Standard Features and Their Impact on 3D Engineering Graphics

K. M. Waldenmeyer
N. W. Hartman
Purdue University, West Lafayette, IN 47907

ABSTRACT

The prevalence of feature-based 3D modeling in industry has necessitated the accumulation and maintenance of standard feature libraries. Currently, firms who use standard features to design parts are storing and utilizing these libraries through their existing product data management (PDM) systems. Standard features have enabled companies to regulate the way key features are defined across all models and revisions, while helping to speed up the modeling process, but the new features also come with limitations. These advantages and disadvantages can become crucial for a firm trying to make its product lifecycle run more smoothly. This paper will detail how a leading PDM package manages 3D geometry, how a standard feature interacts within a 3D part file, and how those conditions affect the way standard feature libraries are stored in a PDM system. It will also describe some of the general obstacles to storing this conceptual information, and make recommendations on how to define standard features more effectively, and how we can best prepare students facing these challenges upon entering the workforce.

Introduction

Three-dimensional geometric modeling has progressed quickly over the last 30 years. Today, most 3D CAD packages are hybrids of a sketch-based parametric modeler combined with surface modeling features that enable users to create complex, fluid designs while maintaining the level of control offered by sketch-based modeling. Since the geometric definition of a product has become so complex, equally complex ideas have emerged on how to classify the different elements that make up a part (Brown, n.d.). This also affects how the part is modeled and how track design intent is tracked over time and over different designs (Dean, 2008).

One of the emerging tenets of modern part definition is the use of standard features. The word “features” is used here to describe a certain set of geometry like points, lines, planes, profiles and even positive and negative solids that are related in a semantic way. Similar to the way that an assembly model is made up of several related part models, these features are composed of several dependent geometric entities that, when combined correctly, form an individual feature with its own properties. Multiple feature operations can also be combined in this fashion to make a collection of features that can be used as a group on multiple models at a future time (Figure 1). These “user-defined” features can speed part geometry creation at a later date and can also enable standard modeling procedures within a corporate environment. A simple (but common) example of a standard feature is the Hole command that many CAD systems have now offered as a command included in the basic capability of the system.. By instantiating a Hole command, the user enters in a few parameters such as diameter, depth, and position relative to other geometry, and the CAD system places a hole in the specified location with those parameters. It remains a "hole" in the feature tree and the user is able to return to it at a later time and change the parameters as needed.

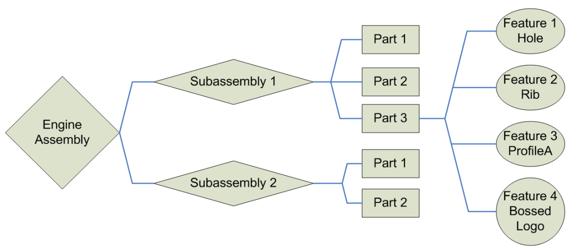

Figure 1. *Standard Features in the product hierarchy.*

This example of a standard feature (that was ultimately so useful that it was added to the default command structure) is just the beginning of an increasingly prevalent and useful tool for 3D modeling. Standard features represent a change in modeling technique and product data management support which differs from more traditional methods of creating and editing geometry. With the use of standard features also come many issues that must be dealt with before using them to their full potential.

Defining Standard Features

In the first example of a hole, it is apparent that the geometry that makes up a hole should belong together and be editable in such a way that the hole parameters can be changed at a later time without a major time reinvestment (Figure 2). However, a hole is just the beginning of a methodology of modeling using standard features.

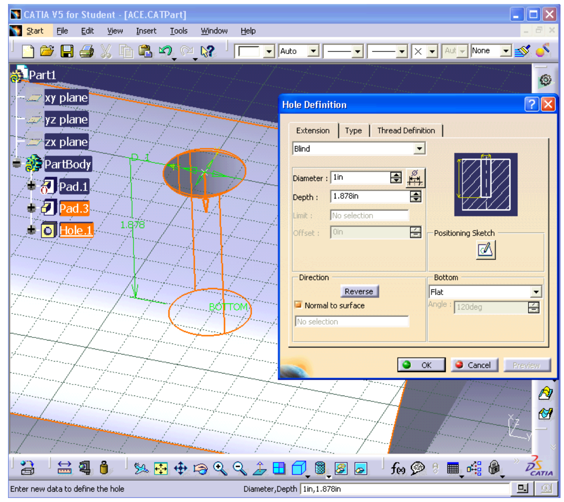

Figure 2. *The Hole command in CATIA V5.*

The idea behind using standard features is their reusability and stability from part to part. Standard features are meant to be reused from one model to the next to maintain uniformity and help simplify the modeling process. Typically standard features are defined by a user during the initial time when those geometric entities are needed, then when the user must add that specific feature to another model, they simply place it into the new model using common geometric references within the new part model, set the correct parameters, and move on to the next feature that must be created. By using this method, the user not only saves time, but also reduces the possibility of error by not having to recreate the same feature over and over again.

A second reason for using standard features is just as their name implies – to implement and enforce corporate modeling standards. Users are able to choose critical design elements from a library that is stored in a common location on a server and include those features in their models, thereby consistently applying and using geometry that is common to multiple product offerings. Modeling geometry one time and capturing the knowledge and design intent used to do in a format that is available throughout the organization offers a competitive advantage, reduces the potential for error, an potentially saves time.

The elements of standard features, as shown in Figure 3, fall into three major categories – static, dependency-driven, and variable-driven. All three of these types of features are usually present in a standard feature and therefore should be treated carefully when planning for how the feature will be implemented into future models.

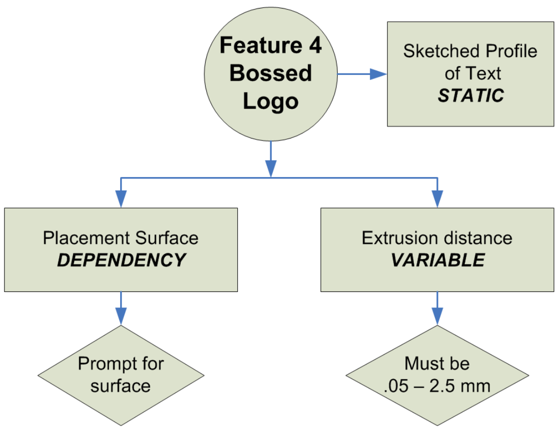

Figure 3. *Possible elements of an embossed logo feature.*

An example that uses all three of the major feature elements might be the modeling of an embossed logo that is part of every product a company manufactures (Figure 4). The logo itself never changes, so the sketch that outlines that text would be considered static. Variables like size and depth can be changed on every new product without affecting the logo outline itself. This represents a major time savings in terms of the amount of repetitive work it would require a designer to do each time they wanted to add the same logo to a new product. It is also possible to apply rules to variables to control their values under different conditions. Designers can set a range of possible values for a variable, a few preset options, or remove the limits entirely for added freedom if necessary.

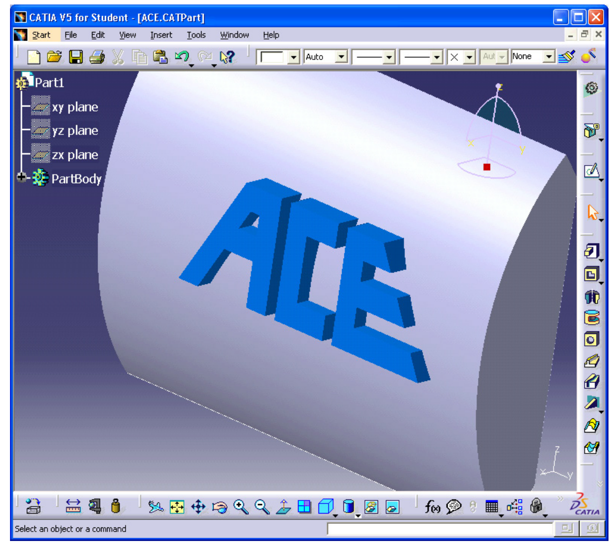

Figure 4. *An embossed logo on a surface that could be made into a standard feature in CATIA V5.*

Another characteristic of standard features is their inherent use of dependencies to adjust to changes in a model, similar to the way parts behave in an assembly. When a standard feature is predefined with planes, surfaces, or other features it is dependent on, the user is then prompted for these inputs upon adding the feature to the current model. These inputs enable the feature to be positioned and configured correctly each time with relatively little work on the part of the modeler. In the future, if the dependent features are changed, the standard feature will adjust accordingly and alert the user if the change makes the standard feature unable to compute and update correctly.

Most standard features are unique and defined by the user. This definition stage is the most complex because it requires the most thought and planning. Designers must be able to anticipate what parts of a standard feature will need to be changed later on, as well as to what degree each of the specified dimensions and parameters will change. This can be difficult at first, but will save time later in the design process because the user will be limited to the predefined rules for each modifiable dimension. The advantage is that these rules are the same no matter where the standard feature is added, so a user in another country or a new hire will add a feature to their model with the same properties and dimension rules, regardless of the circumstances. This works to remove much of the human error that is inherent when modeling features over and over again across parts.

Standard features in PDM

Product data management (PDM) systems are typically used to store part and assembly data for an engineering firm that must track several products over the course of their design iterations and lifecycle. In PDM systems, it is possible to view an assembly and see a list of all the parts that make up the assembly, without ever being required to launch the CAD software and view the model. Engineers can also look at a part and see which assemblies it is included in, and how many instances of that part are included. In a related fashion, it would be useful to have this sort of functionality with standard features in respect to the parts they are used in.

A leading PDM package, Siemens PLM's Teamcenter Engineering, provides the ability to store standard feature data within its database, giving a level of security and backup that is not possible when just storing standard features in the CAD system on a local computer, or on a file server in a commonly accessible location. However, due to the dependencies and variables inherent in standard features, Teamcenter Engineering does not have the capability of tracking these features the same way it tracks parts and assemblies. This is likely due to the geometry-intensive nature of standard features – parent/child references within the standard feature are established between geometric elements of that feature, and all references are wholly-contained within the standard feature. Whereas parent/child references within an assembly model are established between parts. However, it would still be useful for a firm that makes use of standard features often to be able to analyze that usage like a part inside an assembly. This would allow for information dissemination on a wider scale within the organization.

The biggest advantage of handling standard features inside a PDM system would be the archival mechanisms built in to those systems that would enable users to pull a feature from a group of related features in a database, and add the selected standard feature to the model with minimal effort. This would ensure that every user is using the same collection of standard features. Typically once a feature has been created with all its variables, dependencies, and static elements established, it will not change due to the typical reasons for using standard features in the first place – common geometric elements across the product line and to promote consistent modeling methodologies. Therefore, using Teamcenter Engineering as a database to store these "static" standard features is useful and more secure than other methods. However, because of the way Teamcenter Engineering handles product data, the system is lacking in a mechanism to handle standard feature data that accounts for its unique properties and challenges.

Teamcenter Engineering treats a standard feature as another part, on the same level as a part that is stored inside an assembly. This is due to the concept the PDM system uses to define all assemblies and parts: as Items, which then have physical CAD models attached to them. The system is proficient at handling these Items and uses them to show product structure data without the user having to drill down into the geometry file to see and manage that information. However, it does not offer the capability to relate standard features to parts in the same manner. Standard features are stored as Items but are not included in the product structure outside the CAD system; they are only instantiated inside the CAD file and are not referenced back to the PDM system in any way. While this may work for simple product data structures and applications, it is conceivable that a need for more control and reporting on the usage of standard features would find the current implementation inadequate.

Recommendations for Defining Standard Features

Standard features are best described as being user-defined. They require more work at the beginning during definition than when they are used later in the design process. As a result, it is critical to have an understanding of what a standard feature must do before it is ever created and instantiated in a model. It is also important to understand company policies and modeling practices and procedures to ensure that the definition and usage of standard features conforms to rules that have already been established.

In the previous example of an embossed company logo, there may be hundreds of different dimensions that could be editable when converting that logo into a standard feature (Figure 5). Knowing which ones are most important is critical to successfully using standard features as they were intended. Therefore, it would be useful to do a short study on where the standard feature will be used and in what ways. If a rib will be used in a variety of different applications, it would be best to make more of its dimensions editable to the modeler. However, for an embossed logo, only a few dimensions need to be made editable due to the nature of the usage.

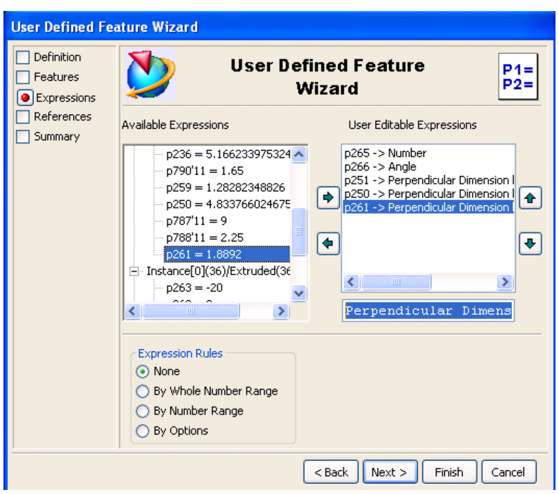

Figure 5. *NX 4's Standard Feature Wizard, where a user is able to pick out all dimensions that will be editable, and to what degree.*

It is also of the utmost important to properly name all dimensions that will be editable, as well as provide documentation on which dimensions refer to which portion of the model. This is an area where organizational standards would be helpful in managing a user's work. In Figure 3, users are able to edit the names for the dimensions they are about to make editable, which makes the instantiation of the standard feature later on easier and faster. Siemens PLM's NX 4, which is the CAD system most compatible with Teamcenter Engineering, enables the user to add a documentation page to the standard feature information for future use. Although the examples here are given in Siemens PLM's NX 4 and Teamcenter Engineering, these recommendations apply in a similar fashion across any CAD program that possesses a system to handle standard features.

Dependencies between feature elements are also a concern for defining standard features. Providing too few dependencies may allow the user to place a feature without the necessary constraints to fully define the feature within the CAD system, meaning the feature will not update correctly when another related feature is changed. Adding too many dependencies could produce errors unless all reference geometry was chosen correctly according to the standard feature's requirements. Therefore it is important to determine which dependencies are needed most and which dependencies are not needed (Figure 6). Again, naming conventions are critical to maintaining a consistent and useful standard feature.

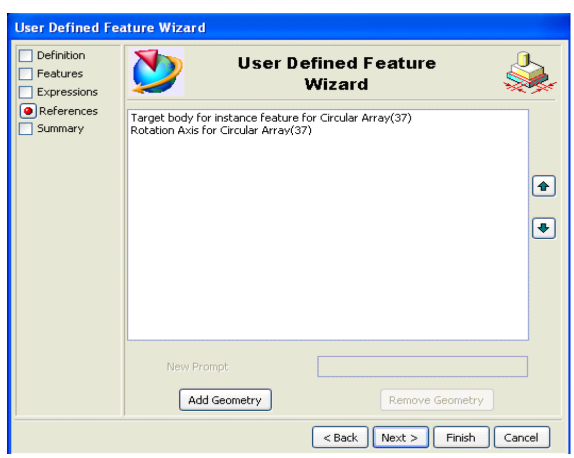

Figure 6. *NX 4's references wizard allows the user to select dependent geometry and add names for prompts.*

An organization that has used primarily sketch-based modeling techniques for a long time may have a difficult time switching to a feature-based technique. This is because it is a different approach to creating geometry; feature-based modeling is more intuitive since it defines specific features inside a part's geometry as being their own entity. This is similar to the difference between considering a hole as a discrete entity with its own unique properties, instead of a circle with a diameter that is extruded and then subtracted to create negative geometry. A rib is just a rib with a thickness, an angle to its surface, and a starting and ending point, instead of a triangular profile that is extruded. This change in modeling technique can be gradual or more abrupt depending on the organization, but enables users to be more consistent across time, location, and culture, while maintaining a higher quality model.

Ideas for storing standard features in PDM systems

As mentioned earlier, Teamcenter Engineering's handling of standard features has room for improvement. Although it is basic functionality for handling standard features, a few recommendations can be made for how it and similar PDM systems could manage standard features in the future. Since these intelligent features are merely elements of geometry, it is appropriate to store them within a PDM system as standalone geometry, as they are currently stored. More metadata could be extracted from them that would aid designers and managers in using them.

For example, a manager might want to check on how many instances are instantiated of a specific feature. A standard feature might be used multiple times within one part, and that part may be used multiple times within one assembly. Therefore, a bill of materials of sorts could be helpful, but to deal with standard features only. This would allow a manager to more easily gauge what a design change to a standard feature that is used across many products might affect. This capability would exist in the PDM system itself, and would not affect the geometry unless changes were made. By having a list of affected parts, managers can then go in and make sure that all affected parts are updated to the newest revision of the standard feature, if needed.

It could also be helpful to be able to extract the geometry's dimensional information within a standard feature to analyze within the PDM system. A report could be run to show which configurations of a standard feature are most common, and which values never change so that new iterations of a standard feature can change based on previous work. Currently this functionality is not present in Teamcenter Engineering, but if this capability were, it could significantly speed up the process of feature-based modeling

Ramifications for Education

Going to a feature-based modeling strategy will have an effect on how students are taught to create geometry in modern CAD tools. The most important difference is that students must model objects while thinking about how the geometry could change downstream, and accounting for those changes by defining the geometry differently at the start. This is a useful educational exercise in that students need to understand the organizational and procedural impacts of their geometry creation techniques. The prevalence of hole commands and similar feature-based tools provided in CAD packages is just the beginning. Allowing students to create and control their own unique standard features in their familiar CAD tool would help them to understand some of the complexities inherent in the technology as well as its potential for engineering design. Adding a standard features element to engineering graphics curricula would help students understand why and how standard features are used, as well the strengths and weaknesses of using the new method over more traditional modeling techniques.

Workforce training is probably one of the biggest issues standing in the way of going to a feature-based modeling paradigm. Users are typically resistant to change and therefore it will take time for modelers to become efficient at modeling using a feature-based system rather than a sketch-based system. However, it is important to reinforce the value that is added by going to a standard features paradigm of modeling, and that ultimately while more work is done up front when creating standard features, less work is done towards the end when changes are more costly and time consuming.

Conclusions

For many years CAD vendors have added feature-based tools such as hole commands, rib commands, and other such geometric entities that are intended to shorten the design process and make changing geometry easier. Because each company is different, vendors have added functionality for users to make their own customized standard features. These standard features have creation possibilities that are almost endless, which makes them a good idea to utilize for larger companies where many products have shared characteristics.

There are still limitations in how to store and manage standard features, but these limitations are slowly being addressed by software vendors. The main driver for the software changes have been companies who requested new functionality for standard features, so as the industry moves towards a feature-based modeling technique, more feature-based support will most likely be added. The move to feature-based modeling represents a change from less focused modeling techniques to slower, higher quality models that are created with future changes in mind. Education to train users, both present and future, in these techniques is critical for ensuring the continuation of higher quality modeling techniques.

References

(2007). NX User Defined Features. Retrieved Oct 1, 2008, from http://www.plm.automation.siemens.com/en_gb/Images/nx_user_defined_features_fsW1_tcm642-8682.pdf

Brown, B. Is CAD a done deal? Retrieved Oct 7, 2008, from http://www.cambashi.co.uk/research/articles/ear_is%20CAD_a_done_deal_Jan03.htm

Dean, A. (1 Oct, 2008). Solid Edge gains efficiency from synchronous technology. Retrieved Oct 9, 2008, from http://www.deskeng.com/articles/aaamap.htm

Fuh, J. Y. H., Li, W. D. (2004). Advances in collaborative CAD: The-state-of-the-art. *Computer-Aided Design*, 37, 571-581.

Hoffman, C. M., Joan-Arinyo, R. (1998). On user-defined features. *Computer-Aided Design*, 80(6), 821-882

Szykman, S., Sriram, R. D., Regli, W. C. (2001). The role of knowledge in next-generation product development systems. *Journal of Computer and Information Science and Engineering,* 1, 3-11.

Wang, F., Mills, J. J., Devarajan, V. (2002). A conceptual approach managing design resource. *Computers in Industry,* 47, 169-183.

Zha, X. F., Du, H. J., Qiu, J. H. (2001). Knowledge-based approach and system for assembly-oriented design, Part II: The system implementation. *Engineering Applications of Artificial Intelligence*, 14, 239-254.