INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

DESIGN 2010

GEOMETRY INTERACTIONS IN CONFIGURABLE PLATFORM MODELS

P. Edholm, H. Johannesson and R. Söderberg

Keywords: geometry interactions, geometry interfaces, platform model

1. Introduction

For many companies, product and production platforms are the strategy to get economy of scale and reduce time to market for new customized products. Production platforms do, however, often limit the design freedom for the developer. Making both product and production platforms fully configurable would be very beneficial for adapting product concepts to various customer requirements. To realize such platforms, new ways to design and configure platforms are needed [Gershenson et al. 2006].
In this paper, a platform based on configurable autonomous subsystems has been used to describe the product platform. This concept has been described and developed in several different areas [see [Claesson, 2006], [Gedell, 2009], [Wahl et al., 2009] and [Edholm et al., 2009] for more information].
When developing a product, the geometry interfaces are of great importance to achieve a geometrical fully compatible product. To achieve robust and compatible geometry interfaces, an interaction activity needs to take place between the different geometry interfaces of the components within the platform. To fully describe a configurable product and production platform, this interaction activity and the definition of geometry interfaces need to be clarified. In this paper, a new internal model element is introduced to handle interactions between interfaces, and geometrical interfaces have been categorized.

2. CC Platform description approach

Several researchers have defined alternative platform descriptions to get more variability in the platform. One such more abstract configurable platform model, proposed by [van Veen, 1996], consists of generic bills-of-materials (BOMs) that provide possibilities to describe large varieties of product types and structures. The idea is to define products as "sets of product types", instead of defining "individual product types". This concept has been applied and explored further by [Erens, 2001] when developing product families and synthesizing product variants. [Männistö et al., 1990] also propose a strategy related to the idea of a generic BOM. They describe a "master BOM", which is a generic description for many product variants that can be defined and manufactured on the basis of the platform.
The 'CC platform', used in this work, is based on autonomous knowledge-carrying configurable generic subsystems, so-called Configurable Components (CCs) [Claesson, 2006]. This approach provides much more configuration flexibility than a part-based defined platform. Such a configurable and knowledge-based platform architecture is also much more robust, as reuse of configurable subsystems instead of reuse of parts makes it possible to have the complete product knowledge contents available for redesign in order to meet new demands. Geometry interactions using

Configurable Components in such a knowledge-based platform architecture context are focused upon in this paper.
When adopting the CC approach, a product or a product platform is described by a linked set of configurable components; see Figure 1. Each CC has relations to other CCs and is instantiated by setting values on its variant parameters (VPs) in the 'control interface (CI)'. Composition of variants is defined by the 'composition set (CS)' with its 'Composition Elements (CEs)', which use variant parameters, design concept definitions, and design rules to compose an instantiated variant. The result of a composition is, if necessary, transferred to other CCs used or to external applications, like PDM and CAD systems, to fully generate variant-defining information needed to prepare for production. This kind of functionality can be achieved if each CC has the following necessary knowledge about itself:

1. Knowledge about its origin, i.e. what it should do and be, how this is realized, and why
2. the chosen solution is what it is.
3. Knowledge about its interactions with the external environment.
4. Knowledge about how to compose variants of its design solutions by means of its internal resources or by using other external CCs.

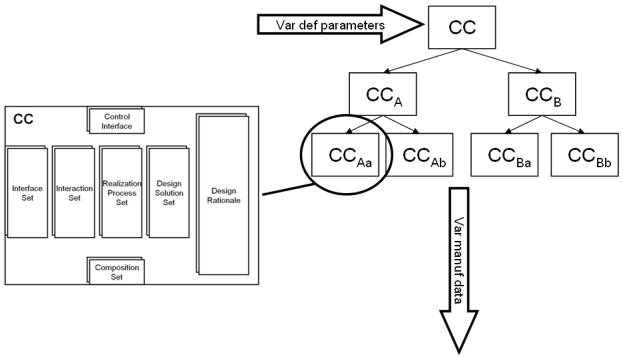

Figure 1. Example of a CC platform

The aim of the performed research has been to answer the following research question:
Which elements are needed in a CC model to ensure the geometrical location and the geometrical relations to the surrounding systems, and how can this be realized in a PDM/CAD environment?

2.1 CC definition

The CC object is mainly defined by three elements: Control Interface, Composition set and Design Rationale. The Control Interface is used to enable external communication with the CC object, and the Composition Set is used to compose the variants using the CC's internal design solutions or by using other external CCs. Finally the design rationale is described with a set of function-mean trees, defined by functional requirements, FRs, and design solutions to solve them, DSs. See the left side of Figure 2 for a schematic view of a CC object. All design solutions are structured in the same manner, but are categorized for convenient purposes.
In the function-mean tree, all design solutions are governed by a specific functional requirement, which means that the design solution is defined as the realization of a specific functional requirement by using, for example, an engineering solution. See the example in Figure 3 of a function-mean tree for the design solutions 'Interface' and 'Interaction'.
A design solution object contains two main elements (see the right side of Figure 2) and references to external information-carrying items (not shown here):
Parameter set, which describes the design solution with full bandwidth to enable configuration flexibility. The design solution can carry several sets of parameters.
Parameter map, which describes where and how the different parameters should be set.

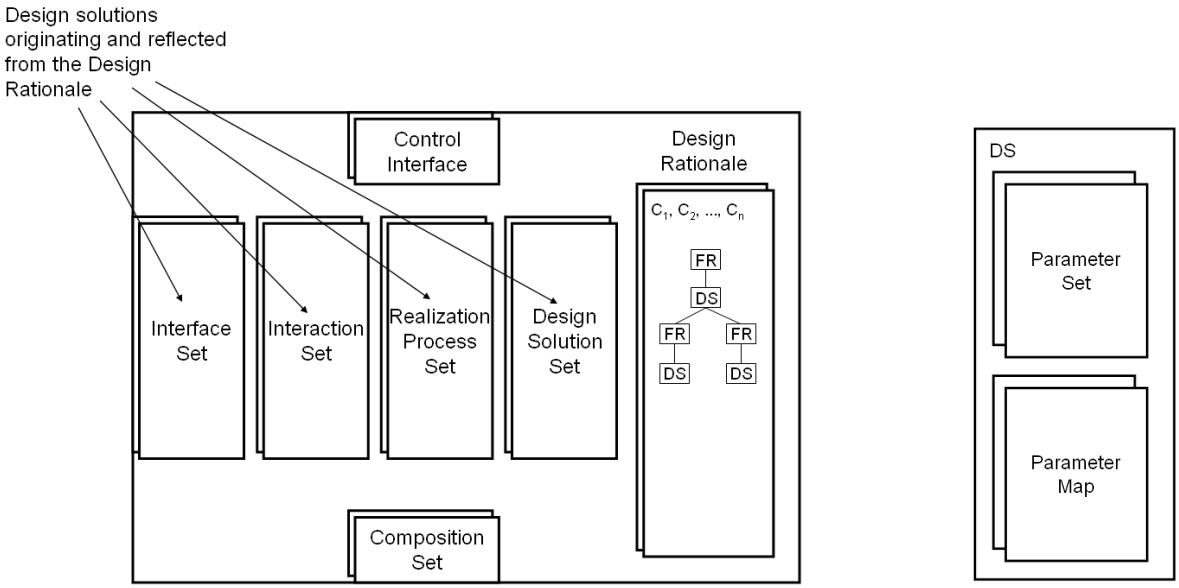

Figure 2. The different subsets within a CC object and a generic design solution definition

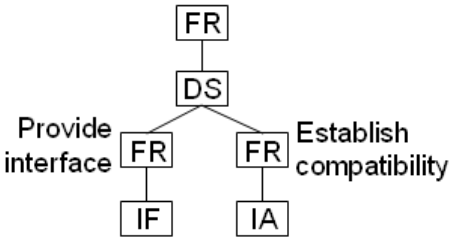

Figure 3. Function-mean tree

The following four different categories of design solutions are defined at present (see also Figure 2):
IF - Interface ("to the surroundings")
These design solutions are required to define the criteria for compatibility with the surrounding parts.
Examples: locating schemes, hydraulic transmission, data protocols.
IA - Interaction ("between interfaces"), defined in this paper
The interaction design solutions are used to create the compatibility between the different interfaces.
Examples: critical packaging sections, location of parts and orientation of locating schemes.
RP - Realization process ("engineering events/methods")
This kind of design solution differs slightly from the others, because it describes events in a process more than traditional designs. The solution is in this case the path to achieve a result.
Example: optimizing a locating scheme.
DS - Design solutions realizing required functional behaviour
Examples: car door, software for ABS brakes.
In this paper two types of design solutions, *interface* and *interaction*, are focused upon. The use of these two design solution categories is limited in this work to handling only geometry interfaces and geometry interactions.
See [Edholm et al. 2009] for a more detailed description of the CC object. (Note that the definition of the CC object is continuously developed and some parts may have changed.)

3. Geometry interfaces

Two types of geometry interfaces (IF design solutions) are defined in this paper: locating schemes and mating geometry. These two types of geometry IFs are used to describe the interface solutions

between the CAD geometry as well as the locating constraints both in the virtual assembly in the CAE systems and in the physical assembly in manufacturing. Below follows a detailed description of the two geometry IF types. Like all other types of objects within the CC concept, the geometry IFs are described with a bandwidth and are fully scalable and possible to parameterize.

3.1 Locating schemes

Locating schemes are used to locate the physical part in the coordinate system. Six discrete points are often used to represent the locating scheme for a part. Figure 4 shows the concept of the six-point locating scheme, often referred to as the 3-2-1 locating method.

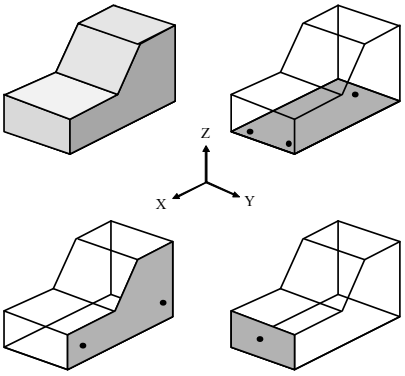

Figure 4. The 3-2-1 locating scheme

The 3-2-1 locating method described in Figure 4 locates the part such that: first, three points lock three degrees of freedom (TZ, RX, RY); second, two points lock two degrees of freedom (TY, RZ); and, finally, the last point locks the remaining one degree of freedom (TX).
In [Edholm et al. 2009] the locating scheme has been defined and used in a configurable component model.
See Figure 5 for an example of a locating scheme for a part.

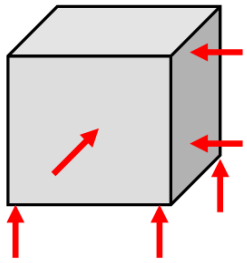

Figure 5. Locating scheme for a part

Locators have many different uses in the product development process and manufacturing [Edholm and Söderberg 2009], but the main purpose of locators is to locate the physical part in the coordinate system. Fixtures are used for this purpose to clamp the points shown in Figure 5. In this case an interaction occurs between the geometry interface (locating scheme) of the part and the clamps of the fixture (the physical realization of the locators).
The locating scheme could, for example, be described as six 3D-points in the CAD model.

3.2 Mating geometry

Mating geometries are defined as the geometry to the surroundings. These mating geometries are critical when designing assemblies. This could be due to packaging issues (i.e. making all parts fit together within a specific volume without clashes) but there could also be functional requirements involved (e.g. gears in a transmission need to be in contact to be able to transfer torque, or mating

surfaces must have a specific gap before welding). See Figure 6 for an example of mating geometries, A and B, for a part.

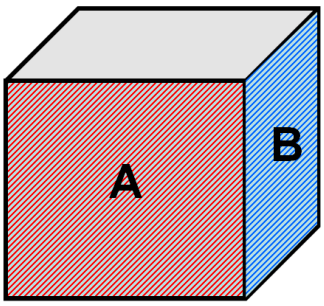

Figure 6. Mating geometries for a part

The mating geometry may interact with mating geometries on other parts. The interaction between different mating geometries could consist of packaging issues or functional requirements as described above. The mating geometries are defined in the CAD model as features, for example surfaces.

4. Configurable geometry interactions

The geometry interactions (IAs) are interactions between two geometry interfaces (IFs). To enable full bandwidth according to the CC concept, the geometry IAs need to be fully configurable as well. This means that a specific geometry IA object can be used in several different ways and is scalable to adapt to different geometry interface design solutions (IFs). Figure 7 shows an example where the locating scheme of a part interacts with the physical positioning clamps in a fixture. This is an example where two geometry IFs (locating schemes) are configured together. The IA object will in this case ensure that the clamps on the fixture will be in the same location as the locators (or if the manufacturing system is the base, ensure that the locators are located in the same coordinates as the clamps).

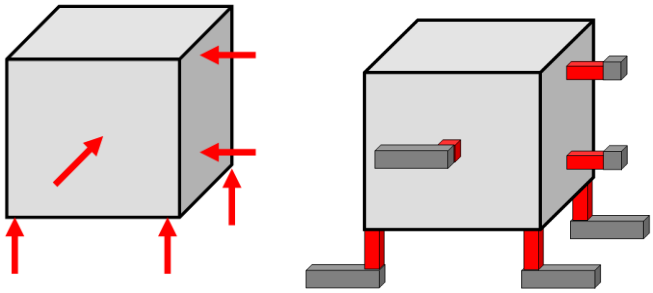

Figure 7. Geometry interaction between locating scheme and fixture

Like all other types of design solutions in the CC object, the IA design solution carries its own parameter set, which defines how the interaction between two interfaces (IFs) can be configured, and a parameter map that describes how the different parameters could be set.

An IA may define which IF acts as master or slave. For example, if one surface on a CAD part should be shaped according to a surface on another part, the IA is defined to handle the master part IF (surface) as prerequisite for the slave part during configuration, and to adapt the IF (surface) on the slave part to the master part.

In the fixture example above (Figure 7), the part is already designed when the fixture design starts. In this case the clamps on the fixture will be configured by the IA design solution to suit the already defined surfaces in the locating points.

In Figure 8 an interaction between surface interfaces is shown. It is not necessary to have a defined master/slave relationship in these cases. The IA design solution could, for example, handle the function to ensure fulfilment of a specific demand such as a gap between the two parts. In this case the

configuration flexibility is equal for the two IFs (surfaces) on the two parts. This is also often the case when designing complex assemblies with many parts where packaging of parts in a specific volume is the issue. The criterion for the IA design solution is then to ensure that there are no clashes between the two parts.

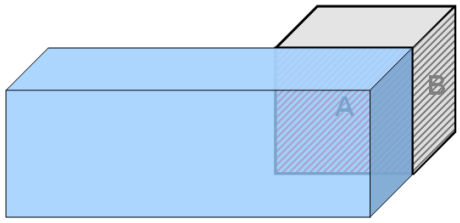

Figure 8. Interaction between two mating geometries on two different parts

See Figure 9 for a schematic view of how the IA design solution in one CC object, *Part 1*, can be used to configure the IF of another CC object, *Part 2*. The CC object Part 1 contains different interaction (IA) objects, one for configuring the locating scheme, IA_1, and one for configuring the mating geometries, IA_2. Even if Part 1 holds the information about the interaction criteria, this does not mean that the interface (IF) design solutions in Part 1 are master for Part 2. It means that Part 1 controls the interaction between Part 1 and Part 2.

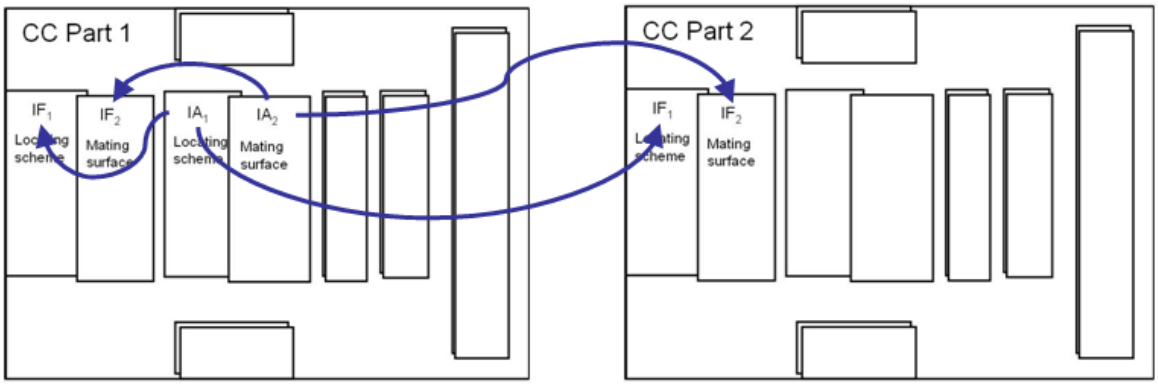

Figure 9. The interaction design solutions configure the interface design solutions

The IF design solution and parts of the CC definition have been implemented in a demonstrator software, CCM (Configurable Component Modeller), where it is also possible – so far with limitations – to describe platforms on the basis of the CC concept and to configure specific product variants by using given input parameters. See [Edholm et al., 2009] for more details about this demonstrator software and the CC definition used.

As said before, the interactions (IAs) in the CC platform model are fully configurable and are described with a bandwidth. They do not need to be fixed during the development, but are set during configuration. This is the key to enabling efficient development without suboptimizations of specific requirements.

5. CAD/PDM realization

TeamCenter (PDM solution from Siemens) and Catia V5 (CAD system from Dassault) have been used to create a technical model of the CC concept. This model is not enough to fully reflect the CC concept, but is used to visualize the potential in a CAD/PDM environment. See the left side of Figure 10 for the structure in TeamCenter that is used to describe a door module CC object (see section 6 for the application example of this door module). Beside the CAD model of the part, there are interface and interaction objects connected to the door module CC object. An open text format is used to

describe the interfaces. This format simplifies the connections to applications using the interfaces (for example, simulation tools or connections to CAD models created in other CAD systems). The interaction objects are here defined in Microsoft Excel format. This format can also be read by several applications, and Excel is easy to use to implement intelligence (formulas etc.) in the object.

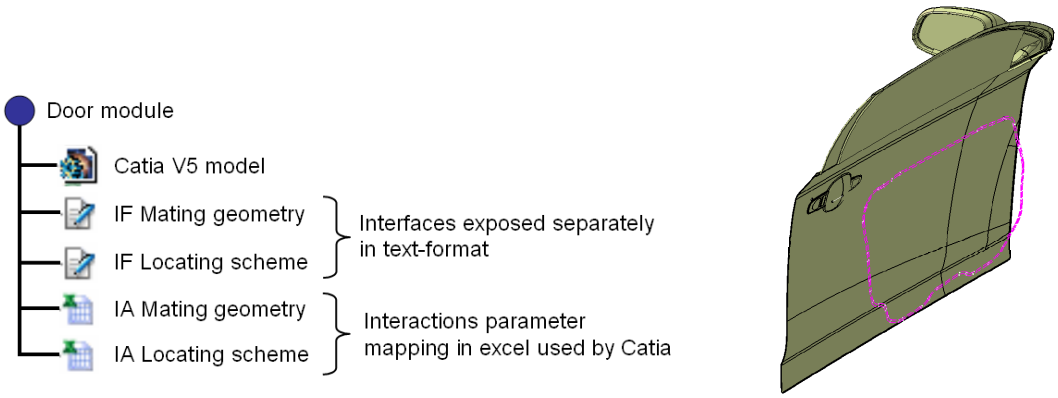

Figure 10. The object structure in TeamCenter to the left and Catia V5 model of start model for door frame to the right

The actual function of the Excel interaction objects is in this case to automatically create the interface geometries; see the right side of Figure 10 for an example. In this example a restriction model for the door frame is created using the product design data for the door and the mating geometry for the door module. These data define pre-requisites for the engineering work. In similar way other pre-requisites could be added to fully configure the door frame.

6. Application example

An industrial case is used to illustrate the concept of geometry interactions (IAs) in a real-life product development environment.
A platform for car doors is to be used to define door variants. The doors consist (in this case study example) of a door frame and a door module.(see Figure 11.)

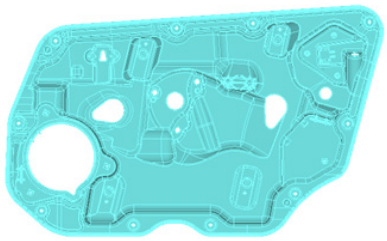

Figure 11. Door module

The platform should enable several different door variants, but by using the exact same door module. See Figure 12. for a basic schematic view of the car door platform described by using the CC concept.

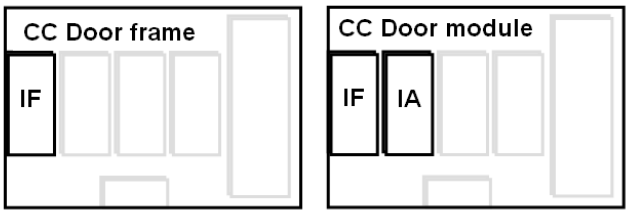

Figure 12. CC Car door platform

In this example the CC object *Door module* carries the IA design solution, meaning that the interaction between the interfaces (IFs) of door module and door frame is controlled by the door module.
The door module and door frame IFs in this example are defined by both a locating scheme and a mating geometry on each part. To simplify the case, the designs are already completed for all parts except for their geometrical IFs.
The geometrical IFs on the door module are here fixed, meaning that the locating scheme and the mating geometry to the surrounding parts are already defined. See Figure 13 for the definition of the geometry IFs for the door module.

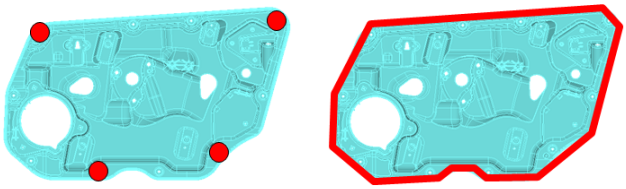

Figure 13. Door module locating scheme (indicated by red points) to the left and mating geometry (indicated by red surface) to the right

The schematic view of the part of interest here, of the car door platform with design solutions defined, looks like Figure 14. This shows the content of geometry IFs and geometry IAs respectively in the different CC objects before configuration.

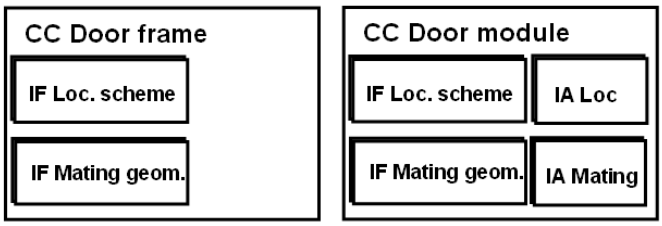

Figure 14. Schematic view of the content of the CC objects in car door platform

When two different car door variants from the platform are to be defined, the configuration starts by giving input criteria according to the two different variants. In this case, engineering data (CAD models) for the two door frame variants are given as input, and the first step of configuration is to instantiate door frame CC objects to create two – except for the IFs – configured door variants; see Figure 15. Note that these two variants could be using different instantiations from the same generic door frame CC. The instantiated door frame CC objects are still unconfigured and defined with full bandwidth. The second step of configuration in this case is to configure the geometry IF design solutions in the two door frames (the geometry IFs in the door module are given), by using the interaction (IA) design solutions in the door module CC.

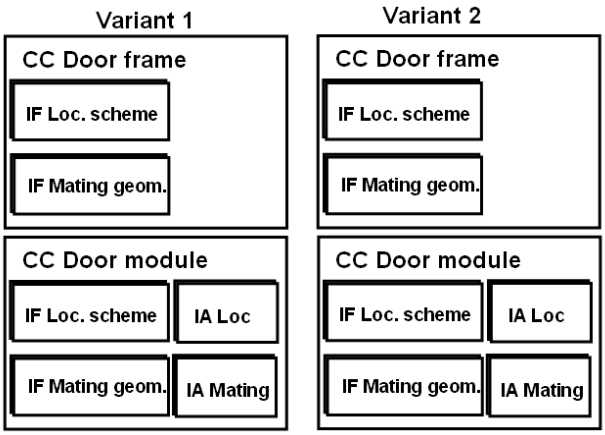

Figure 15. Instantiated CC structure of two door variants

The geometry IA for locating scheme will create features on the two door frame variants according to the locating scheme of the door module. This will enable the door module to be located and fixed to the door frame in the coordinate system in production. See Figure 16 for a schematic view of the locating scheme features on the two door frame variants.

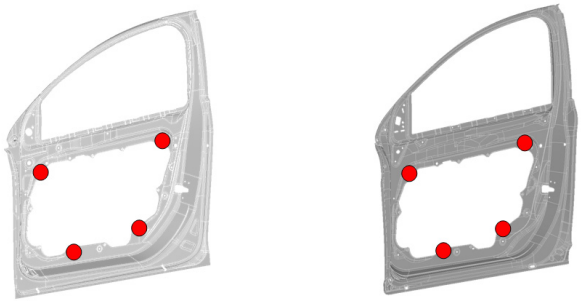

Figure 16. Geometry interfaces (IFs), locating scheme

The geometry interaction for mating geometries creates mating surfaces on the door frames to ensure that the door module will fit in without clashes and according to the requirement specifications. See Figure 17 for a schematic view of the two door frames with mating surfaces defined according to the IF design solution in the door module CC object.
When the geometry IF design solutions are set in all CC objects, the configuration is done in this case. The result is two door variants that use the same door module with full geometric compatibility.

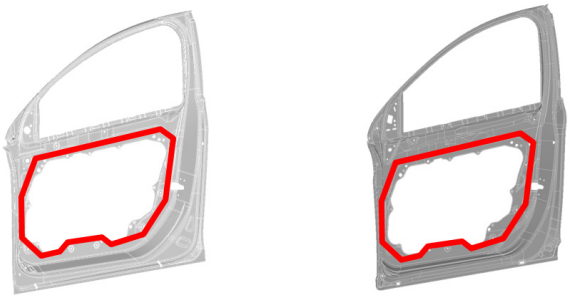

Figure 17. Geometry interfaces (IFs), mating geometry

7. Discussion and conclusions

The work in this paper has its basis in the CC concept of describing configurable product and production platforms. In such platforms the interfaces between different parts described as CC objects and the interaction between the interfaces, are critical to achieve a high level of compatibility between parts and assemblies. The interface (IF) component within the CC object is already defined in previous work. As a result of the work described in this paper, the interaction (IA) object is defined as well. The interaction (IA) definition has been tested in an industrial application example with focus on the geometry IA. Even though the test is narrow and limited to a single example, one can see the need of such an object in order to be able to manage the geometry compatibility between parts. The test also shows that the IA definition could, in a logical way, be used in a real-life application with part CAD data. However, both the interface (IF) and the interaction (IA) definitions need to be extended to involve not only geometry, but also other areas to ensure full compatibility in configured variants from a CC-described platform.
The answers to the research question stated in section 2 are:

1. *Interaction objects in a CC model can be used to ensure the geometrical location and the geometrical relations to the surounding systems.*
2. *It is possible to describe the interaction objects separately in the PDM/CAD environment to enable the functionality needed by the objects.*

Acknowledgement

This work was carried out within the Wingquist Laboratory VINN Excellence Centre at Chalmers University of Technology. It was supported by the Swedish Governmental Agency for Innovation Systems.

References

Claesson, A. 2006, "A Configurable Component Framework Supporting Platform-Based Product Development", PhD Thesis, Chalmers University of Technology, Göteborg.

Edholm, P., Söderberg, R., "Locator management for platform-based architectures in PLM systems", 11th CIRP International Conference on Computer Aided Tolerancing, Annecy, France, March 26-27, 2009.

Edholm, P., Wahl, H., Johannesson, H., Söderberg, R. 2009, "Knowledge based configuration of integrated product and process platforms", Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, San Diego, California, August 30 - September 2, 2009.

Erens, F.J. "The synthesis of variety: developing product families", PhD thesis, Eindhoven University of Technology, Eindhoven, the Netherlands, 1996.

Gedell, S., "Platform-Based Design - Design Rational Aspects within the Configurable Component Concept", Licentiate Thesis, Report No. 46, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, 2009.

Gershenson, J.K., Khadke K.N. and Lai X., 2006, "A Research Roadmap for Robust Product Family Design", Dept. of Mechanical Engineering-Engineering Mechanics, Houghton, Michigan.

Johannesson, H., Claesson, A., "Systematic product platform design: A combined function means and parametric modeling approach", Journal of Engineering Design, 16, 1, 2005, 25-43.

Männistö, T., Peltonen, H., Soininen, T. and Sulonen, R., "Multiple abstraction levels in modelling product structures", Data and Knowledge Engineering, Vol. 36(1), pp. 55–78, 2001.

van Veen, E. A., "Modeling Product Structures by Generic Bills-Of-Material", PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 1990.

Wahl, A.L., Birgersson, M. and Johannesson. H., 2009, "Configurable Geometrical Platform System Interfaces", NordPLM'09 2nd Nordic Conference on Product Lifecycle Management, Göteborg, Sweden.

Peter Edholm
Chalmers University of Technology
Department of Product and Production Development
SE – 412 96 Gothenburg, Sweden
Telephone: +46 708 167924
Email: edholmp@chalmers.se
URL: http://www.chalmers.se/ppd/EN/