Implementation Insights

When planning a top-down project in Pro/ENGINEER, what are the two most difficult concepts to implement? Defining the project structure and capturing design intent.

In the last issue of Pro/files, David Paulson of Mechanical Advantage LLC explained the why's and how-to's of using Layouts to capture design intent. This article follows up with a valuable guide to defining structure, taking you through a series of six planning steps with an actual product design project.

If you'd like to learn more about top-down design techniques, be sure to attend Thomas Braxton's presentation at the 2000 Pro/USER International Conference.

Mapping Your Good Intentions

By Thomas Braxton of Motorola

Intent maps can make the difference between a good and great top-down design project.

This planning tool is a graphical representation of the external references used to capture design intent. With this information, project planners have everything they need to develop a project structure that supports the design process. The intent map also leads to the synthesis of design intent that follows the workflow of the project. This concept is scalable and supports projects of any scope. Note that the intent map does NOT define the design process or workflow.

Employing intent maps lets you create "design assemblies" that provide access to the relevant design information required to build a Pro/ENGINEER object. These design assemblies make managing external references less cumbersome. You only have to retrieve the design assembly for a component to make available much of the data required to complete the design. This relieves the designer from the overhead associated with other parts of the project.

Common Planning Pitfalls

External references create a parent/child relationship between two or more Pro/ENGINEER objects. Most of you are familiar with the problem of unwanted dependencies. Where planners often make mistakes is in populating a vertical (as opposed to flat) project space with design manager skeletons, creating unnecessarily long dependency chains. These dependencies can interfere with realizing the full potential of a top-down design approach.

Another common pitfall in defining project structure is to build assemblies (structure) based on geometric constraints. These are usually a reflection of how a design fits together physically or possibly an arrangement derived from manufacturing considerations. In many cases, this approach may also impede effective top-down technique.

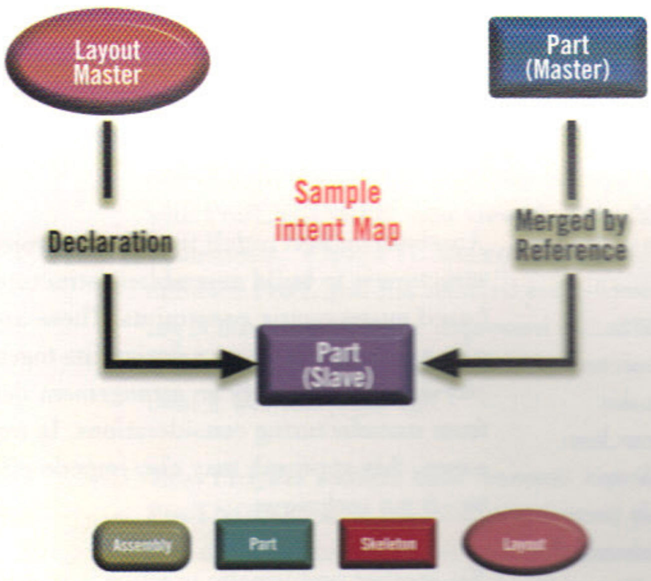
Yet another problematic practice is to use a top-level (all-encompassing) assembly. Because design intent is passed through a conduit of design manager skeletons, this approach fre-

Continued on page 4

Continued from page 3

quently results in long dependency chains—precisely what we want to avoid. In fact, having a single Pro/E assembly serve as "the" top-level object for a design is not only unnecessary but also undesirable.

Layouts are one answer. If equations can define high-level design intent (affecting two or more components), using the layout as a top-level object will enable the effective application of relations to Pro/E objects. The layout is a repository for global parameter definitions, which can be accessed by other Pro/E objects directly through declaration. Alternatively, if you have to capture geometric information that is not readily defined by equations, you can use a reference geometry object—e.g., master models and skeleton parts.



Master/Slave Paradigm

Top-down design assumes that the project is structured in tiers, with each tier having at least one Pro/E object. If there is more than one object in a tier, they become "peers" in the structure. It's important to avoid situations where peers drive one another since this creates circular references.

The answer is to establish master/slave relationships to manage design intent. In the sample intent map shown below, the Pro/E objects are linked by lines indicating the functionality used to capture and propagate design intent from the master object(s) to its slave(s).

To develop this structure, you first need to gather some information for each component of the design.

1. ***Identify and list the subsystems to be designed.***
2. ***Identify and list the components of each subsystem design.***
3. ***Identify and list known relationships between the subsystems and components using the design specification.***
4. ***Identify information that will drive the design data requirements (which establishes the master/slave paradigm for design intent).***
5. ***Referring to the known relations documented in step 3 and the master/slave paradigm established in step 4, determine the best tool in the Pro/E environment to capture design intent.***
6. ***Create an intent map for each subsystem.***

Applying the Process

In my product development group at Motorola, the design phase of a new project typically begins when marketing supplies engineering with a product description (PD) and an industrial design (ID) concept. The PD outlines the volume, weight, finish and the like, while the ID concept attempts to capture a form factor that will appeal to end users. These two bodies of information provide the design specification used to build an intent map for the subsystems of the phone.

The following example maps the intent for a new monolithic or "candy bar" form factor. It is an established part of our design process to build a master model that captures the form or shape of the product based on the industrial design concept. We also use the master merge process to produce derivative models from the master model. I point this out to reinforce the importance of having a grasp on the design process and workflow to building a useful intent map.

1

Identify and list the subsystems to be designed.

We can immediately identify the following subsystems as requiring design activity:

1. *Front housing*
2. *Back housing*
3. *Battery door*
4. *Display system*
5. *Antenna*
6. *Circuit board*
7. *Keypad/controls*

2 ABC

Identify and list the components consisting each subsystem.

At this point, it's not necessary to have even a preliminary design—you only have to identify the components of a given subsystem. As more information becomes available, you can add to this.

Based on the design specification, we have a good idea that the following components will be part of the front housing system.

Subsystem: Front housing

Components

1. *Front housing*
2. *Lens*
3. *Bezel*
4. *Keypad/controls*

3 DEF

Identify and list known relationships between the subsystems and components using the design specification.

The goal of this critical step is to list the information required to design the components. Capturing relatively high-level design inputs is sufficient at this point.

Component: Front housing

Design data requirements

1. *Outer shape*
2. *Mating geometry*
3. *Parting surface*
4. *Control locations*
5. *PCB orientation and location*
6. *Location of key components on the PCB*
7. *Display location*

Tip

Users that know Pro/E well and have a grasp of how the team will accomplish the design should supervise step 3.

Intent Maps

Continued from page 5

4

Identify information that will drive the design data requirements.

The idea here is to determine the source of the required information in the context of the Pro/E environment. Given that we will have a master model of the industrial design concept available, that's clearly where the outer shape geometry resides. As a result, the ID master model (master) will be driving the parts (slaves) that need to access this geometry.

On cursory inspection, the good news is that most of the required data comes from the ID master model. There's no source, however, for some data regarding the printed circuit board (PCB).

Component: Front housing

Design data requirements	*Source of design data*
1. *Outer shape*	*ID master model*
2. *Mating geometry*	*ID master model*
3. *Parting surface*	*ID master model*
4. *Control locations*	*ID master model*
5. *PCB orientation and location*	*Unknown (possibly layout)*
6. *Location of key components on the PCB*	*Unknown (possibly layout)*
7. *Display location*	*ID master model*

Since we know that Layouts created with Pro/NOTEBOOK can capture intent described with equations (relations), we decide that the PCB component location data will be captured using a layout. We can ascertain the location with two measurements relative to a coordinate system and knowledge of the board thickness. The layout can also serve as the source for PCB orientation and location.

5

Choose the Pro/E functionality to capture the design intent.

In many cases, Pro/E offers more than one tool for accomplishing this, so you can simply base your choice on the design process you'll be using.

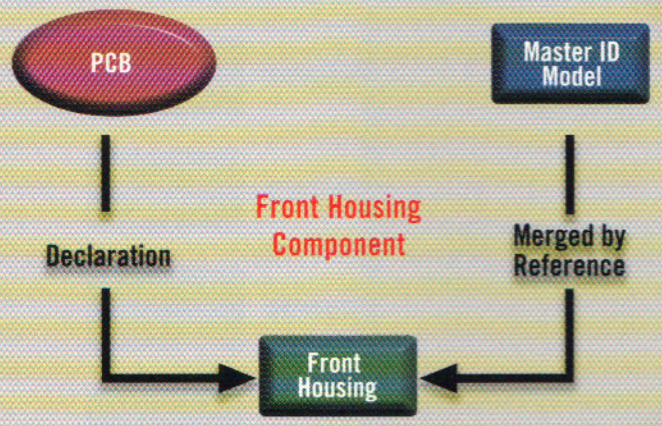
Component: Front housing

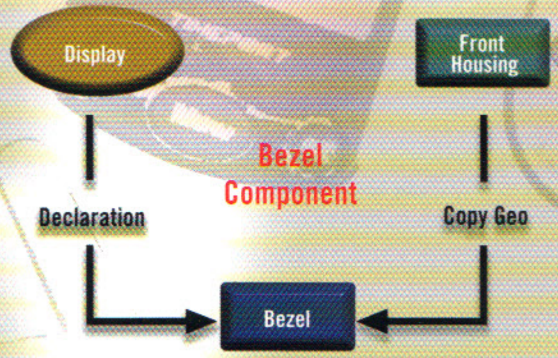
Design data requirements	*Source of design data*	*Capture method*
1. *Outer shape*	*ID master model*	*Merge by reference*
2. *Mating geometry*	*ID master model*	*Merge by reference*
3. *Parting surface*	*ID master model*	*Merge by reference*
4. *Control locations*	*ID master model*	*Merge by reference*
5. *PCB orientation and location*	*PCB layout*	*Declare layout/relations*
6. *Location of key components on the PCB*	*PCB layout*	*Declare layout/relations*
7. *Display location*	*ID master model*	*Merge by reference*

As the intent map takes shape, notice that only two master objects—the ID master model and the PCB layout—are driving design of the front housing. This eliminates the need for an assembly or a design manager skeleton to pass the design intent to the front housing. (Although you could do this with skeletons, it requires more work to create and modify.)

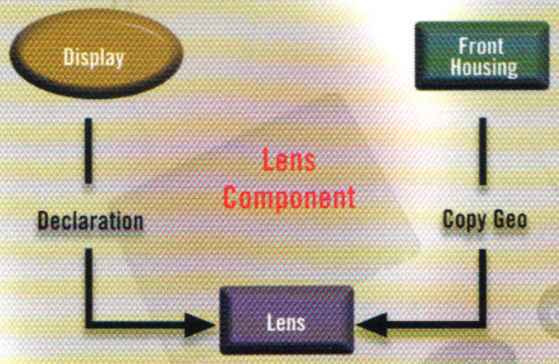
6

Create an intent map for each subsystem.



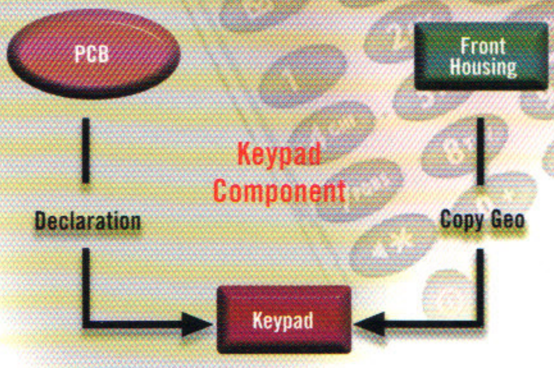


The power of intent mapping becomes obvious when you've applied this process to all components of the subsystem. When completed, the intent map yields one or more design assemblies that will support the design and development of the front housing subsystem.



The intent map thus reveals a structure that provides a propagation path from the master objects to the slaves. Each design assembly also enables the exchange of design intent between peers. The design assemblies should represent a minimized (although probably not globally) set of components needed to work on a given subassembly or part in Pro/E.

Assuming the same person will handle the design of the front housing, bezel, and lens, we should make all these parts available in a design assembly. This is where knowledge of the design process is necessary to build the appropriate structure. The intent maps for the bezel and lens reveal that the front housing is a master object to these parts—another good reason for them to exist in the same design assembly.



Continued on page 8

Intent Maps

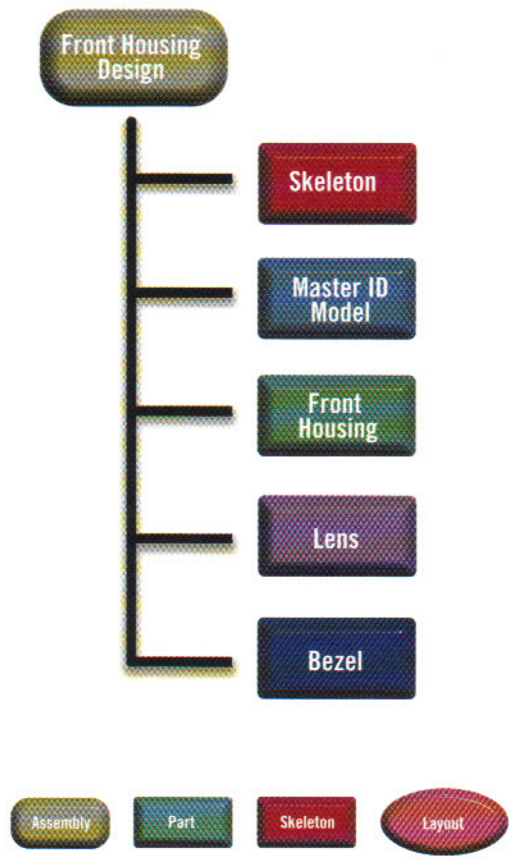
Continued from page 7

Using the intent maps, we can now construct the design assembly that will be used to develop the three parts. We'll add the design manager skeleton in case it's needed later. We'll also make an arbitrary decision not to assemble the keypad, since another team member will be designing this component.

Note that the master model (reference geometry for the phone shape) is assembled to the design assembly for the front housing. This minimizes the dependency chain of references made by peers in the assembly.

The front housing has an established reference through the master-merge technique. By assembling the master model in any working assembly where it may be referenced, other components have direct access to the geometry of the master ID model—a big improvement over placing it in a higher-level assembly and then passing the geometry through design manager skeletons.

Design Assembly for the Front Housing, Bezel, and Lens



Reusing the Assembly

You can readily reuse the components of this design assembly without the legacy of a huge assembly. To do so, though, it's worthwhile to define the reference scope of the components. For example, if you want to have the assembly stand alone so that it can be reused in another design space with predictable behavior, you should probably limit the reference scope of these objects to the current-level assembly. The design manager skeleton will be able to reference the next-higher-level assembly if one exists in the structure with the scope control set to current-level assembly.

Reconsider Conventional Wisdom

If I could impart a single idea to Pro/ENGINEER users about top-down design, it is to minimize dependency chains. By doing so, you will almost certainly end up with a better environment in which to design collaboratively using Pro/E.

The intent mapping process challenges the notion that design manager skeletons are the preferred solution for managing external references in a top-down environment. After applying some of the techniques presented here, you may agree that you have created a project structure that is better tailored to both the people and process involved. ◆

Thomas Braxton is an engineering section manager with Motorola, working as part of a team developing next-generation wireless handsets and information appliances. A Pro/E user since Release 13, he also serves on the Pro/USER Industrial Design and Surfacing Technical Committee. Some of his current research interests are in the areas of surfacing, process improvement, and customization of the Pro/E environment. He can be reached via e-mail at thomas.braxton@motorola.com.