Title:
Towards a Declarative Modeling Approach built on Top of a CAD Modeler

Authors:
Dorian Decriteau, dorian.decriteau@gadz.org, Arts et Métiers ParisTech, France
Jean-Philippe Pernot, Jean-Philippe.Pernot@ensam.eu, Arts et Métiers ParisTech, France
Marc Daniel, marc.daniel@univ-amu.fr, Aix-Marseille University, France

Keywords:
Declarative modeling, semantic description, encapsulating CAD operators, shape design

DOI: 10.14733/cadconfP.2015.94-98

Introduction:

Today, industrial CAD software rely on an incremental B-Rep (Boundary Representation) modeling paradigm where volume modeling is performed iteratively using planar sketched contours subjected to mainly extrusion or revolution operations. Even if CAD modelers provide operators (e.g. pad, pocket, shaft, groove, hole, fillet) to get rid of the direct use and manipulation of canonical surfaces and NURBS, working with a CAD modeler is almost procedural with a lot of intermediate operations required to obtain the desired shape of an object. Actually, all those intermediate operations are time-consuming and generate complex construction trees that are not particularly needed to describe the final shape. Moreover, using such a procedural approach, the designers have to make a mental gymnastic to break down the object body into several basic shapes linked to the different operators of the CAD software.

Clearly, an approach closer to the designers' way of thinking is missing and there is still a gap between the ideas designers have in mind and the available tools and operators used to model them. Ideally, it would be more convenient to enter a semantic description of the shape, the CAD modeler being in charge of generating it. This is the aim of the approach proposed in this paper. More precisely, in order to stay compatible with existing CAD modelers widely used in the industry, but also to take full advantage of their efficient geometric modeling kernels and features, our attempt was to define a high-level declarative modeling approach implemented in the form of a plugin built on top of these modelers. From an initial high-level description, the plugin generates the CAD model and its building tree. The main idea is to encapsulate several operators and/or features to answer to a partial description of the shape. Since it is built on top of actual CAD modelers, such a declarative modeling approach can be integrated within the Product Development Process (PDP) and the traditional way of manipulating CAD models obviously remains accessible to more experimented users/designers. This top-down approach is illustrated on Figure 1. At the top level, within our plugin, designers manipulate a semantic description that is transformed into a procedural description, i.e. a sequence of traditional CAD functions and operators (also called features) which use the traditional Euler operators to act on the low-level geometric entities (faces, edges, vertices) defining the underlying B-Rep model.

Finally, it is clear that the output of this declarative modeling approach is not a final CAD model instantiated with accurate numerical values, but rather the output is to be considered as a first draft. As a matter of fact, the description must not be tedious and can remain incomplete so as to leave the possibility to refine the description during the next steps of the PDP [2]. Of course, handling incompletely defined shapes may generate not expected but valuable solutions. It is also a good mean to take into account the uncertainties the designers have when defining complex shapes.

Proceedings of CAD'15, London, UK, June 22-25, 2015, 94-98

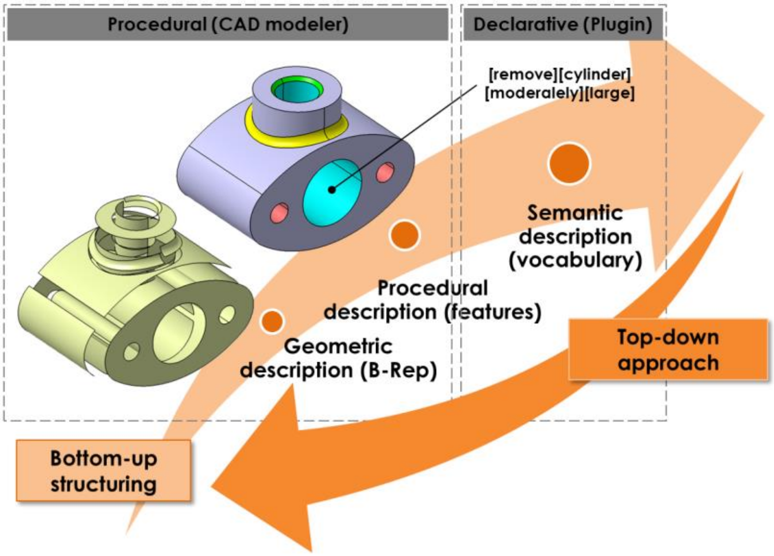


Fig. 1: Declarative modeling approach built on top of a procedural CAD modeler.

Related work:
As suggested on Figure 1, at a low level, CAD modelers generally consider purely geometric description using curves and surfaces represented by means of B-splines and NURBS. Invented more than 40 years ago, these mathematical models are well known and their fundamental concepts can be found in several reference books [3],[9]. Since the expected shapes are generally complex, the designer often has to decompose them into elementary shapes themselves subdivided into several surfaces. Each elementary surface is defined by means of a network of control points, weights and knot sequences. Finally, the elementary surfaces are assembled together to produce a manifold solid, i.e. a B-Rep representation expressing the relationships between the vertices, the edges and the faces of the topological model as described in [4]. However, interacting at this very low level is restricted to experts and generally at a final step of the modeling process or to address specific aesthetic issues. Thus, several attempts have been made to try to overcome the limits inherent to the manipulation of low-level geometric entities.

Feature-based modeling introduced in [12] falls into this category of higher-level approaches. By using features to build their CAD models, designers do not anymore act at a low level but rather on shape primitives that can be parameterized and pre-defined. Depending on the complexity of the shapes to be represented, different approaches and associated features can be used: form features, semi free form features, free form features or fully free form features [8]. If form features are now well known and implemented in most of the existing CAD systems, it is not true for free form features which are not yet fully available in commercial solutions. It is the designer's responsibility to evaluate all methods that can produce the targeted shape. There is no uniqueness in the way a 3D shape can be described and modeled within CAD software. Thus, even if features are a real improvement in the way designers interact with the CAD models, there is still a gap between the available tools and functionalities, and the way designers think in 3D.

Declarative modeling appeared in the early nineteens' and aims at constructing objects by means of a set of properties rather than by entering geometrical information like point coordinates. It is thus mainly based on semantics and not on mathematical properties so that this approach is closer to the way designers think. This modeling process in three steps [1]:

- First, the designer has to make a description of the shape by using an adapted vocabulary. This description must be transformed into a set of constraints that can be solved. The description can be adapted to any given trade assuming that a list of synonymous exists.

- Second, the modeling software models the shape from the description by exploring through adapted and specific algorithms the space of solutions. As already introduced, this implicitly suggests that such an approach is more devoted to explore a set of solutions than to solve well-constrained or over-constrained problems.
- Third, the designer has to browse between the different produced solutions and choose one.

It is evidently easier to study objects belonging to discrete spaces of solutions since these sets can be explored (or described) and the tree of solutions can be pruned with the given properties. Spaces of solutions depending on real values are more difficult to study. Declarative modeling has been studied on curves [11] and surfaces [6]. Actually, declarative modeling is one form of the generic variational design approach. The latter has been introduced in CAD in [7]. Different attempts have been made to develop this approach. Like for example in [5], the associated work quickly concentrates on the important problem of constraints modeling and how to split the set of constraints into smaller independent sets. However, this approach was ambitious and it has been slowed down by the huge amount of work mandatory to redevelop all the basic operators and primitives required to generate shapes from descriptions.

In this paper, the idea is to combine the advantages of the traditional B-Rep and feature-based modeling approaches with the advantages of a more advanced and high-level declarative modeling approach. As a consequence, this combination takes astutely advantage of today's robust commercial geometric modeling kernels without redeveloping everything starting from scratch. At the end, the proposed approach can be seen as a plugin of a CAD software that transforms a high-level description into a building tree gathering together all the operations and functions used to get the final 3D shapes.

Declarative modeling framework:

The challenge of this approach is to be able to provide a description of a part. We made different tests to describe existing parts. We asked designers to model the parts corresponding to these descriptions and this process proves that a designer who does not know the part is able to construct it with a CAD software from its description. Assuming a vocabulary defined, the main problems of a description are the localization of elements and the relative dimensions.

The first step of our plugin is to create the initial solid. For that, the user has to choose into a list a kind of solids (parallelepiped, cylinder, sphere…). The positioning system is based on the absolute axis of the CAD modeler, always visible by the user. To describe it, the user can use absolute positioning words like **[above]**, **[below]**, **[to the right]**, **[to the left]**, **[fore]**, **[back]**. He can also use relative positioning, for example **[Above] [parallelepiped 1]**, **[on the right hand size]**. To help the user to make his/her description, all the names of each shape already designed are written on the model.

After this, the user has to complete the description with some adjectives called "quantifiers". The aim of these quantifiers is to describe the different sizes of the shape. For the initial solid, the quantifiers must be absolute quantifiers. These absolute quantifiers are: **[extremely-few]**, **[very-few]**, **[few]**, **[moderately]**, **[rather]**, **[very]**, **[extremely]**. For example, the description can be **[parallelepiped] [moderately] [wide]**, **[very][long]**, **[very-few] [high]**. One must remember that the purpose is to create a draft and not a perfect shape so the real size values are not important but we have to get the good proportions between dimensions. We choose an order of magnitude (OoM) and we apply some factors in function of the vocabulary. The final size can be obtained by any homothetic transformation if required. Table 1 summarizes these factors. To keep a good ratio for the shape, we do not apply the same size factor for a straight line or a radius.

Our first implementation is based on the Dassault System software CATIA. All the interfaces are programmed in VBA (Virtual Basic for Applications) and the construction of sentences are currently made by menus. But we can imagine creating a more user-friendly interface.

We developed volume operators like slice, bump, bending which automatically call a sequence of the modeler operators, but also surface operators.

We validate the approach through examples. We propose in this abstract to design the part represented on Figure 2. Firstly, the designer has to make its description. According to our experiments about vocabulary, we have selected several words to make a description with the right syntax.

The description should be different (using symmetry) but it is not yet programmed. Based on the developed operators, we can observe the result of the declarative modeling on Figure 3. Its interest can be judged by the number of “clicks” required to produce it compared to the classical modeling process.

- **[Start with] [sphere] [moderately][voluminous]**
- **[Above] [sphere 1], [remove] [slice] [very-few] [high]**
- **[Below] [sphere 1], [remove] [slice] [very-few] [high]**
- **[To the left of] [sphere 1], [remove] [slice] [extremely-few] [high]**
- **[To the right of] [sphere 1], [remove] [slice] [extremely-few] [high]**
- **[Above] [slice 1], [add] [cylinder] [few] [wide], [extremely-few] [high]**
- **[Below] [slice 2], [add] [cylinder] [few] [wide], [extremely-few] [high]**
- **[Above] [sphere 1], [remove] [cylinder] [very-few] [wide] [through all]**
- **[To the left of] [slice 3], [add] [cylinder] [extremely-few] [wide], [few] [high]**
- **[To the right of] [slice 4], [add] [cylinder] [extremely-few] [wide], [few] [high]**

For the straight size:	For the size like radius:
• **[Extremely-few]** → OoM x 1/10 • **[Very-few]** → OoM x 1/5 • **[Few]** → OoM x 1/2 • **[Moderately]** → OoM x 1 • **[Rather]** → OoM x 2 • **[Very]** → OoM x 5 • **[Extremely]** → OoM x 10	• **[Extremely-few]** → OoM x 1/20 • **[Very-few]** → OoM x 1/10 • **[Few]** → OoM x 1/5 • **[Moderately]** → OoM x 1/2 • **[Rather]** → OoM x 1 • **[Very]** → OoM x 2 • **[Extremely]** → OoM x 5

Tab. 1: Link between quantifier and numerical values.

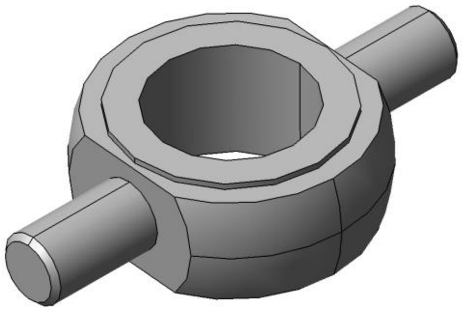
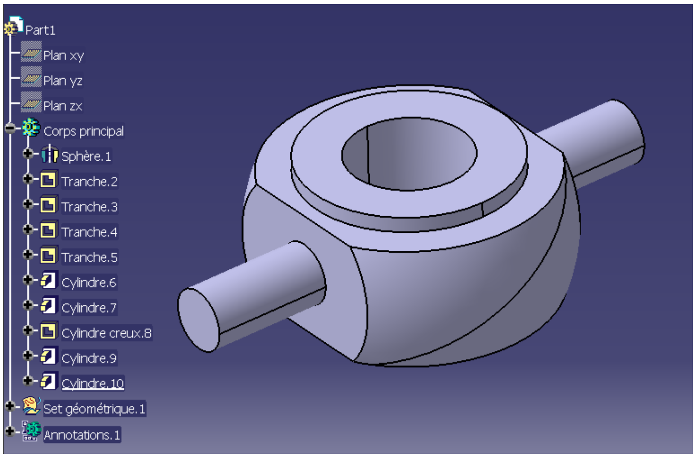


Fig. 2: Mental image of a mechanical part.

<u>Conclusion:</u>

The proposed approach proved to be relevant even if limits exist due to the description: a description could be difficult to produce for a beginner is not unique and some details could be obtained more easily through traditional operations. For our first work, a rigid framework based on menus to enter the description has been developed but a different way to handle the man-interface is more a problem of computer science than design. Anyway, this approach offers a gain in time and the generated model can easily be updated with CAD operators since it is directly embedded in the CAD environment.

Finally, entering such a description means that some semantic information are inserted into the CAD modeler. Today, those information are solely used to create the CAD model but one can imagine that it could be used in any other steps of the PDP. In the future, this interesting possibility will evidently require large modifications of actual CAD modelers to give rise to a new generation of modeling environments.



References:
[1] Desmontils, E.; Lucas, M.: Les modeleurs déclaratifs, Revue de CFAO et d'Informatique Graphique. 10(6), 1995, 559-585.
[2] Falcidieno, B.; Giannini, F.; Léon, J-C.; Pernot; J-P.: Processing free form objects within a Product Development Process framework, in ASME-Press Advances in Computers and Information in Engineering Research, Michopoulos, J. G.; Paredis, C. J.J.; Rosen, D. W.; Vance, J. M. (Eds.), 1, 2014, 317-344.
[3] Farin, G.: Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, fourth edition, Academic Press, San Diego, 1997.
[4] Hoffmann, C.: Geometric and Solid Modeling: An Introduction, Morgan Kaufmann, San Mateo, 1989.
[5] Hoffmann, C.; Juan, R.: Erep: An editable, high-level representation for geometric design and analysis, Proceeding of IFIP TC5/WG5.2 Working Conference on Geometric Modeling for Product Realization, North-Holland Publishing Co., 1992, 129-164.
[6] La Gréca, R.; Daniel, M.: A Declarative System to Design Surfaces, WSCG 2006 Conference, ISBN: 80-86943-03-8, Pzen, Czech Republic, 2006, 17-24.
[7] Lin, V.C; Gossard, D.C.; Light, R.A.: Variational Geometry in Computer Aided Design, ACM Computer Graphics, 15(3), 1981, 171-177. http://dl.acm.org/citation.cfm?doid=800224.806803
[8] Pernot, J-P.; Falcidieno, B.; Giannini, F.; Léon, J-C.: Incorporating free-form features in aesthetic and engineering product design: State-of-the-art report, Computers in Industry, 59(6), 2008, 626-637. http://www.sciencedirect.com/science/article/pii/S0166361508000377
[9] Piegl, L.; Tiller W.: The NURBS Book, Springer Verlag, Berlin, 1995. http://dx.doi.org/10.1007/978-3-642-97385-7
[10] Regli, W.C.: Geometric algorithms for recognition of features from solid models, PhD dissertation, Univ. Maryland, College Park MD. 1995. http://www.cs.umd.edu/~nau/others-papers/regli_thesis95.pdf
[11] Rossignol, V.; Daniel, M.: A Declarative Modeler for B-spline Curves, in Curves and Surfaces Design, Saint-Malo 1999, P.J. Laurent, P. Sablonnière, L.L. Schumaker (eds.), Vanderbilt University Press, 2000, 353-262.
[12] Shah, J.J.; Rogers, M.T.: Expert form feature modeling shell, Computer Aided Design, 20(9), 1988, 515-524. http://www.sciencedirect.com/science/article/pii/0010448588900413