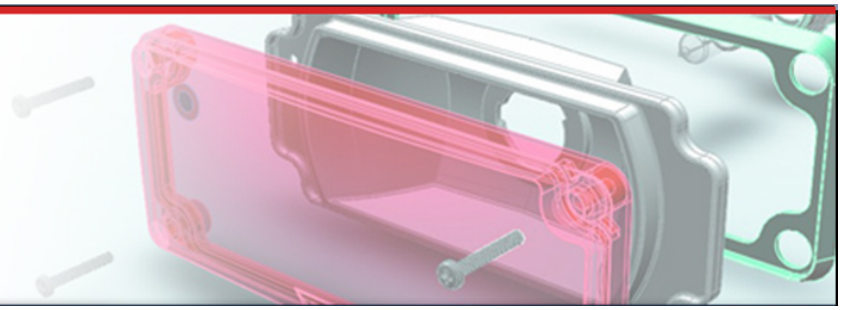
CAP
Computer-Aided Products

KAP's Corner



Thinking Inside the Envelope

- ***SolidWorks***
  - *COSMOS*
  - *PDM*
  - *Rapid Prototyping*
  - *Office Productivity Tools*

Creating envelope components in SolidWorks can be one of the most powerful tools in your repertoire when working with assemblies. KAP invites you to expand your thinking about the use of envelopes any time you need hard geometry to design or mate against, but want the geometry treated as "phantom".

This discussion examines the way envelope components actually behave in an assembly. We show you how to maximize their use to build smarter, more robust assemblies, using interesting cases from the real world.

About KAP

Keith A. Pedersen, Principal Engineer

Keith Pedersen has a BSME from Clarkson College and an MSME from Boston University. After a stint at General Electric in Burlington, VT, Keith was the lead Applications Engineer for Advanced Surfacing products for Matra Datavision USA, including EUCLID-IS, UniSurf, and STRIM. He joined CAP in 1998 to support advanced surfacing applications in SDRC I-DEAS and joined our SolidWorks group one year later. Keith has extensive industry and consulting experience in non-linear Finite Element Analysis and Computational Fluid Dynamics in addition to surfacing applications. He is a Certified SolidWorks Professional (CSWP) and certified to train and support COSMOSWorks.

About CAP

Computer-Aided Products

With offices throughout New England, Computer-Aided Products is the leading regional reseller of CAD/FEA/PDM software, rapid prototyping systems, training, consulting, and support. Our engineering staff includes experts in design, analysis, drafting, PDM, and Windows networks. Our partnership with the best solutions in the industry, along with over 16 years of practical experience, provides a solid foundation for your mechanical engineering and design process.

Your Source for CAD Excellence: Products, Training and Service

www.capinc.com

Introduction to Envelope Components

I consider the ability to create Envelope Components in a SolidWorks assembly as one of the most under-advertised, and under-deployed, capabilities in the system. The training course material, and the on-line help files, will tell you that creating an Envelope solid inside your assembly, will allow you to use the Advanced Selection Tool. You can filter selectively for components that are inside, outside, or crossing the faces of an Envelope.

Also, most of the examples show the envelope solid being created top-down in the assembly. These depictions are true, but they only scratch the surface of the utility of this type of component.

Phantom (Envelope) behavior

I wish we could forget the word "Envelope". It will only limit our thinking. Instead, I want you to think of these components as being "Phantom" parts. I will outline four examples of how a phantom-state component allows you to build smarter, more robust assemblies. I like the term "phantom" because it is a better description of how these types of components actually behave in the assembly.

KAP's Tip: Understanding how Envelopes behave in an assembly gives you the ability to maximize their use as design objects.

1) ANY SolidWorks part file can be added to an assembly as an Envelope. The same part file can exist in the assembly as several Component instances, and as several Envelope instances. There is nothing special within the part file itself that makes the part an envelope.

You can build it any way you want, and configure it as well. Effectively, adding a part file as an Envelope simply gives it different traits compared to a regular Component. The Envelope part behaves as if it exists geometrically in space, but does not exists logically within the assembly hierarchy.

2) The Envelope adds no mass to the assembly

3) The Envelope adds nothing to the Bill of Materials

4) If the assembly is then used as a sub-component in a larger assembly, the Sub's envelopes disappear – they are only visible within the owning assembly.

5) The Envelope will not interfere with other parts via Tools – Interference Detection

6) It will, however, cause interference when using Dynamic Collision Detection.

7) Its faces and edge geometry can be mated to, sketched on, and used to create or locate reference geometry.

8) It displays by default in a watery translucent blue color.

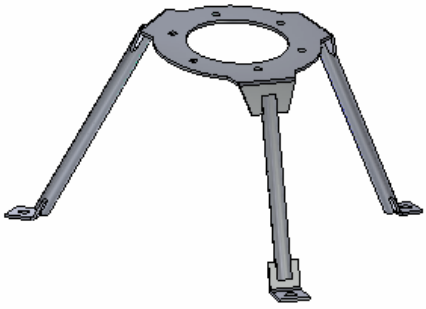
We can exploit this behavior in a large number of real-world, interesting cases. Here are my favorite examples:

Case 1: Tooling

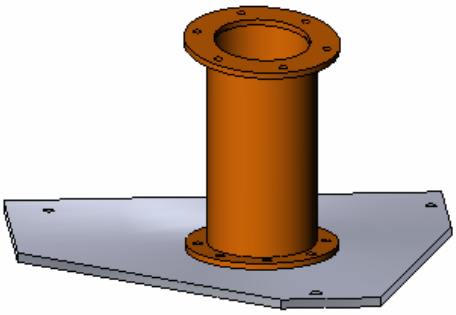
KAP's Tip: Our use of the tooling parts as Envelopes has helped us capture important fit-up information, and correctly documents the fabrication process, without confusing the final Bill of Materials for the end-product.

Many assembly processes require custom jigs and fixtures. Welding processes are a good example. Consider the picture below, left. This is a frame where the three 'feet' must be carefully positioned and aligned relative to the center plate. To achieve this, the weld fixture pictured below, right, is fabricated first. The center spool maintains the desired height and alignment. The beauty of this approach is the automatic updating.

Of course, the Spool and the Base plate need to be designed, ordered, and fabricated in their own right, so you will likely create CAD geometry for these. However, when creating the CAD assembly for the finished Frame, the first parts you should add are the Spool and Base Plate, added as Envelopes. Then the Feet and the Center Plate can be mated to the tooling, and the three Legs can be designed Top-Down to exactly fit-up as needed.

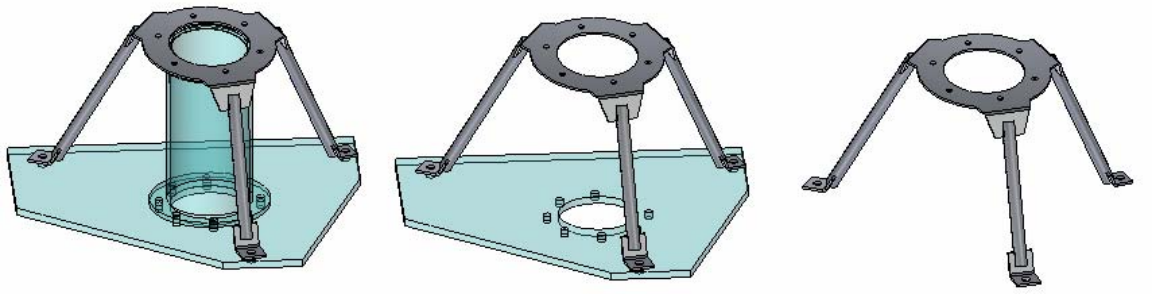


Frame



Weld Spool and Base Plate

In the example I have chosen, the center spool gets removed as soon as the welding is completed, but the base plate remains attached to protect the Frame during shipping and to facilitate sub-assembly. At final assembly, the base plate is removed.



As Welded *As Shipped* *In Service*

Case 2: HARDWARE KITS

Usually the hardware, (nuts, washers, PEMs, etc) associated to a sheet metal plate or a sub-assembly is included at the same assembly level as the items being fastened. However, some companies like to create separate line-items for Hardware Kits in their Bill of Materials, often as an item at a different level of the assembly.

KAP's Tip: Use Envelopes to customize your Bill of Materials while keeping the correct relationships in your assembly components.

How can we de-couple the Fasteners from the Fastened, and yet have the hardware all positioned correctly, and update associatively if the models are changed? Make the first component(s) in your hardware kit, be Envelope(s) of the affected drilled or punched components. All the fasteners can be mated to this (phantom) part. When the hardware kit is added to the main assembly, the Envelope part vanishes, and you will need only 3 mates to put the entire array of hardware into position.

Sometimes this idea is applied simply because it allows you to group large numbers of Mates into 'kit' sub-assemblies, where they have no impact on top-level assembly performance.

Case 3: SNEAKY PATTERNS

In a Part file, we have lots if cool ways of making Pattern features, including Sketch Driven Patterns, Curve-Driven Patterns, Table-Driven, etc. But in an Assembly file, you have only three choices: Linear, Circular, and Feature-Driven. What if you really want to create a Sketch-Driven pattern, of Components, at the Assembly level?

You can, if you first create a 'dummy' part, and add the part to the assembly as an Envelope. Inside the 'dummy' part, create the positioning sketch, and create a simple 'dummy' feature to pattern. (The positioning sketch can be created while working top-down, of course, so that your positions can be related to interesting locations in the assembly). Then create the solid feature pattern in the dummy part.

KAP's Tip: Extend part Pattern functionality into an assembly via Envelopes.

Now you return to the assembly level, and create a Feature-Driven pattern in the assembly, that references the pattern in the dummy part. Finally, just hide the envelope. Because the dummy part was added as an envelope, it will not mess up your weights, BOM, or be confused by a later CAD user as being a design element.

Case 4: SKELETONS

In the Advanced Assembly class, we talk about how you can enforce Top-Down design intent on an assembly by use of Reference Planes, Axes, and Sketches. In the lab examples, reference sketches are the first thing added, as assembly features. But we can expand on that thinking and have the first element be an Envelope, and use it as a Reference Part. This will enable the Top-down design to be collaborative.

Consider a design project where three users must work together in an assembly within limited space. Only one of them at a time can open the top-level assembly with read/write access. And, suppose that the top assembly is composed of thousands of parts, so that you want to limit the number of times that any of them needs to even have the top assembly open. How can we use Envelopes to coordinate these users' work, at lower levels of the assembly?

KAP's Tip: Have an assembly requiring collaboration? Use Envelopes to facilitate and simplify top down design.

The first thing you do is create a Part file that will lay-out the design. It could have a simple solid body to define the overall assembly size, plus planes and sketches to define function. Then it could have three (or more) additional solid bodies that serve as the volume-budgets that each designer must stay within. On the faces of these working volumes, we place more sketches, showing critical interfaces or bolt patterns where

the work of the three designers must fit up to each other. This one part file now serves as the 'backbone' for the assembly design.

If we assign responsibility for a sub-assembly to each of the three designers, then we should put our "Layout Part" as the first Envelope in each of their sub-assemblies. Now the three designers are all working to a common plan, and can guarantee their parts will mate together, without any of them needing to open the top-level assembly.

Conclusion

In summary, consider using the Envelope functionality any time you need hard geometry to design or mate against, but do not want to treat that geometry as a 'real' component of the assembly.

I have a few other examples I sometimes use when discussing this topic in our CAP University sessions, but these four should give you the basic idea, and they are my easiest examples to describe when you are limited to print format. If you come across any cool additional uses that I can add to my bag of examples, I'd love to hear from you – you can Email me at:

KAP@CAPINC.COM