this technique as well, although this requires integration with action-oriented histories. New shape reuse methods [Vergeest et al. 2001] might also benefit from this type of interaction, as these explicitly deal with (re) application of geometric information.

Furthermore, capturing snapshots and bookmarks provides an excellent basis for design research experiments - the design history can be managed as a directed acyclic graph and attributed with other information.

Future issues include collaborative use, visualization technique for snapshots, and concurrent display of model and the application's settings.

Bookmarks and snapshots allow a rich support of collaborative use, in sharing intermediary and alternative designs, both in distributed and single-machine settings. Considering the fact that we have used Java as an implementation platform, the step to a collaborative design system is easily done.

The chosen visualization method for snapshots is straightforward, yet is of limited use in large design sessions. Other visualisation methods will be considered, in which both navigation and graphic display offer better solutions (e.g. using a fish-eye effect).

As was said in the discussion, there is a variety of information that can be presented to the user. All aspects of a modelling session could be kept in separate displays, while each aspect might require a different visualization method. This integration adds extra complexities that should be carefully considered.

## Acknowledgement

This research has been performed as part of the Integrated Concept Advancement (ICA) project, at the Delft University of Technology.

## References

- Do, E. "VR Sketchpad, Create Instant 3D Worlds by Sketching on a Transparent Window", *Proceedings of CAAD Futures 2001 (Eindhoven)*, pp. 161-172
- Himsolt, M., "GML: A portable Graph File Format", <http://www.infosun.fmi.uni-passau.de/Graphlet/GML/gml-tr.html>, July 1997
- Igarashi, T., Matsuoka, S., Tanaka, H. "Teddy: A Sketching Interface for 3D Freeform Design", *Proceedings of SIGGRAPH'99*, pp. 409 - 416.
- Leifer, L.J. "Instrumenting the Design Process for real-time text-graphic design process records", *Proceedings ICED'91, Zurich*, pp. 314-321.
- Mäntylä, M. "An Introduction to Solid Modeling, Principles of Computer Science" Computer Science Press, Maryland, U.S.A.. 1988.
- McCreary, C. [http://www.eng.auburn.edu/department/cse/research/graph\\_drawing/graph\\_drawing.html](http://www.eng.auburn.edu/department/cse/research/graph_drawing/graph_drawing.html), april 1998.
- Myers, K.L., Zumel, N.B., Garcia, P. "Automated Capture of Rationale for the Detailed Design Process" *Proceedings of the Eleventh Conference on Innovative Applications of Artificial Intelligence, 1999 (IAAI99)*, pp. pp. 876-883.
- Qin, S.F., Wright, D.K., Jordanov, I.N. "From on-line sketching to 2D and 3D geometry: a system based on fuzzy knowledge", *Computer Aided Design, Elsevier Science, 2000*, vol 32, pp. 851-866.
- Vergeest, J.S.M., Spanjaard, S., Horváth, I "A Methodology for Reusing Freeform Shape Content", *proceedings ASME DETC 2001, DTM-21708*.
- Verlinden, J.C., <http://dutoce.io.tudelft.nl/~jouke/teddy>, January 2002.

Jouke Verlinden

Delft University of Technology, Subfaculty of Industrial Design Engineering

Jaffalaan 9, Delft, the Netherlands

Phone: +31-152789321, fax: +31-152781839, [j.c.verlinden@io.tudelft.nl](mailto:j.c.verlinden@io.tudelft.nl)