Design Automation: An Overview

Automatic Configuration

Parametric CAD uses interactive feature-based modeling techniques to create a 3D model of your part or assembly geometry. It allows you to define geometric constraints in such a way that modifications to the model (changing dimension values or suppressing/un-suppressing features) will automatically update the parts, assemblies, and any drawings made from them.

Product Configuration is the process of identifying which product, out of a family of pre-engineered products, is the best fit available for a given set of requirements. It is, quite simply, automating a product catalog lookup.

Traditional knowledge-based design (KBE) or "design automation" add-ins are simply a combination of a product configuration tool with a parametric CAD system. After the "Configurator" selects a product, the necessary dimensions are used to update a pre-built Master Assembly to display the geometry in 3D and update the drawings.

The marriage of a Configurator to a parametric CAD system allows you to calculate and set all your parameter values at once. While this may appear to be the next logical step for CAD, have we really done anything to automate what an Engineer does? Is this really design automation?

To answer this question, we must take a step back and attempt to describe what it is that a design engineer does.

Engineering Design

What characterizes a design problem is that there is more than one approach to solving the problem, and for any given approach there is typically more than one solution. The role of the design engineer is to find the optimal solution given the available resources. Resource constraints can include things like performance requirements, cost, raw materials, tooling, and time.

Consider the design of a gearbox. The designer begins with some basic requirements from the customer: torque, power, and speed on the input and/or the output, loading conditions, and size limitations. There are many different types of gearboxes (offset, cycloidal, planetary, right angle, sequential, etc.) and one or more of them may be appropriate for the given problem. After selecting the most promising type, there are still decisions and trade-offs to be made. If the tooth counts are in the correct proportion to each other, the gearbox reduction ratio can be met. However, if the gears are too small, they may break or fatigue, rendering that design invalid. Larger gears lower the load on the tooth, but increase the size and cost of the gearbox. The most promising designs undergo additional analysis, until the best design is found.

Clearly, any tool that claims to automate the engineering design process must provide the tools necessary to generate design variations, eliminate invalid designs, and establish the criteria for and select, the best design.

Organizing Design Rules

All design automation tools claim to capture and reuse your design rules, but what are these rules? How do you gather, document, and maintain them? Before you can automate a design, you must understand it. The simplest way to begin is to decompose the problem into manageable pieces.

Design rules can be sorted into three categories. The first are those related to the physical design of the product itself. These rules facilitate component selection, type and quantity. They determine geometric constraints and calculate dimension values. These rules are organized into what we call a Product Model. The second are those related to design procedures. They establish the design space and describe the sequence of steps that are followed in the search for the best design. These rules are organized into what we call a Process Model. The third are those related to the design of the user interface. They allow you to collect the design requirements, and verify that the set of inputs are complete and consistent. These rules are organized into what we call a Context Model.

So, why three models? Because using three models allows you to solve three different problems, each in its own efficient way.

The Product Model is structured to parallel the physical assembly. Each component is documented in-place. Identify when the component is used, what type it is, and what dimensions need to be set. Provide formulas for the values that can be calculated locally. Treat the rest as parameters that are to be passed to the Product Model from the Process Model or the Context Model.

The Process Model is structured sequentially. It is a list of tasks to be performed and the order in which they are to be done. Each step in the process can introduce new design alternatives, or eliminate an alternative from consideration. Each alternative that passes every step is a valid solution, which consists of the set of parameters necessary to drive the Product Model. A final set of rules must be added to sort the valid solutions and identify the best design.

The Context Model is organized based on how the user is to be prompted to supply the inputs to the design process. This event-driven model is defined by triggers that sequence the order in which values are to be prompted for, determine the default values, and validate the entered values. Values collected by the Context Model are made available to the Process Model, or the Product Model, as needed.

True design automation is not possible without the proper tools to capture all three models.

Design Automation with Parametric CAD

Using a parametric CAD system, it is possible to pre-build a master assembly containing a set of potential variations of the design. An instance of a design can be created by opening the master assembly, suppressing or deleting any components or features that are not used, and appropriately setting any dimension values. However, if the variation is not pre-built, it cannot be reparameterized.

The parametric CAD system provides a limited framework to capture some of the Product Model rules, in the form of geometric constraints, equations, or design tables. It typically contains an application programmer's interface (API), which can be used by a knowledgeable programmer to further manipulate the geometric model. The result is an incomplete and fragmented Product Model, which is at best difficult to build and difficult to maintain.

A parametric CAD system is fundamentally an interactive tool, used by an engineer to document the results of his design efforts. As such, the Process Model rules regarding what to do and when to do it remain in the engineer's head.

The ability to construct a custom user interface within a CAD system is extremely limited. With no place to store any Context Model rules, it falls on the operator to be sure that the inputs entered are accurate, complete, and consistent.

The result is an application that is written to be run by the engineer, and is typically limited to creating and reusing a library of parts or features, or to creating and running simple macros designed to automate a repetitive task. This allows the engineer to work more quickly, but does not automate design.

Design Automation with a Configuration Engine

Given the complexities involved with attempting to automate design using just CAD, and the relative success of using a Configuration Engine to automate product catalogs, it is no surprise that there are several add-in products available that drive a parametric CAD system using a Configuration Engine.

A Configuration Engine provides limited support for managing Product Model rules. The assumption is that the Product Model is the responsibility of the parametric CAD system. The Configuration Engine relies on a pre-built master assembly, containing a set of possible design variations. This master assembly can be reconfigured by changing dimension values and by suppressing or deleting unwanted components, but there is no mechanism to insert and mate new components. The creation and maintenance of the master assembly is arguably the greatest source of complexity when attempting to automate design using a Configuration Engine, which provides no assistance in this area.

There is no way to incorporate Process Model rules directly into a Configuration Engine. Instead, each potential alternative which the process might generate must be identified, along with the rules to determine when it is used. In other words, the design process must be reduced to a catalog lookup before it can be "automated". Depending on the complexity of the design problem, this may not be possible, and when it is, significant compromises must be made to achieve a solution.

The Configuration Engine was designed to create and manage a Context Model. This makes it relatively easy to create an interface that does not require the expertise of an engineer to operate. However, the attempts to include Product and Process rules into the Configuration Engine often result in an interface that configures the available product variations, not the problem statement.

The result is an application with a pretty interface that is run by a salesman or a customer that has an in-depth understanding of the product line. It returns exactly one solution, the closest match from the product catalog. The geometry, drawings and reports output by such a system are limited to a reconfiguration of pre-engineered alternatives. This allows the salesman to work more quickly, but does not automate design.

Design Automation with Genus Designer

Genus Designer was created to provide engineers with a complete set of tools to quickly and easily collect, organize, and maintain all three types of design rules.

Product Model rules are captured in a tree-like diagram that represents the structure of the assembly. When the model is run, decisions about the type, existence, and quantity of components are made, and only then are the appropriate parts selected, parameterized, and assembled. The use of a Product Model, independent of the CAD model, reduces the complexity of automating design by eliminating the need to create a pre-built master assembly, and by explicitly documenting all the decisions that create the final assembly.

Process Model rules are captured in a flow diagram containing a sequence of steps to be followed when a product is designed. Each step can either propose one or more alternatives, or test the current alternative to see if it is viable. When the model is run, every alternative that completes each step is identified as a valid solution. The solutions are then sorted to identify the best solution. Any solution can then be generated by the Product Model, producing the necessary geometry, drawings and reports. The use of a Process Model, independent of the Product Model, reduces the complexity of automating design by eliminating the need to write Product Model rules that iterate over the assembly, and by making it possible to extend the search space to include new alternatives without fear of breaking any of the other Process Model rules already in use.

Context Model rules are captured in a layout diagram that identifies the questions that need to be asked, and the order in which they are asked, to properly configure the problem statement. When the model is run, changing an input may update the default value of other inputs, but such a change will never modify another user-specified value. Instead, a warning is issued if the inputs are inconsistent, and the user cannot generate a design until all the warnings are cleared. Only then are the inputs passed to the Process Model, or Product Model, as needed. The use of an independent Context Model reduces the complexity of automating design by guaranteeing that the inputs fed to the other models are accurate and complete.

The result is an application that can be run by anyone. It does not require an engineer or detailed knowledge of the product, it simply requires an understanding of the problem one is trying to solve. Through the use of systematic search and optimization techniques, it returns multiple custom solutions, and identifies the best design. Genus Designer is the only tool that truly automates engineering design.