Title:
Review of CAD-model Capabilities and Restrictions for Multidisciplinary use

Authors:
Tim Heikkinen, tim.heikkinen@ju.se, Jönköping University
Joel Johansson, joel.johansson@ju.se, Jönköping University
Fredrik Elgh, fredrik.elgh@ju.se, Jönköping University

Keywords:
Computer-Aided Design, Simulation-ready CAD-models, Knowledge re-use

DOI: 10.14733/cadconfP.2017.337-341

Introduction:
Product development is an iterative process, partially due to changes in both company internal and external product requirements, resulting in changes to the product under development. These changes might require recapitulation of design rationale and result in re-doing assessments and syntheses of different kinds. One way to support this work is to proactively model in such a way that as much as possible of the previous work can be re-used. Modelling for re-use can be done by documenting design rationale and formalising performed activities as design guidelines or computer scripts. To be able to find and effectively re-use design activities, the added support could be attached to or otherwise linked to the product features they relate to. This paper focuses on the native CAD-models which also has been utilized by others as carriers of information for different purposes. For instance, [7] describes the use of annotations on CAD-models, an example is presented where both FEM and CAM specifics were added to the geometry enabling automatic FEA and blank casting geometry creation. The approach was later used to support both constraint based redesign activities [2] and encapsulation of in-development/in-service information for throughout product lifecycle retrieval [1]. The information was finally stored outside the CAD-models together with lightweight representations of the geometry [3]. Since CAD-models are such a central part of the PD process, relating to many design activities within multiple disciplines, the questions we asked ourselves were: *(1) How can CAD-models be used as carriers of multidisciplinary information? (2) Which of these approaches are most common in industries today? (3) And what differences can be found between them?* The rest of this paper starts by introducing the research method. The results are then structured in three parts, starting with an exploration of an existing method using the CAD-model as a carrier of multidisciplinary information, followed by a summary of extension techniques found, and finally the results from interviews within three Swedish companies. The paper then concludes with discussions and conclusions drawn.

Method:
This work is part of a research project, called ChaSE (Challenge Fluctuating and Conflicting Requirements by Set-Based Engineering), where the main objective is to provide “A novel method to develop and describe adaptive technology solutions with an ability to manage changing and conflicting requirements in the development of customised products.” Within this project, the Design Research Methodology has been employed. The general approach is to define scientifically and industrially relevant goals through literature and empirical data analysis followed by descriptive and prescriptive loops where support methods are developed and evaluated. In the previous phases, we found an industrial and scientific need for improved flexibility within product and technology development. A method, called simulation-ready CAD-models, was developed as a means to enable flexibility with respect to Finite Element Analyses (FEA) [5], [6]. As a result of the successful evaluation [4], the

method is now attempted to be generalised. To answer the questions above the work was divided in two main tasks; exploration of the existing method utilizing the CAD-model as a carrier of multidisciplinary information, and interviewing employees at three Swedish industrial companies.

Method utilizing the CAD-model as a carrier of multidisciplinary information:

As a starting point for this study an existing method utilizing the CAD-model as a carrier of multidisciplinary information was explored. This method is an updated version of the work presented in [5]. The method enables formalization of FEA from within CAD software and had been realized in Solidworks, a feature-based CAD-modeller. ANSA was used as a pre-processor and LS-dyna as a post-processor. The approach was to enable the formalisation and execution of pre-processor operations from within the CAD environment, providing a link between geometry and FEA features. Examples of FEA features are mesh geometry such as beam, tetrahedron, pyramid, and triangle elements as well as FEA properties such as material, constraints, and boundary conditions. To do this the traditional CAD-model geometry was not always enough and was therefore extended with additional geometric entities, created only for FEA. As an example, see surface within bolt in Fig. 1(a) which is used to create a structured triangular mesh utilising the rotation operation. Programmed features (known as macro-features in Solidworks) had been made which when instantiated provided an easy to use user-interface with the necessary inputs, such as geometry, strings, and numerals, illustrated on left-hand side in Fig. 1(a). Just as any other CAD-feature it was placed in the feature-tree which can be seen in the centre of Fig. 1(a) called MeshPart1. In this way, it was a part of the CAD-rebuild process and updated according to any related changes made. The execution of the programmed-features was separated to run all in one go in contrast to the other functional features however. To automate pre-processor operations within ANSA a template macro-script had been made where each FEA operation had a specified code-section. In this way, each programmed-feature could copy the necessary code-sections and replace the needed inputs at the given position. Geometric inputs were exported in a neutral CAD-file format (Parasolid) and the components names were used as identifiers.

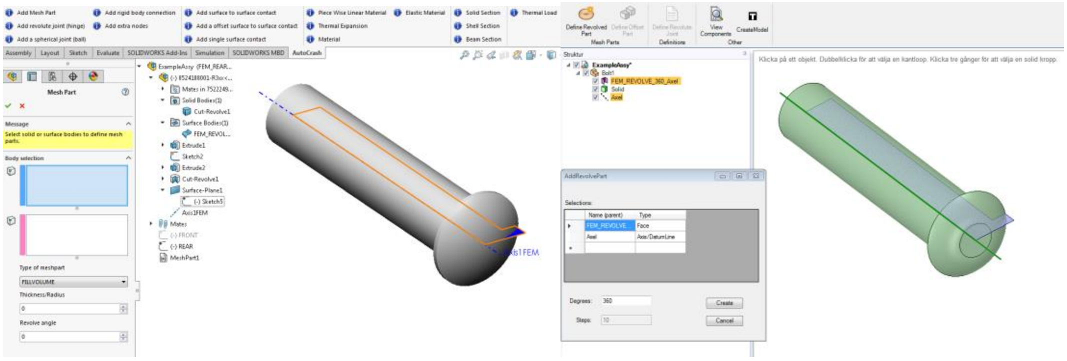

Fig. 1: From left to right, a) SolidWorks GUI b) SpaceClaim GUI.

To further understand CAD-model capabilities and restrictions the method was applied to a direct modelling software (SpaceClaim), see Fig. 1(b). Direct modelling software does not save the operations history and therefore needed another approach to the parametric feature based implementation. Instead of programmed features, a naming convention was used on both geometric entity name attributes and additionally added attributes. Name attributes could be used to store all the information necessary, but quickly become incomprehensive for human users as the information content size grows (see section “Extension techniques and their implications”). Only the links between geometric FEA features (e.g. beam, tetrahedron, triangular etc.) are therefore using this approach, see example name in tree-structure left-side of Fig. 1(b). FEM properties (e.g. material, constraints, boundary conditions etc.) instead use additional attributes. Remembering a naming convention is however not reasonable and support tools were therefore created working as CAD operators for renaming and adding attributes to the CAD components, see example user-interface in lower-left side of Fig. 1(b).

Extended CAD-model techniques:
There are several ways in which the information can be associated with the product features it relates to. For CAD-models, PDM/PLM systems can do this very well on a file-level by adding meta-data. Neutral CAD-file formats are another approach, such as STEP, which can do this very well on more detailed levels. This paper focuses on the native CAD-models. Tab. 1 describes several techniques, available in most CAD systems, to extend the information further.

CAD extension technique	*Description*
Additional attributes	User-defined attributes, i.e. additional attributes/properties attached to objects in the CAD-model.
Names	Names are attributes noted separately due to their commonality among all software and its special characteristic of being visible in the tree-structure.
Parameters	Parameters are similar to attributes in that they have an identifier or variable associated with a value, either string or numeral, but is a separate class either associated with a feature, part, or other parameters, or is completely independent.
Annotations	Annotations are a separate class which can be used to link text and symbols to instances of mainly geometric entities.
Bundled features	Bundled features are user-defined collections of functional features, commonly referred to as User-Defined Features (UDF).
Programmed features	Programmed features are created utilising the software application programming interface (API). They are a way for the user to attach code to the model rebuild loop.
Additional geometry	Modelling geometry is the main objective of a CAD system and can of course also be used to represent multidisciplinary information.

Tab. 1: CAD extension techniques.

The extension of CAD-models can be either contained inside the CAD-model which requires the CAD software to have such functionalities or it can be stored outside of the CAD-model which requires standalone software or add-ins to establish a link. Most often they are hybrids, as the method above.

Interview results:
Within the case-companies interviewed almost all the techniques presented above for extending the CAD-models were used, except Annotations (although this was planned for in the future in Company B). See Tab 2 for a full summary, and challenges expressed beside it (see Tab. 1 for information about the different techniques). Companies B and C used Siemens NX and Company A used SolidWorks.

Extension technique	*Company A*	*Company B*	*Company C*
Names	-	+	+
Attributes	-	+	+
Parameters	-	+	+
Annotations	-	-	-
Bundled features	-	+	-
Programmed features	+	-	+
Geometry	+	+	+

Tab. 2: Utilized CAD-model extension techniques at case-companies.

The challenges expressed were:

- Administration of the added multidisciplinary information
- Finding correct information in method descriptions.
- Keeping track of nomenclatures
- Dealing with software updates.
- Modelling robust and flexible CAD-models.
- Managing geometric simplicity and realistic representation as well as mesh quality and size.

Proceedings of CAD'17, Okayama, Japan, August 10-12, 2017, 337-341

Extension techniques and their implications:

Almost all CAD software reviewed had all the extension techniques available, with exception to feature-bundles as well as programmed features in SpaceClaim. The noteworthy differences were with respect to the range that information could be manually inserted to the model objects and to which degree executable information could be modelled and embedded. Tab. 3 shows an overview of some differences in the extension techniques presented in Tab. 1 with respect to additional aid requirements, comprehensiveness, and possibility to model executable information. The thought-process is further described below.

Extension technique	*Requires additional aid*	*Comprehensiveness*	*Can contain executable-information*
Names	No	Low	No
Additional attributes	Depends	Low	No
Parameters	No	Low	To some extent
Annotations	No	Medium	To some extent
Bundled features	No	High	To some extent
Programmed features	Yes	High	Yes
Additional geometry	No	No	No

Tab. 3: CAD extension techniques and their individual properties.

Some of the extension techniques required additional aid in the form of custom applications. These could be either added to the CAD system as add-ins or stay as standalone applications. Creating custom applications for adding the information required in the CAD-models is not particularly difficult but will require maintenance. For instance, in Siemens NX additional attributes could be manually inserted to all geometric entities, from points, curves, surfaces, to bodies, whilst in Solidworks and SpaceClaim they had to be programmatically inserted. Interesting to note also that it is an extensively used technique in both Company B and C who used Siemens NX. Programmed features also need user-interfaces unless the users are supposed to meddle around in the code themselves.

To be comprehendible the semantics, or meaning/purpose, of the added information needs to be understood by a human-being. This is important since many of the extension techniques produce information which is openly editable by others, such as names. The semantics behind additional geometry is not always understood, it usually requires one or several of the other techniques as well. Take for instance the added surface within the bolt in Fig. 1(a) and 1(b), the purpose or meaning of this geometry is not clear unless additional information is attached to it. Attributes and parameters can be comprehensive for humans if the information they convey can be separated into a variable-value relation, e.g. thickness, density, material name etc. Parameters can sometimes be associated with additional notes as well, which increases its comprehensiveness potential. Names can be used as in the example above, where the name "FEM_Revolve_360_Axel" was used to convey the purpose of the added surface in Fig. 1(b). For a human to fully understand the meaning of this information it probably needs to be expanded however, and at some points it becomes too much. Annotations have in contrast to attributes (including names) and parameters the advantage of being visible in the graphical user interface (GUI) and the possibility to model symbols which greatly add to its comprehensiveness potential. Feature bundles and programmed features are the only ones described as highly comprehendible, as seen in Tab. 3, because they utilize custom GUIs. When feature bundles or programmed features are used, a GUI shows the input such as parameters, pointer to geometric objects etc. in a collective, easy to understand, format.

Executable-information in this paper is information which expresses logical reasoning and is understood as well as used by a computer-based interpreter, such as programming code or mathematical expressions. It requires an interpreter and can in contrast to other pieces of information express an action, to alter or add more information and notify to different kinds of events. Parameters can contain executable-information in the form of mathematical equations to describe relations to other parameters. An interpreter in this case understands the formulation of mathematical equations and can produce new information by retrieving the values from other parameters and performing different kinds of mathematical operations. Feature bundles are pieces of information which can contain executable information, such as shape features which produce new information in the form of geometry. Programmed features are built upon programming code, either as macro-scripts built with visual basic for applications or standalone application built with any language supported by the .NET platform or Java utilizing the CAD software's API. Any programming code requires an interpreter and with the ones utilized by programmed features the possibilities for creating new information is vast.

To embed executable-information inside the CAD-model you can for instance embed macro-features in Solidworks, knowledge fusion rules in Siemens NX, or write Knowledge-ware rules in Catia.

Conclusion:

There is a growing trend in manufacturing companies to utilize CAD-models as potential communication hubs to support re-usability and enable multidisciplinary exploration of their products. We have reviewed several techniques to facilitate such communication in several CAD software to identify pros and cons of these techniques. Our answers to the initial questions which guided this work are based upon both interview studies within three major Swedish companies and close examination of four CAD vendors. The answers can be phrased as follows; *(1) How can CAD-models be used as carriers of multidisciplinary information?* We found seven different techniques for adding information to the CAD-models, they were; name, additional attributes, parameters, annotations, bundled features, programmed features, and additional geometry, see Tab. 1. Out of these seven, parameters, bundled features, and programmed features had interpreters allowing them to also model executable-information. *(2) Which of these approaches are most common in industries today?* Within the three companies investigated, two of them used naming conventions applied to both attributes (name and additional ones) as well as parameters. Additional geometry was also used by all of the companies, see Tab. 2. *(3) What differences can be found between them?* It can be concluded that there is a trade-off between comprehension and extendibility. Even if name conventions applied to attributes and parameters may be the most commonly used of the extension techniques, because of its availability, it has drawbacks in terms of comprehension and extendibility. Programmable features on the other hand provides a flexible and extendible way of extending CAD-models, but requires programming skills, time, and could affect systems maintenance.

References:

[1] Ding, L.; Matthews, J.; McMahon, C. A.; Mullineux, G.: An information support approach for machine design & building companies, Concurrent Engineering Research and Applications, 17(2), 2009. https://doi.org/10.1177/1063293X09105325

[2] Ding, L.; Matthews, J.; Mullineux, G.: Capturing constraint evolution: A technique to preserve and handle design knowledge, International Journal of Product Development, 15(1–3), 135–155, 2011. https://doi.org/10.1504/IJPD.2011.043665

[3] Ding, L.; Davies, D.; McMahon, C. A.: The integration of lightweight representation and annotation for collaborative design representation, Research in Engineering Design, 20(3), 185–200, 2009. https://doi.org/10.1007/s00163-009-0077-2

[4] Heikkinen, T.; Johansson, J.; Elgh, F.: Assessment of Simulation-ready CAD Models in a Set-Based Concurrent Engineering context, DESIGN 2016, 1–10, 2016.

[5] Johansson, J.: A Feature and Script Based Integration of CAD and FEA to Support Design of Variant Rich Products, Computer-Aided Design and Applications, 11(5), 552–559, 2014. https://doi.org/10.1080/16864360.2014.902687

[6] Johansson, J.; André, S.; Elgh, F.: Simulation Ready CAD-models as a Means for Knowledge Transfer, ICED 15, 195–205, 2015.

[7] McMahon, C. A.; Davies, D.: The use of annotation in design representation, Design Methods for Practice - 5th International Seminar and Workshop of Engineering Design Integrated Product Development, EDIProD 2006, DS 37, 2006.