COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 4, 558–569
http://dx.doi.org/10.1080/16864360.2015.1131552

A knowledge-based framework for integration of computer aided styling and computer aided engineering

Severin Stadler and Mario Hirz

Graz University of Technology, Austria

ABSTRACT
The collaboration of industrial design and engineering design traditionally represents a potential area of conflict. In view of virtual product development, industrial styling-technology processes provide significant gaps on the way from industrial to engineering design. The present paper introduces a novel knowledge-based engineering framework for an effective and efficient integration of virtual styling and engineering disciplines including management of data quality, data structuring and meta-information of Class-A data. As a key aspect of the research work, Class-A verification configurations for implemented checks were developed and applied to the framework. The paper closes with an exemplary application of the framework on a typical automotive styling integration workflow including a short exemplary description of the developed software prototype.

KEYWORDS
KBE; computer-aided styling; class-A; product data quality

1. Introduction

The appearance of consumer products is one of the main reasons for customers' buying decision. Looking at automotive industry, the styling of a new car still states an important aspect besides costs, reliability, safety and fuel consumption. New mobility concepts and propulsion systems provide more degrees of freedom for new shapes and appearances of future vehicles. Based on these facts, customer visible components earn a particular attention. With regards to the development stage of a product, there is a wide range of influencing factors that have to be considered during intense convergence processes between styling and technology. These so-called styling integration starts from the initial definition of several styling shapes and is finalized after the design freeze of economical producible parts.

Fig. 1 gives a coarse overview of the main disciplines in early automotive development with focus on the exterior styling integration. The styling development is part of initial phases of product development and therefore it is characterized by a dynamic behavior. Based on initial specifications and a so-called styling briefing, a competition between several styling teams is performed, where different prospective vehicle outlines are developed. Step-by-step, those outlines are filtered by both objective and subjective criteria, until a proper styling concept is found. In general, there are different procedural development methods performed simultaneously during styling development. After creation of two dimensional drawings - physically as well as on virtual way - the three dimensional styling development is continued using computer-aided industrial design (CAID) software. This technology is a subset of CAD, but offers more tools for conceptual and aesthetic development instead of those provided by technology-oriented CAD [10]. In the following, several computer-aided systems like CAD, CAID and CAE are combined under the term CAx. Particularly direct geometric modeling of curves and surfaces and the possibility of including particular analysis tools, e.g. continuity and optical reflection assessment, are main reasons for using specific CAID tools. Simultaneously to virtual modeling in CAID, a physical mock-up, a so-called clay model, is built up and modified manually with scrapers. In some projects, the physical model is built up by milling machines or 3D-printer, using already developed 3D-CAID data. These hardware models are optimized by hand, and brought back into virtual environment by use of 3D-scanners, e.g. laser-scanners. Finally, the hardware-based method delivers raw data of vehicle surfaces in form of point clouds.

The quality, content, structure arrangement and format of raw data are often inapplicable for downstream performed process tasks, like engineering-oriented development in CAD or even virtual reality (VR) representation.

CONTACT Severin Stadler severin.stadler@magna.com

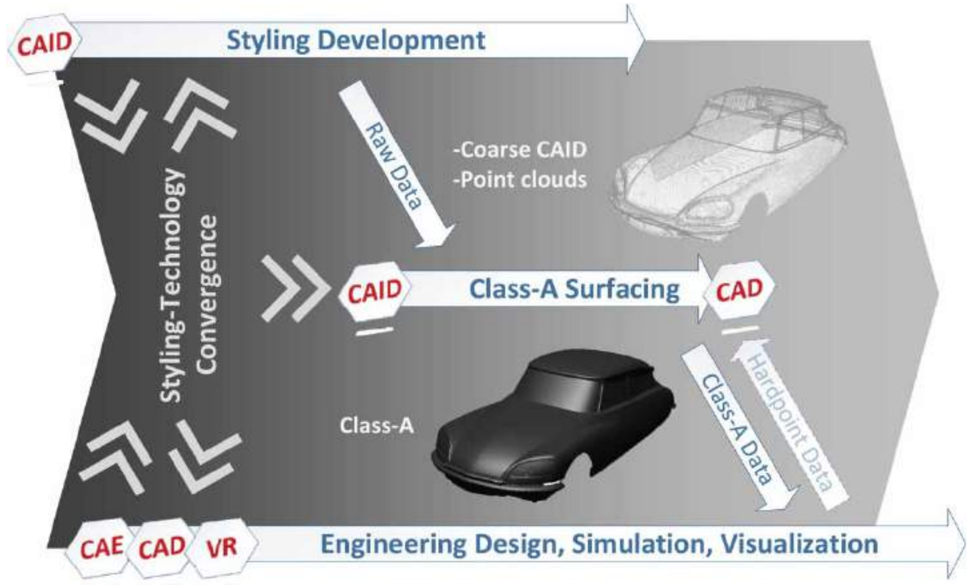

Figure 1. Styling integration process and different virtual environments.

In this way, raw data, which serve as a proposal in view of vehicle styling, have to be built up new to reach harmonious freeform surfaces with a high surface quality in terms of mathematical accuracy and geometrical continuity. In automotive industry and increasingly in consumer goods industry, these high quality freeform surfaces are called Class-A surfaces. Representative for Class-A surfaces are the visible surfaces of automobiles either in exterior or interior. Based on enhanced visual requirements on Class-A surfaces, like light reflection characteristics, specific required mathematical boundaries can be derived. As an example in automotive engineering, Class-A surfaces have to achieve curvature (G2) or even continuity in terms of change-of-curvature (G3) between boundaries of patches or segments. The differences between tangency (G1) and curvature (G2) continuity can be seen in the varying surface acceleration on the left side and in the highlight analysis by using so-called isophotes on the right side of Fig. 2. Isophotes are one possibility of virtual surface interrogation methods.

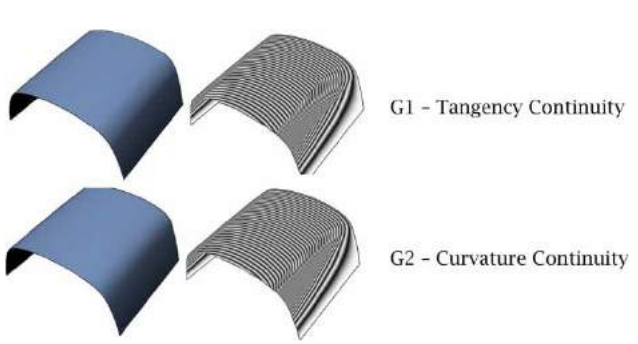

Figure 2. Comparison of surface continuity.

The big advantage of this method is that discontinuity in the surface is displayed in the isophote as a discontinuity magnified by an order of one. In other words: if the surface is G^n continuous, the isophote at that location will only be G^{n-1} continuous, [4]. The definition of continuity of course doesn't mean a good quality of the surface at all, as there are some more factors to be considered regarding the reflection characteristics, e.g. surface waviness due to high degrees and number of segmentations. Amongst other reasons, explicit modeling and refining of Class-A surfaces in CAID software is exclusively defined using Bézier shapes or sometimes simple B-Splines; the application of NURBS would offer a wide range of degrees of freedom, which could result in wavy surfaces.

As mentioned, Class-A surfaces are created within surface modeling and refinement tasks in CAID environments including periodic fairing or smoothing processes in a so-called "Class-A Surfacing" discipline, as seen in Fig. 1. A main challenge in Class-A Surfacing includes the conversion of styling inputs into producible shapes that simultaneously consider several requirements from engineering and production points of view. In this way, minimum and maximum values of curvature radii, draft directions for die cast components as well as possible grains on the surface have to be considered. Class-A data are manually checked for quality in the CAID author system, converted to CAD and subsequently they are released to downstream engineering disciplines like design, simulation or even high performance visualization.

The engineering design for example uses these data as boundary conditional outer surfaces for the creation of technical parts within CAD, e.g. vehicle fender incl. wall thickness, flanges and flared tube ends. Furthermore, there are several simulation disciplines in engineering to do validations according to aerodynamics. It's important to emphasize that downstream disciplines, particularly those from engineering design, are prohibited to do changes on released Class-A data.

In general, Class-A surfacing is mainly driven by styling (aesthetical) and by engineering demands and thus it requires a strict regulation in terms of communication and steering processes for a target-oriented alignment. These intensive and iterative alignment processes in early concept phases between aesthetic requirements of styling and technical boundaries from engineering point of view can be summarized in the term "Styling-Technology-Convergence" (STC). Typically, there are regular STC meetings during the entire development process, where the requirements and responses from styling as well as from engineering are discussed and aligned. The surfacing discipline receives orders to change or create the Class-A data according to the results of the STC alignment meeting, as shown by the arrows in Fig. 1. If there are suggestions from engineering point of view, so called "Hardpoints" CAD-data can be delivered to support the ordered changes within the Class-A surfacing. In this way, the consecutive status of these convergence processes is represented by the corresponding Class-A surface data.

2. Challenges of styling integration

As part of the research work, state of the art styling-technology-convergence (STC) processes were analyzed regarding effectivity and efficiency within virtual development. As can be seen in Fig. 1, there are different CAx system technologies used during the styling integration process. Due to the fact, that Class-A surfacing works with the same virtual technology (CAID) as the virtual styling development, the interoperability between styling and surfacing disciplines provides no serious problems. In particular, Class-A surfacing uses the different types (physical or virtual) of raw data and drafts from styling just as boundary geometry to match and refine the Class-A surfaces; so there is no direct usage of these data. In the same way, responses or proposals of technical engineering (hardpoints) in common CAD format can be easily transferred to CAID to use them as boundaries too. This direction of conversion works well because CAID is also able to represent NURBS under the precondition that CAD part documents are not too big.

Before Class-A data are released to downstream engineering processes, the surfaces are checked in the CAID system according to prescribed requirements, like geometric continuity, gaps or draft angles. In general, CAID systems enable performant check functions like isophote or reflection analyses, self-intersection analysis of surfaces or even gap analyses. However, all these checks have to be done manually in time consuming tasks and there is no possibility within CAID systems to create check profiles to reach consistent and standardized checks for archiving. A considerable fact is, that particularly checks regarding topology like topological surface compounds in the CAID source system cannot deliver the force of expression to ensure an efficient downstream applicability in target CAD environments. The degree of freedom in CAID, which allows an explicitly creation of geometry by definition of the spline order, the direction of u- and v-parameters, the orientation of the surface normal or even the tolerances for connection continuities between surfaces, may lead to profoundly problems after conversion into CAD. A conversion of surface data from CAID to CAD can induce changes in structure arrangement and geometric representation in the target system, especially when using neutral geometry exchange formats like IGES; however, state of the art are direct data converters. As an advantage, these converters are able to detect errors during conversion, if geometry cannot be converted accurately and to provide user response.

Fig. 3 shows an example of a typical Class-A surface after conversion into CAD. The picture on the left side shows discontinuities between patches, whereby the red labeled borders describe gaps or overlaps. Particulars can be seen in the magnified detail on the right side. These errors are often results of erroneous data conversion, especially into system with a higher accuracy. In the course of visual checks in the CAID source system these errors may not be recognized.

Practice in industry shows that, based on those reasons, converted Class-A data from CAID are often not applicable within CAD. Despite of a prohibition of changing Class-A data in downstream disciplines, they have to be prepared manually before they can be considered in engineering-related works, which may result in different derivations of one release Class-A status. In the example of Fig. 3, the creation of essential topological surface compounds like the feature 'Join' in CATIA V5 of not parameterized surfaces states a precondition for several technical-oriented design steps in CAD, e.g. the creation of offsets for sheets in part design.

As already mentioned, the geometrical surface quality is checked manually in CAID environment with a relatively high effort. Besides the visual check of aesthetic shapes using isophote or reflection functions in CAID, there are comprehensive additional process-oriented criteria regarding downstream applicability, which have to

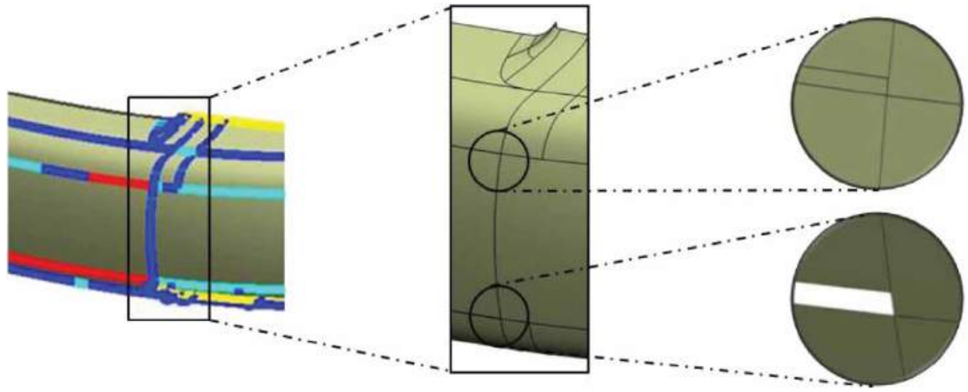

Figure 3. Gaps and overlaps between patches.

be checked in the target CAD system environment before they are released to engineering design. Some of these criteria are consolidated in product data quality guidelines like from SASIG [7]. In industry, there is a lack of knowledge how Class-A data should be checked for. In this way standardized check criteria configurations depending on the data maturity have to be defined. Those checks are not acceptable if they would be performed in time consuming manual steps. Furthermore, documentation and archiving of check results including associated data states a gap to be closed in industry processes.

Besides problems with geometrical and topological data quality, another challenge appears concerning data structuring. In engineering design, CAD-based assemblies of modules and components are state of the art representation techniques on module- and full vehicle-level, enabling complex product structure-oriented processes. In contrast to accurate and deep arranged structures in engineering design, styling development does not consider an adaptable data structuring. One reason is that CAID systems are not able to build up surface assemblies out of surface parts. In addition, surfaces are drawn on the whole for a bigger region before they are separated by gaps to surface sets that cover specific areas. These surface sets allow structure arrangement within a CAID document, but the technology is strongly limited to one or two layers. Thus, one question is how to get a transition of raw styling data to dynamical arranged surface data including a part wise separation of the geometry. This separation of Class-A data to singular grouped CAD elements is furthermore a precondition for the previously described quality checks in CAD. Fig. 4 shows the scope of a typical CAID document in automotive surfacing development, while the engineering design requires separated surfaces in its working tasks. In this way, a disassembling or separation method is required which could be ruled by a Class-A master-structure.

While the definition and storage of technology-related attributes within CAD-documents and data management systems is commonly used in engineering design, this methodology is currently missing during styling integration. This originates from the fact, that CAID systems including their documents are not able to store meta-information. Typical STC processes in industry do not consider the storage of attributes in PDM systems. As there are several impacts on Class-A surfacing, the

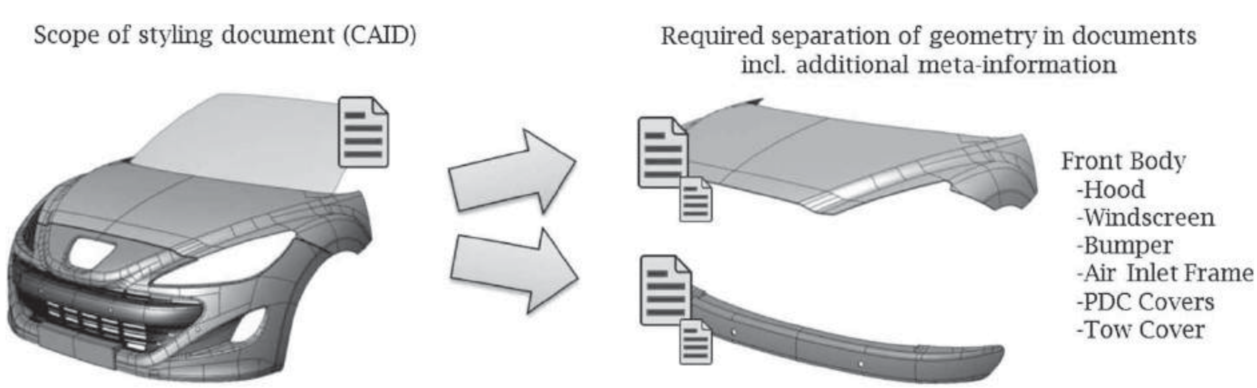

Figure 4. Lack of structure granularity and meta-information.

storage of attributes states an essential demand for effective styling integration.

The stated challenges of today's styling integration processes can be divided into three key aspects: data quality, structured arrangement of data and storage of meta-information. A review of the state of the art shows, that the interaction of styling and technology demands for new methods. In literature, some information about the general processes in this area can be found [2], but there is only marginal information available, which discusses the topic process-oriented in view of the required level of detail, e.g. surface quality and data structuring. On the other side a lot of research is done focusing on data exchange between CAD and CAD [9] or CAD and CAE (computer-aided simulation) [6], [8] systems. Furthermore there are several works in the area of quality assessment of freeform surfaces like [4]. There are also contributions related to new methods in CAD for fairing or surfacing functions to improve the freeform surface quality for styling applications: e.g. [5]. Regarding the aesthetic evaluation of freeform surfaces and contours in car styling, Bluntzer et. al. [1] introduces a knowledge-based CAD approach to automatically identify, extract and interpret characteristic lines of styling. However the process view from styling creation until integration to modern strongly structured PDM-oriented engineering disciplines including fairing processes for Class-A data in not considered sufficiently until now.

3. KBE framework for styling integration

The goal of the present work is to reach an effective and efficient integration of styling and Class-A surfacing disciplines into well regulated engineering processes. Considerable potential for improvement can be detected in the administration, check and transition of Class-A data between surfacing-processes and downstream engineering & visualization disciplines. The main idea includes a knowledge-based engineering (KBE) framework, which provides several methods and procedures to face these specific challenges, in particular in view of data quality, structure management and meta-information enrichment. An overview of KBE including different levels of knowledge integration is discussed in [3].

This framework mainly addresses and supports the Class-A designer due to the fact that Class-A surfacing discipline plays a central role and CAx-interface between styling and engineering. Additionally the framework is adequate for styling-technology (STC) engineers in order to support communication and steering between these heterogeneous disciplines. Fig. 5 gives a schematic overview of the developed framework, which is separated into a problem-oriented KBE system to control and administrate processes and data including automated routines, and a predefined CAx/PDM system environment, which contains the required systems to be controlled.

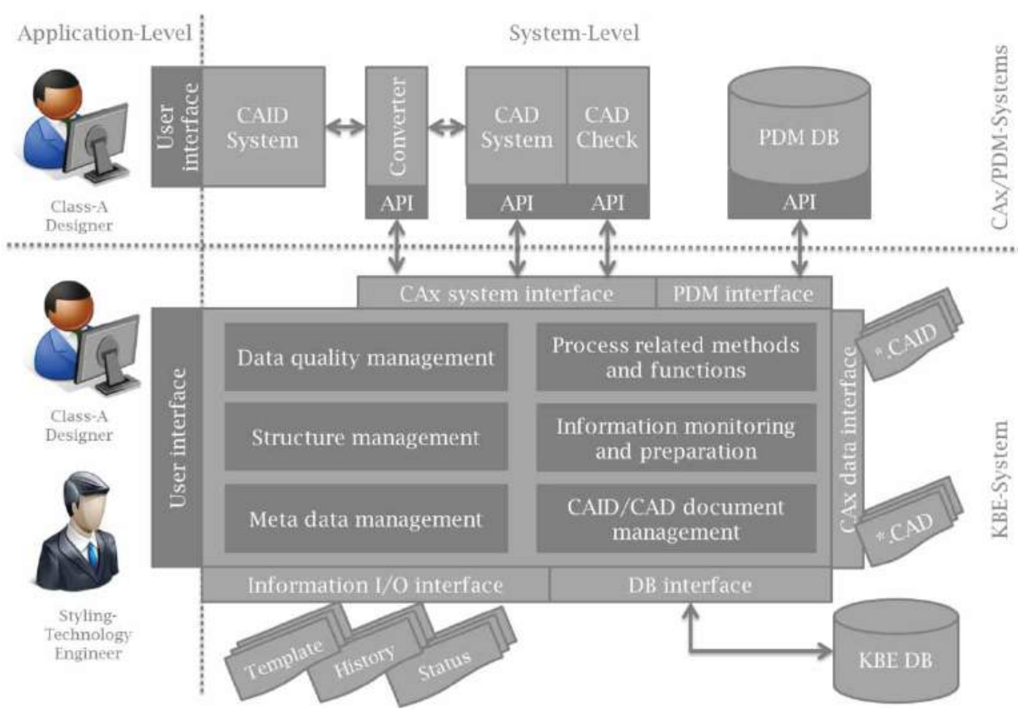

Figure 5. Scheme of KBE framework for integration of CAID into engineering processes.

Furthermore, the framework is divided into a system level in the background and an application level related to the user accessibility. As one key aspect of this framework, the strongly limited application level allows the user to work in a primary CAID system by applying specific expert knowledge. In this way, time-consuming and non-creative work, like data conversion, manual checks and preparation or even expert knowledge in terms of surface check in CAD systems is neither required nor part of the user's work anymore. For example, a Class-A design engineer is then enabled to keep its focus on the creation and modification of Class-A data in CAID, while data conversion, check and preparation tasks including the storage in global engineering PDM databases are performed by the framework system in background. Several systems, as there are the CAID/CAD data converter, the target CAD system including a CAD check system as well as the product data management (PDM) system are accessed bidirectional by a problem-oriented KBE system using the supported application programming interfaces (API) of these systems.

Besides these CAx and PDM system interfaces, the KBE system approach provides further operations to administrate and control processes and data. Process-related methods for example support and automate the time-consuming release procedure of Class-A data conversion to prepared downstream-ready data. Of course there are further functionalities implemented for efficient process integration. An information monitoring and preparation unit enables the engineer, Class-A designer or STC-engineer, to monitor the actual status of Class-A data on demand. In view of data quality, the status of the previous Class-A releases can be retrieved efficiently in relation to selected predefined milestones. These milestones include quality criteria depending on the data maturity, which is changing during the development process. So, the system also contains functionalities for project-management to view actual development status with required meta-information. Several geometry-related results, meta-data and additional information as well as documents are stored in a coupled relational KBE database (e.g. SQL) which is controlled by a specific database interface. As a key element, this KBE database contains company specific expert knowledge like for example thresholds for Class-A verification as well as process and project related settings, e.g. for disassembling and preparation of data. So the KBE database contains adaptable knowledge in terms of settings to meet varying development requirements. This fact leads to a high practicality of the framework for application in industry.

The KBE system contains additional interfaces to handle external CAID and CAD data, to connect application programming interfaces (API) of CAD and PDM systems, as well as an interface for export and import of Class-A status, templates in terms of initial structure arrangements for different car types or even the history documentation of specific CAID and CAD processes. While APIs have become standard for many CAD and PDM systems to enable high professional knowledge-based engineering methods, today's CAID software do not yet support this potential opportunity for process integration. Nevertheless, the presented KBE framework is able to process the CAID data efficiently as they are on the way to CAD environment as described later in this paper.

As core elements, the KBE architecture contains methods to manage data according to the key aspects, as mentioned in Section 2. First, there is the quality management of Class-A data in terms of surface quality and preparation for data applicability in downstream engineering disciplines. Second, the management of structures during the transition from styling is provided to enable compatibility to downstream engineering master-structures and their accurate degree of resolution, which simultaneously is required for the quality checks on part and feature level. Third, the enrichment, consistent definition and maintenance of meta-information of Class-A surfaces are also parts of the KBE architecture.

4. Application of the framework in process

In Section 3, the developed framework is described from a static point of view to show the architecture including the system and application level. As mentioned, this framework is mainly addressed to Class-A designers and STC-engineers, but is not limited to them. For example, engineers of technical visualization can access the KBE user interface to add meta-information in view of textures or material description. The KBE system is implemented into a KBE software tool including a coupled KBE database. Subsequent sections show the application of this tool in automotive styling-integration processes, starting with the creation of an initial master-structure within the KBE tool for Class-A surface design until checked and downstream-ready Class-A data in CAD format for upload to the PDM system, as illustrated in Fig. 6. The main part of this illustrated release process is performed automatically by the control of the developed KBE software tool and thus it leads to tremendous savings in time and effort by simultaneously newly introduced and enhanced tasks like Class-A checks, preparation or documentation. As the introduced framework management leans on three main pillars, the following description is arranged adequately in relation to Fig. 6.

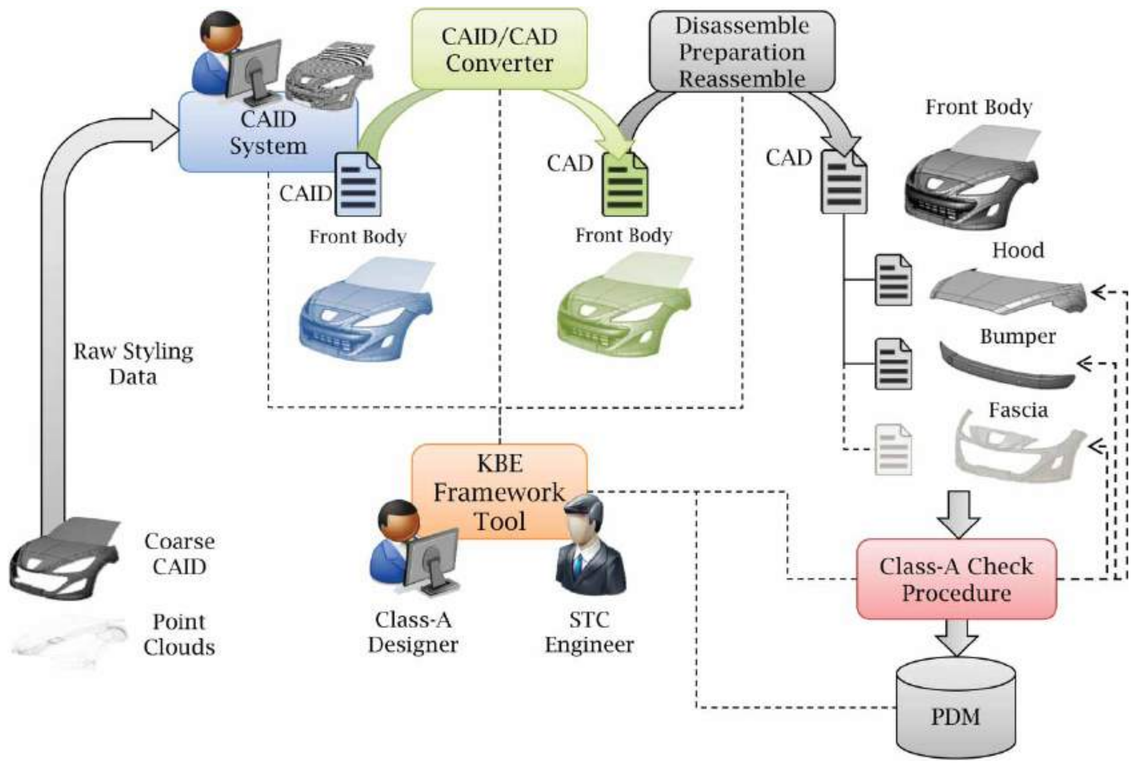

Figure 6. Application of KBE framework in styling release process.

4.1. Structure management

The scope of one CAID document contains very big areas, like a whole front end of a car exterior as illustrated in Fig. 6 (blue). The conversion into a CAD part document results in extensive CAD data, which may lead to structure and granularity problems due to the fact, that all data are within one CAD part document, as seen in Fig. 6 (green). Direct converters are able to convert also simple structure information to CAD, but this would mean that a master-structure has to be defined in CAID environment, which is difficult to hold consistent and flexible throughout the integration process. The idea of the framework's structure management is to define and administrate a master-structure for Class-A data in the KBE system, see Fig. 6 (orange). After an automated conversion of the CAID document into CAD controlled by the KBE system, the structure of the resulting CAD document is checked automatically in comparison to the master-structure regarding structuring and nomenclature of the elements. After this structure check is performed successfully, the master-structure enables a disassembling of the extensive CAD document, containing all sub-elements in groups, into single CAD models, see Fig. 6 (gray). In this way, the Class-A data are separated into different documents. The KBE system allows the definition of the master-structure in a more detailed granularity as on part level (sub-parts), which may be advantageous for downstream disciplines or for check operations of specific elements.

In a next step, a standardized sub-structure within the CAD part document is created, as seen in Fig. 7. This sub-structure is divided into an additional level containing these sub-parts by use of group features of the CAD system. A sub-part is defined as a couple of surfaces, which can be connected together in order to create a surface compound. A sub-part for example could be the sign on a button of the climate control, or the button itself; the example of Fig. 7 contains the Beam and the Tow Cover as sub-parts of the Bumper part. This level is required by several downstream disciplines; sub-parts are not declared as separated entities or structure nodes in the detailed engineering structure. In this way, meta-information can be assigned in a very detailed level; also checks regarding surface compounds are able to be performed. Within these sub-parts a standardized sub-structure, as seen in Fig. 7 (right), is set up containing non-parametric Class-A data which are created after conversion, possible automatically created topological compounds, created offset surfaces based on the compounds as well as additional pre-defined geometric information, which could be advantageous for downstream processes, e.g. derived basic surfaces without details to support meshing process for simulation. The present KBE tool is able to find the erroneous converted geometry

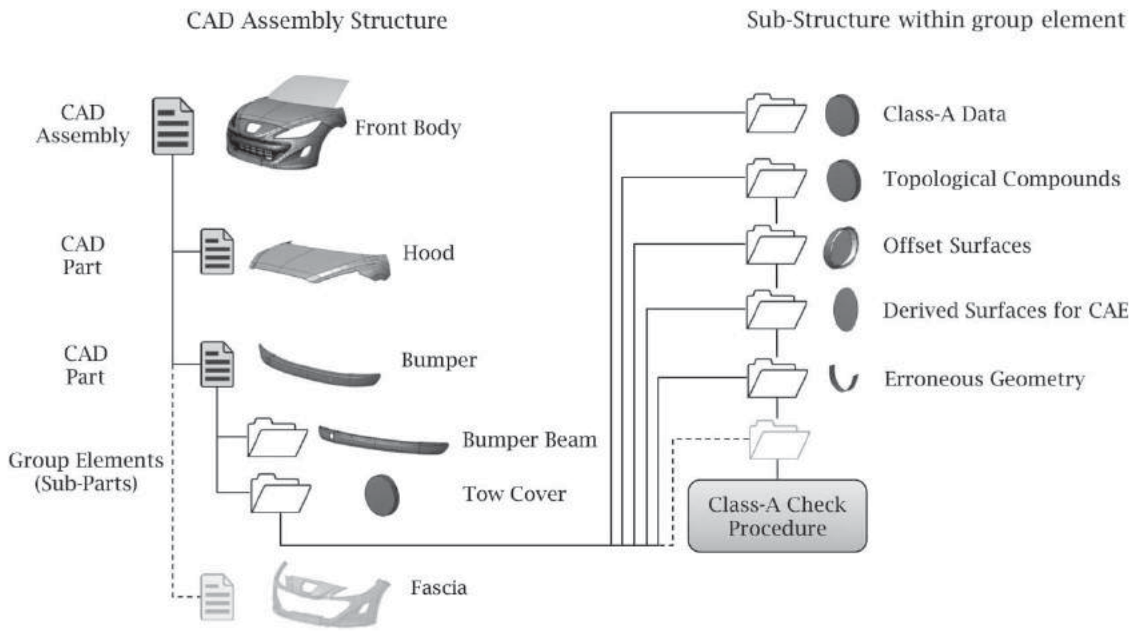

Figure 7. Sub-parts including prepared sub-structure for check procedure.

from the conversion protocol and to store them into the sub-structure's group Erroneous Geometry. Finally the tool assembles the part-documents to a CAD assembly according to the master-structure and releases the data incl. meta-information to a PDM system. Several tasks of the procedure are performed automatically by the KBE system; the Class-A designer is able to keep his focus on the modeling and refinement work within the CAID system and on the administration of the structure.

Besides supporting Class-A designers with the KBE system, STC-engineers have the possibility to manage Class-A structure and engineering structure efficiently. As a key benefit of this method, the elements of the Class-A structure can be connected to the further technical pendants of the engineering structure, which enables an adoption of the Class-A structure to the engineering structure in traditional PDM system. This assignment is enabled by the coupled relational database of the KBE tool and considerably supports the styling-technology convergence processes.

4.2. Data quality management

The data quality management of the framework focuses on the quality of Class-A surface data. This includes checks in view of the described traditional tasks, like geometric continuity, but also the applicability in downstream disciplines. In course of the present research work, the data exchange between CAID and CAD systems was analyzed in detail. As a result, CAD technology reflects the highest sensitivity on CAID data quality. This sensitivity mainly derives from the characteristics of geometry description in CAD systems, which are based on topological relations to create compounds of geometric elements. Small gaps in surfaces, overlaps or self-intersections of geometry can appear when working in CAID systems. After conversion into CAD, these failures lead to incompatible surface elements, which furthermore conduct problems within CAD environment, e.g. merging errors, topological errors. Creating compounds (e.g. surface of engine hood) is necessary in engineering-related design to apply technical features, e.g. wall thicknesses of a sheet metal. In this point of view it's a precondition to check Class-A data in the target CAD system regarding objective quality and downstream applicability, while perceived and subjective quality as well as aesthetic surface characteristics still have to be checked visually prior in CAID environments supported by use of virtual reality power walls.

Before checking Class-A data, the quality requirements have to be defined. These requirements are different between branches, companies, products or even within a product. Additionally the geometric quality of customer visible product data is relatively low in the beginning of product development (Pre Class-A) and increases until the desired quality (Final Class-A) is reached. In automotive engineering there are no fix defined standards that describe the quality of Class-A surfaces. An analysis of different Class-A data in automotive area gives some approximate values as Tab. 1 shows. Due to the fact, that in initial Class-A creation, “Pre Class-A”, the high quality of final Class-A is practically

Table 1. Short extract of common Class-A criteria in automotive engineering.

Criteria	Pre Class-A	Final Class-A
G0 Continuity	< 0.02mm	< 0.01mm
G1 Continuity Exterior	$< 0.5°$	$< 0.05°$
G1 Continuity Interior	$< 0.5°$	$< 0.1°$
G2 Continuity	-	$< 10\%$
G3 Continuity	-	Optional
Minimum Radius	0.1mm	0.1mm
Surface Normal Orientation	Consistent	Consistent
Waviness	Without	Without

not possible to reach, there are coarser requirements on the quality.

For example, a G0-Continuity of less than 0.02 mm is required in the beginning of development, while in a final stadium of development gaps that are larger than 0.01 mm are prohibited. The check of gaps in this range assumes a more accurate system tolerance, which typically is about 0.001 mm considering actual CAD systems. Due to the tight system tolerance of today's CAD systems, the checked level of accuracy can be kept constantly along the whole process from styling to engineering tasks. Besides G0- and G1-continuity, G2-continuity is checked for Class-A quality in automotive engineering and must be less than 10%. This means, that the quotient between the doubled difference of the two radii and the sum of the two radii at one position must be smaller than 10%. For example, the criteria waviness limits the amount of inflexion points along predefined isoparameter curves or within surface segments. The illustrated quality requirements do not deliver a statement whether the Class-A surface is applicable in downstream processes. For example surface patches with a gap of 0.001 mm can fulfill the Class-A requirements but they still can have intersections, which can lead to topological compound problems even when the merging tolerance of the topology is 0.1 mm. Especially the G0-Continuity is one significant parameter regarding downstream applicability and thus it is already strongly restricted for Pre Class-A. Besides the parameters in Tab. 1, there are more, often company specific requirements, like the prohibition of convex or concave patches or restrictions of the size different of neighboring patches. Besides the collected Class-A requirements from practice and the problems in downstream processes, additional adequate criteria can be elaborated out of product data quality guidelines, e.g. SASIG [7]. Following list gives an extract of criteria, which are checked in the integrated check procedure of the framework.

- G0-, G1-, G2- and G3-Continuity
- Closed surface/curve boundary
- Identical surface/curve features
- Inconsistent orientation of face or surface to surface compound
- Mini-element of curve or surface
- Narrow surfaces/curves
- Sharp edges in surfaces/curves
- Self-intersection between geometry
- etc.

After the checks of these criteria are performed, additional checks in relation to applicability in the target system have to be done. Those automatically check routines contain the following tasks:

- Creation of a surface compound without consideration of continuity
- Creation of a surface compound under consideration of continuity
- Creation of offset surfaces (wall thickness)

During the check of those three criteria, a specific hierarchy has to be considered: a criterion is only checked if the previous one was checked positively. The results are stored in the group elements of the sub-structure as seen in Fig. 7 (right). Several criteria for Class-A have been tested and evaluated according to their significance and practical application in industry processes, including different degrees of data maturity (Pre Class-A, Final Class-A). Depending on the criterion, selective professional commercial CAD check tools or internal routines using the API of a CAD system are applied in the present approach.

With regards to the realized KBE software prototype, three maturity dependent check profiles were predefined which can be adapted by the KBE project administrator and which are stored in the coupled KBE database of the tool. The user is able to create milestones in the beginning of the project which can be assigned afterwards with a predefined check criteria configuration. If the Class-A designer starts a surface release procedure, elements in the master-structure and a certain milestone have to be selected. After starting the release function of the KBE tool, the converted and disassembled Class-A data are checked according to the criteria of the milestone. After the sub-structure of Fig. 7 was created, first the objective check criteria are performed, before the CAD internal checks regarding creation of topological compounds and offset test of surfaces continue. Several results of the checks including the checked CAD data are archived in the KBE database and are clearly represented to the user to get an overview of entire Class-A data quality status including the option to look for detailed check results of the geometry. The release functionality is described in more detail in section 4.4.

4.3. Meta-data management

As described above, the capabilities of structure and data quality management of the presented KBE framework already enable an efficient integration of styling into engineering disciplines. In addition, an introduced meta-data management within the KBE system supports the assignment of any information to a specific entity, of the defined master-structure - on part level or even below on sub-part level. This approach provides an important step forward due to the fact, that today's CAID data do not support user-specific attributes with the exception of material definition. The users, Class-A designer, STC-engineer or even users from downstream disciplines, like visualization or engineering simulation, are able to assign meta-information by the use of the KBE system. Different rights of access of the users are set up and considered when working with the KBE user interface. In this way, attributes can be assigned to the finally created Class-A CAD parts also without the necessity of sharing the owner rights of a part, which is strictly defined in collaborative working methodology using PDM systems. Meta-information, like product variants, definition of mirror-parts, carry-over parts, engineering material, shading information, work effort for cost-estimation, screenshots of parts and sub-parts for documentation, etc. are part of the styling integration framework. Example: When the checked CAD parts are reassembled afterwards, simultaneously a renaming of the CAD documents for consistent nomenclature can be executed under consideration of predefined rules using the meta-information. Thus, CAD data, which are created and uploaded to the PDM database, are able to include meta-information in addition to the CAD attributes.

4.4. KBE software prototype

The presented KBE framework is implemented as Windows form application software which is installed and executed at the clients' workstations. Depending on the category of engineer or user (STC engineer, Surface designer, etc.), the modular graphical user interface (form) automatically adapts itself to offer the required functions and access rights to the user. The KBE software is programmed in object oriented Visual Basic .Net language and is connected to a server-based relational database (KBE database) that actually contains about 30 tables to handle the information and especially their relations. As database of the developed prototype, Microsoft's Access SQL-database is used with stored SQL procedures. This enables a quick change of the database system, exemplary in case of an increased number of simultaneous client accesses to the database and to be flexible according to different IT infrastructures. As illustrated in Fig. 5, the KBE software addresses the application programming interfaces (APIs) of the CAD system, the CAID/CAD-converter, the CAD checker software as well as the interface of the master product data management system. The present software prototype is programmed to access CATIA V5 as CAD system, ICEM Surf as CAID system, a CATIA V5 integrated direct converter and a professional check software which is accessed by use of the CATIA API. As PDM system the software Teamcenter is connected by use of available import and export interfaces. The software prototype contains some workarounds due to limited access of the APIs of today's CAID and PDM systems. In particular CAID systems do not support knowledge-based design features und they are furthermore not equipped sufficiently with an automation interface. The exchange with PDM systems is actually realized using specific XML-schemata but in contrast to CAID system they generally offer programming interfaces. With regards to the applicability of the software prototype these circumstances can be seen as minor limitations.

Due to its ability of addressing the APIs of different CAx systems, the software prototype reaches an acceptable applicability. Especially the styling development is often performed in a competition of different external styling teams. With regards to this situation the software prototype shows its practicality because it delivers a required structure arrangement for the styling development which has to be followed. In this way the interface between styling and surfacing disciplines is suitable described. The software prototype's own connected KBE database enables a flexible handling and control of the styling and surfacing data and meta-data. Considering the requirements of downstream processes, the software enables problem-oriented configurations. For example, the preparation functionalities of the release procedure can be configured to perform an automated extract of basic Class-A surfaces without detailed flanges. This knowledge-based defeatured CAD data furthermore would support an efficient meshing process exemplary for aerodynamics simulations and therefore it would lead to short simulation loops in early development of the outer shape.

The developed software prototype includes a couple of individual functionalities like the conversion of data, the export of information out of the database or even the import of templates for predefined vehicle styling structures. The system also enables the definition of different check profiles for the check of Class-A data. Moreover, these check profiles can be associated with milestones of the actual project. In this way, the actual release status of Class-A data can be tracked relating to the projects

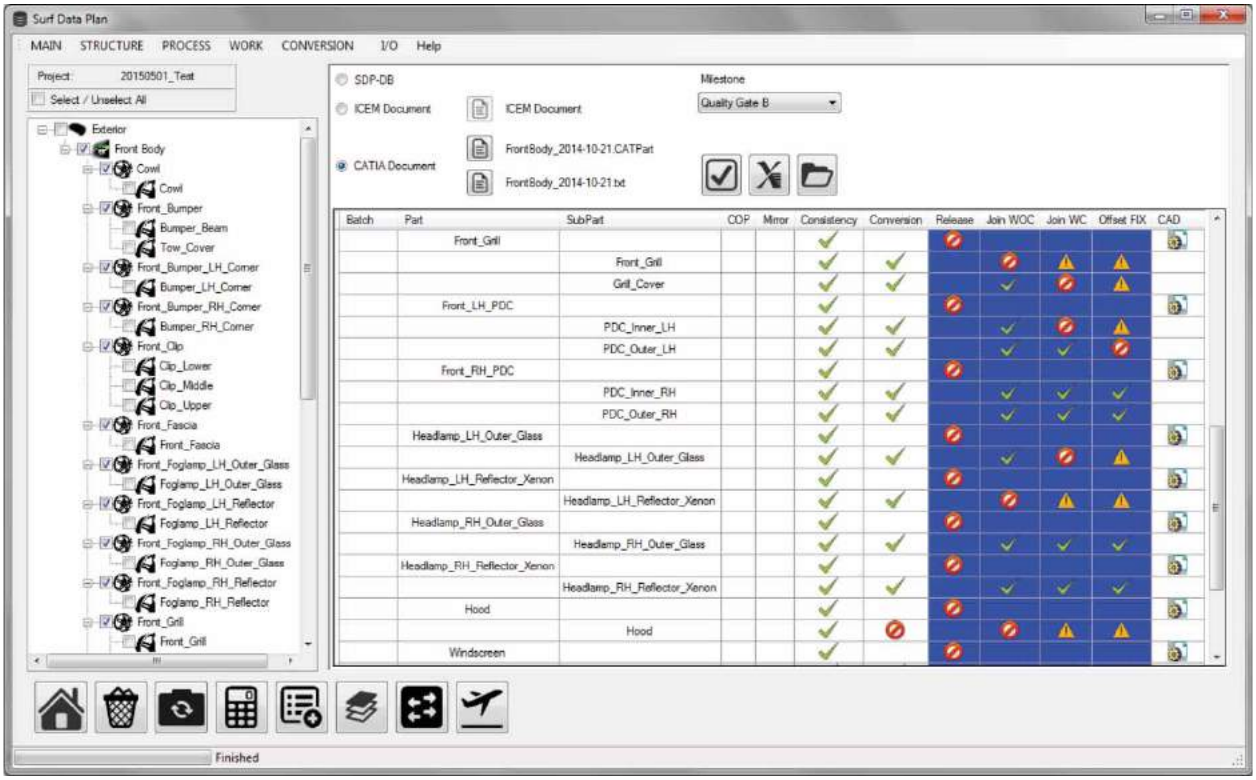

Figure 8. GUI of software prototype exemplarily illustrating the results of the release procedure.

timeline. The prototype not only offers these individual functionalities that enable a flexible usage in practical industry processes. It also supports the users with assembled functions exemplary to perform a fully automated Class-A release procedure as shown in Fig. 6.

Fig. 8 shows the user interface of the software prototype directly after finishing a release procedure. To perform this release procedure, first of all the user has to select the region, batches, parts of sub-parts of the styling surface master-structure in the left panel of the user interface. In the next step, the user has to select the documents if they are not yet stored in the KBE database. Then the user selects a predefined milestone that defines the reference for the release procedure. After the user has started the procedure, the system performs the release procedure fully automated. As a result, the user gets an overview of the release checks with regards to the selected entities of the structure. This overview allows the user to get detailed information like Class-A check protocols by clicking on the adequate symbols in the table of Fig. 8. In addition, several results, and protocols and of course the correlating CAD data are archived in the KBE database. Finally, the user is able to again select specific elements for upload to the PDM system and a final release for usage in downstream processes.

This release procedure commonly is characterized by several repetitive and non-creative tasks that commonly take several days under consideration of the enhanced check and preparation steps. The application of the software prototype in industry projects enables a tremendous time reduction. For example the presented release process of a car's front exterior like in the example of Fig. 7, is performed fully automated within five minutes including documentation and archiving in the database.

5. Conclusion

The paper points out the challenges of styling-technology integration processes in view of the involvement of different computer-aided styling, design and engineering systems. The presented KBE framework is able to face these challenges of styling integration into engineering processes and therefore it manages the interaction between surfacing and engineering disciplines. Beyond that, the framework enables the surfacing disciplines to introduce structure arrangement and designation of data for styling development without limiting their creative and aesthetic work. Actually, a prototype software solution of the framework is applied in automotive full vehicle body development projects, where it first of all has led to transparent and documented representation of Class-A data. A continuous view on the actual status of maturity and quality regarding specific development milestones states a key issue of the solution.

Integrated automation procedures, like an automated Class-A release procedure, which provides data conversion, disassembling, checking, documentation as well as storage in PDM database, lead to a considerable time-reduction. In this way, the framework supports styling-engineering change-management and shortens the duration of technical validation loops of styling data. The KBE solution is able to check the quality of data, but reparation of Class-A data or the evaluation of harmonious surface reflection has still be done interactively or physically by the responsible designers in the source CAID system. Furthermore, the development of Class-A verification methods and criteria has shown, that subjective and perceived quality issues, like the check of freeform surface regarding harmonious or adequately accelerating, has still to be done in CAID or VR environments optionally accompanied by milled prototypes. In this way, the presented approach is able to support the styling-technology convergence processes efficiently by handling non-creative and repetitive tasks, but of course the creative tasks of vehicle design remain in the charge of stylists and engineers.

ORCID

Severin Stadler http://orcid.org/0000-0001-9867-0552
Mario Hirz http://orcid.org/0000-0002-4502-4255

References

[1] Bluntzer, J.; Ostrosi, E.; Sagot, J.: Car styling: a CAD approach to identify, extract and interpret characteristic lines, Procedia CIRP, 21, 2014, 258–263. http://dx.doi.org/10.1016/j.procir.2014.03.174

[2] Bonitz, P.: Freiformflächen in der rechnergestützten Karosseriekonstruktion und im Industriedesign, Springer, Heidelberg, 2009.

[3] Hirz, M.; Dietrich, W.; Gfrerrer, A.; Lang, J.: Integrated Computer-Aided Design in Automotive Development, Springer, Berlin Heidelberg, 2013.

[4] Lennings, A. F.; Peters, J. C.; Vergeest, J. S. M.: An efficient integration of algorithms to evaluate the quality of freeform surfaces, Computers & Graphics, 19(6), 1995, 861–872. http://dx.doi.org/10.1016/0097-8493(95)00070-4

[5] Mullineux, G.: Improvement of free-form surfaces for product styling applications, Computer-Aided Design, 31(12), 2002, 871–880. ttps://dx.doi.org/10.1016/S0010-4485(01)00143-9

[6] Prenner, M.; Stadler, S.; Hirz, M.; Mayr, J.: Measures for the optimization of the CAS/CAD/CFD-process in conceptual vehicle aerodynamics development, NAFEMS: Concept Design Driven by Simulation, Wiesbaden, Germany, 2013.

[7] SASIG – Product Data Quality Guidelines for the Automotive Industry, Strategic Automotive Product Data Standards Industry Group, http://www.sasig.com.

[8] Tierney, M.; Nolan, D.; Robinson, T.; Armstrong, C.: Managing equivalent representations of design and analysis models, Computer-Aided Design and Applications, 11(2), 193–205. http://dx.doi.org/10.1080/16864360.2014.846091

[9] Troll, A.: CAx-Datenaustausch mit neutralen Datenformaten: Prozessgetriebene Konzeption eines Assistenzsystems für die Produktentwicklung, Ph.D. Thesis, Universität Bayreuth, Deutschland, 2011.

[10] Westin, S.: Computer-aided industrial design, ACM SIGGRAPH Computer Graphics, 32(1), 1998, 49–52. http://dx.doi.org/10.1145/279389.279457