An Approach to Capturing Structure, Behavior and Function of CAD Artifacts

Lisa Anthony* William C. Regli† Jon E. John‡ Santiago V. Lombeyda§

Geometric and Intelligent Computing Laboratory
Department of Mathematics and Computer Science
Drexel University
3141 Chestnut Street
Philadelphia, PA 19104
http://gicl.mcs.drexel.edu

Abstract

This paper presents an approach to Computer-Aided Design that unites ideas from design with 3D layouts and knowledge engineering. Our goal is to capture the structure, behavior and function of CAD artifacts. We describe a software tool based on this approach, the Conceptual Understanding and Prototyping (CUP) environment, for capturing the design intent inherent in the design process and authoring design semantics in previously created artifacts. CUP records design ideas, based on functional, geometric and knowledge-based relationships among components in an electro-mechanical assembly. This design knowledge is stored using ontologies defined in the eXensible Markup Language (XML). The goal of this work is to enable users to navigate intricate product data and design knowledge-bases.

Keywords: Conceptual Design, Computer-Aided Conceptual Design, Assembly Modeling, Virtual Environments, Computer-Aided Design, Engineering Ontologies.

1 Introduction

Over the past two decades, Solid Modeling and 3D Computer-Aided Design have become critical elements of the product realization process, leading to successive generations of increasingly robust software tools for *detailed design*, simulation and engineering analysis. This paper presents our approach

*Email: lpa22@drexel.edu.
†URL: http://gicl.mcs.drexel.edu/regli. Email: regli@drexel.edu.
‡Email: jej22@drexel.edu.
§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

to the support of *knowledge-based* and *conceptual design* and a software environment, **Conceptual Understanding and Prototyping** (**CUP**), that embodies this approach.

Conceptual design is the initial phase of the product design process, when product development teams, consisting of design engineers, manufacturing engineers, marketing and management personnel, perform the “back-of-the-envelope” calculating, sketching and planning—fleshing out—of their initial product concept. Existing conceptual design tools fall, for the most part, into one of two categories: graphical tools for layout and sketching, or functional tools for textually describing desired product behavior. Further, the existing CAD paradigm is founded on techniques for the representation and manipulation of detailed geometric information. CUP provides a knowledge-based approach to support small teams of engineers in defining the semantics and layout for new products.

With CUP, users specify a spatial layout of components and sub-assemblies, as well as **structural**, **behavioral**, and **functional** (S-B-F) information about components and sub-assemblies. CUP also provides mechanisms for capturing textual information about the designer’s intent and preferences. In this way,

1. CUP is an environment that enables users to document structural information in a top-down fashion and record the function, structure and behavior of the intended artifact and its subcomponents;

2. CUP provides a tool for design teams to digitally describe the high-level semantics of an artifact, simultaneously creating the logical description of the device using domain ontologies and the 3D spatial layout of the individual components.

CUP combines several new trends in information technology from the areas of product data representation and product knowledge sharing with Java, Java 3D and XML. CUP creates a structured way in which product data semantics can be described as designs evolve. CUP extends computer support to the very early phases of product development, as well as improving interfaces between downstream CAx tools and applications.

2 Related Research

Conceptual Design The engineering design process encompasses all stages of a product’s lifecycle, from speculative exploration during conceptual design to maintenance and review during its evolution. Different sources in literature partition the process into different phases, but in general it is agreed that there are three main blocks of design: early design, detailed design, and design analysis. Shah and Mäntylä [32] consider the early phase of design to include functional design, *conceptual design*, and embodiment design. Pahl and Beitz [24], Ullman [39] and Buede [4] all place conceptual design in the early design phase, before the geometry of the product becomes important. Throughout the conceptual design process, the designer is more concerned with the function and overall structure of the design than with its detailed geometry. A basic engineering structure is agreed upon that will allow extension for the specific problem at hand [32]; as a result of this preliminary commitment to cost-bearing decisions, the design phase of the product realization process has a tremendous impact on the lifecycle cost of a product.

Pahl and Beitz, often considered the definitive treatment of engineering design [24], present conceptual design as an iterative, evolutionary, top-down process that ultimately yields a principle solution or concept for the product. The most effective way of developing the functional requirements of a product in great enough depth is by modularizing the problem via a process known as "functional decomposition" [39]. Given an overall function of the product, designers and engineers will break this into multiple sub-functions to be treated as individual subproblems.

Computer-Aided Conceptual Design A recent special issue of the *Journal of Computer-Aided Design* [18] and survey article [19] presents a contemporary assessment of existing research. Broadly speaking, existing approaches to and software tools for conceptual design fall into one of two categories: those based on *functional modeling* and those based on *sketching and layout*. The *functional modeling* approach finds its roots in traditional mechanical design, described above. *Sketch-based* conceptual design [22, 26, 16, 13, 12, 40, 20] incorporates ideas from computer graphics, user interface design and computer vision in order to facilitate the speedy creation of detailed 3D designs, usually from elementary 2D sketches. The goal of both methods is to improve support for the flexible generation of alternative design ideas at the early stages. The conceptual designs can then be considered for aesthetics, or used for engineering analysis, cost estimation or any number of other preliminary design assessments.

Many of the core ideas used in the *functional modeling* approach to conceptual design have been noted in the previous section. The mainstream CAD and design research communities have used this as a basis to develop software environments to support conceptual design. Much of the work in this area comes from academia as research in *Computer-Aided Conceptual Design* (CACD), and has lead to several prototype systems. These systems most often take a graph-based (nodes and edges) [31, 2, 9] approach to describing the engineering relationships among the elements in a mechanical assembly model, or other CAD-based structure [23].

Sketch and layout-based approaches to conceptual design cover a diverse collection of domains and involve research contributions from CAD, computer vision, computer graphics and a number of engineering disciplines. *Conceptual sketching* has been explored as a way to develop user interfaces [16], architectural drawings [26, 11] and 3D solid models for CAD [12, 26, 22, 34, 13, 20, 8]. One approach is to make these systems *feature-based*, where domain-specific features can be exploited to drive the recognition and refinement of the conceptual design into a detailed 3D model [3, 41, 42, 16].

Relationship of CUP to Existing Research. While there are many tools for detailed design (e.g., the major commercial CAD suites) there are few tools that support digital conceptual design or capturing formal, knowledge-based, descriptions of design characteristics. Having access to such information aids future designers in developing similar products (variant design) and in developing a completely new product (innovative design). Access to the design rationale of the original designers can help the efficiency of future design processes or shed light on something maintainers may not see the cause for and consider eliminating [30].

CUP unites the *graph-based* and *sketch-based* approaches to conceptual design and combines them with ideas from current work on knowledge-based systems. Within CUP's 3D virtual environment, users evolve the knowledge-level semantics of the product they are defining. From the knowledge-based design perspective, most relevant to our work is the CONGEN system of Gorti and Sriram [15]. Their work presented a design framework for symbol-to-form mapping which consists of deriving spatial relationships between objects as a consequence of the functional relationships, instantiating alternative feasible solutions subject to these relationships, and presenting the evolving descriptions of geometry. From the graphics and modeling perspective, Wallace and Jakiela [43] developed an

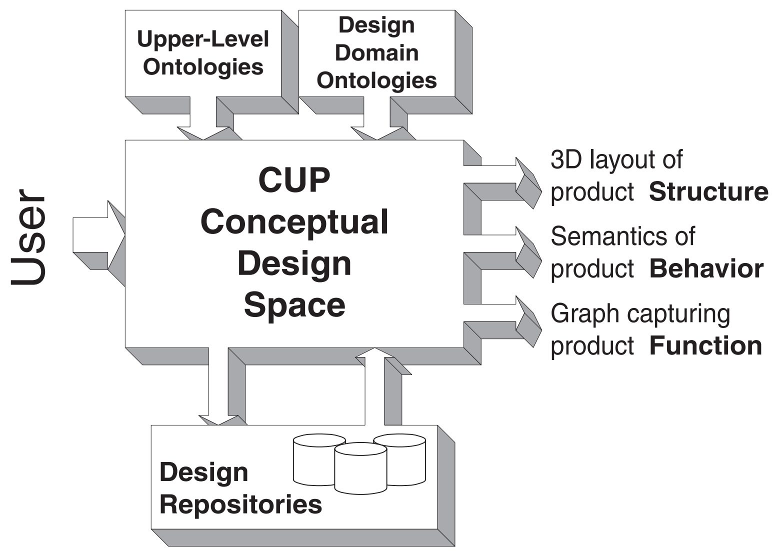


Figure 1: The conceptual design framework used in CUP.

experimental conceptual design tool for industrial designers capable of generating alternative design suggestions based on user input. Other relevant work includes the Virtual Assembly Design Environment (VADE) of Jayaram et al. [21]; and the ConceptualCAD© [1] project at Arizona State University, which is studying how to use knowledge-based templates, called design exemplars, to create conceptual models of product structure and function.

3 Approach

Our approach to conceptual design considers the design process parallel to the artificial intelligence problem of how to create shared semantics for knowledge-based systems. In the past, shared semantics were usually imposed *a priori*, before the development of agents could begin. In the expert systems era, semantics were often hard-coded—resulting in brittle systems and narrow problem solutions. The need for common knowledge-sharing standards has been cited as a major obstacle to achieving true concurrent engineering [7].

The current trend is to support the development of shared semantics. Paralleling the evolution of the Internet and its protocols, contemporary work in agent-based computing and knowledge representation seeks to create formal frameworks and tools for the rapid formation, modification and integration of ontologies and knowledge-bases. In much the same manner as a team of digital agents works toward a common goal, a team of designers progresses toward a final design in part by achieving collective group semantics.

CUP takes an analogous approach: a team of designers engages in a collaborative problem-solving exercise in order to define the semantics of a new product. With CUP the design process is one in which designers author an assembly design, specifying its structure and layout in 3D while annotating the relationships among all the entities in the assembly with reference to base ontologies. The base, or upper-level, ontologies CUP employs describe fundamental structure-behavior-function properties common in mechanical design systems. Hence, as designers create the 3D "back of the envelope" sketch of their design, they also are authoring the set of logical sentences that describe the semantics of the assembly with reference to these base ontologies. It is important to note that this approach, and the CUP system, is not specific to any one design domain. To modify CUP to perform design for the Architecture/Engineering/Construction (AEC) domain, one would simply need to provide suitable 3D conceptual design primitives and references to appropriate base ontologies for describing relationships in the AEC world. This approach is similar to that favored by the knowledge engineering community for development of large, shared knowledge-bases [5].

Shown in Figure 1, users of CUP author design knowledge using 3D primitives and existing components from design repositories while describing engineering semantics with a domain ontology and a shared upper-level ontology. The output of a completed CUP session is a product knowledge-base that includes an XML-based description of product structure, behavior and function with respect to the shape primitives and their relationships as described with the ontologies.

3.1 Technical Background

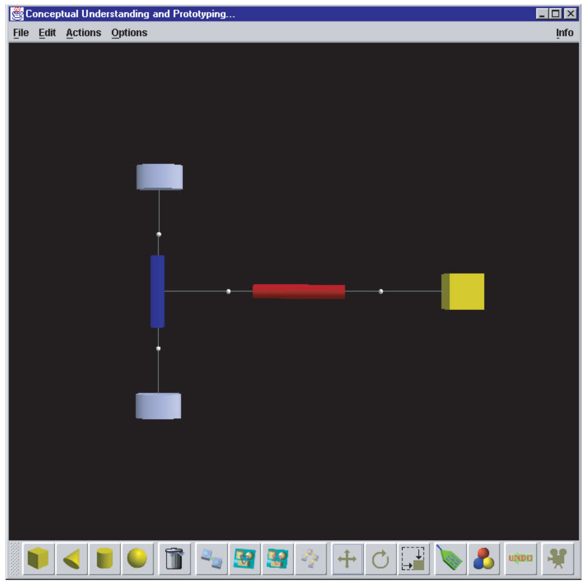
Designers often model preliminary designs without having detailed specifications for every component and feature, being more concerned with specifying design intent and the interrelationships between the components and features. CUP's basic approach to the support of this activity involves some preliminary CAD capabilities: a user defines the assembly's major components in three dimensions, arranging the primitive shapes in a layout which makes logical sense at this early stage of design. As discussed in Section 2, the emphasis is on the function of the components in the assembly and/or the flow of a substance or material from one or more components to some other. That is, the laying out and defining of the structural, behavioral, and functional relationships in the assembly are more salient at this point in the design process than geometry, size, shape, rotation, even relative orientation. It is the structure-behavior-function knowledge, more 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 this search of the design space.

To represent the concept that lies behind the preliminary layout the user constructs, users "mark up" the design as they create the 3D layout of physical components. In this way, CUP captures a semi-formal notion of design intent. For example, why does the `shaft` connect to the `motor`? What does the link between the `axle` and `wheel_1` represent? The semantics of these queries and their answers can be expressed in terms of the incorporated ontologies.

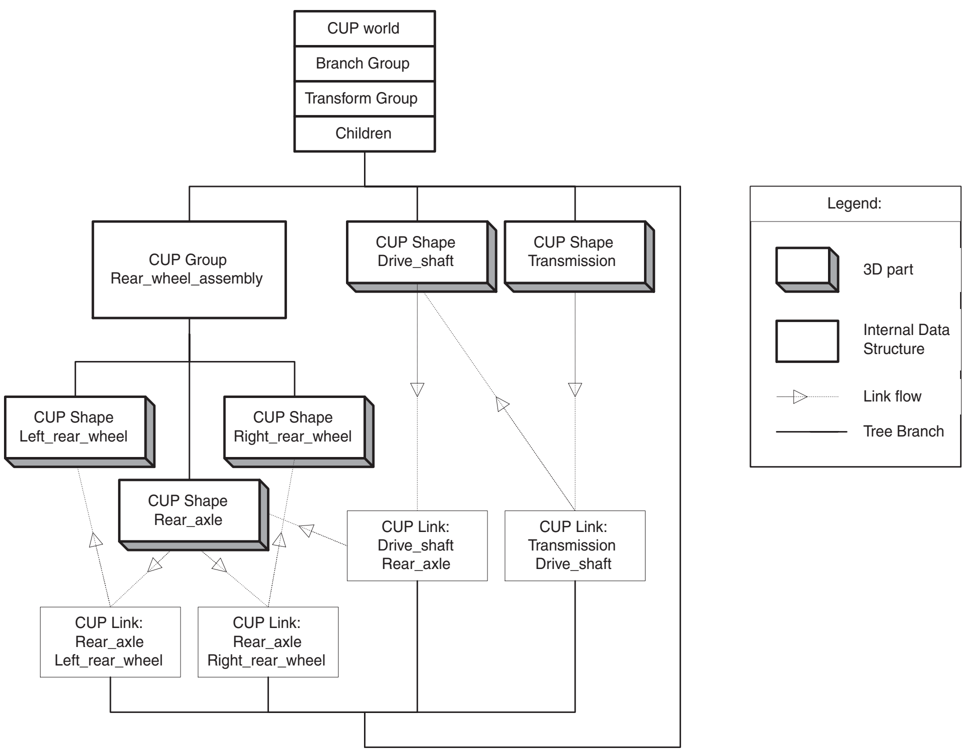
A perfunctory survey of the literature regarding functional mapping and representation in assembly design [37, 15, 14, 19, 33, 17, 25, 36] presented three basic, common entities: *function*, that is, what an object (component, group of components, or subassembly) does within the overall assembly; *behavior*, the observable actions by which it accomplishes this function; and *structure*, specifically what attributes of the physical object (e.g., physical constraints, materials, and so on) are related to helping it achieve this behavior. With function, some literature [37, 36] also mentioned an entity known as *flow*, the substances or materials which are the inputs and/or outputs of the object's function.

3.2 System Overview and User Interface

CUP consists of a very elemental CAD system in which the user designs without focusing on details, yet still defines enough information that a full design can evolve from this work. For this purpose, the CUP environment supports the creation and manipulation of a set of primitive three-dimensional



(a) A simple drive train assembly.



(b) Internal data model in CUP representing the drive train.

Figure 2: A design model of a drive train/transmission assembly built in CUP.

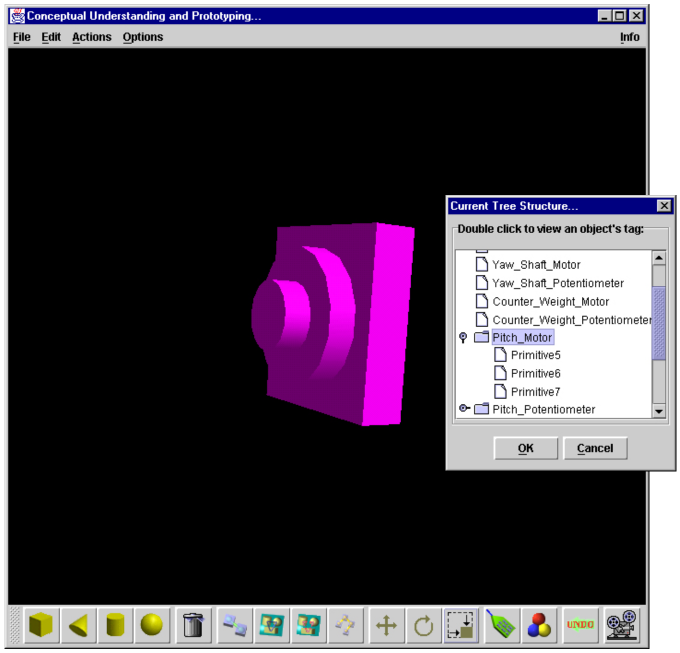


Figure 3: The CUP Tree View navigates and manipulates the design in a hierarchy.

objects (i.e., blocks, cylinders, spheres, and frustums) and the importation of more complex shape entities (i.e., VRML97 data structures, such as indexed `FaceSets`). The user can assign further characteristics of these objects textually, as well as correlate relationships between these objects. This is accomplished by offering to the user a limited CAD environment with the abilities to **tag**, **link**, and **group** the objects. A simple example of an object created in CUP and the relationships among its components is shown in Figure 2 (a). In part (b) of that figure, the internal data model of the assembly is depicted. Arrowed-edge connections indicate function-flow relationships among the components; plain-edge connections show the internal structure of the 3D world and its objects.

The CUP interface consists of the central pane and a Java 3D-based viewer capable of displaying a design (the CUP "world"). There are controls in menus and toolbars for creation and deletion of primitives, groups, and links; editing of object properties such as scale, rotation, color, and position; and editing of object tags. Also available is a two-dimensional **tree view** of the current world. This tree view, shown in Figure 3, displays the hierarchical structure of the world's objects and groups, and offers the capability to view and edit tags from this window.

All objects in the design space have a one-to-one relationship with a tag containing information about their structure, behavior, and function, as well as text-based descriptions about the components. Objects may be linked to each other with either uni-directional or bi-directional links that establish functional ties among components. An example of the linking and tagging facility is shown as part of Figure 4. Finally, a selection of multiple objects may be grouped to create a conglomerate which has a tag of its own.

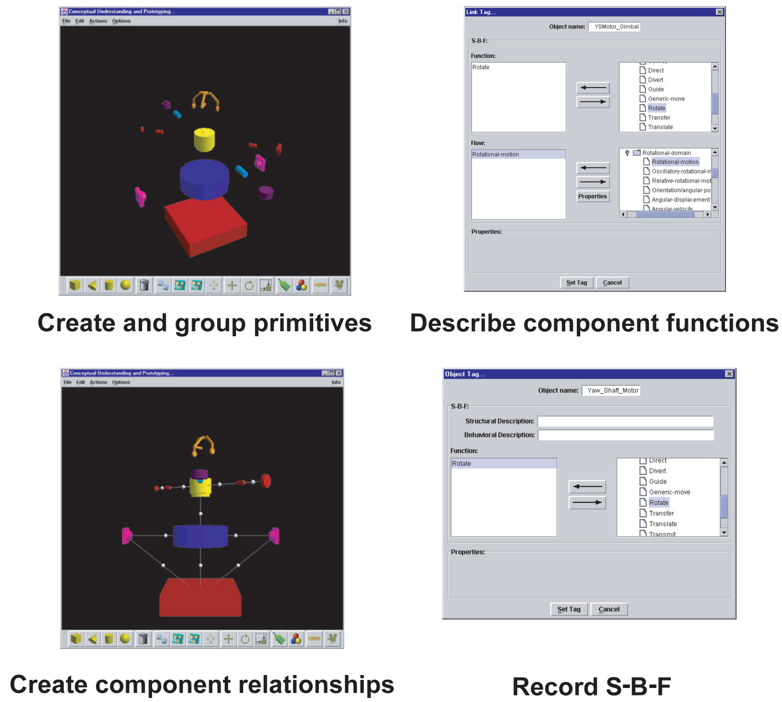


Figure 4: A step-by-step view of CUP's support of design activities.

3.3 Representation of Structure, Behavior and Function

CUP has been developed so that users can "plug in" their own representation schema. Our implementation was tested using a taxonomy proposed by Szykman et al. [37] of the National Institute of Standards and Technology (NIST) regarding the representation of function in engineering design. In this pre-standard, **function** and **flow** are defined as the two main quantities that are necessary in order to represent design intent properly. The novel aspect of the NIST work is that it defines a representation based on the separation of function and flow into two distinct categories. Reasons given in their work for this decision are to eliminate duplicate entities within the taxonomy itself, to facilitate changing any component of an artifact's function independently of the others, and to promote the inclusion of functions which are not associated with a flow. These three reasons are referenced in their work as weaknesses in taxonomies which appeared in the literature prior to their own work.

NIST has specified a relatively complete schema for both function and flow within the mechatronic domain. **Function** is the actual function performed by any given component of an assembly, or by a link between two or more components. Most functions have an associated flow. **Flow** is the definition of what material or quantity is being transferred by its associated function, and in which direction it is being transferred (i.e., *from* which component *to* which component).

At present, we use NIST's taxonomies for both function and flow, which provide over 130 functions and over 100 flows, respectively. In our system's data representation, we have included two additional quantities: structure, or the construction or shape of the component, and behavior, or the observable action of the component. Thus, users of CUP specify any or all of these quantities in an S-B-F representation of their designs.

Once an assembly is created and its pertinent S-B-F information defined, it becomes necessary

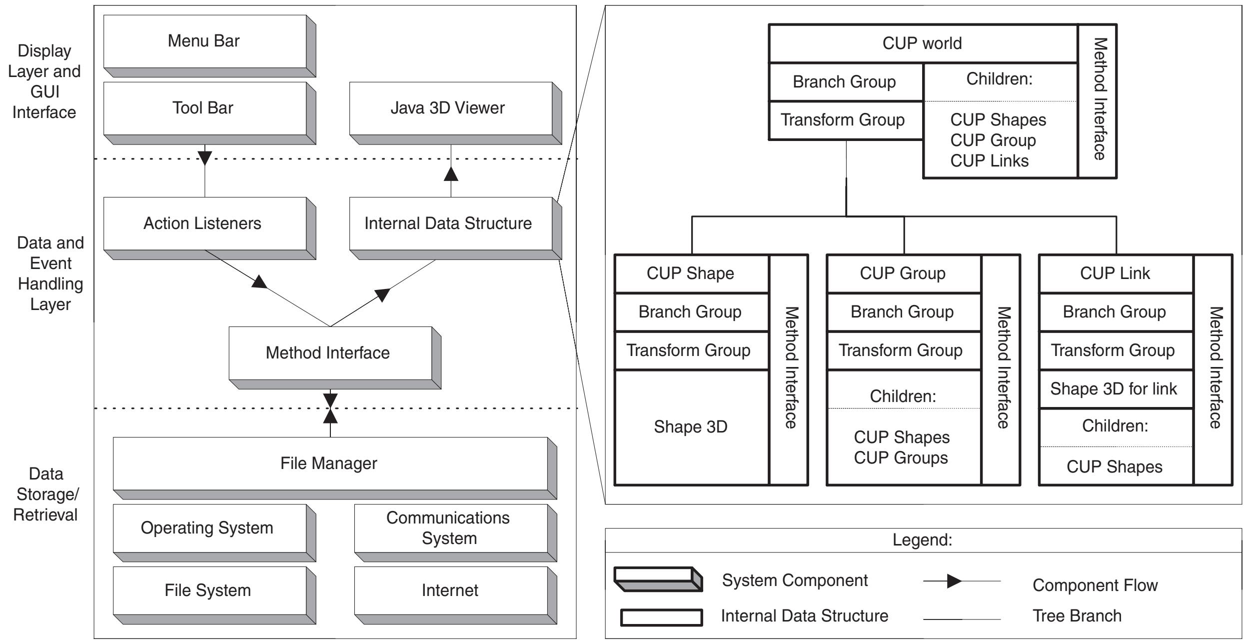


Figure 5: CUP's architectural internals: modules, data types, and their relationships.

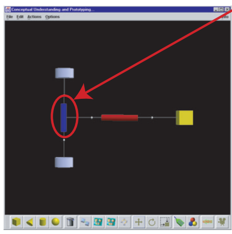
to consider how this information could be stored. Our system uses the eXtensible Markup Language (XML) [6], a dynamic language similar to HTML and SGML for the structured representation of data with user-defined tags. The robust extensibility of XML makes it a perfect fit with our need for a hierarchical semantically embedded representation of design data. Figure 6 shows two pieces of XML code that were created by CUP in reference to the drive train model shown in Figure 2. In part (a), the code shown pertains to the circled primitive, a cylindrical shape whose function is to perform rotational motion. In part (b), the circled link is the representation of the rotational motion transferred by the relationship from the axle to the rear wheel.

3.4 Implementation Specifics

The Java 3D internals, which are based on the VRML (Virtual Reality Modeling Language) structure, are shown in Figure 5. At the top level there is a root `Branch Group` with all of the "children" in the VRML world attached to it. These children may be other `Branch Groups`, `Transform Groups`, and various leaf nodes, which may contain `CUP Shapes`, `CUP Groups`, and `CUP Links`.

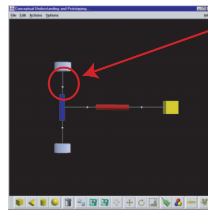
CUP is based on cross-platform technologies: the Virtual Reality Modeling Language (VRML) and Sun Microsystems' Java language. The intention is to create a downloadable tool which the user can run inside the Java Runtime Environment software from Sun on a wide variety of platforms. CUP currently runs on Pentium II/III-based machines with Microsoft Windows NT 4.0 and Sun Ultra workstations with Solaris 2.6/7 that are equipped with Java Runtime Environment version 1.2 and the VRML97, Java3D, Java Help, and XML 1.0 extensions.

CUP model worlds are saved in XML format (`.xml`), structuring the S-B-F information about each component of the model within the model's file. CUP can also import geometries defined in



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE World SYSTEM
"http://edge.mcs.drexel.edu/GICL/projec
ts/CUP/application/cup.dtd">
<World>
...
<Shape name="Axle" type="Cylinder">
    ...
    <TagInfo>
      <StructureList>
        <Item value="shaft"/>
      </StructureList>
      <FunctionList>
        <Item value="Rotate"/>
      </FunctionList>
      <BehaviorList>
        <Item value=""/>
      </BehaviorList>
    </TagInfo>
  </Shape>
...
</World>
```

(a) Description of the rear axle of a drive train.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE World SYSTEM
"http://edge.mcs.drexel.edu/GICL/projec
ts/CUP/application/cup.dtd">
<World>
  <Link name="axle_wheel_1">
    <TagInfo>
      <FunctionList>
        <Item value="Transfer"/>
      </FunctionList>
      <FlowList>
        <Flow value="Rotational-motion"
          source="Axle" destination="Wheel_1"/>
      </FlowList>
    </TagInfo>
    <Targets>
      <Target ref="Wheel_1"/>
      <Target ref="Axle"/>
    </Targets>
  </Link>...
</World>
```

(b) A link for the function-flow pair relating the axle to the rear wheel.

Figure 6: XML code describing the drive train assembly modeled in CUP.

VRML format (`.wrl`)—such as those created by other CAD systems—and act as an S-B-F editor for CAD models which otherwise would not store this information. This import functionality is shown in the later examples and Figure 8.

4 Using CUP: Potential Scenarios

The following scenarios demonstrate the applications of CUP to the design world.

Using CUP During the Design Process A simplified missile seeker assembly [10] might be one of many dozens of designs stored in a corporate design database or 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—starting points for variant design.

The design scenario shown in Figure 7 demonstrates how CUP might be used by a designer to describe quickly in 3D the major components and structural relationships in the assembly. The figure lays out the step-by-step process from component design to functional description to final assembly. The resultant conceptual design then becomes the basis for detailed CAD (in this case, in Bentley's Microstation Modeler). Rather than performing detailed CAD modeling to create a draft design (which for this model took several days), designers can build a design in a matter of minutes. In this way, CUP could help designers capture the design intent and build a structure-behavior-function (S-B-F) model of the artifact to be created. Finally, this CUP design can be used as a starting point for further refinement or as a query-by-example to a design knowledge-base, in which the components and relationships in the query are compared against those digitally archived [28].

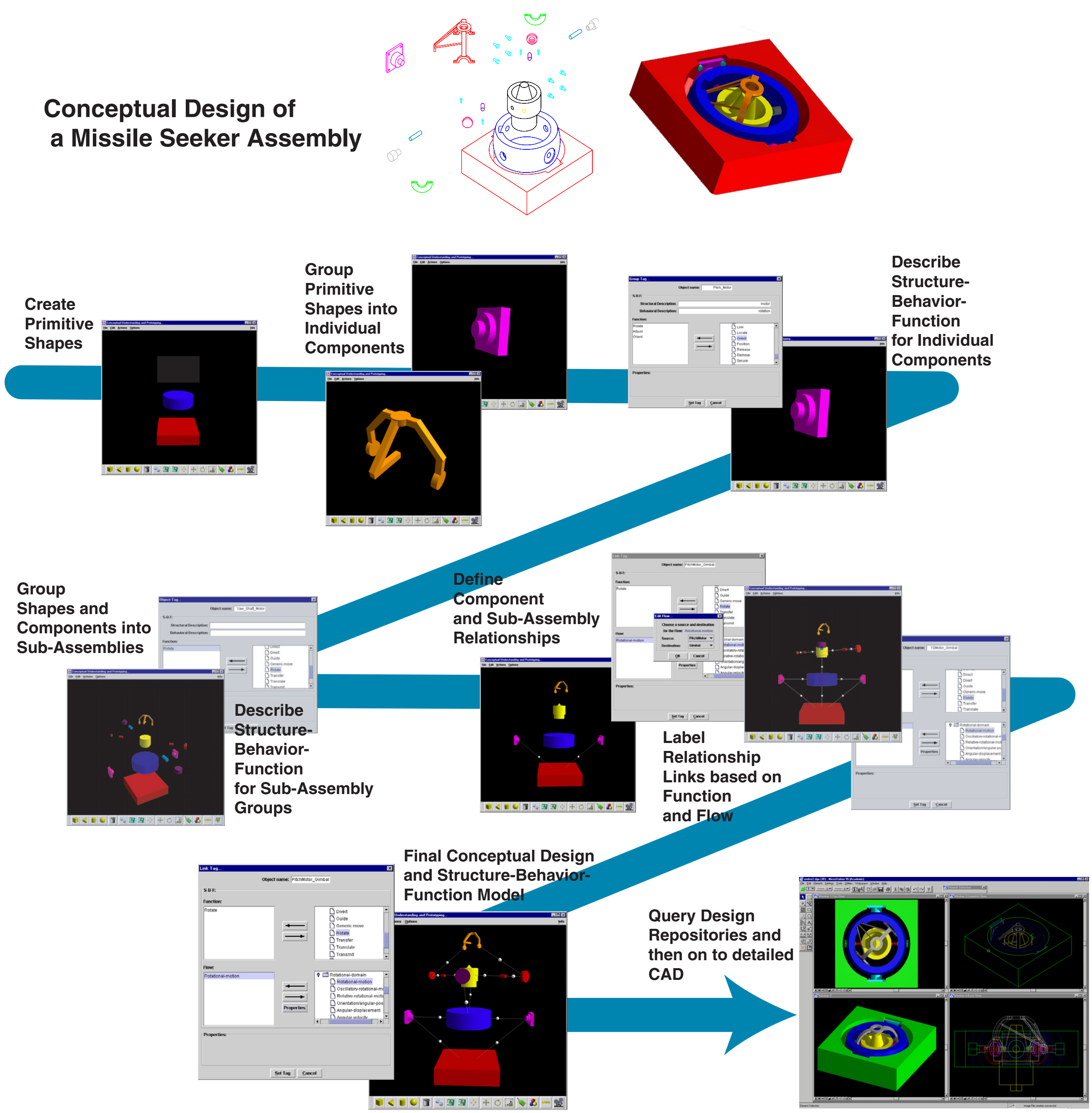
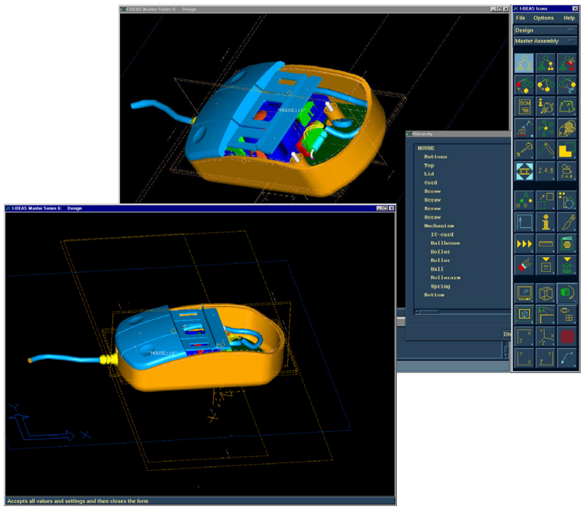
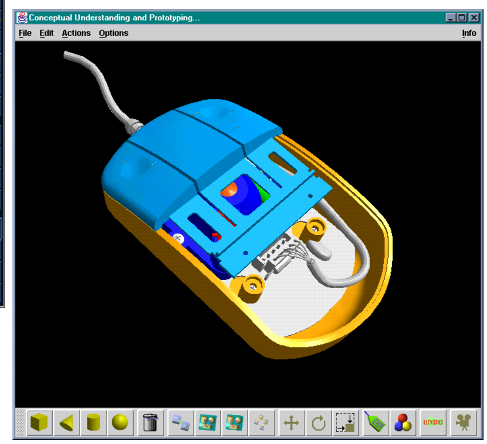


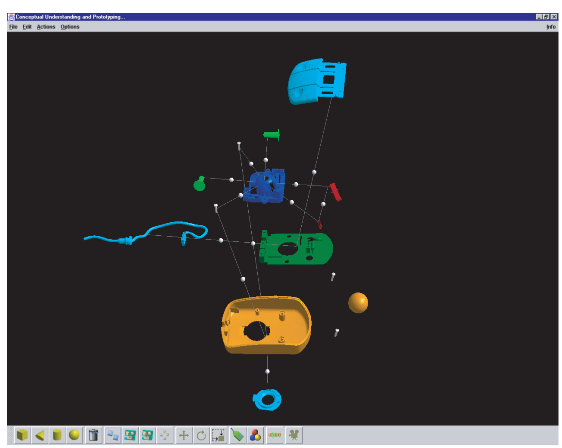
Figure 7: The design process of the missile seeker within CUP.



(a) A computer mouse designed in SDRC I-DEAS.



(b) The mouse as it appears imported into CUP.



(c) The mouse with S-B-F markups created with CUP.

Figure 8: The use of CUP to annotate existing CAD models.

Using CUP to Augment Traditional CAD Most traditional CAD environments support *detailed design*, where the precise product shape is specified. For these environments, CUP can be used to augment shape and assembly data with engineering semantics. CUP is able to import complex geometries (i.e., in VRML format), and annotate them with descriptions of structure, behavior and function. In this way, CUP is used to describe the engineering knowledge implicitly authored in traditional CAD tools. Shown in Figure 8 is an example of annotating the design of a computer mouse from SDRC's CAD suite, I-DEAS Master Series. The output of I-DEAS is a detailed solid model of the design, along with assembly relationships and kinematic constraints. CUP can then be used to import the shape models and allow users to author the knowledge-level S-B-F information for these designs. In Figure 8 (a), the electro-mechanical design of a computer mouse is shown in SDRC's I-DEAS. The SDRC system encodes the shape and assembly design knowledge; when imported into CUP (Figure 8 [b]) the semantics of this design can be described (Figure 8 [c]) using the domain ontologies loaded into CUP.

5 Conclusions

This paper described and demonstrated an approach to knowledge-based design that supports small teams of engineers in defining the semantics and layout for new products. We have implemented a tool to demonstrate this approach: **CUP**[1], a system for the computer-aided design and semantic annotation of assemblies. In the development of this tool, we have created a 3D conceptual modeling system and introduced methods for integrating the description of formally represented engineering knowledge (function and behavior) with graphical conceptual modeling. We believe that CUP, and tools like it, will become essential components of future design environments, enabling users to create a knowledge-level description of the design without having to perform detailed CAD and solid modeling [38].

We are extending our approach and the CUP system in several ways. Future work includes enabling multi-user design with CUP, with shared 3D JavaSpaces; and enabling access to design repositories [28, 29, 27, 35], helping designers navigate large-scale engineering digital libraries and knowledge-bases of engineering information and solid models. It is our eventual goal to integrate CUP with the National Design Repository (`http://www.designrepository.org`) and provide facilities for users to add their own ontology information to the Repository's knowledge-base.

While continuing our research, it is important to note that there are many open research issues in conceptual design that will require input from the wider community. One problem in particular is that of design validation: given the high-level functional description, such as that provided by a tool like CUP, can we create design validation or compilation software that will check the composed system with respect to design requirements? A second major community-wide problem is the lack of shared semantics and knowledge representations [38]. While standards such as STEP have begun to provide reliable means of data exchange, the community lacks detailed ontological descriptions of design and manufacturing knowledge. While the approach in this paper, demonstrated with the NIST S-B-F taxonomy, is completely generalizable, there are no other sufficiently detailed design ontologies (from any design discipline) with which to test it.

[1]CUP is available for download and use at `http://edge.mcs.drexel.edu/CUP`. Some example models are also available.

Acknowledgements. 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 AT&T Labs. 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. The authors would also like to thank the ASME JCISE reviewers for their many helpful comments.

References

[1] ConceptualCAD©. Arizona State University, Design Automation Lab, 2001.

[2] L. Al-Hakim, A. Kusiak, and J. Mathew. A graph-theoretic approach to conceptual design with functional perspectives. *International Journal of Computer Aided Design*, 32(14):867–875, December 2000. Special issue on Conceptual Design.

[3] G. Brunetti and B. Golob. A feature-based approach towards an integrated product model including conceptual design information. *International Journal of Computer Aided Design*, 32(14):877–887, December 2000. Special issue on Conceptual Design.

[4] Dennis M. Buede. *The Engineering Design of Systems - Models and Methods*. John Wiley and Sons, New York, NY, 2000. ISBN 0-471-28225-1.

[5] Paul Cohen, Robert Schrag, Eric Jones, Adam Pease, Albert Lin, Barbara Starr, David Gunning, and Murray Burke. The DARPA high-performance knowledge bases project. *AI Magazine*, 19(4):25–49, Winter 1998.

[6] World Wide Web Consortium. The eXtensible Markup Language (XML). `http://www.w3c.org/XML`, 2000.

[7] Mark R. Cutkosky, Jay M. Tenenbaum, and Jay Glicksman. Madefast: Collaborative engineering over the internet. *Communications of the ACM*, 39(9):78–87, September 1996.

[8] T. H. Dani and R. Gadh. Covirds: A new approach to concept shape modeling via a virtual environment. In *High Performance Computing Conference*, pages 348–353, April 1997. Atlanta, GA.

[9] Y.-M. Deng, G.A. Britton, and S.B. Tor. Constraint-based functional design verification for conceptual design. *International Journal of Computer Aided Design*, 32(14):889–899, December 2000. Special issue on Conceptual Design.

[10] A. Diaz-Calderon, C. Paredis, and P. K. Khosla. A modular composable software architecture for the simulation of mechatronic systems. In *ASME Computers in Engineering Conference*, Atlanta, GA, Sept. 1998.

[11] Dirk Donath and Holger Regenbrecht. VRAD — virtual reality aided design in the early phases of the architectural design process. In *CAAD Futures*, pages 313–322. Center for Advanced Studies in Architecture, National University of Singapore, 1995. ISBN: 9971-62-423-0.

[12] 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.

[13] L. Eggli, C. Y. Hsu, B. D. Bruderlin, and G. Elber. Inferring 3d models from freehand sketches and constraints. *Computer-Aided Design*, 29(2):101–112, February 1997.

[14] J. S. Gero. Design prototypes: A knowledge representation schema for design. *A.I. Magazine*, pages 26–36, Winter 1990.

[15] S. R. Gorti, A. Gupta, G. J. Kim, R. D. Sriram, and A. Wong. An object-oriented representation for product and design processes. *Computer-Aided Design*, 30(7):489–501, June 1998.

[16] 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.

[17] B. Henson and N. Juster. Towards an integrated representation of function, behavior, and form. In *Computer Aided Conceptual Design*, pages 95–111, Apr 11-13 1994. Lancaster, UK.

[18] W. Hsu and B. Liu. Conceptual design: issues and challenges. *International Journal of Computer Aided Design*, 32(14):849–850, December 2000. Special issue on Conceptual Design.

[19] Wynne Hsu and Irene M. Y. Woon. Current research in the conceptual design of mechanical products. *International Journal of Computer Aided Design*, 30(5):377–389, 1998.

[20] T. Hwang and D. Ullman. The design capture system: Capturing back-of-the-envelope sketches. *Journal of Engineering Design*, 1(4):339–353, 1990.

[21] Sankar Jayaram, Uma Jayaram, Yong Wang, Hrishikesh Tirumali, Kevin Lyons, and Peter Hart. VADE: a virtual assembly design environment. *IEEE Computer Graphics and Applications*, 19(6), November-December 1999.

[22] H. Lipson and M. Shpitalni. Conceptual design and analysis by sketching. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)*, 14(5):391–402, November 2000.

[23] C. J. Moore, J. C. Miles, and D. W. G. Rees. Decision support for conceptual bridge design. *Artificial Intelligence in Engineering*, 11(3):259–272, July 1997.

[24] G. Pahl and W. Beitz. *Engineering Design—A Systematic Approach*. Springer, London, UK, 2nd edition, 1996.

[25] S. Prabhakar and A. K. Goel. Functional modelling for enabling adaptive design of devices for new environments. *Artificial Intelligence in Engineering*, 12(4):417–444, October 1998.

[26] S.F. Qin, D.K. Wright, and I.N. Jordanov. From on-line sketching to 2d and 3d geometry: a system based on fuzzy knowledge. *International Journal of Computer Aided Design*, 32(14):851–866, December 2000. Special issue on Conceptual Design.

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

[28] William C. Regli and Vincent Cicirello. Managing digital libraries for computer-aided design. *International Journal of Computer Aided Design*, 32(2):119–132, Februrary 2000. Special Issue on *CAD After 2000*. Mohsen Rezayat, Guest Editor.

[29] 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.

[30] William C. Regli, Xiaochun Hu, Michael Atwood, and Wei Sun. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 2001. To appear.

[31] David Serrano and David Gossard. Tools and techniques for conceptual design. In Christopher Tong and Duvvuru (Ram) Sriram, editors, *Artificial Intelligence in Engineering Design: Design Representation and Models of Routine Design*, volume I, chapter 3, pages 71–116. Academic Press, 1250 Sixth Ave, San Diego, CA, 1992. ISBN 0-12-660561-0.

[32] Jami J. Shah and Martti Mäntylä. *Parametric and Feature-based CAD/CAM*. John Wiley and Sons, Inc., New York, New York, 1995. ISBN 0-471-00214-3.

[33] Y. Shimomura, M. Yoshioka, H. Takeda, Y. Umeda, and T. Tomiyama. Representation of design objects based on the functional evolution process model. *Journal of Mechanical Design*, 120(2):221–229, June 1998.

[34] Malgorzata Sturgill, Elaine Cohen, and Richard F. Riesenfeld. Feature-based 3-d sketching for early stage design. In A. A. Busnaina, editor, *ASME Computers in Engineering Conference*, pages 545–552, New York, NY 10017, September 17-20, Boston, MA 1995. ASME.

[35] S. Szykman, C. Bochenek, J. W. Racz, J. Senfaute, and R. D. Sriram. Design repositories: Engineering design's new knowledge base. *IEEE Intelligent Systems*, 15(3):48–55, May/June 2000.

[36] S. Szykman, J. W. Racz, and R. D. Sriram. The representation of function in computer-based design. In *ASME Design Engineering Technical Conferences, 11th International Conference on Design Theory and Methodology*, New York, NY, USA, September 12-16, Las Vegas, NV 1999. ASME, ASME Press. DETC99/DTM-8742.

[37] S. Szykman, J. Senfaute, and R. D. Sriram. Using xml to describe function and taxonomies in computer-based design. In *ASME Design Engineering Technical Conferences, 19th Computers and Information in Engineering Conference*, New York, NY, USA, September 12-16, Las Vegas, NV 1999. ASME, ASME Press. DETC99/CIE-9025.

[38] Simon Szykman, Ram D. Sriram, and William C. Regli. The role of knowledge in next-generation product development systems. *ASME Transactions, the Journal of Computer and Information Science in Engineering*, 1(1), March 2001.

[39] David G. Ullman. *The Mechanical Design Process*. McGraw-Hill, Inc., 1997. ISBN 0-07-065756-4.

[40] Casper G. C. van Dijk. *Interactive Modelling of Transfinite Surfaces with Sketched Design Surfaces*. PhD thesis, Delft University of Technology, 1994.

[41] P. A. vanElsas and J. S. M. Vergeest. Displacement feature modelling for conceptual design. *Computer-Aided Design*, 30(1):19–27, January 1998.

[42] P. A. vanElsas and J. S. M. Vergeest. New functionality for computer-aided conceptual design: The displacement feature. *Design Studies*, 19:81–102, January 1998.

[43] D. R. Wallace and M. J. Jakiela. Automated product concept design: Unifying aesthetics and engineering. *IEEE Computer Graphics and Applications*, pages 66–75, July 1993.