12

Object Modeling to Localize Knowledge for Feature Interrelationships

T. J. Peters, S. A. Demurjian,
R. McCartney, D. M. Needham
Department of Computer Science and Engineering
University of Connecticut
Storrs, Connecticut 06269-3155 USA
(860) 486 - 3719, FAX (860) 486 - 4817
{tpeters,steve,robert,needham}@eng2.uconn.edu

Abstract

Effective object-oriented software support of knowledge intensive CAD requires extension of the object-oriented paradigm ***because*** its hierarchies are fundamentally based upon ancestral relations. ***Propagations*** are proposed as an extension to facilitate modeling of interrelationships between features. Such object modeling provides an effective software tool for localized perspectives upon design knowledge stored in rule and case bases. These local views show promise as a means to avoid the combinatorial explosions common to knowledge based systems, as will be discussed with respect to a completed industrial example and to ongoing work on rapid prototyping data.

Keywords

Object modeling, knowledge bases, CAD, features, case-based reasoning.

1 INTRODUCTION: OBJECTS, KNOWLEDGE

Features are fundamental CAD knowledge representations. Feature-based rules and cases can embody semantic design information. It is well known that at-

M. Mäntylä et al. (eds.), *Knowledge Intensive CAD*

tempting to capture all such knowledge within a monolithic knowledge base can lead to combinatorial explosions.

Providing broadly applicable object-oriented software tools to allow designers to explicitly represent important semantic information *between* features is the motivation for this work. Our view is that such interrelationships need to be supported via a generic software language construct, defined below.

Definition: A *Propagation* is a software construct for representing interrelationships between two non-ancestrally related objects.

Our work contributes beyond constraints, in its modularity and expressive power. Namely, constraints systems frequently permit modeling only within a restricted language, having both permitted and prohibited interrelationships. For example, a geometry constraint system might represent points, lines and arcs, with parallelism allowed between two lines, while excluding two lines from being simultaneously parallel and perpendicular. In our approach to providing modular expressive power, we do not predetermine any such restrictions. The user has both more freedom and more responsibility to consistently model appropriate interrelationships. To afford such broad flexibility, we forsake the consistency protections that can be provided within specialized domain constraint systems.

As such, each propagation is an independent, third-party object, and causes mediating software to fire whenever a related object is altered. These propagations can effect a local partitioning of a knowledge base in that the mediating methods associated with a particular propagation could invoke only some subset of the design knowledge.

For instance, a monolithic knowledge base would contain references for holes, slots, fillets, bosses, drilling, milling, etc. A significant value of object-oriented techniques is the encapsulation of methods within their associated objects. Our work extends that metaphor by allowing those methods associated with interrelationships between holes to be encapsulated within a propagation object for holes. This provides the basis for the extraction of the knowledge about holes from the rest of knowledge base. Thus, access for knowledge about holes would need not traverse aspects related to other types. While any knowledge base could be so segmented by a knowledge engineer, our software environment (ADAM) provides a rigorous methodology to support this process, which, otherwise, might be a tedious, error-prone software development exercise. Our efforts so far have concentrated upon prototyping user interactions and object-oriented extensions necessary to support such modeling. We have completed 'proof of concept' studies on industrial data with practicing engineers to validate that our methodology is supportive of their conceptual design process. We report briefly here on that success, while emphasizing its role as the first step relative to effective partitioning of large knowledge bases.

To begin our consideration of such large knowledge bases, we have chosen the domain of rapid prototyping. Our reasons for this choice were that the *de facto* standard representation for rapid prototyping (known as the .STL file format), typically has hundreds of thousands of features and that the authors are simultaneously developing specialized techniques to consistently maintain the conceptual design knowledge implicitly captured within .STL files.

The remainder of this paper is divided into five sections. In Section 2, related work is reviewed. Section 3 describes ADAM. Section 4 discusses a completed prototype. Section 5 outlines ongoing work with rapid prototyping data to test the ability of these tools to 'scale up'. Section 6 explores how techniques from artificial intelligence can be integrated with propagations. Finally, Section 7 offers concluding remarks. Biographical sketches of the authors follow.

2 RELATED WORK

Our citations to the supporting literature on features and intelligent design are admittedly quite terse, so as to focus our emphasis upon the specific new issues expressed herein.

The literature on features is quite voluminous at this juncture in time. Hence, there will be no attempt to effectively summarize it here. Rather, the reader is referred to a recent monograph (Shah and Mäntylä, 1995), a comprehensive survey (Shah 1991) and selected papers (De Floriani 1989), (Drake and Sela, 1989), (Glovin and Peters 1987), (Gossard, Zuffante, and Sakurai 1988), (Gupta, Regli, and Nau 1994), (Henderson 1984), (Hernandez et al 1991), (Mäntylä, Nau, and Shah 1996), (Requicha 1996), (Rosen 1993), (Shah and Rogers 1993), (Vandenbrande and Requicha 1990). Collectively, these sources provide a rich set of references for the interested reader. Similarly, there exists a wealth of literature on applications of artificial intelligence to design and we note only those that have most strongly influenced our own work (Dixon 1986), (Dixon et al 1987), (Orelup et al 1988), (Hayes and Sun 1995), and (Prabhakar and Goel 1996).

The growing recognition of the need to express interrelationships for design is expressed in the diverse contexts of chain models (Palmer and Shapiro 1993) and meta-models (Henderson and Taylor 1993). Also, constraints are another form of expression of interrelationships in design (Chen and Hoffmann 1995), (Dighe, Jakiela, and Wallace 1993), (Fu and de Pennington 1993), and (Hoffmann and Juan 1993).

The distinctive aspect of our research is our focus upon feature interrelationships as a path towards powerful integration of object-oriented modeling and knowledge based support.

3 SOFTWARE ENGINEERING: ADAM

This work has used our software engineering environment, Active Design and Analyses Modeling (ADAM) to facilitate our prototyping of non-ancestral interrelationships within the object-oriented design paradigm. ADAM is a graphic/textually based object-oriented software engineering environment (El Guemhioui 1994) and (Ellis 1994). From language-independent designs, ADAM can automatically generate code in multiple object-oriented languages, including: C++, Ontos C++, Ada83, Ada95, Eiffel, and a LISP-based dialect. The software designer uses ADAM to define object types and to perform analyses on the emerging software design. ADAM has been our software engineering test-bed for examining the language support issues needed to develop propagations as a generic design model capability, so as to represent *any* non-ancestral interrelationship between objects. Within this modeling test-bed, our CAD test cases have been on industrial examples.

4 PROOF OF CONCEPT INDUSTRIAL STUDY

Our proof of concept test part was a disc, with multiple holes (Brett et al 1996). Our discussion here will focus upon one instance of feature interrelationships. Figure 1 illustrates the propagation entity which enforces the minimum spacing between two holes types, the Heel Screw Hole feature and the CB Screw Hole feature.

The radius of either hole is changed by calling a SetRadius method which in turn triggers the associated propagation (depicted by a message arrow). The propagation then gets the value of the radii of each hole by their respective GetRadius methods (depicted by an access arrow). If the new radius does not violate the distance test of the propagation, then the radius is set by the propagation with a call to its SetRadius method (depicted by a modify arrow). This propagation only accesses the design rule for hole spacing, which is encapsulated as a method of the propagation object depicted by a circle in Figure 1. Rules can be localized to different propagations, thereby partitioning the rule base. Such localized views into an industrial knowledge base have been advantageous within this prototype.

5 SCALING UP

The previous successful industrial experiment was admittedly of modest complexity and leaves open the issue of applicability to partitioning of large knowledge bases. Because rapid prototyping is quite data intensive, we look there for testing scalability of the interaction of propagations and knowledge bases.

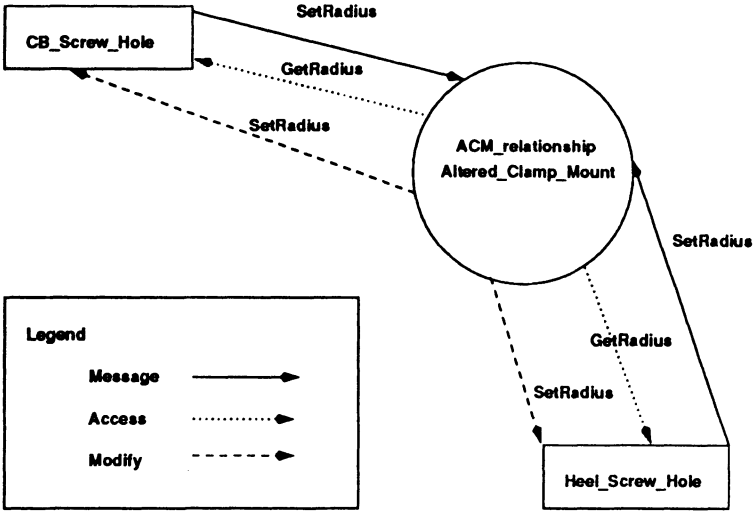

Figure 1 ADAM Propagation Entity between CB Screw Hole and Heel Screw Hole.

Propagations are now being investigated in support of design advisors for virtual prototyping, providing a test of their capability for more data intensive environments. These design advisors would have the canonical geometric data representation for solid freeform fabrication as their input. This .STL file format consists of a triangulation of the boundary surfaces of a solid. Typically, there will be hundreds of thousands of triangles, and we consider each such triangle to be a feature in this domain. Any geometric design modifications can then be accomplished by local changes to these features.

A fragment of an .STL file is shown in Figure 2. Let p designate the vertex common to the 90 degree angles of the black and grey triangles at the center of the image of Figure 2. A representative local feature editing could be accomplished by perturbing p, subject to rules to preserve the topological form of the artifact (Andersson et al 1994), (Boyer and Stewart 1991), (Boyer and Stewart 1992), (Dorney 1994), (Peters et al 1996) and (Stewart 1993).

Rule: For a .STL file, let the parameter ν be the minimal separation between disjoint pairs of vertices, edges and faces. If for each j and each vertex p_j, the corresponding perturbation δp_j is such that $\|\delta p_j\| < \nu/2$, then the perturbed object preserves the topological form.

Each proposed design edit would have an interrelationship to the .STL file, as depicted by the propagation testing for topology preservation in Figure 3. The disc example had only a modest number of propagations, but a typical

.STL file contains hundreds of thousands of vertices, where each such vertex might have associated propagations.

Figure 2 Local View of .STL Geometry

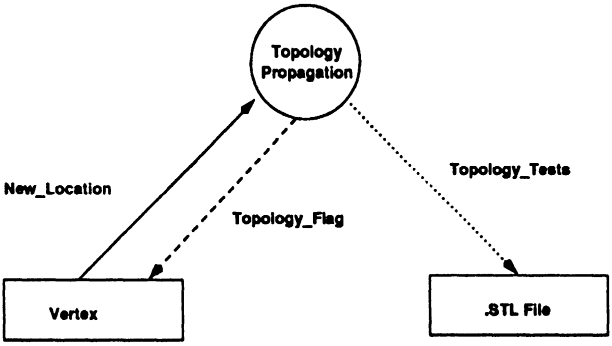

Figure 3 Topology Propagation

6 ARTIFICIAL INTELLIGENCE SYNERGY

To frame critical issues, the following broad research question is posed relative to knowledge base partitionings by propagations, as discussed in Section 4.

- In localizing knowledge within a propagation method, what is the complementary role of global knowledge relative to logical consistency and computational efficiency?

To explore such issues with realistic test cases, we consider propagations for .STL files, requiring the perturbation of many vertices. There could be pairwise interrelationships between many of the vertices, as shown abstractly in Figure 4.

Since the geometry of a .STL file is triangulated, design changes which would preserve the same topology could be executed by a sequence of perturbations of individual vertices and adjacent edges, each with its own propagation to appropriately limit the vertex movement. If all intended perturbations are less than $\nu/2$, then the Rule applies and ν can be computed once (even, off line) and only updated at the end. However, this $\nu/2$ upper bound can be needlessly conservative. More aggressive edits are possible within specific neighborhoods of a vertex and for a particular direction of movement (Dorney, 1993). However, these more aggressive edits require a propagation of the type shown in Figure 3 to fire after each such chosen vertex, requiring an incremental updating of the value of ν.

Since the computation of ν is of quadratic complexity over all disjoint pairs of vertices, edges, and faces, it is desirable to see if some of the geometry for this quadratic computation can be culled for better performance without effecting the validity. This raises more specialized questions within the scope of the broad research question articulated, above, namely

- If the number of such aggressive edits is small, might it be best to do the incremental updates without resorting to any culling? Namely, might the overhead of culling exceed any over performance advantage gained?
- If the number of such aggressive edits is large, must the order of their execution be considered to optimize the culling and performance.

The issues of firing a particular propagation then become subject to the artificial intelligence concerns typically associated with large rule based systems and complex case based reasoning systems. Among them are firing order, efficient transversal, avoidance of cycles, conflict resolution. Addressing all three questions posed in this section will form the basis for our submission to the planned 1997 workshop.

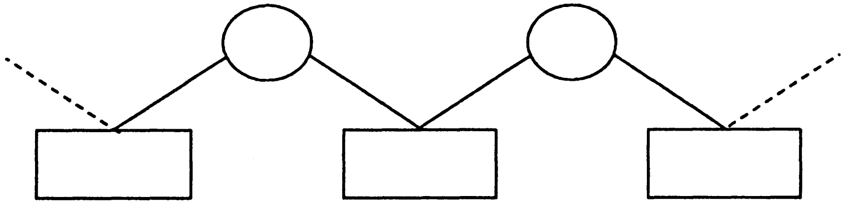

Figure 4 Propagations as Transitive Relationships

7 CONCLUSIONS

A prototype of our object-oriented construct of propagations has been employed to model interrelationships between features in industrial designs. This modeling exercise has demonstrated that this initial prototype of propagations holds promise to deliver the following advantages:

1. improved abstraction of the conceptual design, capturing the semantic information of feature interrelationships, and
2. local views of knowledge bases to address combinatorial complexity.

Acknowledgements: The authors acknowledge, with appreciation, partial funding for this work received under National Science Foundation Grant Number MII-9308346. The views expressed herein are of these authors, not of the National Science Foundation. The authors thank J. Damon, R. Disa, and E. J. Mintel of Pratt and Whitney Aircraft for supplying us with industrial examples and practical guidance. However, any remaining errors or omissions are solely the responsibility of the authors.

REFERENCES

Andersson, L-E., Dorney, S. M., Peters, T. J., Stewart, N. F. (1995) Polyhedral perturbations that preserve topological form, *Computer Aided Geometric Design*, **12**, 785 - 799.

Brett, B. D., Peters, T. J., Demurjian, S. A., and Needham, D. M. (1996) Relations between features–prototyping object-oriented language extensions on an industrial example, *ASME Design Conference Proceedings*, released only on CD-ROM.

Boyer, M., Stewart, N. F. (1991) Modelling spaces for toleranced objects, *International Journal of Robotics Research* **10**, (5), 570-582.

Boyer, M., Stewart, N. F. (1992) Imperfect-form tolerancing on manifold objects: a metric approach, *International Journal of Robotics Research*, **11**, (5), 482-490.

Chen, X. and Hoffman, C. (1995) Design compilation for feature-based and constraint-based CAD, *Proceedings of the Third Symposium on Solid Modeling and Applications*, (eds. C. Hoffman and J. Rossignac), Salt Lake City, Utah, May 17 - 19, 13 - 20.

De Floriani, L. (1989) Feature extraction from boundary models of three dimensional objects, *IEEE Trans. PAMI*, **11** (8), 785 - 798.

Dighe, R., Jakiela, M. J., and Wallace, D. R. (1993) Structural synthesis under manufacturability constraints, a CAD system for the design of injection-molded product housings, *Research in Engineering Design*, **5**, (3 & 4), 185 - 201.

Dixon, J. R. (1986) Artificial intelligence applied to design: a mechanical engineering view, *Proceedings AAAI-86*, August.

Dixon, J. R. et al, (1987) Expert systems for mechanical design: examples of symbolic representations of design geometries, *Engineering with Computers* **2**, 1 - 10

Drake, S. and Sela, S. (1989) A foundation for features, *Mechanical Engineering*, Jan., 66 - 73.

Dorney, S. M. (1994) *Maintaining Equivalent Topological Form in Tolerance Modeling*, Doctoral Dissertation, Department of Computer Science and Engineering, University of Connecticut, Storrs, CT.

El Guemhioui, K. (1994) ADAM Users Guide, Version 3.4.0, The University of Connecticut, Department of Computer Science & Engineering.

Ellis, H. (1994) *An Information Engineering Approach to Unified Object-Oriented Design and Analyses*, Doctoral Dissertation, The University of Connecticut, Storrs, CT.

Fu, Z., and de Pennington, A. (1993) Constraint-based design using an operational approach, *Research in Engineering Design*, 5, (3 & 4), 202 - 217.

Glovin, S. R. and Peters, T. J. (1987) Features as a basis for intelligent CAD/CAM, *Proceedings of the Artificial Intelligence in Manufacturing Conference*, Society of Manufacturing Engineers, Long Beach, CA, Oct. 14-16, 2-1 to 2-5.

Gossard, D. C., Zuffante, R. P. and Sakurai, H. (1988) Representing dimensions, tolerances and features in MCAE systems, *IEEE CG & A*, 8 (2), 51 - 59.

Gupta, S.K., Regli, W., and Nau, D. (1994) Integrating DFM with CAD through design critiquing, *Concurrent Engineering: Research and Applications*, 2 (2) 85-95.

Hayes, C., and Sun, H.C. (1995) Using a manufacturing constraint network to identify cost-critical areas of designs. *AIEDAM*, 9 (2), 73-87.

Henderson, M. R. (1984) *Extraction of Feature Information from Three Dimensional CAD Data*, Doctoral Dissertation, Purdue University, West Lafayette, IN.

Henderson, M. R. and Taylor, L. E. (1993) A meta-model for mechanical products based upon the mechanical design process, *Research in Engineering Design*, 5, (3 & 4), 140-160.

Hernandez, J. A. et al (1991) Intelligent decision support for assembly system design, in *Innovative Applications of Artificial Intelligence*, (eds. R. Smith and C. Scott), AAAI Press, 135-156.

Hoffmann, C. M, and Juan, R. (1993) High-level representation for geometric design and analysis, in *IFIP Transactions, Geometric Modeling for Product Realization*, (eds., P.R. Wilson, M.J. Wozny, and M.J. Pratt), Elsevier Science Publishers B. V. (North-Holland), Amsterdam, 129 - 164.

Mäntylä, M., Nau, D., and Shah, J. J. (1996) Challenges in feature-based manufacturing research, *Communications of the ACM*, 39, (2), 77 - 85.

Orelup, M. F., Dixon, J. R., Cohen, P. R., and Simmons, M. K. (1988) Dominic II: meta-level control in iterative redesign, *Proceedings AAAI-88*, August.

Palmer, R. S. and Shapiro, V. (1993) Chain models of physical behavior for engineering analysis and design, *Research in Engineering Design*, 5, (3 & 4), 161 - 184.

Peters, T. J., Demurjian, S. A., Needham, D. M., Peters, R. J., and Dorney, S. M., Propagating topological tolerances for rapid prototyping, accepted *Proceedings of ASME IMECE Conference*, Atlanta, November 18 - 21, 1996.

Prabhakar, S. and Goel, A. (1996) Extending functional models for non-local adaptations in engineering device design, *Proceedings of the Ninth Florida AI Research Symposium*, Key West, FL, 369-374.

Requicha, A. A. G. (1996) Geometric reasoning for intelligent manufacturing, *Communications of the ACM*, **39**, (2), 71 - 76.

Rosen, D. W. (1993) Feature-based design: four hypothesis for future CAD systems, *Research in Engineering Design*, 5, (3 & 4), 125-139.

Shah, J. J. (1991) Assessment of features technology, *CAD*, **23**, (5), 331 - 343.

Shah, J. J., and Mäntylä, M. (1995) *Parametric and Feature-Based CAD*, Wiley, N.Y.

Shah, J. J., and Rogers, M. T. (1993) Assembly modeling as an extension of feature-based design, *Research in Engineering Design*, 1993, 5, (3 & 4), 218 - 237.

Stewart, N. F., (1993) Sufficient condition for correct topological form in tolerance specification, *CAD*, **25**, (1), 39-48.

Vandenbrande, J. H., and Requicha, A. A. G. (1990) Spatial reasoning for automatic recognition of interacting form features, *Proceedings of the International Computers in Engineering Conference, American Society of Mechanical Engineers*, 251 - 256.

AUTHOR BIOGRAPHIES

Thomas J. Peters: Associate Professor of Computer Science and Engineering, Ph.D., Wesleyan University (Topology). He has been Senior Technical Staff at Computervision Corporation and Technical Staff at Draper Laboratories. His CAGD interests emphasize applied topology and features.

Steve A. Demurjian: Associate Professor of Computer Science and Engineering, Ph.D., The Ohio State University (Computer and Information Sciences). His research area is software, security, and information engineering, with a concentration on the object-oriented paradigm and model. He is a member of the IFIP WG11.3 on Database Security.

Robert McCartney: Associate Professor of Computer Science and Engineering, Ph.D., Brown University (Computer Science). He was a statistician at the Smithsonian Institution. His work ranges from diagrammatic reasoning to planning and plan recognition to puppotics.

Donald M. Needham: Assistant Professor, Computer Science, Franciscan University of Steubenville, Doctoral Candidate, M.S., Naval Postgraduate School (Computer Science). He was a master instructor at the U. S. Naval Academy.