Constraint Schemas for Sketch Parameterization: The Case of Centrality, Symmetry, and Inner Loops Design

Sofia Kyratzi[1], Philip Azariadis[2] and Zoi Koutkalaki[3]

[1]University of the Aegean, skiratzi@aegean.gr
[2]University of West Attica, fazariadis@uniwa.gr
[3] University of West Attica, zkoutkalaki@uniwa.gr

Corresponding author: Philip Azariadis, fazariadis@uniwa.gr

Abstract. During parametric modeling, the strategy that is employed for sketch parameterization depends on the ability of the designer to understand and express the design intent through the available modeling and constraining tools. A key challenge is to recognize and understand the design rationale behind constraining decisions so as to reduce subjectivity in parameterization. In this work, we handle deficiencies in sketch parameterization by exploiting the different perspectives of designers in constraining tasks. We study various sketches to identify different patterns of constraints that imply the same design intent, with the aim of providing a homogeneous basis to convey and establish design intentions in the context of IDI Architecture.

Keywords: Parametric modeling, Design Intent, Sketch-based modeling, sketch constraining.
DOI: https://doi.org/10.14733/cadaps.2027.87-100

1 INTRODUCTION

In a typical parametric, feature-based MCAD modeling process, a 3D model is constructed across three levels, starting from the 2D profiles (or sketches) to the features (or modeling operations), up to the final model [12]. The geometry of the sketches and features is defined with the use of parameters, constraints, and relations, which form a constraining schema. The constraining schemas determine the flexibility and robustness of the 3D model towards modifications of parametric values and convey its geometric design intent [3][22]. Design intent refers to all design decisions that are taken during the modeling process so that the model can be successfully regenerated without leading to inconsistencies [2]. A challenge of parametric design process is that there are different modeling strategies and constraining schemas for building the geometry of the 3D model, resulting in multiple design alternatives for approaching product performance and design requirements [20]. The modeling decisions and dependency management relies mainly on the designer and his/her ability to understand and express the design intent through the available modeling and constraining tools.

Constraints and parameters can be introduced in all three design levels. In feature-based modeling, sketch-based features have a dominant role. The performance of features relies strongly on the constraint schema that is employed for the parameterization of the feature's profile. For that reason, various research works [3][11-12][16][20] focus on design intent communication at the sketch level by studying and analyzing different constraints and the design intent they infer. A challenge of these research works is to recognize and understand design rationale behind constraining decisions so as to reduce subjectivity in parameterization. In this research path, the IDI Architecture [17-19] provides a direct and structured correspondence between different constraining schemas (called meta-constraints) and their inferring design intent (called intention regularities). The IDI Architecture approaches constraining tasks as a design-intent-oriented problem and provides an infrastructure for establishing design intent through a proper constraining schema.

The objective of this paper is to handle deficiencies in sketch parameterization, generated specifically by inexperienced users. By exploiting the different perspectives of designers in constraining tasks, we study various sketches to identify different patterns of constraints that imply the same design intent. Our study focuses on the ability of designers to express design intent through constraining tools independently of the software that is used. The identified patterns are then associated with an intention regularity and its corresponding meta-constraint implementation, in order to bridge the gap between standard constraining tools and design intentions. Under this approach different constraining schemas can be interpreted and expressed using the same meta-constraint tools, providing a homogeneous basis to convey and establish design intentions. The proposed approach offers a roadmap for design intent communication in MCAD education, improves model reusability and flexibility, and provides a perspective framework for recognizing and implementing design intent in 2D sketches.

2 BACKGROUND, SCOPE AND RESEARCH OBJECTIVES

In sketch parameterization, constraints, parameters and design variables determine a constraint schema that implements a design intent. When a parameter or a design variable belongs in at least two different constraining schemas that implement two or more design intents, then those design intents are called coupled. Otherwise, the design intents are called uncoupled, and the modification of one parametric value has an effect only on the corresponding design intention [20]. The Independence Axiom underlies the creation and preservation of uncoupled design intents, while Information Axiom indicates the minimum information content for the establishment of a design intent, i.e., the minimum number of available design variables for modification tasks. Sketch parameters are characterized as dimensional and ground. Dimensional parameters define the geometry of the shape, and ground parameters place the shape with respect to reference entities (i.e., axes, planes, edges, faces). The parameters that can be changed independently to capture design intents are called design variables or variable parameters.

The majority of the researchers approach design intent communication at the sketch level by either studying appropriate CAD education practices or by analyzing different constraining schemas in terms of the design intent they convey. These research paths explore under different perspectives the same problem, that of the development of an effective constraining and modeling strategy for the creation of 3D parametric objects that effectively convey their design intent.

Research on efficient parametric modeling practices and knowledge-based approaches to CAD education is closely related to studies on how designers realize the design problem, consider and analyze their different options, and how they finally process during modeling and modification tasks [4][6][14-15][22-23]. Concerning CAD education, researchers agree that declarative knowledge of modeling tools should be enhanced with strategic knowledge methodologies that will facilitate designers to adopt a more intent-based design approach [4][6][14]. Ottey et al. [22] conducted an extensive study on the use of design intent in CAD education and modeling practices, revealing a strong correlation between the effective representation and communication of design intent in MCAD models and the successful acquisition of MCAD tools by learners. Rynne and Gaughran [21] emphasize the importance for designers to understand the semantics of the modeling tools they use, so as to effectively create and manipulate geometry in MCAD software. They highlight as a key skill the ability to decompose geometric elements and cognitively assemble them in alignment with the design intent. Bodein et al. [6] propose a practical approach for modeling complex parts in parametric CAD systems, which prioritizes the explicit management of relationships between features through explicit references, while Aranburu et al. [2] identify inefficiencies in common industrial practices for constructing 3D models, with the aim of presenting the challenges of achieving robust and flexible designs. Focusing on improving modeling strategies, Contero et al. [11] introduce three indicators for evaluating the modification ability of the model, and in later work [15], they define six main dimensions of CAD model quality along with a set of rubrics to assess modeling expectations and establish best practices, particularly for novice users.

The primary objective of research on constraining schemas is the derivation of a comprehensive and consistent set of constraints that define the geometry of the model while accurately reflecting the designer's intent [5]. This includes studies on the impact of different constraint schemas on 3D models, as well as frameworks for the communication and representation of design intent [9][11-12] [17-19]. In the field of modeling and constraining practices, Yin and Ma [33] proposed feature parameter maps to explicitly manage feature dependencies through their defining parameters, while Contero et al. [10] introduced quality criteria based on rubrics to assess the influence of modeling practices on CAD model quality. Camba et al. [7] further analyzed three major modeling methodologies, horizontal, explicit reference, and resilient modeling, in terms of model reusability. Aranburu et al. [1] investigated these same strategies in design optimization contexts, evaluating their performance based on model behavior under geometric modifications, the success rate of generated design variations, and total processing time required to produce the solutions. Additionally, Company et al. [9] contributed to the field by classifying constraints according to their semantic meaning, addressing the challenge of design intent representation.

2.1 Scope and Objectives of Current Work

The Integrated Design Intent (IDI) Architecture is described in detail in [17-19]. Here, we briefly explain the backbone of the Architecture. The IDI Architecture forms a framework to capture the design intent of a sketch/feature/model as this is generated by the constraining choices of a designer. It sets the design intent, via the pair "meta-constraints" – "intention regularities". Intention Regularities (IR) are defined as geometric or topologic patterns that appear in engineering objects and can be recognized as design intentions. Meta-constraints (MC) are constraints defined by the combination of geometric entities, attributes, and standard constraints that geometrically and/or semantically express an intention regularity. Each of the three design levels, i.e., sketch, feature and model, includes a set of meta-constraints and intention regularities, named respectively as SMC/SIR, FMC/FIR, and MMC/MIR. Meta-constraints and intention regularities are associated to form an integrated design intent from sketch level to the model level and signify which modeling decisions convey an intended design choice compared to the geometric and/or constraining configurations of the sketch/feature/model. The IDI Architecture can support both Bottom-Up or Top-Down approach for capturing the model's design intent. In the Bottom-Up approach, the IDI Architecture assists the designers to constrain the model according to its design intent, which is embedded as an intrinsic property of it. In the Top-Down approach [19], the use of intention regularities and their corresponding meta-constraints facilitate designers to represent and implement design intent within the model. In both approaches, sketch meta-constraints are considered the starting point for model constraining, whereas the standard constraints provided by parametric software are not explicitly incorporated into the Architecture. In this work we aim to associate IDI Architecture with standard constraining tools and constraining strategies.

Considering the plurality of constraining schemas that express or convey a design intent, most research work in the domain of sketch parameterization and design intent focus on introducing different rules/rubrics that a designer should follow to count the robustness of their parameterization. In contrast, the IDI Architecture aims to establish an integrated and consistent framework for sketch parameterization aligned with design intent, by addressing the different ways that designers perceive or adapt constraining process. The first step to follow is to identify as patterns the different constraining schemas that are used to express the same design intention.

This work focuses on the sketch level. At this level, a sketch meta-constraint (SMC) is defined as a higher-level constraint composed of a group of standard constraints (linked constraints), sketch entities (linked entities), or other meta-constraints that collectively express a design intention. A sketch intention regularity is the design intention associated with a given sketch meta-constraint. In [18], we identified and described multiple sketch meta-constraints and their corresponding intention regularities. In the present work, we further concentrate on three sketch intention regularities and their constraining expressions: centrality, symmetry, and inner/outer loops.

In this work, we study a variety of sketches with respect to these intention regularities, with the aim to identify different constraining schemas that either directly or indirectly imply the same design intent. Through sketch analysis, combined with parametric modeling principles and the geometric definition of the corresponding design intentions we identify, for each design intent, a range of different constraining patterns. These patterns are further categorized into "valid constraining rules" and "invalid constraining cases" each associated with the relevant intention regularities. Unlike related research works on design intent communication and expression, which mainly introduce metrics or methodologies for establishing design intent and model robustness, the key contribution of this work lies in providing a homogeneous basis to convey and establish design intent by exploiting the various expressions and deficiencies observed in sketch parameterization with respect to a given design intent.

The association of the identified constraining cases with intention regularities leads to the development of an expandable glossary that bridges the gap between standard constraining tools and design intentions. This glossary is associated with both Top-Down and Bottom-Up approaches of IDI Architecture for its efficient incorporation and implementation within existing parametric CAD software. Furthermore, valid constraining rules provide a methodological foundation for the implementation of meta-constraint schemas. Invalid constraining cases contribute to the recognition of a design intent in a sketch when it is indirectly, inconsistently, or incorrectly expressed. Linking both valid constraining rules and invalid constraining cases with the same intention regularity enables the development of supporting modeling tools that can either assist the parametric modeling process or facilitate the immediate correction of constraining schemas.

The integration of valid and invalid constraining cases into the IDI Architecture can support a learning-by-activity paradigm, thereby improving the effectiveness of MCAD education. Designers can learn to recognize incorrect or insufficient constraining schemas, associate specific sets of constraints with implicit design intents, and understand the underlying semantics of constraining schemas, while also being guided towards more efficient and valid constraining methods. Moreover, within the context of the Top-Down approach, the IDI Ontology [19] can additionally provide designers with appropriate constraining schemas using current parametric modeling tools.

3 SKETCH ANALYSIS OVERVIEW

During the parametric design process, designers employ parameters and constraints to define the geometry of a sketch and capture the design intent of the model. The effectiveness of parameterization towards a design intent lies in the experience of the designer and their understanding of the constraining process. Thus, different constraining schemas may be employed to capture the same design intent. Acknowledging the generated ambiguity and exploiting the different perspectives, we analyzed the parameterization of student sketches in an MCAD course to identify patterns of parameters and constraint arrangements that are used to express the design intent of centrality, symmetry and inner/outer loop design. The study was conducted on a set of about 300 3D modeling projects created using Creo Parametric software by second- and third-year students enrolled in design engineering courses. To ensure that the observed constraining schemas reflected natural modeling practices rather than prescribed strategies, the projects were randomly selected from laboratory exercises in which the primary objective was the creation of a 3D model while no explicit guidance on sketch-level design intent was provided. A total of 223 sketches were selected from the overall set of 3D modeling projects, based on the existence of the three design intent schemas studied in this paper.

Sketch parameterizations and constraining schemas were analyzed using geometric modeling criteria, as well as criteria related to the expression and inheritance of geometric design intent at the three design levels. Within this framework, for each design intent, we define a set of *valid constraining rules* that establish a design intent based on the available parametric modeling tools and the underlying geometric and topological definitions. Valid constraining rules refer to robust constraining schemas that would preserve the desired design intent under changes in parametric values.

Our study showed that similar patterns of constraints are employed to achieve specific geometric properties in sketches. In general, we observed that most inexperienced designers tend to use parameters to express specific instances of design intent, instead of introducing more robust constraining schemas that would preserve the desired design intent. In this context, for each sketch intention regularity, we identified different constraining patterns that create an instance of a design intent or indirectly imply it. These patterns are categorized as *invalid constraining cases*. Two main groups of invalid constraining cases are identified across all design intents, named as "Inconsistent Dimensions" and "Wrong use of constraints". "Inconsistent dimensions" refers primarily to a constraining schema that lacks the necessary algebraic relations to establish appropriate parameter dependencies. "Wrong use of constraints" refers to a constraining schema that captures a design intent indirectly or inefficiently. Since these cases are not always explicitly distinguishable within a constraining schema, the categorization was performed according to the most dominant invalid characteristic observed in each case.

In the context of IDI Architecture [18], when a configuration of geometric entities and constraints that imply an intention regularity is detected, the corresponding meta-constraint is applied automatically or prompts as a choice, depending on whether it implies a straightforward design intention or corresponds to a configuration that accepts multiple interpretations or involves an ambiguous certainty. In the following subsections, we analyze the studied design intents in terms of (a) the constraint patterns that are identified for the recognition of an intention regularity and (b) the implementation strategy that is employed, by the corresponding meta-constraint, to establish the design intent.

4 CENTRALITY DESIGN INTENT

At the sketch level, centrality design intent is described by two intention regularities [18]: SIR_AxesCentered and SIR_BoundaryCentered. SIR_AxesCentered regularity refers to a sketch that is centered about one or both axes (X, Y, or both) and corresponds to the sketch meta-constraints SMC_XCentered, SMC_YCentered and SMC_XYCentered, respectively. SIR_BoundaryCentered refers to a sketch loop that is centered with reference to an outer loop and corresponds to the SMC_BoundaryCentered meta-constraint. We observed that, in complex geometric shapes, designers often seek to center only a subset of sketch entities with respect to one or both reference axes (X, Y, or both). To capture this design intent, we introduce the partial centered intention regularity, denoted as SIR_PartialCentered.

4.1 Axes-centered and Partially Centered Design Intent

Based on the parametric modeling principles, the geometric definition of centrality, and the available constraining tools, we define two primary rules to establish centrality along a single axis or across both axes. We consider these rules as a basis for detecting invalid cases of implemented centrality.

Axes Centrality Rule 1 – Geometric Constraints: aligns axial entities of a geometric shape with internal or external references using geometric constraints. Usually, pre-defined centered objects like circles, centered rectangle, or palette-based entities are utilized (Figure 1 (a-b)).

Axes Centrality Rule 2 – Parametric Dependencies: the sketch is defined with respect to internal or external references using geometric constraints and parameters with relations among them, with the aim to create dependencies that establish centrality (Figure 1 (c)).

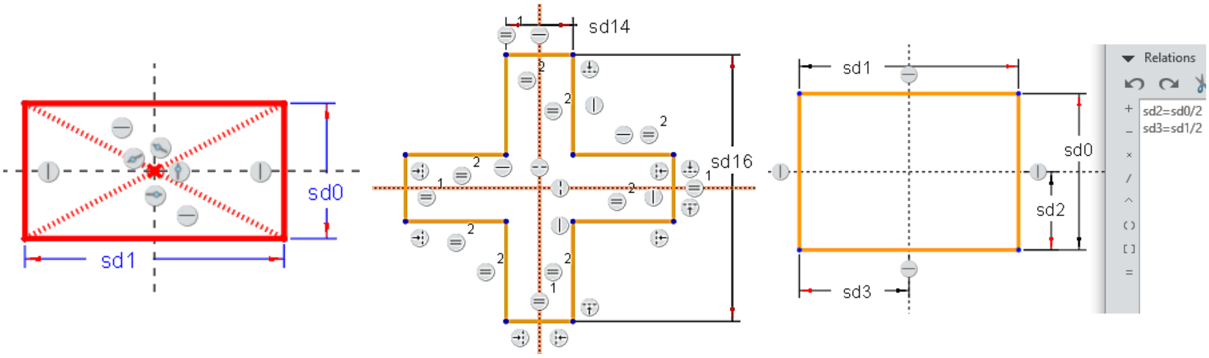

Figure 1: (a-b) Axes Centrality Rule 1: use of a pre-defined centered rectangle and palette entity. (c) Axes Centrality Rule 2: use of dimensional and ground constraints along with relations that create centrality dependencies.

Our study revealed two constraining approaches for the creation of an axes centered sketch. In the first approach, predefined centered objects are used in **56% of cases**, to directly express the SIR_XYCentered intention regularity. This approach complies with the first centrality rule, and it is mainly applied for the definition of a basic feature. Only a small number of sketches, mainly those requiring the use of palette-based entities, adopted this approach for additional features.

The second approach complies partially with the first centrality rule since designers used dimensional and ground parameters in an effort to locate the sketch around the plane axes, without creating dependencies among parameters to establish centrality. The analysis of these constraining schemas resulted in the identification of mainly two invalid constraining patterns.

Invalid Axes Centrality Case 1 – Inconsistent Dimensions: use of constraints and value parameters without dependencies. In this case centrality is invalidated when their parametric values are modified (Figure 2(a-b)).

Invalid Axes Centrality Case 2 – Wrong use of constraints: use of geometric constraints and variable parameters that indirectly result in a centered object (Figure 2(c)).

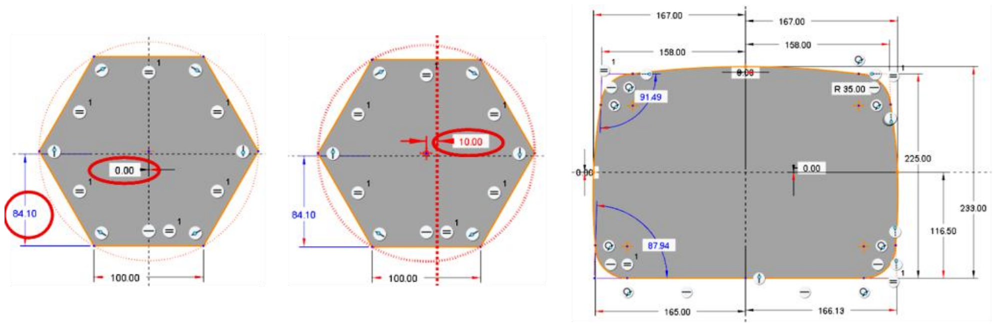

Figure 2: (a-b) Invalid Axes Centrality Case 1: the placement of the sketch around the axes is achieved using dimensional constraints that can break the centrality. (c) Invalid Axes Centrality Case 2: centrality is forced through the placement of independent dimensional and ground parameters and by setting specific values to them.

The above invalid centrality schemas create an instance of a centered sketch around X/Y axes. Therefore, the value change of any independent value breaks the centrality of the sketch. These invalid schemas are common in the majority of the examined sketches, with **40% of cases** appearing in the sketch defining the basic feature. The results of the sketch analysis regarding these cases are summarized in Table 1.

Rule/Case	*Number of sketches*	*Percentage*
Valid Axes Centered Rules		
Geometric Constraints	28	56%
Parametric Dependencies	0	0
Invalid Axes Centrality Cases		
Inconsistent Dimensions	17	34%
Wrong use of constraints	5	10%
Total Number of Sketches	**50**	

Table 1: Results of sketch analysis towards centrality constraining cases.

Partial centrality refers to positioning a section of a sketch relative to the X/Y axes. Therefore, the same axes centrality rules apply for SIR_PartialCentered intention regularity. The sketch analysis indicates that constraining patterns that are used to achieve partial centrality fall into the same invalid cases identified above.

In the context of IDI Architecture, both axes centrality rules and invalid centrality cases are associated with SIR_AxesCentered intention regularity and the corresponding SMC_AxesCentered meta-constraints. The implementation strategy of SMC_AxesCentered includes the generation of a bounding box of the sketch, the calculation of its medial horizontal and vertical axis and their alignment respectively with the X/Y axes (Figure 3(a)). In the case of partial centrality, the SMC_PartialCentered meta-constraint involves the calculation of the bounding box and its medial axes for the section of the sketch that is to be centered (Figure 3(b-c)).

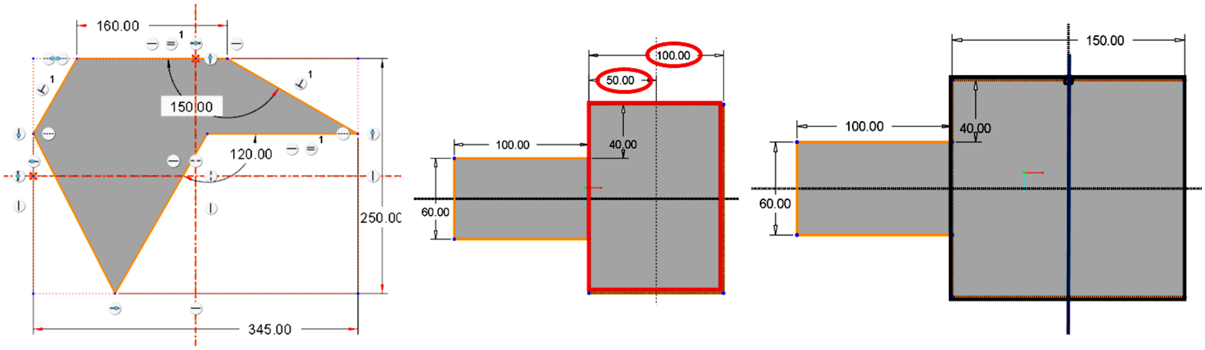

Figure 3: (a) SMC_AxesCentered meta-constraint uses the bounding box of the entire shape and places its center at the origin. (b) Partial centrality about X axis. (c) Implementation of SMC_PartialXCentered meta-constraint using the bounding box of the sketch section and its X-medial axis.

4.2 Boundary Centered Design Intent

Boundary centered (SIR_BoundaryCentered) intention regularity refers to a sketch that is centered to a pre-existing face (Figure 4) or an exterior line loop of the same feature (Figure 5(a)). Boundary centrality is considered as a different centrality type since it combines two or more sketch boundaries. In this context, we define two primary rules for the establishment of boundary centered design intent.

Boundary Centrality Rule 1 – Common References: both shapes are defined with respect to the same internal or external reference using geometric constraints or relations (Figure 4).

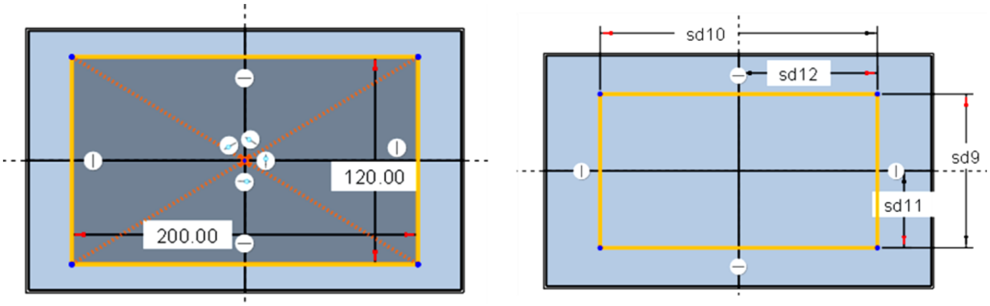

Figure 4: Boundary Centrality Rule 1 (a) both external and internal shapes are snapped to the origin of the WCS. (b) Given a axes-centered external shape, internal sketch is defined with reference to X/Y axes through dimensional parameters and relations (according to Axes Centrality Rule 2).

Boundary Centrality Rule 2 – Dimensional Relationships: the sketch is defined with reference to the geometry of the exterior boundary and relations or constraints among independent variables to create dependencies that establish centrality (Figure 5).

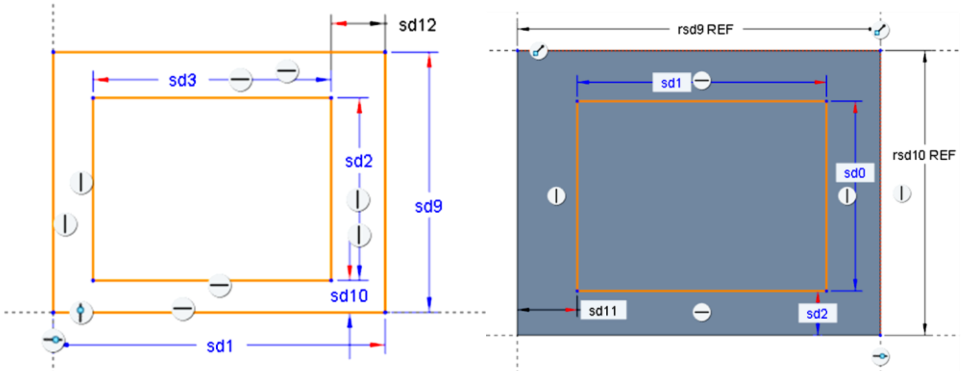

Figure 5: Boundary Centrality Rule 2 - The boundary to be centered is located either (a) within the same sketch as the exterior boundary (sd0Xsd1 parallelogram), or (b) within the boundary of a pre-existing face. In both cases, relationships must be defined between sketch variables to ensure the inner loop is centered relative to the outer face boundary.

Our study revealed two constraining approaches for the creation of a boundary-centered sketch. The first approach is used in 42% of cases and aligns with the first boundary centrality rule. Therefore, it directly expresses the SIR_BoundaryCentered intention regularity. In the second approach, as in the case of Axes Centrality design intent, designers used, in 60% of the cases, dimensional and ground parameters in an effort to center the sketch within a pre-existing external boundary, without creating dependencies among parameters to establish this centrality (Figure 6). This approach complies partially with the boundary centrality rule 2, but it is considered invalid since it creates an instance of a boundary centered sketch, with SIR_BoundaryCentered to be inconsistently or indirectly implied. In this context, we identify two common invalid constraining patterns.

Invalid Boundary Centered Case 1 – Inconsistent Dimensions: use of constraints and value parameters without dependencies. In this case boundary centrality is invalidated when their parametric values are modified (Figure 6(a)).

Invalid Boundary Centered Case 2 – Wrong use of constraints: use of geometric constraints and variable parameters that indirectly result in a boundary centered object (Figure 6(b)).

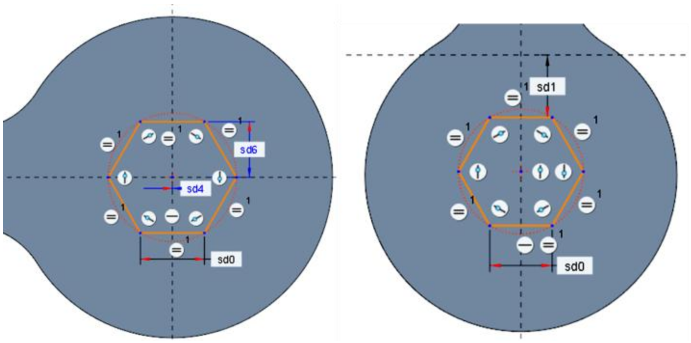

Figure 6: (a) Invalid Boundary Centered Case 1: the sketch is defined through value parameters without using constraints to establish centrality. (b) Invalid Boundary Centered Case 2: centrality is forced through the incorrect placement of independent dimensional and ground parameters.

The results of the sketch analysis regarding these cases are shown in Table 2.

Rule/Case	*Number of sketches*	*Percentage*
Valid Boundary Centered Rules		
Common References	18	43%
Dimensional Relationships	0	0%
Invalid Boundary Centered Cases		
Inconsistent Dimensions	20	48%
Wrong use of constraints	4	9%
Total Number of Sketches	**44**	

Table 2: Results of sketch analysis towards boundary-centered constraining cases.

Following a similar implementation strategy as in the case of Axes Centered intention regularity, SMC_BoundaryCentered involves the generation of the exterior and interior bounding boxes and the enforcement of the medial axes of the inner sketch to align with the medial axes of the outer sketch.

5 SYMMETRY DESIGN INTENT

Symmetry design intent, at the sketch level, is described by three intention regularities, SIR_AxesSymmetric, SIR_PartialSymmetric, and SIR_CenterSymmetric. SIR_AxesSymmetric indicates a sketch that is symmetric about one or both axes (X, Y, or both) and corresponds to the sketch meta-constraints SMC_XSymmetric, SMC_YSymmetric, and SMC_XYSymmetric.

SIR_PartialSymmetric refers to a sketch having a subset of entities symmetric about an axis. SIR_CenterSymmetric implies a symmetry about the center of a circle and is indicated directly by the design of sketch geometry.

Considering the geometric definition of symmetry about a single or both axes and the available constraining and modeling tools of parametric modeling software, we define two primary rules to establish symmetry and partial symmetry design intents about X/Y axes.

Valid Symmetry Rule 1 – Consistent Symmetric Constraints: a consistent parameterization with geometric and dimensional constraints and independent variables that preserve symmetry under value changes (Figure 7(a)).

Valid Symmetry Rule 2 – Symmetrical Dependencies: a symmetric constraint schema that is based on symmetric external dependencies (Figure 7(b)).

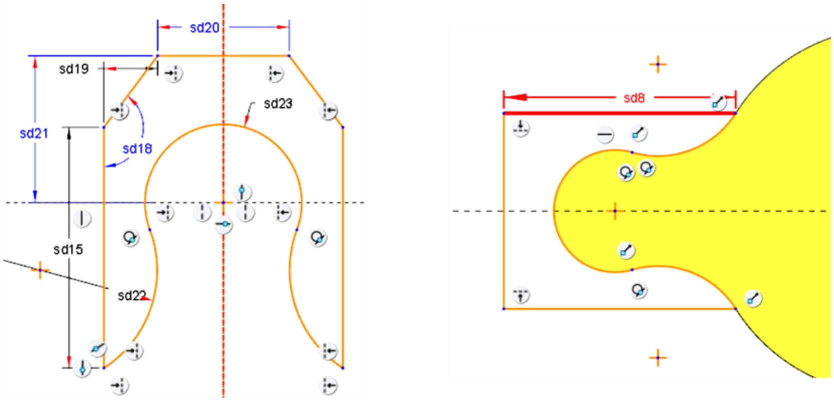

Figure 7: (a) Valid Symmetric Rule 1: a mirroring tool is employed to specify symmetric vertices in the sketch. (b) The sketch is based on a pre-existing symmetric feature (yellow noted feature in the image). In both cases, independent parametric values are used for the definition of sketch design intent.

Only 33% of the examined student sketches followed valid symmetric rules. Most students (57% of sketches) expressed symmetry by drawing redundant construction geometry and/or by applying the same constraining schema to (intended symmetrical) sketch vertices to capture shared geometric properties about the axes (Figure 8(a)). Several of these sketches followed a constraint schema that indirectly establishes symmetry (Figure 8(b)). A small proportion of sketches (10%) misplaced dimensional constraints in a symmetric constraint schema, which invalidate symmetry if their values are modified (Figure 8(c)). In this context, we categorize these invalid constraining patterns into three cases that inconsistently or indirectly imply symmetry by creating an instance of a symmetrical sketch about X/Y axes.

Invalid Symmetry Case 1 – Inconsistent Dimensions: a symmetrical arrangement of geometric entities about a single or both axes, but with an inconsistent constraining schema, with dimensions and constraints that break symmetry when their parametric values are modified (Figure 8(a)).

Invalid Symmetry Case 2 - Wrong use of Constraints: a constraining schema that indirectly establishes symmetry (Figure 8(b)).

Invalid Symmetry Case 3 - Dimension misplacement: a symmetric constraint schema that assigns dimensional parameters on non-symmetric entities, i.e., dimensions on vertices that are not constrained by symmetric constraints (Figure 8(c)).

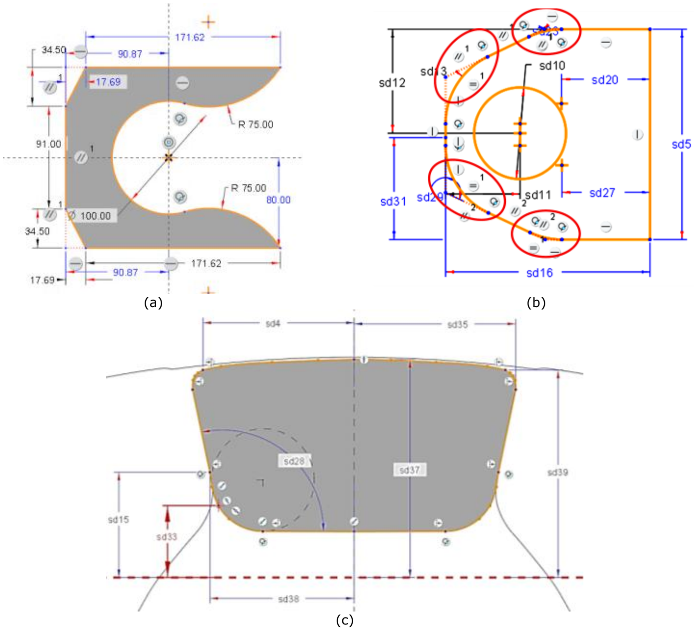

Figure 8: (a) Invalid Symmetry Case 1: dimensions and constraints are used to create an instance of a symmetric sketch which is not preserved when the parametric values are modified. (b) Invalid Symmetry Case 2: various (non-symmetric) geometric constraints are applied to sketch vertices resulting in a constraint schema that indirectly establishes symmetry. (c) Invalid Symmetry Case 3: variable parameters are placed on non-symmetric vertices that will invalidate symmetry if their values are modified.

The results of the sketch analysis regarding these cases are shown in Table 3.

Rule/Case	*Number of sketches*	*Percentage*
Valid Symmetry Rules		
Consistent Symmetric Constraints	44	34.1%
Symmetrical Dependencies	5	3.9%
Invalid Symmetry Cases		
Inconsistent Dimensions	37	28.7%
Wrong use of constraints	25	19.4%
Dimension misplacement	18	14.0%
Total Number of Sketches	**129**	

Table 3: Results of sketch analysis towards symmetry constraining cases.

Valid symmetry rules and invalid symmetry cases are associated with one of the corresponding SIR_AxesSymmetric, SIR_PartialSymmetric, and SIR_CenterSymmetric intention regularities. Therefore, when one of the above invalid constraining schemas that imply symmetry is detected and symmetry is confirmed, the corresponding meta-constraints are offered as an option. When symmetry meta-constraints are applied, the sketch will be regenerated with the minimal set of constraints appropriate for the definition of the sketch, while also establishing and preserving symmetry about a centerline aligned with one or both axes.

6 INNER/OUTER LOOP DESIGN INTENT

At the sketch level, the boundaries of the faces to be created are described as outer or inner loops. The corresponding intention regularities are SIR_FaceInnerLoop and SIR_FaceOuterLoop. SIR_FaceInnerLoop captures the intention to create a sketch that defines an inner loop to a pre-existing face (Figure 9), while SIR_FaceOuterLoop corresponds to a sketch that defines an outer loop to a pre-existing face. The establishment of these design intents is crucial, especially for the robust design of a hole or slot feature. Unlike the other design intents studied, in this case we cannot define a set of valid rules, as their establishment depends entirely on the specific geometries involved. Therefore, we introduce a valid constraining strategy that indicates a general framework for the assessment of student sketches.

Valid Inner/Outer Loop Constraining Strategy: use of geometric constraints, reference entities and relations to create dependencies that limit the values of independent variables so as to prevent intersections between inner/outer loops and the face boundaries on which they lie.

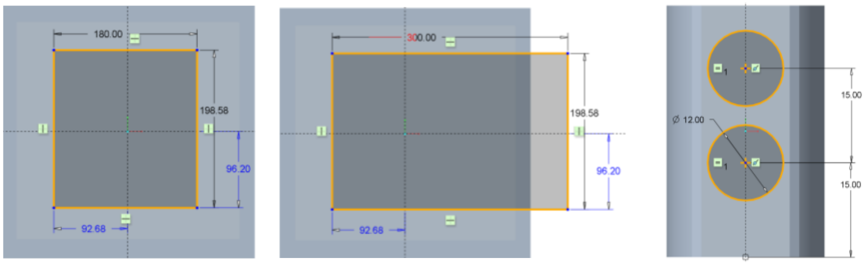

Figure 9: (a) A sketch that defines an inner loop to a pre-existing face boundary. (b) Parametric modifications that break the model coherence. (c) Multiple inner loops co-exist inside the same outer face boundary.

The sketch analysis revealed that although these intentions were clear in all designs (Figure 9(a)), none of the students defined limits to establish the design intent and avoid regeneration failures after setting wrong parametric values (Figure 9(b)). Thus, all constraining schemas were invalid towards these design intentions, resulting in the following common pattern.

Invalid Inner/Outer Case – Inconsistent Constraints: a constraining schema that breaks the model coherence when the values of variable parameters are modified (Figure 9(a)-(b)).

The recognition of this intention regularity is related to parent-child relationships. The sketch analysis also indicates common patterns in the geometry of the outer and inner loop profiles in MCAD designs, e.g., loops of identical geometry (Figure 9(a)), loops with different geometric properties, inner loops located within a portion of the outer face boundary, or multiple inner loops on the same outer face (Figure 9(c)). These arrangements designate different cases for the corresponding meta-constraints and determine the strategy for their implementation. In general, SMC_FaceInnerLoop and SMC_FaceOuterLoop meta-constraints, that correspond to SIR_FaceInnerLoop and SIR_FaceOuterLoop intention regularities, aim to ensure that face boundaries do not intersect and that initial topology is preserved. In the example shown in Figure 9 (a-b), SMC_FaceInnerLoop would prevent parametric modifications that could break model coherence, as illustrated in Figure 9(b), by setting geometric or algebraic boundaries on rectangle's width and length relative to the outer face boundary.

7 CONCLUSIONS

This paper identified sketch constraint schemas commonly employed by inexperienced designers during sketch parameterization that exhibit incorrect or incomplete implementation of design intent.

Their recognition and association with an intention regularity and the corresponding meta-constraint schema, creates a triplet of "standard constraint schema" – "intention regularity" – "meta-constraint" for the establishment of the design intent across alternative constraining strategies.

The advantages of the proposed approach are threefold. First, it enables the creation of robust and flexible 3D models by mitigating shortcomings that arise from inappropriate or incomplete constraining of sketch entities. Second, the proposed framework enhances MCAD educational practices by supporting a learning-by-activity paradigm: students learn to associate specific sets of constraints with implicit design intents, and understand the underlying semantics of constraining schemas. Finally, the proposed framework defines independent sets of constraints for each design intent, aligned with design axioms, thereby supporting the creation and coexistence of uncoupled design intents within a single sketch and enabling flexible and robust model modification.

Regarding sketch parameterization, the analysis revealed a limited understanding of the parameterization processes and the effective use of constraint tools to express design intent. It also highlighted gaps in the geometric understanding of the design process and the algebraic expression of parameter dependencies. Interestingly, since the modeling tools remain the same, inexperienced users tend to adopt similar parameterization approaches. This consistency facilitated the identification and categorization of invalid constraining cases that indirectly or inconsistently imply a design intent.

This study has certain limitations. It adopts an experimental approach restricted to inexperienced participants using a specific parametric software environment. Future research includes the incorporation of additional design environments to examine the generalizability of the findings across different design platforms. Moreover, present results can be further enhanced through a systematic evaluation framework involving standardized exercises and carefully prepared design tasks.

REFERENCES

[1] Aranburu, A; Cotillas, J; Justel, D; Contero, M; Camba, J.D.: How Does the Modeling Strategy Influence Design Optimization and the Automatic Generation of Parametric Geometry Variations?, Computer-Aided Design, 151, 2022, https://doi.org/10.1016/j.cad.2022.103364.

[2] Aranburu, A.; Justel, D.; Contero, M.; Camba, J.D.: Geometric Variability in Parametric 3D Models: Implications for Engineering Design, Procedia CIRP, 109, 2022, 383-388. https://doi.org/10.1016/j.procir.2022.05.266

[3] Barbero, B.R.; Pedrosa, C.M.; Samperio, R.Z.: Learning CAD at university through summaries of the rules of design intent, International Journal of Technology and Design Education,27, 2017, 481–98. https://doi.org/10.1007/s10798-016-9358-z

[4] Barbero, B.R.; Pedrosa, K.M.; Peña, G.K.: The importance of adaptive expertise in CAD learning: maintaining design intent, Journal of Engineering Design, 29(10), 2018, 569-595. http://doi.org/10.1080/09544828.2018.1519183.

[5] 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

[6] 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

[7] Camba, J.D.; Cosin, A.; Contero, M.: An Evaluation of Formal Strategies to Create Stable and Reusable Parametric Feature-Based 3D Models, Proceedings of the ASME 2014, International Mechanical Engineering Congress and Exposition, Volume 11, Montreal, Quebec, Canada, 2014. https://doi.org/10.1115/IMECE2014-37859

[8] Chang, K.H: Product Design Modeling using CAD/CAE: The Computer Aided Engineering Design Series, Elsevier, 2014.

[9] 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.
[10] Contero M, Company P, Salvador-Herranz G. Testing Rubrics for Assessment of Quality in CAD Modelling. Research in Engineering Education Symposium, Kuala Lumpur, Malaysia: 2013, p. 107–12.
[11] Contero, M.; Naya, F.; Pérez-López, D.; Company, P.; Camba, J.D.: A Study on Sampling Strategies to Determine the Variability of Parametric History-Based 3D CAD Models, In Proceedings of the ASME International Mechanical Engineering Congress and Exposition, 2018. https://doi.org/ 10.1115/IMECE2018-87404.
[12] González-Lluch, C; Company, P; Contero, M; Pérez-López, D; Camba, J.D.: 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, 2019, 821-841. https://doi.org/10.1007/s10798-018-9458-z.
[13] Iyer, G.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
[14] 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.
[15] 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
[16] Karadeniz, A.S.; Mallis, D.; Mejri, N.; Cherenkova, K.; Kacem, A.; Aouada, D.: DAVINCI: A Single-Stage Architecture for Constrained CAD Sketch Inference, In 35th British Machine Vision Conference (BMVC 2024), 2024, Glasgow, UK.
[17] Kyratzi, S.; Azariadis, P.: A Constraint-based Framework to Recognize Design Intent during Sketching in Parametric Environments, Computer-Aided Design and Applications, 18, 2021, 545–60. https://doi.org/10.14733/cadaps.2021.545-560.
[18] Kyratzi, S.; Azariadis, P.: Integrated Design Intent of 3D Parametric Models, Computer-Aided Design, 146, 2022, 103198. https://doi.org/10.1016/j.cad.2022.103198
[19] Kyratzi, S.; Azariadis, P.: An Ontology-based Tool for Supporting the Constraining Strategy of MCAD Objects, Computer-Aided Design & Applications, 21(4), 2024, 659–676. https://doi.org/10.14733/cadaps.2024.659-676
[20] Nerenst, T.B.; Ebro, M.; Nielsen, M. H.; Eifler, T.; Nielsen, K.L.: Parametric CAD Modeling: New Principles for Robust Sketch Constraints, Computer-Aided Design and Applications, 20(1), 2023, 56-81. https://doi.org/10.14733/cadaps.2023.56-81
[21] Rynne, A.; Gaughran, W.: Cognitive modeling strategies for optimum design intent in parametric modeling, American Society for Engineering Education, 18, 2007, 12.366.1-12.366.15. https://doi.org/10.18260/1-2--2651.
[22] Otey, J.; Company, P.; Contero, M.; Camba, J.D.: 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