White Paper

The differences between Traditional Product Configurators and Knowledge-Based-Engineering, Design-To-Order™ Systems:

By John C. Lambert
Design Automation Associates Inc.
June 8, 2004

Introduction

What is the difference between a Configurator and a Design-To-Order™ System?

Introduction:

Most companies with designed-to-order, engineered-to-order, custom & special products also have some products that require only simple configuration and product selection. This situation often leads to the question concerning the differences between Traditional Product Configurators and knowledge-based-engineering, Design-To-Order™ systems. Although there are many differences, this short paper intends to address this question by focusing on a single, simple example.

Discussion:

Configurators are primarily "predicate logic" tools. Predicate logic refers to the "if-then" relationships that dominate traditional configuration problems. For example, when configuring an automobile, one might select an option of "power windows". Having selected this option, the "if-then" (predicate) relationships might dictate that power-locks, power antenna and power seats are automatically included. Configurators are not designed to handle "algorithmic logic" or "generative creation and interpretation" of geometry. DTO™ Systems are designed to handle these areas, in addition to predicate logic.

Design-to-order problems are defined as those including algorithmic & predicate logic as well as geometry and geometric assemblies. Let's take a look at Figure 1 for a simple DTO™ example involving a special power transmission drive sprocket. Let's also suppose a custom keyway with an intersecting set screw is called out as special sprocket features. In order to assess this configuration, and the associated cost, the following steps must be taken.

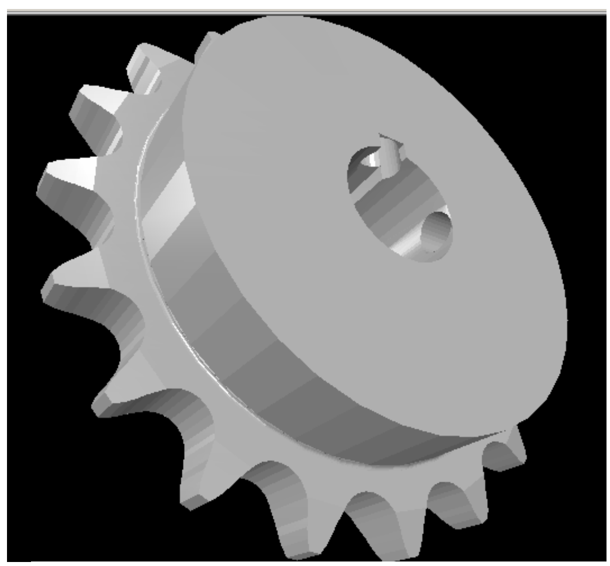

Figure 1: Special Sprocket with Keyway and Set Screw

Engineering standards specify that minimum walls of .05" must not be violated. In order to check for this condition in a DTO system, a 3D solid model would be **generatively** created that contains the unique combination of the keyway, the set screw hole and the various geometric features of the host sprocket. Once the geometry specification is complete, various solid modeling operations must be performed to construct the geometry and assess the resulting minimum walls that result from the intersection of these features*. If the min-wall standards are violated then the user must immediately be notified that the “special” is not valid. In addition, they must be notified why. This conceptually simple geometric assembly problem is easily solved with a DTO™ system but virtually impossible** to solve with a Configurator. The DTO™ system relies on rules instead of pre-stored geometries. Each time a new special configuration is introduced, the DTO™ system “generatively”’ creates that geometry and interrogates every aspect of the new topology. These operations are conducted according to the appropriate engineering and manufacturing standards. Any violations are instantly identified and reported to the user. This identification process involves geometric, geometric assembly, algorithmic and predicate logic as part of the geometry definition process.

In addition to geometry definition, another capability that is easy to provide is an accurate cost estimate for a special component or assembly. To do this, a manufacturing routing is often required. The determination of routing involves the geometric interpretation of a part’s specific features on both an individual basis as well as on a relational basis. These features must then be evaluated relative to the shops operational capabilities, capacities and various section criteria. Given geometry, capabilities and selection criteria, a routing and associated manufacturing process cost estimate can be made. The solution to this seemingly straight forward problem is intensive as it involves geometry, geometric assemblies, algorithmic logic, predicate logic and external data integration. Compromising any of these functional capabilities devastates the ability to determine the appropriate routing, and provide the corresponding routing process cost in an efficient way.

Summary:

Traditional Product Configurators are efficient at handling BOM type problems. Configurator output is generally communicated by textual outputs which may be accompanied by template file display of *pre-stored* graphics. These outputs can be confused with DTO system outputs when a sufficient understanding of the underlying technology and problem domain are not present.

DTO™ systems are highly efficient at solving problems that involve engineered customs and specials with dynamic geometry. The quotation and assessment of these specials often times involves the solution of geometric and algorithmic problems. DTO™ systems are also strong with predicate logic and are uniquely qualified for problems that involve these multiple domains.

DTO™ systems and Product Configurators can be viewed as complementary technologies. They can easily be linked together to provide a complete automated solution where simple configuration and customs & specials coexist in the same environment.

Foot Notes on Configuration:

* - From a traditional Configurator perspective, one might argue that in this simple case various closed form algebraic relationships could be constructed to determine the min-wall thickness in the critical location. In taking this approach one should bear in mind that a specific collection of relationships would have to be constructed for each location to be checked. In addition, these sets of relationships would change for each sprocket configuration. Finally, if the various features were adjusted such that the min-wall calculation becomes a truly 3D problem, the required relationships would increase in complexity at an astounding rate! These relationships would also have to be customized and embedded in each template file as described below. This combination or large numbers of template files combined with large numbers of closed form relationships would create an even more unmanageable template maintenance problem than described below. In general, the “Factorial Rule” of thumb applies which states:

Template Files = (# of Independent Geometric Features)!

For example, 5 independent geometric features require five factorial (5!), or 120 template files. In the case of a special sprocket, the number of required template files would number in the thousands!

** - Configurators solve this problem in the following way. They would select an external CAD system and then set up libraries of pre-configured geometric assemblies (template files). The exact number of geometric configurations corresponding to the number of possible combinations of features and options *expected*. In cases where there are large numbers of possible configurations, these libraries could easily contain many thousands of individual template files which ultimately become a significant maintenance problem. Products involving customs & specials generally involve many thousands of possible geometric topologies, establishing complete coverage with template files is generally impossible.

Once the appropriate template has been selected via predicate logic, the various geometric parameters are fed from the Configurator into the external CAD system. At this point the template file must execute the pre-stored expressions that look for template (configuration) *specific* problems. The fact that these expressions must be template specific increases the complexity of each template and the corresponding level of maintenance required.

In the event that a configuration is specified that does not have a corresponding template, the system will not work and the appropriate template, with its corresponding relationships, must be manually introduced.

