





A Constraint-based Framework to Recognize Design Intent during Sketching in Parametric Environments

Sofia Kyratzi¹  and Philip Azariadis² 

¹University of the Aegean, skiratzi@aegean.gr

²University of the Aegean, azar@aegean.gr

Corresponding author: Sofia Kyratzi, skiratzi@aegean.gr

Abstract. Current research on geometric design intent underlies the importance for the development of effective methods for the interpretation, capturing and enhancement of design intent during geometric modeling of 3D objects. In this paper, we propose a constraint-based framework for the automatic recognition of a set of design intentions during the sketching process of mechanical and industrial products. We introduce the term “intention regularities” to describe common constraining schemes that occur frequently in a product design process. By combining intention regularities and standard CAD constraints we define a set of meta-constraints which can be utilized in a sketching CAD environment, under an integrated approach, to enhance a design-intent-based modeling strategy.

Keywords: Design Intent, Sketching, Parametric Modeling, Constraints.

DOI: <https://doi.org/10.14733/cadaps.2021.545-560>

1 INTRODUCTION

Modern parametric CAD software provide the designers with tools to create 3D models that convey their objectives about the geometry, functionality, engineering and manufacturing of a product. The objectives are incorporated into the model's geometry with the use of parameters and constraints [9]. A single parameterized model represents a family of different objects [3] and if built efficiently, can be easily edited, altered and redesigned. A model is considered efficient and robust when it transfers its design knowledge and it reacts in a predictable way to the changes of the parameters without being rendered into inconsistency [5][7]. The purpose or underlying rationale behind an object [18] and the design decisions that are taken during the development process are summarized under the concept of “Design Intent” [11].

The “Design Intent” concept admits multiple definitions (see [11],[22] and the references therein) that all urge to describe a notion which associates the modeling decisions about the geometry of an object with a higher level of engineering and manufacturing information that is related to shape [20]. Design intent is associated with constraints that define the geometry of an object, captures functional characteristics, as well as engineering and manufacturing properties

[2]. In this work, we focus on geometric design intent that refers to the constraints and parameters for the geometric creation and modification of a model. For a CAD model to be easily modified, the designer must introduce a constraint scheme that best express his/her intention in a geometric and semantic level. This scheme should be flexible enough to permit changes but also retain design intent [8]. Although, research community highlights the importance of a cautiously planned modeling strategy, the need for a quick creation of a 3D model and time pressure can cause designers to sacrifice proper design intent for expediency [14]. A more supportive approach includes the development of design tools that will capture and support design intent during modeling. The goal is to distribute the effort of building efficient and robust models between designers and CAD software.

Most of the commercial CAD software make an effort to embrace the notion of “design intent”. They employ geometric and heuristic criteria to presume/interpret the intention of the designer from the drawn geometry and the selected modeling tools and inform him/her if an operation fails or is not allowed to be executed. The design intent emerges directly from the geometric definition of the modeling tools. This effort reveals the importance for the development of effective methods for the interpretation, capturing and enhancement of actual design intent during geometric modeling. The major challenge is at what extend a parametric software can support a designer to efficiently incorporate design intent in his/her models, so to avoid the “from scratch remodeling” paradox [6]. Company et al., [8] conclude that there is a need for intelligent parametric editors which can support the constraining tasks performed in parametric CAD environments.

Following the aforementioned challenge, in this paper, we propose a framework to enhance parametric constraint editors, which can be utilized to automatically recognize design intentions (such as symmetry, centrality etc.) during the sketching process. We first study the most common types of constraints in CAD software and propose four new classifications to characterize the way they influence design intent. We identify design intention in numerous mechanical and industrial products and we introduce the term “intention regularities” to describe them. Then, we study three popular commercial CAD packages (Creo Parametric, Inventor, and SolidWorks) in order to determine to what extend they assist their users to design on the basis of intention regularities. Finally, by combining intention regularities and standard CAD constraints we define a set of meta-constraints which can be utilized in a sketching CAD environment.

The paper is organized as follows. Section 2 presents the related literature. Section 3 discusses common types of constraints and proposes a design-intent-based classification. In Section 4 introduces the term of intention regularities and discuss their application in CAD software. Section 5 presents the proposed constraint-based framework for the recognition of design intent during sketching. Section 6 presents an example and related discussion, while Section 7 provides final conclusions.

2 RELATED WORK

Parametric CAD software process in three modeling levels. The lower level corresponds to the 2D profiles (or “sketches”), the middle level refers to the modeling operations (or “features”) and the upper level represents the modeling sequence (or “model tree”) of the 3D model. Flexibility and robustness must be achieved at all three levels [12]. Design intent in CAD models is expressed through the sketch constraints that define the 2D profile of a 3D modeling operation, the relationships that are set between modeling operations (or features) and the selection of the modeling operations (features) [22]. The flexibility of a profile does not depend on the amount of constraints, but on the semantic level of those constraints.

Ganeshram & Mills [11] underly the importance of an intent-based design approach for the creation of a robust, modifiable 3D model. Since there is no unique modeling or dimensioning scheme for a part [2], it relies mostly on the designer’s skills to select the most appropriate modeling and constraining strategy that best capture the design intent. Several works [4],[14-15],[22-23] acknowledge this fact and focus on the study and development of strategies and

approaches that improve the expression of design intent into CAD models. All of them emphasize that CAD education should follow a more strategic knowledge schema together with the declarative knowledge of the software tools. Johnson & Diwakaran [15] study modeling procedure and attributes that facilitate the model understanding and alteration. In [4] the authors analyze industrial designer's and student practices on CAD in order to identify key factors that lead to better modeling performance of complex parts and conclude to a methodology that consequently reduces the number of possibilities for the user when creating a model. A methodological framework to assist and preserve design intent in the form of semantics related to component shape is proposed by Otto & Mandorli [23]. The proposed framework is based on the assumptions that shape, dimensions, and required manufacturing precision of a mechanical component are designed to fulfill specific functionalities.

In parallel and in an effort to improve design intent representation in 3D models, numerous researchers [1-3],[8-9],[12-13],[20] study and analyze the constraints that are used in CAD software and their influence in design intent communication. An early research in the field, conducted by Ault [1], discusses methods for constraining geometric models and strategies for capturing design intent in these models. Similarly, in [3], the primary objective is the derivation of a comprehensive and consistent set of geometric constraints for shape definition. The problem that models fail to regenerate under some parameter values is discussed by Hoffmann & Kimb [13]. They propose an algorithm, based on non-linear optimization, that computes valid ranges for dimensional constraints within which the model will regenerate.

According to Camba et al. [5], the ability to alter and reuse CAD models in history-based parametric environments, relies on the parent/child relationships and interdependencies which are captured in a tree-like structure. Based on the traditional way of communicating design intent with functional dimensioning in mechanical drawings, Mandorli et al., [20] propose a method for supporting 3D dimensioning within 3D explicit modeling environments. An approach to communicate design intent on the basis of model annotations is discussed by Camba et al., [6]. They introduce the "extended annotation" structure type, where design information is represented both internally within the 3D model and externally. In [10], Contero et al., evaluate the effectiveness of 3D annotation techniques as a tool to express and communicate design information.

Focusing on the constraints in sketches, González-Lluch et al., [12] study the effect of the fix geometric constraint in reusability tasks and concluded that fix constraints do not convey design intent in an effective way. Recently, Company et al., [8] propose a classification of model constraints with respect to their semantic meaning. The constraints are distinguished between discrete (associative) and continuous (metric) constraints, which are further subdivided into intrinsic (i.e., relate geometrical components of the figure to each other) and extrinsic (i.e., relate the figure to the scene). Another classification of constraints is discussed in [16], where the authors distinguish between the implicit-explicit parameters and the dependent/independent/free parameters. An exhaustive study and analysis on constraint types and classification is given [3]. Their goal is to derive a consistent and comprehensive set of geometrical constraints which enables compatibility in parametric data exchange.

Based on the above, we conclude that the efficient representation of design intent relies mainly on the ability of parametric CAD software to properly support a parametric modeling process. In this work, we employ the latest version of three popular commercial parametric CAD software (Creo Parametric, Inventor, and SolidWorks) and study whether they can interpret design intent as this is conveyed through selected modeling tools. We define a framework of structured constraints (called meta-constraints) that are able to capture and represent the design intent of a model in a parametric CAD environment.

3 2D CONSTRAINTS AND DESIGN INTENT IN SKETCHING CAD ENVIRONMENT

In parametric modelling environments, designers use sketch entities (such as lines, curves, arcs etc.) to define the geometry and the topology of a 2D profile. Constraints are employed to determine the size, position, and orientation of a sketch, and set the relationships for the sketch entities [1]. The values and the arrangement of the constraints produce an instance of a geometric object. Every design decision conveys the intention of the designer. Although the designer recognizes his/her own design intent in the produced sketches, a third party, human or software, has to interpret the underlying design intent through the employed constraints.

Each geometric element in a sketch has a certain number of degrees of freedom (DOF) that is the number of independent parameters that specify its shape and position on the sketch plane. The introduction of constraints in a sketch removes gradually the DOF. The sketch is fully constrained when all the DOF of all sketch elements are removed, under-constrained when there are sketch elements that their position/shape is not fully defined, and over-constrained when there are more constraints than the number of DOF. The majority of CAD software support only fully constrained sketches and requires action from the user for the other two cases.

Each parametric CAD software offers a set of constraints that are classified into four standard types, i.e., dimensional, geometric, ground and algebraic constraints [2]. Dimensional constraints are used to define the size of the profile or the distance between two entities. Geometric constraints impose non-algebraic relationships between sketch entities. This type usually includes the tangent, parallel, coincident, horizontal, vertical, equal, symmetric, and perpendicular constraint. Ground constraints orient and position the sketch with respect to the global coordinate system. Algebraic constraints impose restrictions on dimensional or ground constraints in the form of mathematical equations. Some CAD software include the fix constraint that constrains the position of a sketch element relative to the sketch coordinate system. Fix constraints are geometric ground constraints that do not allow the sketch element to be modified and prevent reusability [12]. Constraints are applied either by the designer or are automatically detected by the sketching software. Although CAD software assist the designer to constrain the geometry of a sketch in a schematic level, it cannot offer efficient support in a semantic level that express design intention [8]. All constraints, geometric elements and parameters formulate a system of equations and the constraint solver is employed to provide a comprehensive solution [1].

For the development of the proposed framework, we utilize the characterization of constraints as explicit/implicit and strong/weak, because these classifications directly designate a design choice. Explicit constraints are created by the user. Implicit constraints are not represented as entities in the sketch, but they are inferred from the values of other constraints [16]. Strong and weak classification refers, correspondingly, to the constraints that their value is set by the user and those that their value is calculated automatically by the CAD software. Strong constraints indicate design intent and are given high priority by the constraint solver. Furthermore, we distinguish between global and local constraints. A global constraint captures a relationship that concerns all the area of the sketch by considering the sketch as an entity itself. For example, a mirror constraint or the dimensional constraint with value 250 in Figure 1 (b). Local constraints refer to the constraints that involve a part of the sketch.

Considering the classification of constraints according to their semantic meaning [8], we propose four new classifications of constraints with respect to the way they convey design intent: (a) Automatic/ manual constraints: Automatic are the constraints that can be either applied by the designer or by the CAD system. Manual are the constraints that are only applied by the designer, like symmetric or fix constraint; (b) Autonomous/associative constraints: Autonomous are the constraints that are applied to one sketch entity and associative are the constraints that relate two or more entities or other constraints; (c) Direct / Lateral constraints: A constraint is characterized as "direct" when the design intent is straightforwardly expressed by the semantic meaning of the constraint, such as parallelism or tangency. Lateral constraints, if grouped together may indicate a design intent that is not directly expressed (e.g., multiple vertical constraints indicate parallel sketch entities), and (d) Strict / soft constraints: Strict constraints are explicitly defined by a

designer or the CAD software to express a certain design intention, e.g., tangent constraint explicitly indicates that two entities should be tangent. Soft constraints do not express a design intent.

The above classifications indicate at which level a constraint expresses a certain design intention. The majority of classification approaches focus on general constraint objects in an effort to support designers to capture their design intent. This classification approach offers a limited interpretation with respect to the extent a constraint carries a design intent, especially when combined with other constraints. In contrast, with the classification approach proposed in this paper, we are able to characterize the constraints as part of a meta-constraint entity and express design intent with more clarity.

Considering the geometric, topologic and semantic meaning of each constraint, we analyze the four standard constraint types in terms of the introduced classifications. Dimensional constraints are in general manual, autonomous constraints with soft intention. They become associative, lateral and strict constraints when there are algebraic equations that correlate them. Algebraic constraints are manual, associative, strict constraints that express an implicit intention. According to the commercial software, all geometric constraints are automatic (except of symmetric constraint), direct and strict. But a substantive classification of the geometric constraints depends on the constraining schema of each sketch. For example, a horizontal constraint directly and strictly expresses an "horizontal line" intention, but when combined with horizontal and vertical constraints it is considered as lateral and strict constraint that implies "parallel", "equal", or "perpendicular" intentions. Finally, ground constraints are classified according to the constraint type that express them.

4 INTENTION REGULARITIES

In the context of 3D reconstruction from a single sketch, the term "image regularities" is used to express the spatial relationship among a pair, a group or all the elements in a sketch that correspond to a feature in the 3D model [17][19][21]. Image regularities refer to multiple different geometric relationships between elements that occur frequently in objects, like perpendicularity, orthogonality, skewed symmetry etc. Extending this concept, we introduce the term "intention regularities" to describe the geometric or topologic patterns (e.g., symmetry, orthogonality etc.) that appear in engineering sketches and can be recognized as design intentions. Intention regularities are related to the modeling strategy that is followed for the creation of a 3D object and are expressed through a set of entities and constraints in 2D sketches. For example, an object that includes symmetric parts, perpendicular faces and cubic corners reflects the design intention of a (partial) symmetric orthogonal object. The choice of the appropriate sketch elements combined with symmetric, vertical and horizontal constraints comprise the modeling strategy that captures this intention. In this research work, we introduce the intention regularities of "partial symmetry", "centered sketch", "side sketch" and "closed loop" as defined in the following.

"Partial symmetry" intention regularity refers to the design intention of a partial symmetry around an axis. It is expressed through the partial reflection symmetry between two or more sketch entities around X-/Y- axis (Figure 1(a)). It concerns only a part of the sketch and corresponds to a local geometric property. The design intention of creating a centered object around coordinate axes is described by the "centered sketch" intention regularity, and it is expressed through a centered sketch with respect to the X or/and Y axes (Figure 1 (b)). Centered sketch and partial symmetry intention regularities correspond to different design intents, even if a centered profile exhibits symmetric properties. Thus, they are studied separately.

The "side sketch" intention regularity refers to a sketch that is placed on one quadrant of the projected planes (Figure 1 (c)). A side sketch can be a strategic selection or a modeling necessity in the case of revolve operations around an axis or existence of adjacent features. A closed loop refers to a sketch whose boundary entities form a closed, non-self-intersecting loop. The "closed

loop” intention regularity describes a common design intention that is used for the creation of a 3D object. Side sketch and closed loop intention regularities associate all sketch elements.

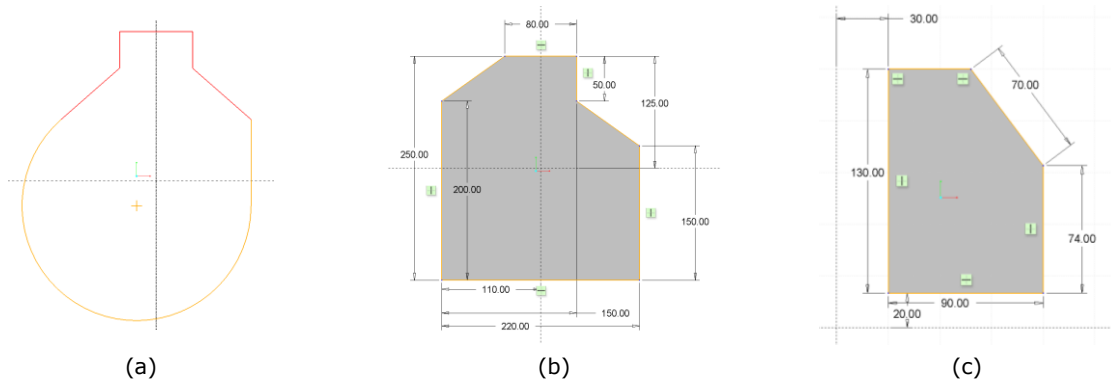


Figure 1: (a) A sketch that exhibits partial symmetry. (b) A centered non-symmetric profile. (c) A side sketch lies on one quadrant of the sketch plane. All profiles designate a closed loop.

We used the latest versions of Creo Parametric, Inventor, and SolidWorks to test whether they can recognize the above four intention regularities. The test cases and the results are presented in the following subsection.

4.1 Case Studies and Results

Partial symmetry intention regularity: we study whether existing CAD software is able to recognize this intention regularity without the use of a mirror/symmetric operations (which require multiple design steps). For the oblique line in

Figure 2, none of the tested software is able to suggest a symmetric constraint, although they are able to propose other “plain” constraint options, like equality, perpendicularity or collinearity.

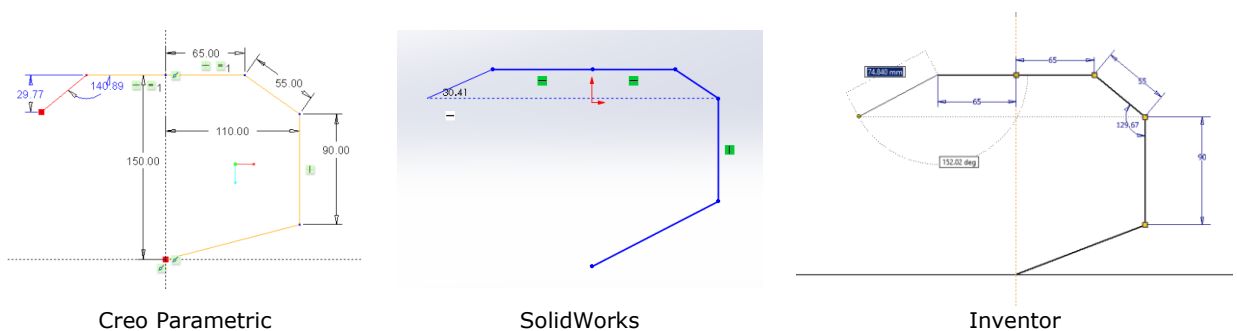


Figure 2: Partial Symmetry Intention Regularity. None of the employed CAD software are able to recognize the symmetric design intention of the user.

Centered sketch intention regularity: this regularity can be tested only when a sketch is fully constrained with strong constraints. In this case, the user constrains the sketch in such a way that implies a centered sketch intention (Figure 3). However, the implied intention is not maintained after a small modification of the vertical dimension indicating that the tested software is unable to capture and interpret this design intent.

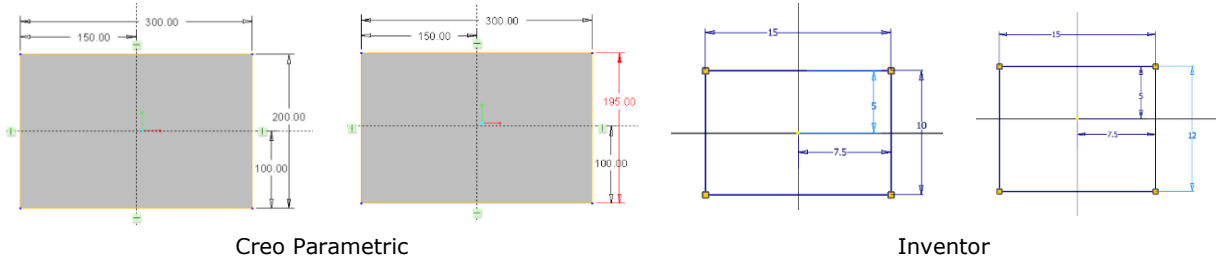


Figure 3: The evaluation of centered sketch intention regularity shows that the CAD software creates a centered sketch if only explicitly defined with a corresponding algebraic constraint.

Side sketch and closed loop intention regularities: Both regularities can be recognized after the sketch is completed, during the constraining phase (see 'initial sketch' in Figure 4). Our evaluation study shows that none of the CAD software considers the constraints imposed by the topology and geometry of the sketch, since dimension values are allowed that break either the side sketch or closed loop intention regularity (Figure 4).

CAD software do not recognize closed loop or a side sketch as design intentions, thus, neither they compute valid value ranges for the dimensional constraints nor they handle constraints in order to maintain the closed loop or the side sketch intention regularities. The employment of the fix constraint may partially solve the above problems, but it ends up in a non-flexible sketch [12]. Our experimental research showed that the employed CAD software deals with constraints as independent entities which are not implicitly related to each other. Dimensional constraints are associated only if the designer defines an algebraic relationship or inserts an associative geometric constraint (e.g., equal lengths). Valid values of dimensional constraints are computed only at the level of geometric validity of the profile and not on the basis of an interpreted design intent.

5 THE PROPOSED CONSTRAINT-BASED FRAMEWORK TO CAPTURE DESIGN INTENT

A constraint-based framework is proposed to capture the abovementioned four intention regularities, that follows an "integrated design intent" approach. The term "integrated design intent" is used to describe the design intent of a sketch as this emerges from all constraints' categories and from the geometric and topologic information that are conveyed in it. Within this framework we introduce the term "meta-constraint", which is a constraint object that implies a specific intention regularity. Four meta-constraint types are proposed that correspond to the four intention regularities. A meta-constraint is comprised by a group of standard constraints (that are called linked constraints), sketch entities or other meta-constraints. Different arrangement and association of constraints and entities may represent the same meta-constraint. Meta-constraints can be applied over the standard constraints, in an upper layer forming an object-oriented structure. They are able to capture and re-apply a predefined design intent and gradually establish the integrated design intent of a sketch. The meta-constraint is a design intent enforcement that associates a group of entities and their constraints, so they cannot be modified independently.

A sketch may include distinct groups of the same meta-constraint or different meta-constraints. Sketch elements and standard constraints may be associated with more than one meta-constraint. Each meta-constraint admits a weight during sketching that indicates its plausibility. The estimation of each weight is discussed in the following sections. According to its weight, a meta-constraint may be applied automatically or provided as a prompt choice. A meta-constraint exists in two modes, the synchronous and asynchronous. In the synchronous mode, it is applied during sketching. In the asynchronous mode the detection and implementation of the meta-constraint occurs after the sketch profile is completed. When a meta-constraint is applied, existing constraints will be replaced with new ones (grouped under the corresponding meta-constraint) but the sketch will remain fully constrained. When a designer deletes a meta-constraint

the corresponding design intent is removed and the sketch remains with the existing constraint scheme (fully constrained).

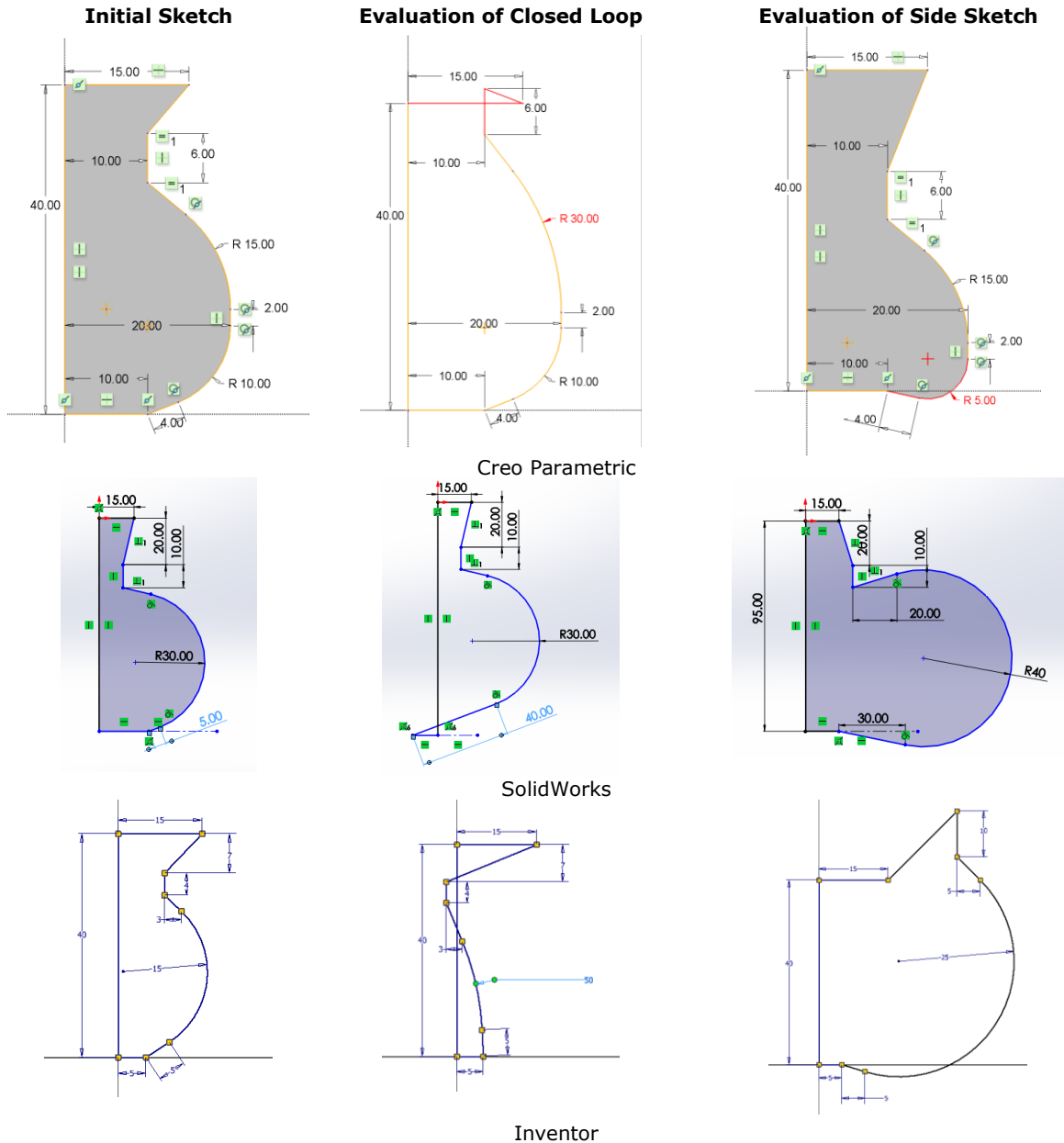


Figure 4: The constrained sketches in the first column indicate a closed loop and a side sketch intention. Existing CAD software is unable to recognize these two design intentions.

Meta-constraints are associative constraints of direct and strict intention. These attributes grant the meta-constraints higher priority. Every modification of the sketch should be validated by the active meta-constraints, otherwise it is not allowed. In the following, all proposed meta-constraints are presented and discussed in detail. In this work, we focus on sketches that do not include

internal loops and we define the meta-constraints accordingly. The examples are illustrated using Creo Parametric 5.0 environment.

5.1 Symmetric Entities Meta-Constraint

Symmetric Entities meta-constraint is related to the partial symmetry intention regularity. It is a local constraint and is defined from at least two sketch entities and the X/Y axis. It occurs in both synchronous and asynchronous mode. In its synchronous mode is linked with the equal, coincident, horizontal and vertical constraints. It also considers implicit or explicit ground constraints that control the position of sketch elements around an axis. In its asynchronous mode is linked with standard symmetric constraint or the mirror operation and is employed only as a design intent carrier. During synchronous mode, when a potential symmetry (with respect to X or Y axis) between existing sketch entities and a new entry is identified, a weight is calculated according to the incremental scheme shown in

Table 1 (examples of the weight computation are depicted in Figure 6):

Case		New Entity	Existing Entity	Weight	Status
Entity Type		Line	Line	0.1	OK
		Curve type	Same Curve Type	0.2	OK
		Line/Curve	NO	0	STOP
Constraint		Equal Lengths Constraint		0.2	OK
For each		Coincident Points Constraint on X / Y		0.1	OK
		Horizontal / Vertical Points Collinearity		0.2	OK
Axis	X	Horizontal Ground Distance < TOL		0.2	OK
	Y	Vertical Ground Distance < TOL		0.2	OK
Constraint		Horizontal Line (H)	Horizontal Line (H)	0.1	OK
Constraint		Vertical Line (V)	Vertical Line (V)	0.1	OK

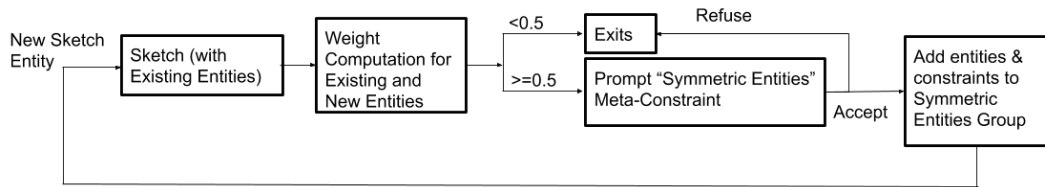
Table 1: The Symmetric Entities meta-constraint weight computation method. Parameter TOL implies a tolerance threshold predefined by the system.

The weight value depends on the constraint type and the corresponding design intent. Equal length is an autonomous, direct and strict constraint, so it is anticipated to convey symmetric intention. The equality of ground distances around an axis also convey direct and strict design intent regarding symmetry. Horizontal and Vertical constraints are autonomous, strict but lateral constraints, so they are assigned with lower weight. They can convey a symmetry intention if they are combined with other direct constraints. The same holds for coincidence constraint (on X/Y axis). Although it can initiate a geometric symmetry, in the context of symmetric entities meta-constraint is considered as lateral and soft.

If the weight is higher than a predefined threshold, a symmetric intention is detected for the new entity. In that case, the meta-constraint appears as a prompt choice to the designer (by highlighting the paired entities). We set the weight threshold equal to 0.5 to ensure that at least three criteria of

Table 1 are taken into account. Upon confirmation of the prompt choice, the mirror operation is applied to the new entity to establish the geometric symmetry. The involving sketch entities and the constraints of the initial entity are grouped and are associated with symmetric properties. The procedure continues detecting more symmetric entities for the same "symmetric entities" group. If the procedure exits with no new pair of entities, the group locks. Any new detection of symmetric entities establishes a new group. The flowchart of the above procedure is shown in

Figure 5.

**Figure 5:** The flowchart of the Symmetric Entities meta-constraint creation.

If there are more than one existing entities of the same type, the editor calculates a weight factor for all of them. Although different pairs of entities can be identified, the symmetric entities meta-constraint can be applied to only one of them. Utilizing the symmetric entities meta-constraint, there is no need to insert a centerline or to apply the standard symmetric constraint to each entity, facilitating the design process.

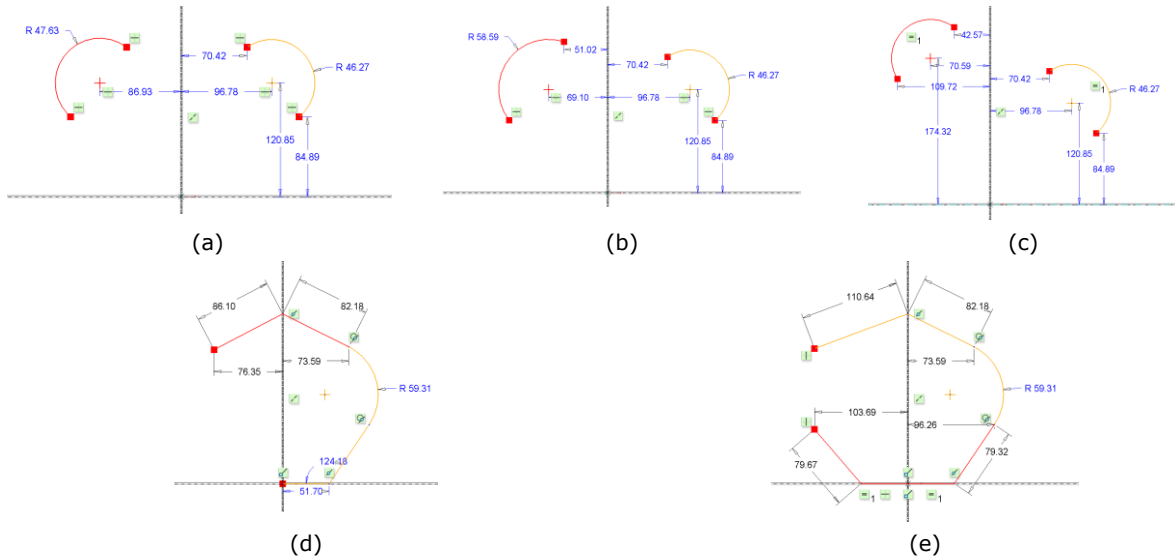


Figure 6: Application of symmetric entities meta-constraint examples. The associated weight is (a) 0.8 and (b) 0.6 due to horizontal points collinearities. In these cases the sketch editor prompts the choice of symmetric entities meta-constraint. (c) The weight equals to $0.4 < 0.5$ and the designer is never asked for symmetric intention. (d) The weight equals to 0.6 due to coincident points on Y axis, Horizontal Ground Distance and Horizontal Point Collinearity. The sketch editor prompts the choice of symmetric entities meta-constraint. (e) The upper lines of the sketch receive weight equal to 0.2 and the editor will not recognize a symmetric intention. For the lower lines of the sketch, the horizontal pair of lines receives weight equal to 0.9 and the adjacent oblique pair of lines weight equal to 0.7. If the user accepts the prompt choice, all four lines will be added in the same symmetric entities group.

5.2 Side Sketch Meta-Constraint

The Side Sketch meta-constraint corresponds to the side sketch intention regularity. It is a global constraint that relates all the sketch entities and their constraints. All standard constraints are

linked to the meta-constraint group together with the sketch elements and the axes that define the quadrant that the sketch is placed on. The position of the sketch is fixed with respect to these axes. Side Sketch meta-constraint operates in an asynchronous mode and receives the maximum weight value if all sketch entities are located on one quadrant of the sketching plane. In that case it is prompted as a choice to the designer. Upon confirmation, the constraint solver allows only for those value changes that do not result in sketch entities to cross the X/Y axes. If a value modification violates the meta-constraint, the software offers the choice to remove the meta-constraint and permit the value change. In the example of Figure 4 the linked axes are the positive X and Y axes. The value changes that resulted in the sketches of the third column will not be allowed if the designer insists on preserving the meta-constraint.

A sketch can include only one side sketch meta-constraint. Current CAD software do not include neither a manual nor automatic constraint that can certify a side sketch. Its establishment depends on the constraining strategy that the designer follows, and our experimental results shows that it is not a trivial task.

5.3 Axes-Centered Meta-Constraint

The Axes Centered meta-constraint refers to the centered sketch intention regularity. It is a global constraint and involves all sketch entities and their constraints. It is linked to the medial X-axis and the medial Y-axis of the sketch. It operates in an asynchronous mode when the designer finalizes the sketch design. A simple sketch can include only one axes-centered meta-constraint. The axes-centered meta-constraint includes three cases. The X-centered case, the Y-centered case and the XY-centered case that correspondingly indicate a sketch that is centered around X, Y and both X and Y axes.

The axes-centered procedure begins if the sketch spans around X- or/and Y- axis (i.e., there are sketch elements in two or all four adjacent quadrants of the sketching plane). Two subgroups are generated to capture axes centrality, named as Group-X and Group-Y and include, respectively, the sketch elements and the constraints that indicate centrality around X- and Y-axis. If the medial X-axis (resp. Y-axis) of the sketch identifies (within tolerance) with the X-axis (resp. Y-axis), the weight of Group-X (resp. Group-Y) becomes equal to 1 (Figure 7).

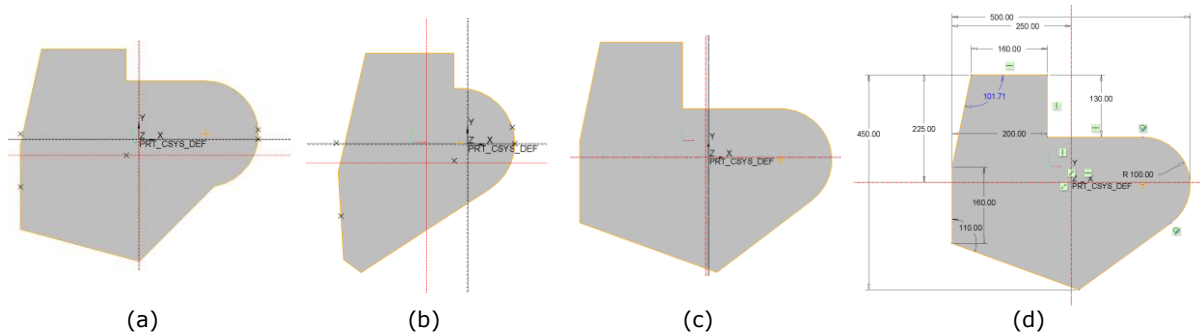


Figure 7: (a) Application of the Y-centered meta-constraint. The medial X-axis is not close enough (within given tolerance) to the X-axis and the software does not interpret a X-centered sketch intention. (b) The distance of the medial axes from the X- and Y-axes is out of the predefined tolerance and no axes-centered meta-constraint is applied. (c) Application of the XY-centered meta-constraint will result to the sketch depicted (d).

When either Group-X or Group-Y take a weight equal to 1, the sketch editor prompts the choice of X-/Y-/XY-Centered sketch meta-constraint (Figure 7(c)). The designer can select one of the prompt choices, and upon confirmation the axes-centered procedure coincides the medial axes with the coordinate axes generating a centered sketch (Figure 7(d)). In the case that the designer explicitly defines the proper algebraic constraints, the corresponding case of the meta-constraint is

automatically applied. The sketch is marked as either X-centered, Y-centered or XY-centered. Each time a value changes the axes-centered procedure should maintain the coincidence of the medial axes with the coordinate axes. This meta-constraint can co-exist with symmetric entities and closed-loop meta-constraints.

5.4 Closed-Loop Meta-Constraint

The Closed Loop meta-constraint corresponds to the closed loop intention regularity. It is a global constraint that relates all the sketch entities and their constraints. It is linked to all the constraints of the sketch. Closed Loop meta-constraint operates in an asynchronous mode and receives the maximum weight value when a closed loop of sketch elements is created. It is applied automatically, and the user can only remove it. All standard constraints are linked to the meta-constraint group together with the sketch elements. After a change of a parameter value, the CAD software checks if there are intersecting sketch entities. In the latter case, the software informs the user and provides him/her with the choice to keep or remove this meta-constraint.

Based on our experimental research, all tested CAD packages allow parameter values that produce a self-intersecting sketch. The Closed-Loop meta-constraint will enforce non self-intersections during sketching.

6 A DESIGN EXAMPLE OF THE META-CONSTRAINT FRAMEWORK

In this section we present a design example that employs the proposed framework. The wall mounted lamp with hangers of Figure 8(a) exhibits partial symmetry in the upper part while it is centered around X axis where the hangers are placed. The geometric intention of the designer is a partial symmetric X-centered sketch that will be the profile of the base feature. During sketching of the upper part of the profile (Figure 8 (b)-(c)), the sketch editor detects a symmetric entities intention. The horizontal lines receive a weight equal to 0.9 because of line type (0.1), equal lengths (0.2), horizontal points collinearity (0.2), coincidence on Y axis (0.1), equal horizontal ground constraint (0.2) and horizontal constraint (0.1). Accordingly, the weight of the oblique lines pair equals to 0.7 because of line type (0.1), two pairs of horizontal (collinear) points (0.4), and equal horizontal ground constraint (0.2). The symmetric entities meta-constraint is prompted as a choice to the designer, and if he/she confirms it for both pairs, a geometric symmetry is established (Figure 8(d)) and the corresponding entities are placed in the same symmetric entities group. For the sketch in Figure 8(e), the symmetric entities procedure exits without detecting another plausible symmetry.

When the designer starts the constraining process (Figure 8(e)), the sketch editor detects a closed loop and prompts for the corresponding meta-constraint. Simultaneously the axes-centered procedure begins because there are sketch entities in all four quadrants of the planes. The axes-centered procedure prompts for a X-centered meta-constraint because the medial X-axis of the sketch is in a close distance to the X-axis (Figure 8(f)). When the X-centered meta-constraint is confirmed, a X-centered profile is created (Figure 8(g)). The integrated design intent of the sketch includes the applied symmetric entities, closed loop and X-centered meta-constraints. These appear as visible properties to the designer (Figure 9(a)). The designer finalizes with the constraints and concludes with the generation of the base feature (Figure 9(b)).

The profile of the lamp object is placed on the up-right quadrant of the sketching plane in order for the designer to complete a revolve operation around Y-axis (Figure 10(a)). Since all sketch entities form a closed loop and are placed on one quadrant of the sketching plane, the weight of both the closed loop and the side sketch meta-constraints takes the highest value. Both meta-constraints are prompted as a choice. If they are confirmed, the sketch editor will never allow for parameter values that violate them (Figure 10(b-c)) unless the designer removes these meta-constraints from the sketch. The designer completes with the sketch constraining by inserting proper dimensional values (Figure 10(d)) and the final 3D object is created (Figure 10(e)).

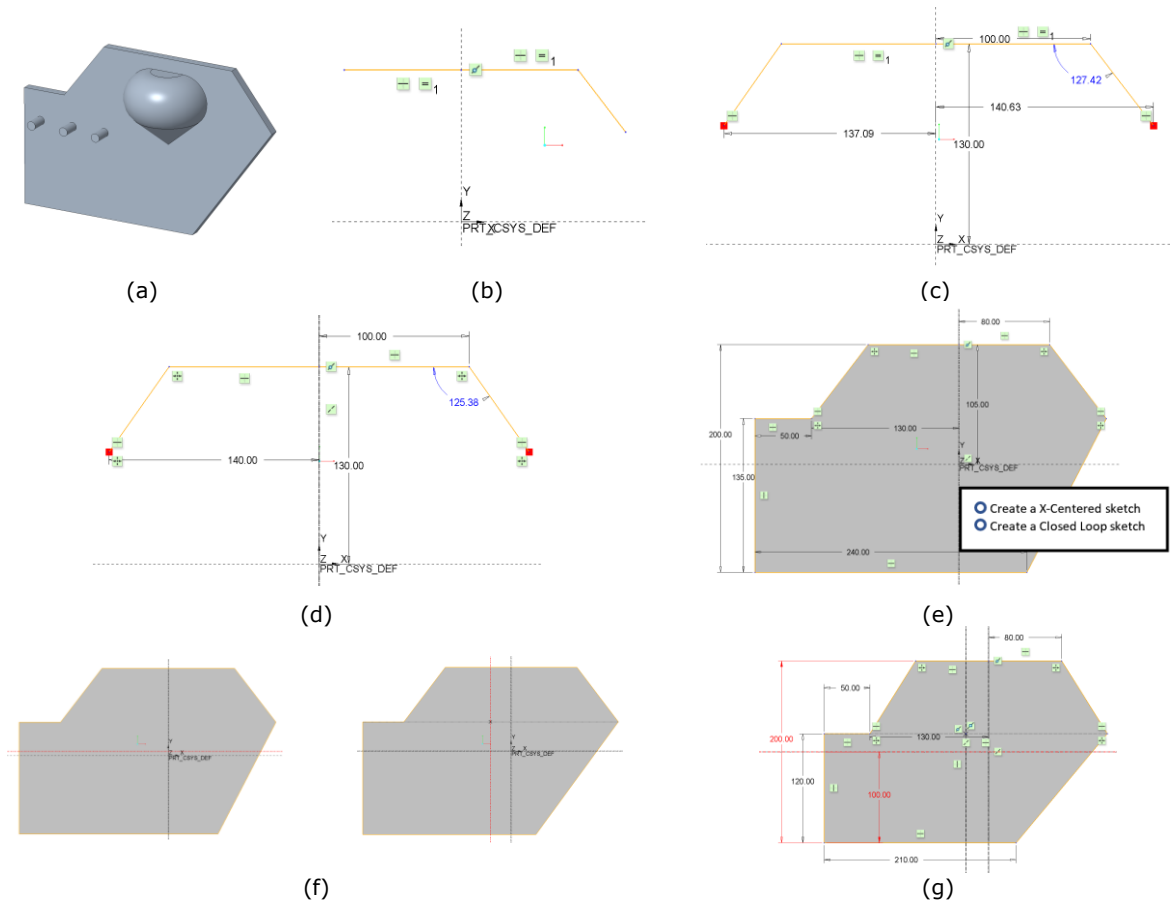


Figure 8: (a) The object to be designed. (b - c) The pairs of horizontal and oblique lines admit a weight >0.5 and the symmetric entities meta-constraint is prompted as a choice. (d -e) The confirmation of the symmetric entities meta-constraint results in a profile that exhibits partial symmetry in the upper part of it. (e-f) The axes-centered procedure detects an intention of the designer to create a profile that is centered around X axis and, (g) the confirmation of the X-centered meta-constraint results in a X-centered sketch.

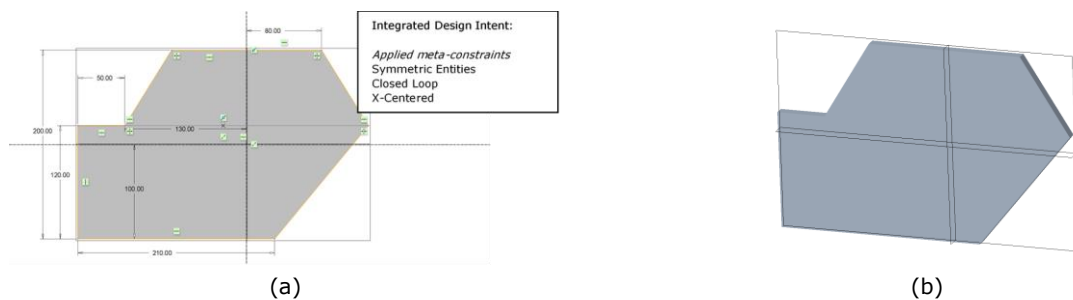


Figure 9: (a) The integrated design intent properties of the sketch. (b) The partial symmetric X-centered base feature.

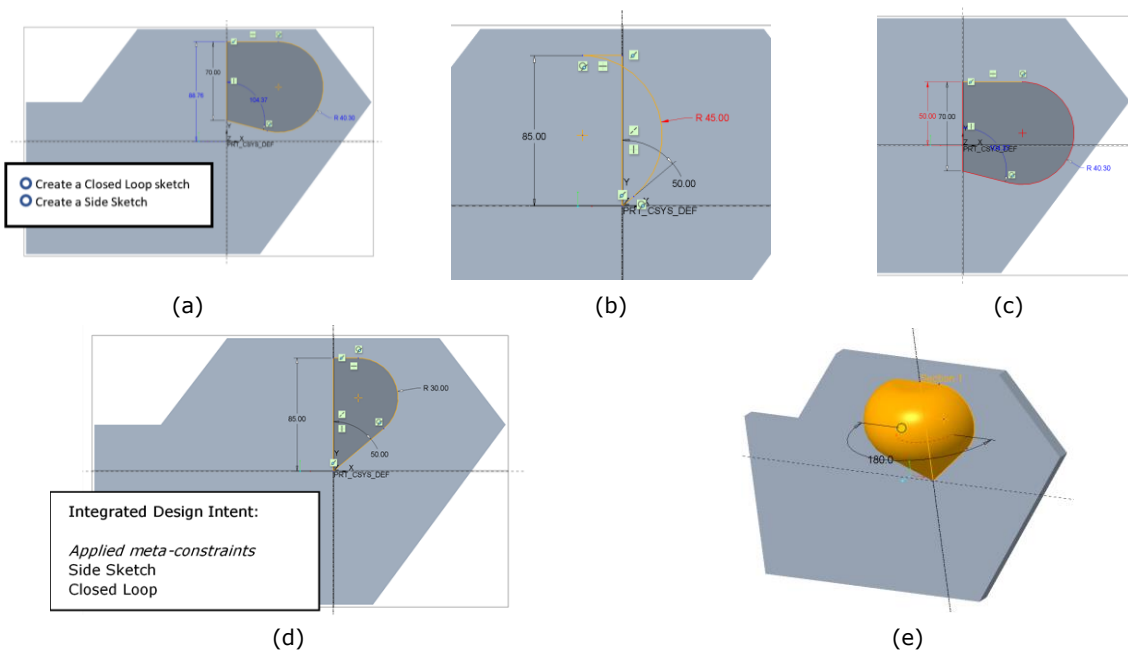


Figure 10: (a) During the development of the initial profile of the lamp object, the closed loop and side sketch intentions are detected. (b-c) The closed loop and side sketch meta-constraints will not allow such sketch solutions that violate the underlying design intentions. (d) The integrated design intent properties that are associated with the sketch. (e) The final step of the revolve operation using the designed profile.

7 CONCLUSIONS

The major advantage of the proposed framework is that it considers all sketch entities and constraints as interrelated elements that convey an integrated design intention. The intention regularities and meta-constraints establish a robust fundamental base for the design intent capturing and communication problem. In this work, we study four intention regularities and we propose a framework for the implementation of the corresponding meta-constraints. In this framework each intention regularity can be explicitly selected or removed by a designer during sketching. The proper management of meta-constraints can efficiently designate an integrated design intent of an object and preserve it during model modifications.

Future research focuses on a thoroughly study on engineering objects and modeling strategies in order to determine more intention regularities and define the corresponding meta-constraints. Moreover, more complex sketches with internal loops will be considered.

Sofia Kyratzi, <https://orcid.org/0000-0003-2266-9998>
 Philip Azariadis, <https://orcid.org/0000-0002-8687-5777>

8 REFERENCES

- [1] Anderl, R.; Mendgen, R.: Modelling with constraints: theoretical foundation and application, *Computer-Aided Design*, 28(3), 1996, 155-168. [https://doi.org/10.1016/0010-4485\(95\)00023-2](https://doi.org/10.1016/0010-4485(95)00023-2)

- [2] Ault, H.: Using geometric constraints to capture design intent, *Journal for Geometry and Graphics*, 3(1), 1999, 39-45.
- [3] Bettig, B.; Shah, J.: Derivation of a standard set of geometric constraints for parametric modeling and data exchange, *Computer-Aided Design*, 33(1), 2001, 17-33. [https://doi.org/10.1016/S0010-4485\(00\)00058-0](https://doi.org/10.1016/S0010-4485(00)00058-0)
- [4] Bodein, Y.; Rose, B.; Caillaud, E.: Explicit reference modeling methodology in parametric CAD system, *Computers in Industry*, 65(1), 2014, 136-147. <https://doi.org/10.1016/j.compind.2013.08.004>
- [5] Camba, D.-J.; Cosin, A.; Contero, M.: An Evaluation of Formal Strategies to Create Stable and Reusable Parametric Feature-Based 3D Models, In *Proc. of ASME International Mechanical Engineering Congress and Exposition*, 2014. <https://doi.org/10.1115/IMECE2014-37859>
- [6] Camba D.-J.; Contero M.; Johnson M.; Company P.: Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability, *Computer-Aided Design*, 57, 2014, 61-73. <https://doi.org/10.1016/j.cad.2014.07.001>
- [7] Cheng, Z.; Ma, Y.: A functional feature modeling method, *Advanced Engineering Informatics*, 33, 2017, 1-15. <https://doi.org/10.1016/j.aei.2017.04.003>
- [8] Company, P.; Naya, F.; Contero, M.; Camba, D.-J.: On the Role of Geometric Constraints to Support Design Intent Communication and Model Reusability, *Computer-Aided Design and Applications*, 17(1), 2020, 61-76. <https://doi.org/10.14733/cadaps.2020.61-76>
- [9] Contero, M.; Naya, F.; Pérez-López, D.; Company, P.; Camba, D.-J.: A Study on Sampling Strategies to Determine the Variability of Parametric History-Based 3D CAD Models, In *Proc. of the International Mechanical Engineering Congress and Exposition*, 2018. <https://doi.org/10.1115/IMECE2018-87404>
- [10] Contero M.; Otey J.; Company P.; Camba D.-J.: Explicit Communication of Geometric Design Intent in CAD: Evaluating Annotated Models in the Context of Reusability, *ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2014*, 2014. <https://doi.org/10.1115/DETC2014-34527>.
- [11] Ganeshram, I.-R.; Mills, J.-J.: Design Intent in 2D CAD: Definition and Survey, *Computer-Aided Design and Applications*, 3(1-4), 2006, 259-267. <https://doi.org/10.1080/16864360.2006.10738463>
- [12] González-Lluch, C.; Company, P.; Contero, M.; Pérez-López, D.; Camba, D.-J.: On the effects of the fix geometric constraint in 2D profiles on the reusability of parametric 3D CAD models, *International Journal of Technology and Design Education*, 29(4), 2019, 821-841. <https://doi.org/10.1007/s10798-018-9458-z>
- [13] Hoffmann, C.M.; Kimb, K.-J.: Towards valid parametric CAD models, *Computer-Aided Design*, 33(1), 2001, 81-90. [https://doi.org/10.1016/S0010-4485\(00\)00073-7](https://doi.org/10.1016/S0010-4485(00)00073-7)
- [14] Johnson, M.; Diwakaran, R.: Assessing the Effect of Incentive on Computer-Aided Design Intent. In the *Proc. of the ASME Design Engineering Technical Conference*, 2019, 523-532. <https://doi.org/10.1115/DETC2009-86644>
- [15] Johnson, M.-D.; Diwakaran, R.P.: An educational exercise examining the role of model attributes on the creation and alteration of CAD models, *Computers & Education*, 57(2), 2011, 1749-1761. <https://doi.org/10.1016/j.compedu.2011.03.018>
- [16] Kim J.; Pratt M.-J.; Iyer R.-G.; Sriram R.-D.: Standardized data exchange of CAD models with design intent, *Computer-Aided Design*, 40(7), 2008, 760-777. <https://doi.org/10.1016/j.cad.2007.06.014>
- [17] Leclerc, Y.; Fischler, M.: An optimization-based approach to the interpretation of single line drawings as 3D wire frames, *International Journal of Computer Vision*, 9(2), 1992, 113-136. <https://doi.org/10.1007/BF00129683>
- [18] Lee, J.; Lai, K.: What's in design rationale, *Human-Computer Interaction*, 6(3-4), 1991, 251-280. https://doi.org/10.1207/s15327051hci0603&4_3

- [19] Lipson H.; Shpitalni M.: Optimization-based reconstruction of a 3D object from a single freehand line drawing, *Computer-Aided Design*, 28(8), 1996, 651-663. [https://doi.org/10.1016/0010-4485\(95\)00081-X](https://doi.org/10.1016/0010-4485(95)00081-X).
- [20] Mandorli, F.; Otto, H.-E.; Raffaelli, R.: Explicit 3D functional dimensioning to support design intent representation and robust model alteration, *Computer-Aided Design and Applications*, 13(1), 2016, 108-123. <https://doi.org/10.1080/16864360.2015.1059201>
- [21] Marill, T.: Emulating the human interpretation of line-drawings as three-dimensional objects, *International Journal of Computer Vision*, 6(2), 1991, 147-161. <https://doi.org/10.1007/BF00128154>
- [22] Otey, J.; Company, P.; Contero, M.; Camba, D.-J.: Revisiting the design intent concept in the context of mechanical CAD education, *Computer-Aided Design and Applications*, 15(1), 2018, 47-60. <https://doi.org/10.1080/16864360.2017.1353733>
- [23] Otto H.-E.; Mandorli, F.: A Framework to Support 3D Explicit Modeling Education and Practice, *Computer-Aided Design and Applications*, 12(1), 2015, 104-117. <https://doi.org/10.1080/16864360.2014.949581>