Proceedings of the
1999 ASME Design Engineering Technical Conferences
September 12-15, 1999, Las Vegas, Nevada

DETC99/DFM-8943

CONCEPTUAL DESIGN FOR ASSEMBLY

Santiago V. Lombeyda[1] **William C. Regli**[2]
Geometric and Intelligent Computing Laboratory
Department of Mathematics and Computer Science
Drexel University
3141 Chestnut Street
Philadelphia, PA 19104

ABSTRACT

This paper presents an approach to support computer-aided conceptual design of mechatronic assemblies in a collaborative, multi-user environment. We describe a system, Conceptual Understanding and Prototyping (CUP), that allows a team of design engineers, collaborating over the Internet, to develop a high-level structure-function-behavior (S-B-F) description of an assembly in a VRML-based virtual environment. Our goal is to enable users to navigate intricate product data management (PDM) and case-based design knowledge-bases, providing the ability to perform design at conceptual level and have intelligent CAD tools that can draw on details from large repositories of previously archived designs.

This work furthers research efforts in computer support for collaborative design activities—drawing on work in Human-Computer Interaction (HCI) and Computer Supported Collaborative Work (CSCW). We envision CUP to be a network interface to next-generation of engineering PDM systems and CAD databases. We are deploying CUP as query interface to the National Design Repository (`http://repos.mcs.drexel.edu`). This will enable CAD users to interrogate large quantities of legacy data and identify artifacts with structural and functional similarities—allowing designers to perform case-based and variant design.

[1] Present Address: Center for Advanced Computing Research, Mailstop 158-79, 1200 E. California Blvd. California Institute of Technology, Pasadena CA 91125 USA. Email: `slombey@caltech.edu`

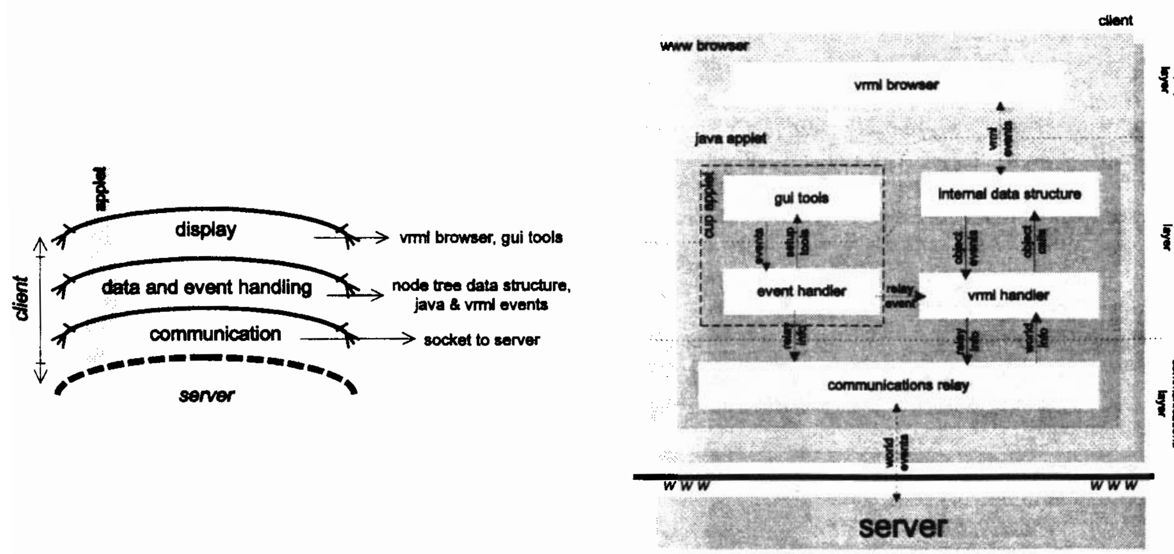
[2] Web: `http://www.mcs.drexel.edu/~wregli`; Email: `regli@drexel.edu`.

INTRODUCTION

It is conservatively estimated that more than 75% of engineering design activity is case-based design (27). An engineer or team of engineers, when posed with a new problem, synthesize solutions through a combination of analogical reasoning, experience, and search through design information archived in the corporate files and described in product and component catalogs. Over the past decade, 3D Solid Modeling has become a critical element of the product realization process—leading to successive generations of increasingly robust software tools for *detailed* design and engineering analysis.

The goal of this research is to develop novel techniques and tools to support *Conceptual Design.* Conceptual Design is the initial phase of product development, when product development teams (consisting of design engineers, manufacturing engineers, marketing and management personnel) determine the essence of a new product. As has been often documented, and shown in Figure 1 (a), the design phase of the product realization process has a tremendous impact on the life-cycle cost of a product. However, as illustrated in Figure 1 (b), there are few tools to support 3D conceptual design.

During Conceptual Design, teams of designers may begin to develop a new product by sketching its general shape on paper. This "back of the envelope" approach is key aspect of the creative thought process—once completed, one has a clearer idea of what is being created and can proceed to drafting or CAD activity. Previous research in areas such as conceptual design of graphical user interfaces (GUIs) (13; 18) and cased-based reasoning and case-based design (26; 25; 20; 10; 1) are relevant to



(a) (b)

Figure 4. CUP internals.

common thread that brings these layers together is a Java applet[4] which is responsible for creating sockets for communication with the server, up-keeping the representation of data, and communicating with the VRML browser. To elaborate:

1. The *communications layer* performs asynchronous communication between agents in the environment and the session server, as well as to the internal representation layer and to the display layer.
2. The *internal representation layer* represents a hierarchy of tree which includes environment variables and the design structure. The design structure is a hierarchy of objects which includes such information as properties as well as control and ownership information.
3. The display interface is responsible for managing the environment, and thus event driven. This layer has several states available and dictates what the user is able to do and see at each of them.

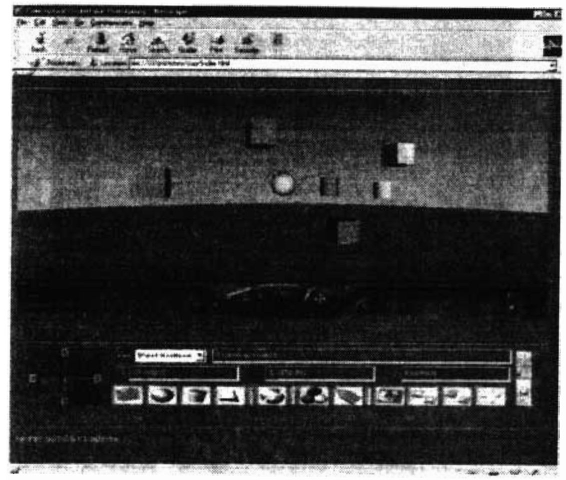
Input and Output Interface. The interface is divided into two areas: the first is the top, a VRML browser capable of displaying a design world. All object/design output will be displayed through this browser. The second area of the interface consists of the tools applet—responsible for most of the interaction to and from the user. This is implemented by using the External Authoring Interface (EAI) between the VRML browser and Java applets running inside the web browser. The VRML EAI provides a set of Java classes to establish connections between the VRML browser's objects/events and the outside world (in the browser). Through the EAI, Java applets can modify, add, or delete nodes in the VRML world; or handle changes or interaction among nodes within the VRML world—as shown in Figure 4. For example, the EAI handles creation, manipulation, destruction, tagging, linking, and grouping of objects. The VRML browser is the display mechanism for the world and the event handler for interactive selection and manipulation events.

Data Flow. Data flows across this interface as structures of objects and nodes, pictured in Figure 5. This information flows across the data structure according to actions taken by the user, or outside the interface and relayed by the server. Thus, much of the data flow in terms of design objects is between a class `VrmlHandler` and the browser. This will be triggered by actions caught within the interface, or relayed to the interface from the browser, as well as from the communications `Relay` class, which gets mapped to the applet as well.

The main data structures being handled represent trees of data structures that closely resemble VRML nodes. In fact these data structures have direct connections with actual objects events. These objects comprise all the information needed for the purpose of conceptual and collaborative design, such as tags with the structural, behavioral, and functional descriptions, as well as information of ownership, control and protection. Links and groups are treated as objects as well, with the difference that they have specific characteristics which allow them to point to two (in the case of links) or more objects (i.e., groups).

The upper most generic class from which this structure in-

[4] An applet is a Java application running within a Web browser



(a)

(b)

Figure 3. Screen shots of CUP and some of the facilities for linking and tagging objects.

herits is the `VrmlObject` class. This class is abstract, which means that no object may be instantiated as being of type `VrmlObject`, but must be of some derived class. A `VrmlObject` has these characteristics: a central node, a position accessor method, a tag, and a parent pointer.

There are four main classes which derive from `VrmlObject`: `Geometry`, `Root`, `Link`, and `Group`. The `Geometry` class is abstract as well, and is the base class for all shapes to be created and handled in this package. This class has an initializer which is responsible for creating transformation nodes, appearance nodes, and sensor nodes, all of whose change events are stored to be used by the `VrmlHandler`.

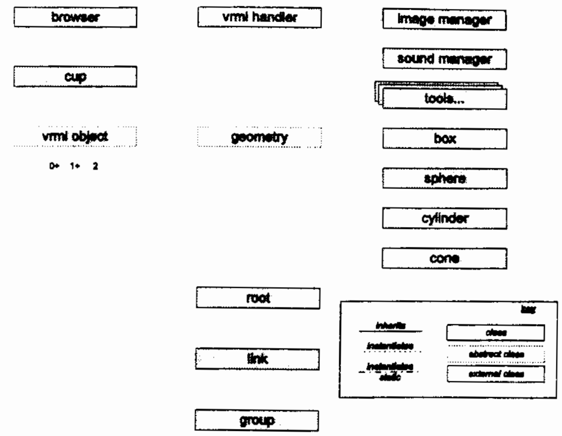


Figure 5. The information classes and methods developed in Java to support CUP.

The `Link` class is similar to that of a `Geometry`, the difference being that the parent pointer is always the root, and that it has two more pointers for `from` and `to` links to `VrmlObjects`. The tag in this case to be used will be of the type `LinkTag`, rather that the generic `Tag` class. Finally the root and group nodes contain vectors of object nodes. This hierarchy of classes is described in Figure 5.

As objects are created, they are introduced into the world by adding them as children of either the root, or an already existent parent. From these objects, events are captured so as to be able to modify them at a later period, or to be able to capture a selection or drag events. It is important to add that these objects are not single nodes, but usually a composition of several embedded nodes such as transform nodes, sensor nodes, group nodes, geometries, appearances, and shapes (VRML nodes). Existence of objects may easily be traced by inspection of the display browser.

Implementation Specifics. CUP runs on Pentium II-based machines with Microsoft Windows NT 4.0 and Sun Ultra workstations with Solaris 2.6. The current implementation is based on Java Version 1.6, Netscape Navigator Version 4.05, and CosmoPlayer Version 2.1. Functionality on the Solaris platform is limited due to the lack of a plug-in compatible VRML browser (we are currently using the VRWave stand-alone browser).

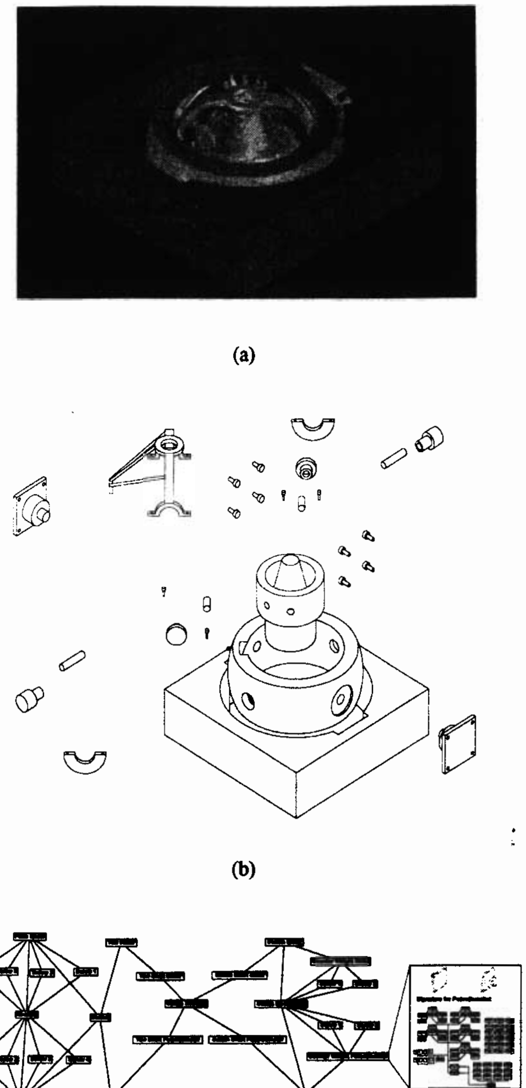
TESTING AND EXAMPLE

The final interface went through a process of re-design and reevaluation. The testing was done by members of the National Design Repository Project (`http://repos.mcs.drexel.edu`) and Geometric and Intelligent Computing Laboratory (GICL, `http://gicl.mcs.drexel.edu`) at Drexel University. This initial testing was largely informal—done on the basis of interview and feedback from members of the laboratory team using the tool.

CUP is primarily concerned with the creation of conceptual designs of *mechatronic systems*, electro-mechanical systems that combine electronics and information technology to form both functional interaction and spatial integration in components, modules, products, and systems. Typical examples of mechatronic systems include automatic cameras, miniature disk drives, missile seeker heads, and consumer products like CD players, camcorders, and VCRs. These designs include mechanical and electronic components—such as the CAD model (simplified) of a missile seeker assembly pictured in Figure 6, and related meta-data (process and assembly plans, documentation, etc).

This simplified seeker assembly might be one of many dozens stored in a corporate design data/knowledge-base. A design team, faced with the task of creating a new seeker, might want to interrogate the CAD knowledge-base and examine previous design cases that might be relevant to this new problem. Examining this legacy data can prove time-consuming and tedious, unless the device required and its search parameters are known precisely beforehand.

As illustrated in Figures 7 (a) and (b), CUP allows a designer to quickly sketch out, in 3D, the major components and structural relationships in the assembly. Rather than performing detailed CAD to create a draft design (detailed CAD modeling for this model took several days), designers can, in a matter of minutes, build a conceptual design. This conceptual design can then be used as a starting point for further refinement or as a query to the design knowledge-base. CUP also, via the attributing, tagging and labeling features, helps designers capture the design intent and to build a structure-function-behavior (S-B-F) model of the artifact.



(a) (b) (c)

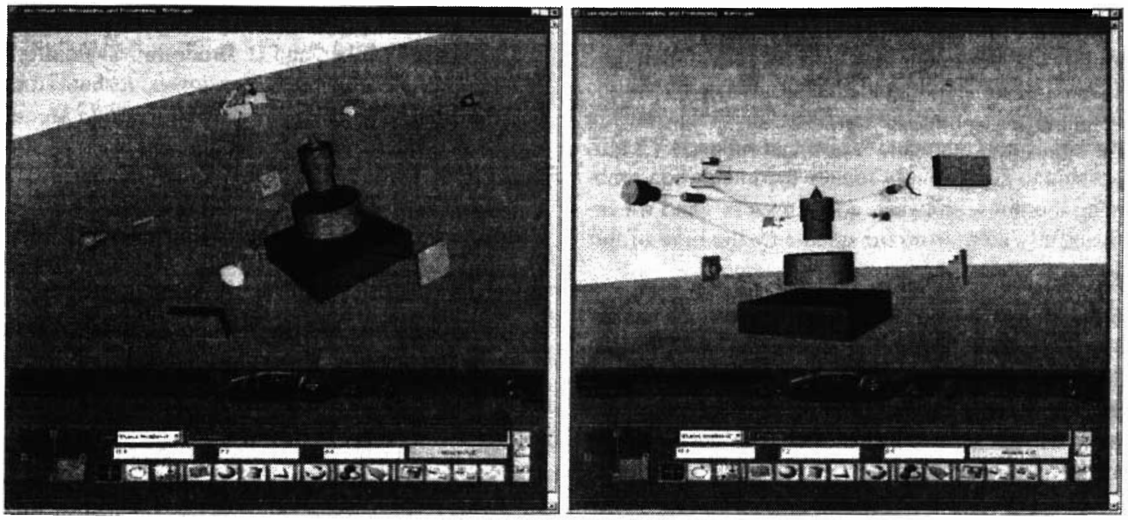
Figure 6. An example of a mechatronic system: a simplified missile seeker assembly along with its assembly structure.

CONCLUSIONS

Research Contributions. This research represents our initial efforts to develop a system for the conceptual design of mechanical assemblies. In doing so, we have created a novel form of modeling system and introduced new object manipulation methods to support the unique user interaction needs in collaborative/conceptual design environments. It is our belief that CUP can been seen as an important component in a conceptual design environment—enabling users to create a knowledge-level description of the design without having to perform detailed CAD and solid modeling.

Future Directions. We are extending this work in several significant ways. At the time of this writing, these thrusts are each in the development stages:

Multi-User Conceptual Design: Building multi-user collaborative design functionality via CUP's network-based



(a) (b)

Figure 7. Illustrations of a conceptual design for the missile seeker pictured in Figure 6.

Virtual Reality environment could prove to be a powerful complement to traditional CAD and CSCW tools. The current CUP does not implement a multi-user environment—adding this functionality is part of our continuing development efforts.

Access to Design Repositories: Increasingly, engineers depend on Product Data Management (PDM) systems, large-scale engineering digital libraries and knowledge-bases of CAD solid models to perform their job (24; 23). This involves searching through vast amounts of corporate legacy data and navigating manufacturer's catalogs to retrieve precisely the right components to assemble into a new product. The complexities of this trend are compounded, as team members from different companies collaborate over computer networks on joint engineering projects. Presently, however, design storage and retrieval systems (such as commercial databases and Product Data Management systems) are limited in their ability to capture design intent and reason about CAD knowledge—often relying on the textual annotations of the designer, or the part's filename, for storage and retrieval purposes. In some cases (15), ontology-based classification schemes have been employed. However, these require that all users share a common ontology for representing the data and know, a priori, the precise classification of the item of interest.

It is our eventual goal to integrate CUP with the National Design Repository (`http://repos.mcs.drexel.edu`) and provide facilities for users to add their own ontology information to the repository's knowledge-base.

CUP as 3D "Freehand Sketching": Other researchers have approached conceptual design as a freehand sketching recognition problem (5). In this work, we have developed a 3D modeling approach to conceptual design that enables teams of designers to embed semantic (S-B-F) information in their models. We believe that our approach offers several unique benefits. First, it is the structure-behavior-function (S-B-F) knowledge, more so than the geometry and topology, that encodes the designers' intent. By capturing this intent, we can search design knowledge-bases for related information and create pro-active design tools that can guide the search of the design space. Second, this approach liberates the designer form the usual restrictions of exact measurements, or precise positioning and orientation. CUP will allow the designers to create a 3D "freehand" sketch and the general structure of the artifact without detailed CAD.

Evaluation: As a complement to our approach to conceptual design, we plan to undertake studies to rate efficiency of this design tool. How much is gained by having multiple users? What is the correlation between time efficiency and design accuracy in such environment? How do multimedia communication tools interact with this interface? What happens without them?

Internet Flux: One technical challenge in this project was posed by the constant change in Internet, Web, and VRML software and technologies. Changes to development tools, Java, and browsers (Internet Explorer and Netscape) necessitated monthly updates to our development suite. Often the new tools were in conflict with older APIs and functionality. Our future plans are to port the current CUP to Java3D, a 3D programming environment for Java that supports VRML. We expect this to significantly reduce the integration problems among the tools, software, and browsers—and we expect to have fully completed the update by the time of this conference.

The immediate use for CUP is as a query interface to the *National Design Repository*. The structures created by CUP will be used to generate a query Repository and extract similar assembly designs. It is our hope that this research expands the understanding of how software tools can support conceptual design and that it lays the foundation for exploring new techniques to enhance our ability to search and retrieve 3D solid model data. Further, we believe that existing approaches to multimedia libraries can be augmented with geometric reasoning techniques that are tightly coupled with engineering knowledge and solid models, future as part of this research.

Acknowledgements. Thanks to GICL members Lisa Anthony, Jon E. John and Yuriy Shapirshteyn

This work was supported in part by a National Science Foundation (NSF) CAREER Award CISE/IRIS-9733545 and Grant ENG/DMI-9713718; additional support was provided by the National Institute of Standards and Technology (NIST) under Grant 60NANB7D0092 and AT&T Labs' Internet Platforms Division.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other supporting government and corporate organizations.

REFERENCES

S. Bhatta and A. Goel. Discovery of physical principles from design experiences. *International Journal of Artificial Intelligence in Engineering Design and Manufacturing*, 8(2), Spring 1994. Special issue on Machine Learning in Design.

D. Bowman and L. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, Providence, Rhode Island, April 27-30 1997.

Thomas A. DeFanti, Maxine D. Brown, and Rick Stevens. Virtual reality over high-speed networks. *IEEE Computer Graphics and Applications*, 16(4):42, July 1996.

Jose Miguel Salles Dias, Ricardo Galli, Antonio Carlos Almeida, Carlos A.C. Belo, and Jose Manuel Rebordao. Mworld: A multiuser 3d virtual environment. *IEEE Computer Graphics and Applications*, 17(2):55–66, March-April 1996.

L. Eggli, G. Elber, and B. Bruderlin. Sketching as a solid modeling tool. In Jaroslaw Rossignac, Joshua Turner, and George Allen, editors, *Third Symposium on Solid Modeling and Applications*, New York, NY, USA, May 17-19 1995. ACM SIGGRAPH and the IEEE Computer Society, ACM Press. Salt Lake City, Utah.

James E. Fowler. Variant design for mechanical artifacts: A state-of-the-art survey. *Engineering with Computers*, 12:1–15, 1996.

Bernd Frohlich, Martin Fischer, Maneesh Agrawala, and Andrew Beers a Pat Hanrahan. Collaborative production modelling and planning. *IEEE Computer Graphics and Applications*, 17(4):13, July-August 1997.

Michelle Gantt and Bonnie A. Nardi. Gardeners and gurus: Patterns of cooperation among cad users. In *Transactions of the ACM Conference on Human-Computer Interaction*, 1992.

A. Goel, S. Bhatta, and E. Stroulia. Kritik: An early case-based design system. In Mary Lou Maher and Pearl Pu, editors, *Issues and Applications of Case-Based Reasoning to Design*. Lawrence Erlbaum Associates, 1996.

A. Goel and E. Stroulia. Functional device models and model-based diagnosis in adaptive design. *International Journal of Artificial Intelligence in Engineering Design and Manufacturing*, 10, 1996.

Ashok Goel. Design, analogy, and creativity. *IEEE Expert and Intelligent Systems*, 12(3):49–56, May-June 1997. Special issue on AI in Design.

Jonathan Grudin and Steven E. Poltrock. Computer supported cooperative work and groupware. *Advances in Computers*, 45:269–320, 1997.

Marti A. Hearst, Mark D. Gross, James A. Landay, and Thomas F. Stahovich. Sketching intelligent systems. *IEEE Intelligent Systems*, 13(3):10–19, May-June 1998.

Chris Johnson. Time and the web. In *Transactions of the ACM Conference on Human-Computer Interaction*, 1994.

Jihie Kim, S. Ringo Ling, and Peter Will. Ontology engineering for active catalog. Technical report, The University of Southern California, Information Sciences Institute, 1997.

Wolfgang Kruger, Christian-A. Bohn, Bernd Frohlich, Heinrich Schuth, Wolfgang Strauss, and Gerold Wesche. The responsive workbench: A virtual work environment. *IEEE Computer*, 28(7):42–49, July 1995.

Wim Lamotte, Eddy Flerackers, Frank Van Reeth, Rae Earnshaw, and Joao Mena De Matos. Visinet: collaborative 3d visualization and vr over atm networks. *IEEE Computer Graphics and Applications*, 17(2):66–76, March-April 1996.

James A. Landay. *Interactive Sketching for the Early Stages of User Interface Design*. PhD thesis, Carnegie Mellon University, School of Computer Science, December 1996. #CMU-CS-96-201.

Valerie D. Lehner and Thomas A. DeFanti. Distributed virtual reality: Supporting remote collaboration in vehicle design. *IEEE Computer Graphics and Applications*, 17(2):13–18, March-April 1996.

M. L. Maher, M. B. Balachandran, and D. M. Zhang. *Case-Based Reasoning in Design.* Lawrence Erlbaum Associates, Mahwah, NJ, 1995.

Deborah A. Mitta and Patricia L. Flores. User productivity as a function of autocad interface design. *Applied Ergonomics*, 26(6):387–396, December 1995.

A. Pang and C. Wittenbrink. Collaborative 3d visualization with cspray. *IEEE Computer Graphics and Applications*, 17(2):32–41, 1997.

William C. Regli. Network-enabled computer-aided design. *IEEE Internet Computing*, 1(1):39–51, January-Feburary 1997.

William C. Regli and Daniel M. Gaines. An overview of the NIST Repository for Design, Process Planning, and Assembly. *International Journal of Computer Aided Design*, 29(12):895–905, December 1997.

K. Sycara and D. Navinchandra. Retrieval strategies in a case-based design system. In C. Tong and D. Sriram, editors, *Artificial Intelligence in Engineering Design*, volume II. Academic Press, July 1992.

K. Sycara, D. NavinChandra, R. Guttal, J. Koning, and S. Narasimhan. Cadet: A case-based synthesis tool for engineering design. *International Journal of Expert Systems*, 4(2):157–188, 1992.

David G. Ullman. *The Mechanical Design Process*. McGraw-Hill, Inc., 1992.

R.C. Zeleznik, A.S. Forsberg, and P.S. Strauss. Two pointer input for 3d interaction. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, Providence, Rhode Island, April 27-30 1997.