Preserving Design Intent in Feature-Based Parametric CAD Data Exchange

Jullius Cho, Detlef Gerhard

Preserving Design Intent in Feature-Based Parametric CAD Data Exchange.

Jullius Cho, Detlef Gerhard

`{jullius.cho, detlef.gerhard}@tuwien.ac.at`

Abstract. The proliferation of 3D technology on many different platforms has made CAD data exchange vital to industrial engineering innovation. The need to share and integrate CAD data, information, and knowledge amongst systems involved in product development has become an important requirement. This requirement is largely determined by a proper communication of design intent, which is usually expressed implicitly within CAD models or tacitly as experiential knowledge from domain experts. However, solutions for CAD data exchange are mostly restricted to the exchange of pure static shape information, restricting their applicability in many downstream processes. This paper suggests methods and strategies for preserving design intent in CAD data exchange. A test case is undertaken to demonstrate the idea and as evidenced in the results, the method proposed helps in capturing and preserving design intent.

Keywords: Engineering Knowledge, Design Intent, Data Exchange, Geometric Parameters.

1   Introduction

With current trends of automation as well as the proliferation of 3D technology on many different platforms, the need to integrate and share CAD data, information and knowledge among systems involved in product development becomes an important requirement. Over time, computer-aided geometric design systems have evolved from modeling 3D solid models containing geometric and dimensioning information to models, which are embodiments of rich engineering data[1]. This data can be analyzed to deduce information and knowledge that is valuable across the entire life cycle of a product. This knowledge includes the degree of flexibility and the ease of alteration of a CAD model which strongly depend on the modeling methodology[2]. Further, the principles of geometric and solid modeling, feature-based parametric and variational modelling form the foundation of computer-aided design. In general, these support the creation, exchange, visualization, animation, interrogation, and annotation of digital models of physical objects [3], [4]. For greater productivity, easier maintenance, scalability, tractability, and efficiency, CAD model reusability is inevitable and therefore, a

dire industrial need. This need is largely determined by a proper communication of design intent[5], [6], which is usually expressed implicitly within CAD models[7] or tacitly as experiential knowledge from domain experts. However, the inherent lack of consideration of design intent and associated parameters, constraints, rules etc. in existing (standard) CAD data exchange systems greatly restricts their applicability in many downstream processes. There is therefore, a need to propagate design intent present in CAD models and in the minds of domain experts to a structured and reusable format.

This paper focuses on the preservation of design intent in feature-based CAD data exchange. Using automatic knowledge acquisition techniques coupled with a structured elicitation of experiential knowledge from experts[8], [9], design intent is captured in geometric parameters, constraints, rules. These are controlled by an inference mechanism based on functional requirements and experiential knowledge. This preservation mechanism is an integral part of a broader framework for the modeling of a high-level neutral representation of engineering knowledge for reuse in feature-based parametric modeling systems described in [8], [9]. A test case is presented to demonstrate the idea and as evidenced in the results, the methods proposed are effective for preserving design intent in CAD data translation and transformation.

2 Background

To set up the scene for this paper, we begin with a brief overview of a scenario where design intent is actively mined and used. One of our industrial partners produces high quality hinges and drawer pull-out systems for kitchens cabinet drawers and shutters. The company works very closely with carpenters, suppliers, and solution providers who are mostly software vendors. This symbiotic association is to provide each associate with the required data, information, and knowledge to be morphed into resources, products, and services to satisfy their individual and collective needs. One pertinent objective of the company is to provide these associates with the right engineering data at the right time to fulfill specific requirements. This multifaceted data therefore, has to be consistent, accommodating updates across all systems that use this data, and across all associates involved. The data requirements are generally broad, including data for visualization, product configuration, analysis, process planning, etc. Because these associates inevitably use different CAD, analysis, visualization and other downstream authoring tools, the data provided has to be translated and/or transformed to meet their needs. To maintain a single source of data, the data has to be presented in a generic way, independent of any underlying authoring system[3]. For the data to be complete and useful, it should be multifaceted, encompassing the static geometry as well as the parameters, rules, constraints and inference mechanism for data manipulation. This inference mechanism is what is referred to as the design intent as it enforces the intention(s) of the designer. With CAD systems and technologies abound, how can the design intent be preserved when translating and transforming CAD data?

The main approach for automatically extracting data in CAD models is explained in [8]. The approach uses applications programming interfaces (APIs) to interrogate CAD

systems to gain access to CAD model data. In addition, the knowledge acquisition and formalization assistant (KAFA) enables the elicitation of experiential knowledge for integration into a "generic" XML CAD representation named XmlCad. While the APIs of CAD systems offer access to the internals of their systems, there still exist some problems. In their work "Data exchange of parametric CAD models using ISO 10303-108"[10], J. Kim, M. J. Pratt, R. Iyer, and R. Sriram, outline the main problems observed in developing translators, which read and write information through the APIs of CAD systems. These include semantic differences, implicitly created information, design rationale, granularity of processing, numerical accuracy, and other implementation issues.

STEP has been established in the CAD / CAM environment for several years as a stable, neutral interface for data exchange including information of the entire life cycle of a product model [11]–[13]. STEP's biggest limitation however, is that, implementers do not implement the handling of features, constraints, or construction history, which are important in a commercial CAD system. After a data exchange, there is largely only information of the model that is presented with the description of B-Rep and contains no more engineering design intentions in the receiving CAD system[10], [13]. The editable representation (Erep) project proposed by Hoffmann and Juan [3] defines a neutral file format that specifies modeling features, constraints, and assembly models thereby providing a means of transmitting the indent of a design. The company Proficiency provides interoperability software for CAD/CAM/CAE systems based on The Universal Product Representation (UPR). The UPR was proposed by [14], exchanges parametric information to support the union of data types supported by commercial CAD systems. Another important aspect related to the transfer of design intent is that oftentimes the lack of tools for the visualization and analysis of relationships between features in CAD models presents an important barrier for understanding design intent [6]. Based on the researches described in [9] and [8], we describe the design intent preservation process by first of all, examining the basic structure of a CAD model.

3   CAD Structure

Parametric or variational modeling systems enable the designer to enforce design intent in a geometric model, which can be either a drawing, part or an assembly. Our approach to design intent preservation in CAD data exchange involves CAD parameters, constraints, and rules of logic, which define and shape solid models. To communicate our approach effectively, a brief discourse on the static CAD structure to be made dynamic is necessary. This includes assembly and/or part representation including topological and geometrical data. For the discussion in this paper however, it suffices to describe a simplified structure of a CAD model, paving the way to describing the preservation of the design intent.

A CAD model is either an assembly, part or drawing. In this write-up however, we will not talk about drawings and will assume that CAD models are assemblies. Using a top-bottom approach describing the level of granularity, a basic CAD model comprises an assembly, parts, features, and entities. The assembly is an amalgam of parts bound

together by set of constraints. Parts consists of basic features such as sketches, fillets, chamfers, holes, etc. which in turn are built from basic entities. The basic entities include a point, a line segment, an arc, a parabola, an ellipse, and a spline, etc. The mathematical definitions (see **Table 1** and **Table 2**) and management of these units to preserve the shape and function of a modeled physical object constitutes design intent. In addition to the description, the composition of each component from its basic low-level entities to high-level representations are structured according to **Fig. 1**. Each entity has a unique identification and is classified according to its type, relating it to other entities. Furthermore, each entity has a geometric description as well as a clear definition of the parameters and relationships that rule coexistence in an assembly. This decomposition is used extensively in the MultiCAD API as shown in **Fig. 2** and **Fig. 3**.
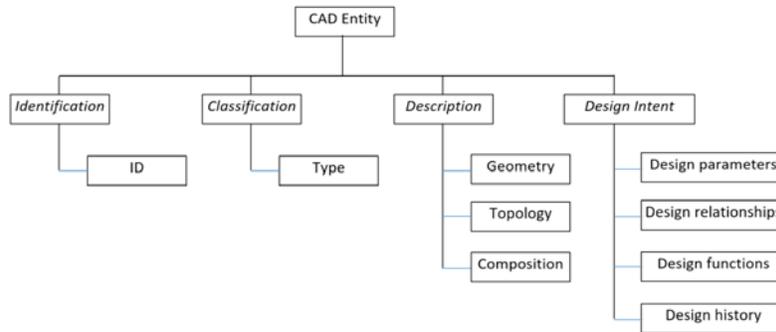
Fig. 1. CAD entity decomposition

3.1 Basic CAD topological objects

Feature-based design is based on functional elements or features having particular significance in the design [3]. This view can be extended to include basic entities, which can be used to drive the defined features. **Table 1** shows possible mathematical representations of some basic CAD entities that are the main enablers of design intent. These basic topological objects can be represented mathematically either in Cartesian (rectangular) or parametric forms. These definitions lend themselves particularly useful for the definition of parameters, which can be changed in an authoring system to model the shape, relationships, and other properties of these entities. For example, a circle in a particular plane can be manipulated when the center point *(h,k)* and radius *(r)* are known. Also, an ellipse can be manipulated if the center *(h,k)* as well as the lengths of the major *(a)* and minor *(b)* axes are known.

Table 1. Cartesian and parametric representation of basic CAD topological objects

Topological object	Cartesian Representation	Parametric representation
Arc	$(x-h)^2 + (y-k)^2 = r^2$	$x = h + r * cos(t)$ $y = k + r * sin(t)$
Circle	$(x-h)^2 + (y-k)^2 = r^2$	$x = h + r * cos(t)$ $y = k + r * sin(t)$
Ellipse	$\dfrac{(x-h)^2}{a^2} + \dfrac{(y-k)}{b^2} = 1$	$x = h + a * cos(t)$ $y = k + b * sin(t)$
Line Segment	$\dfrac{x - x_0}{a} = \dfrac{y - y_0}{b} = \dfrac{z - z_0}{c}$	$x = a + \mu * r_x$ $y = b + \mu * r_y$ $z = c + \mu * r_z$
Parabola	$(x-h)^2 = 4p\,(y-k)$ $(y-k)^2 = 4p\,(x-h)$	$x = 2at$ $y = -at^2$ $x = h + at^2$ $y = k + 2at$

3.2 Basic CAD Features

CAD features form the basis for feature-based modeling. As mentioned in Section 3.1, CAD features can be viewed as functional elements having particular significance in a design. For example, the basic CAD features shown in **Table 2** correspond to functional elements, which from the feature-based modeling view can be represented conveniently by entities and parameters.

Table 2. Basic CAD Features

Features	Composition
Extrusion	sketch entities, extrusion distance
Hole	circle, radius, depth
Chamfer	radius, angle
Fillet	radius, angle
Revolution	axis, sketch
Face	edge, vertex
Edge	line segment
Body	face, edge
Vertex	point, line segment
Sketch	arc, ellipse, parabola, line segment, point, spline

3.3 CAD Constraints

Constraints are associations and restrictions applied CAD entities and features to define shape and functionality. Conceptually, constraints mate or align surfaces, curves and points, until the relative position of the part, with respect to the partial assembly, has been completely specified [3]. There are two general types of constraints: geometric constraints control the relationships of objects with respect to each other and dimensional constraints control the distance, length, angle, and radius values of objects. A list of some major constraints is shown in **Table 3**.

Table 3. Geometric and dimensional constraints

Constraints	Representation
Coincident	A coincident constraint between endpoints of lines/circles/arcs and other segments
Collinear	A collinear constraint between line segments.
Concentric	A concentric circle or arc constraint.
Equal	An equal-length curve constraint on lines and circles/arcs.
Fix	A fixed-geometry constraint for line/circles/arcs.
Horizontal	A horizontal constraint on line segments.
Parallel	A parallel constraint between two line segments.
Perpendicular	A perpendicular constraint between two lines or a line and circle/arc segment.
Tangent	A tangent constraint between lines and circle/arcs or between two circles/arcs.
Vertical	A vertical constraint on line segments.

4 Design intent formalization for feature-based parametric CAD data exchange

One of the initial stages of product development is requirement gathering. These requirements may be the result of business, mechanical, or manufacturing considerations. Integrating these requirements in specific tools to manage the behavior of a model in a particular context is the challenge of capturing the designer's intent. Data translation and transformation entails conveying both the static representation of models in as well as the knowledge required to edit these models in the destination systems. Generally, design intent is a term used to describe how a model should be created and how it should behave when it is changed[5]–[7], [15], [16]. The essence of preserving the design intent in CAD data exchange is that there be a dynamic model available in the receiving system. This provides the user of the receiving CAD system with the possibility to modify the shape and other properties using the information (parameters, geometrical constraints, and features) exchanged during the transfer. This would enhance

an imported model from a so-called "dumb model" to a so-called intelligent model" or "smart model" [11].

One approach is to provide the missing information held by current static models. This information can be divided into geometric definition, parameters, constraints, rules, and an inference mechanism. For this information on the designer's intentions to be effectively shared and used across systems, its representation should be independent of any system. Moreover, any changes made should seamlessly propagate to all systems that interface the data and data-manipulation instruction systems. Our approach to preserving design intent in CAD data exchange is to provide the missing information held by current static models. Using the MultiCAD API Manager proposed in [8] and the KAFA proposed in [17], engineering knowledge can be extracted from existing CAD models and experiential knowledge elicited from domain experts respectively. This information is then structured into an editable representation [8], [9] which can be interfaced by various authoring systems. This representation is both in XML and Json formats. It is best to describe the approach of preserving design intent in CAD data exchange by means of a succinct example. Before we expound on an example, let us first describe the main enablers of design intent: these include parameters (and variables), constraints, rules, and an inference mechanism.

4.1 Design intent in parameters, variables, constraints and design rules

In this paper, the concepts of parameters and variables will be kept simple. A parameter is assumed to be a constant in an equation describing a model while a variable is the way in which an attribute or quantity is represented. In CAD systems, parameter define the scope of variational entities such as the length of a line or the magnitude of an angle or a Boolean condition of a particular process or activity. Parameters and constraints can be used to determine the size, shape, and other properties of the different modeling elements[3]. Constraints are associations and restrictions applied to geometric entities to achieve a desired shape and/or function[18]. Constraints control the relationships of objects with respect to each other or control the distance or length and angle values of entities and features. Constraint based systems allow the position and size of geometric elements or entities to be specified using variables, allowing the designer to create a model which is easily modifiable during the design process. These constraints could be either at the assembly or at geometric levels. Some of the basic constraints include concentricity, tangentiality, parallelism, perpendicularity, etc.

Design rules can be either assignment operations assigning values to attributes, or a sequence of expressions comprising conditions and consequent actions if the conditions are evaluated to be true or false. [17] proposed the Knowledge Acquisition and Formalization Assistant (KAFA) which can be effectively used by experts to define explicit rules describing the functional relationships of entities and features in a CAD model. Rules are expressions of the form:

$$if<conditions> \ then \ <actions> \qquad (1)$$

where conditions are expressions involving attributes and the logical connective *and* [19].

When variables representing parameters are given specific values depending on the designer's requirements, the inference mechanism reacts to the changes, executing instructions that manage the propagated effects of this change. In general, when rules are examined by the inference engine, actions are executed if the information supplied by the user satisfies or does not satisfy the conditions in the rules[20]. The inference mechanism involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied or not. Two methods of inference are often in use - forward and backward chaining. Forward chaining is a top-down method which takes facts as they become available and attempts to draw conclusions leading to actions being executed while backward chaining is the reverse [19].

4.2   Example

Returning to the example, we show how a basic line segment (see **Table 1**) can by interrogated and all associated geometry, parameters, constraints, and rules safeguarded for reconstruction and manipulation in another CAD system or another appropriate authoring system. Let us assume the parametric equation of a line segment in three dimensions consists of a known point $p = (a, b, c)$, a direction vector $\vec{r} = (r_x, r_y, r_z)$ and a parameter μ which determines how long the line should be from the fixed point p in the direction $\vec{r}$. This equation can be written as follows:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \mu \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \tag{2}$$

where $x = a + \mu * r_x$, $y = b + \mu * r_y$ and $z = c + \mu * r_z$. For the sake of this discussion, it is assumed that the direction vector $\vec{r} = (r_x, r_y, r_z)$ is a unit direction vector. This means that it has a magnitude of one (1) : ($|\vec{r}| = \sqrt{r_x^2 + r_y^2 + r_z^2} = 1$). In this case, μ is a variable representing the length parameter that defines the length of the line starting from the fixed point $p = (a, b, c)$ and stretching in the direction of $\vec{r} = (r_x, r_y, r_z)$. In an appropriate authoring system, it should be possible to identify the length *parameter* (μ) through a unique identification. It should also be possible to define a *rule* which will assign a user defined length to the parameter, for example, μ= 10. With this assignment, the length of the line will be increased or decreased to 10 units. *Constraints* attached to the line such as perpendicularity, parallelism, tangentiality etc. will be enforced, and associated constraints propagated accordingly to other entities and features by the *inference mechanism*. All these attributes can be mathematically defined using vector analysis. This procedure can be applied to other geometric entities such as an arc, ellipse, spline or to features such as the extrude length. Because each CAD or authoring system and downstream application does handle parameter allocation, rules, and constraints differently, it makes sense to manage the static geometry, parameters, constraints, rules, and inference mechanism differently from each other. There should however, be a system in place to trace and relate these to each other. Our approach is a persistent naming convention based on a prefix relating to the entity, the @ sign and a Globally Unique Identifier (Guid). For example, the Line Segment

above will have an identification such as **d@ID419** that is unique in the whole representation. This unique identifier associated to the length parameter is used to trace this particular line in the database.

5 Test case

To exemplify the aspects of design intent preservation described above, we provide a simple test case. A simple hinge consisting of three parts – two flaps and a pin holding the flaps together but providing coordinated movement so that a shutter attached could be moved as desired (see the left image in **Fig. 2.**). While the three components have independent geometries, they are bound together by constraints, which enforce the coordinated movement. **Fig. 2.** shows the hinge assembly in SolidWorks including components and constraints. The right side of **Fig. 2.** shows a partial view of the generic xml representation generated with the MultiCAD API Manager. The constraints are defined with a name, a type, a unique identification, and a description.
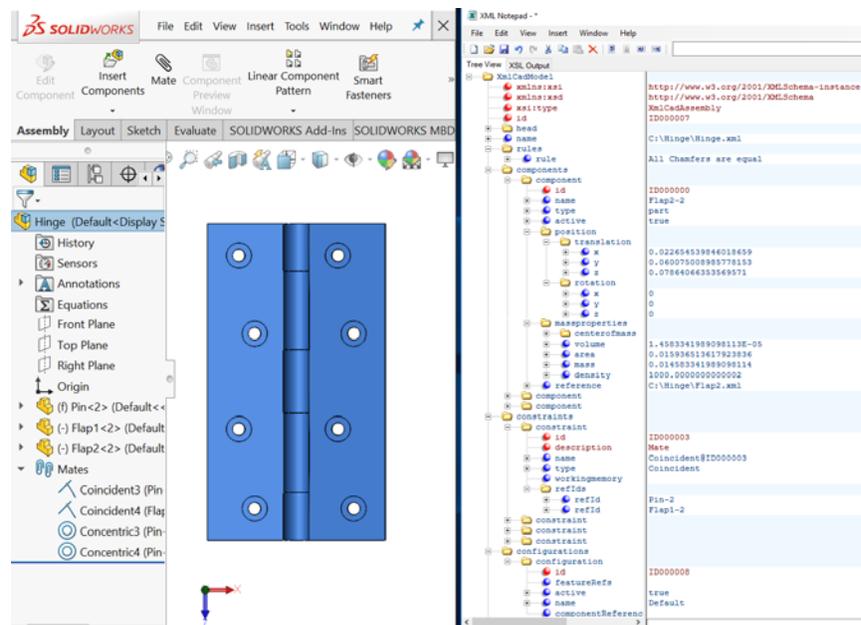
Fig. 2. Simple door hinge with two flaps and a pin in SolidWorks and partial view in XmlCad[1]

Fig. 3 provides more detail by showing the parameter and constraints of one hinge flap. The image to the left shows the base sketch of the flap with three lengths annotated

---

[1] XmlCad is a proprietary Software to represent and manipulate/edit CAD models represented in XML Format. The software is still under development

(ID001683, ID001693, and ID000424), two of which are equal and collinear (ID001683 and ID001693). The image to the right is a partial view of the XML representation of a hinge flap showing amongst other things features, parameters, and constraints.
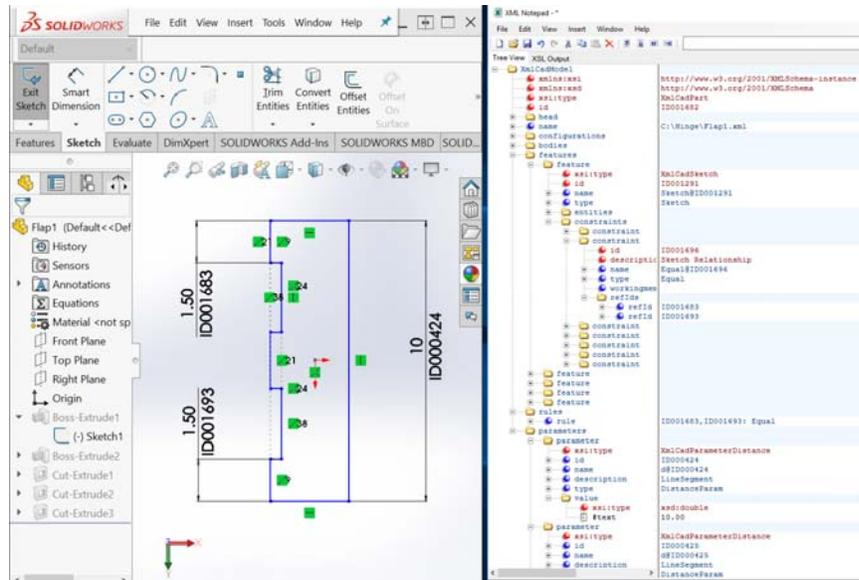

Fig. 3. Base Sketch of a Hinge Flap in SolidWorks showing Parameters and Constraints of plus partial view in XmlCad

6   Results, discussions and conclusion

Although design intent is a well-established required aspect in the context of CAD and downstream application systems, it remains a complex concept with different experts approaching it in different ways[5], [7], [10], [13], [15], [21]. Generally, the anticipated reaction of a CAD model to changes is commonly understood as design intent[5]–[7], [19]. The authoring tools as well as strategies employed do influence design intent communication. Recognizing the fact that tools and strategies do influence design intent communication, we are developing a set of optimized tools to cater for the various aspects of design intent preservation. With the MultiCAD API Manager, the KAFA and the neutral representation, we successfully extract and formalize CAD geometry, parameters, and constraints from existing CAD models. We further enrich this knowledge with experiential knowledge from experts thereby hoarding information that makes it possible for geometry to be editable in the target authoring system.

Different CAD systems represent and store different objects in different ways. However, the mathematics behind these objects are generally translatable. Using the principle of rewrites introduced in the UPR[14], it is possible to translate the data from the

general representation to suit the requirements of particular CAD systems. Our editable representation takes two form – XML and Json – the two main data representation and transportation formats. With these, most authoring systems can parse and get the required data and information for further use.

Parametric or variational modeling systems have been developed to enable the designer to "capture design intent" in the geometric model of a part or assembly. Preserving the designer's intent during CAD translation or transformation" is a challenge. This paper presented a test case, which shows how design intent propagated by the bearers geometric parameters, constraints, rules and an inference mechanism can be preserved during CAD exchange. The goal of the preservation is to enable the transmission of design intent information during a model transfer so that the model can be edited in the receiving system as though it had originally been created there.

In combination with the researches presented in [8], [9], the next stage in this project is to provide a complete software solution to manage engineering knowledge for reuse in different authoring applications.

References

[1] M. Bordegoni and C. Rizzi, Eds., *Innovation in Product Design*. London: Springer London, 2011.

[2] J. Camba, M. Contero, M. Johnson, and P. Company, "Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability," *Comput.-Aided Des.*, vol. 57, pp. 61–73, Dec. 2014.

[3] C. M. Hoffmann and R. Juan, "Erep An Editable High-Level Representation for Geometric Design and Analysis," 1992.

[4] X. Xu, *Integrating advanced computer-aided design, manufacturing, and numerical control: principles and implementations*. Hershey, PA: Information Science Reference, 2009.

[5] J. Camba, M. Contero, and J. Otey, "Explicit Communication of Geometric Design Intent in CAD: Evaluating Annotated Models in the Context of Reusability," in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2014, p. V007T07A014–V007T07A014.

[6] J. Otey, P. Company, M. Contero, and J. D. Camba, "Revisiting the design intent concept in the context of mechanical CAD education," *Comput.-Aided Des. Appl.*, vol. 15, no. 1, pp. 47–60, 2018.

[7] M. Li, F. C. Langbein, and R. R. Martin, "Detecting design intent in approximate CAD models using symmetry," *Comput.-Aided Des.*, vol. 42, no. 3, pp. 183–201, Mar. 2010.

[8] J. Cho, T. Vosgien, and D. Gerhard, "Engineering Knowledge Extraction for Semantic Interoperability between CAD, KBE and PLM Systems," in *Product Lifecycle*

Management and the Industry of the Future., S. Foufou, J. Ríos, A. Bernard, B. Eynard, and A. Bouras, Eds. Sevilla, Spain: Springer International Publishing, 2017.

[9]   J. Cho, T. Vosgien, T. Prante, and D. Gerhard, "KBE-PLM Integration Schema for Engineering Knowledge Re-use and Design Automation," presented at the 13th IFIP International Conference on Product Lifecycle Management (PLM16), Columbia, SC (USA)., 2016.

[10]   J. Kim, M. J. Pratt, R. Iyer, and R. Sriram, "Data exchange of parametric CAD models using ISO 10303-108," *NIST Intergov. Rep. Gaithersburg MD USA Natl. Inst. Stand. Technol.*, 2007.

[11]   S. Bondar, A. Shammaa, J. Stjepandic, and K. Tashiro, "Advances in Parameterized CAD Feature Translation.," in *ISPE CE*, 2015, pp. 615–624.

[12]   F. Maier and M. Stumptner, "Enhancements and Ontological Use of ISO-10303 (STEP) to Support the Exchange of Parameterised Product Data Models," 2007, pp. 433–440.

[13]   M. J. Pratt, "Extension of the Standard ISO 10303 (STEP) for the exchange of parametric and variational CAD models," in *The Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility and the Virtual Enterprise*, vol. 4, Springer, 1998.

[14]   A. Rappoport, "An architecture for universal CAD data exchange," in *Proceedings of the eighth ACM symposium on Solid modeling and applications*, 2003, pp. 266–269.

[15]   H. K. Ault, "Using geometric constraints to capture design intent," *J. Geom. Graph.*, vol. 3, no. 1, pp. 39–45, 1999.

[16]   M. Contero, O. Jeffrey, C. Pedro, and D. C. Jorge, "A Review of the Design Intent Concept in the Context of CAD Model Quality Metrics," p. 10, Jul. 2014.

[17]   D. Gerhard and C. Lutz, "IT-Based Configuration and Dimensioning of Customer Specific Products – Towards a Framework for Implementing Knowledge Based Design Assistant Systems," *68-6 Proc. 18th Int. Conf. Eng. Des. ICED 11 Impacting Soc. Eng. Des. Vol 6 Des. Inf. Knowl. LyngbyCopenhagen Den. 15-19082011*, 2011.

[18]   J. J. Shah, "Designing with Parametric CAD: Classification and comparison of construction techniques," in *Geometric Modelling*, F. Kimura, Ed. Boston, MA: Springer US, 2001, pp. 53–68.

[19]   N. L. Griffin and F. D. Lewis, "A rule-based inference engine which is optimal and VLSI implementable," in *Tools for Artificial Intelligence, 1989. Architectures, Languages and Algorithms, IEEE International Workshop on*, 1989, pp. 246–251.

[20]   K. Orsvärn and T. Axling, "The Tacton view of configuration tasks and engines," in *Workshop on Configuration, Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 1999, pp. 127–130.

[21]   L. A. Piegl, "Knowledge-guided NURBS: Principles and Architecture," vol. 3, no. 6, p. 12, 2006.