Title:

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

Authors:
Sofia Kyratzi, skiratzi@aegean.gr, University of the Aegean
Philip Azariadis, fazariadis@uniwa.gr, University of West Attica
Zoi Koutkalaki, zkoutkalaki@uniwa.gr, University of West Attica

Keywords:
Parametric modeling, Design Intent, Sketch-based modeling, sketch constraining

DOI: 10.14733/cadconfP.2026.87-92

Introduction:

In a typical MCAD parametric, feature-based 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 [1]. The geometry of the sketches and features is defined with the use of parameters, constraints, and relations that 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 [2-3]. Design intent refers to all design decisions that are taken during the modeling process so that the model to be successfully regenerated without rendering to inconsistencies [4]. A challenge of parametric design process is that there are different modelling strategies and constraining schemas for building the geometry of the 3D model, resulting in multiple design alternatives for approaching product performance and design requirements [6]. 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 [1][3][6-8] 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 to reduce subjectivity in parameterization. In this research path, the IDI Architecture [9-11] 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 so to identify different patterns of constraints that imply the same design intent. 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 homogenous 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 an AI training module for recognizing and implementing design intent in 2D sketches.

The Integrated Design Intent (IDI) Architecture – Background:

The IDI Architecture is described in detail in [9-11]. 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 build an integrated design intent from sketch level to the model level.

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 [10], 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 loops.

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 referent design intention [6]. The IDI Architecture supports this approach with the pairs "meta-constraint" – "intention regularity". Meta-constraints comply with the design axioms of Independence and Information [6]. 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.

Detection and Implementation of Sketch Intention Regularities:

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 his/her 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 loop design. The study was conducted on a set of 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 and no explicit guidance on sketch-level design intent was provided.

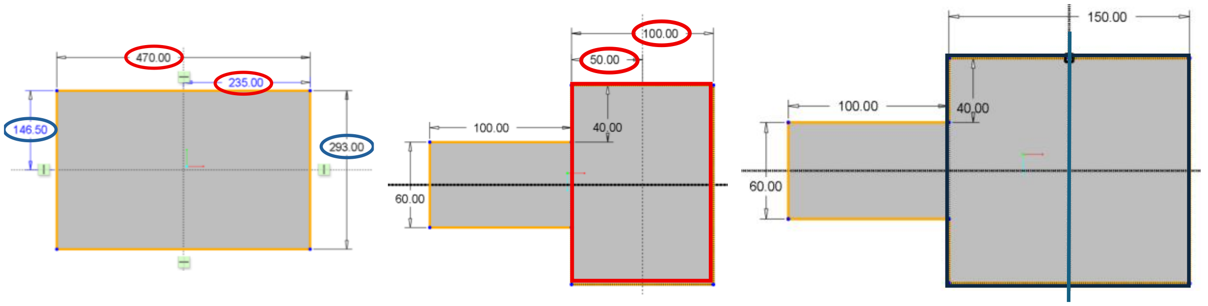
Sketches parameterization and constraining schemas were analyzed using geometric modeling criteria, as well as criteria related to the expression and inheritance of geometric design intent at the three design levels. Our study identified that similar patterns of constraints are employed to achieve specific geometric properties in sketches. These patterns are further discussed in the following subsections. 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 under changes in parametric values.

In the context of IDI Architecture [10], 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.

Centrality Design Intent

According to IDI Architecture [10], centrality design intent, at the sketch level, is described by two intention regularities, 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.

Our study revealed two constraining approaches for the creation of an axes centered sketch. In the first approach, a centered parallelogram tool is used, which directly implies the SIR_XYCentered intention regularity. In the second approach, designers used dimension and ground parameters to locate the sketch around the plane axes (Fig.1 (a)). This constraining schema, that creates an instance of a centered sketch around XY axes, revealed to be common in most of the sketches implying an SIR_XYCentered intention regularity. The same constraining pattern appeared around one axis, implying respectively SIR_XCentered or SIR_YCentered regularity. The association of these constraining patterns with SIR_AxesCentered triggers the corresponding SMC_AxesCentered meta-constraints. For the implementation of SMC_AxesCentered, a bounding box of the sketch is generated where its medial horizontal and vertical axis are aligned respectively with the X/Y axes (Fig. 1(b)). For the implementation of SMC_BoundaryCentered the medial axes of the outer sketch aligns with the medial axes of the inner sketch. For the implementation of SMC_PartialCentered the medial axes of the selected part of the sketch are aligned respectively with the X/Y axes.



(a) (b) (c)

Fig. 1: A constraint arrangement that indicates (a) centrality about XY axes, (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.

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. The analysis of the student sketches revealed that inexperienced users tend to express symmetry by drawing similar geometric entities with equal dimensions (within a threshold), and by applying the same constraining schema to capture shared geometric properties about the axes (Fig. 2).

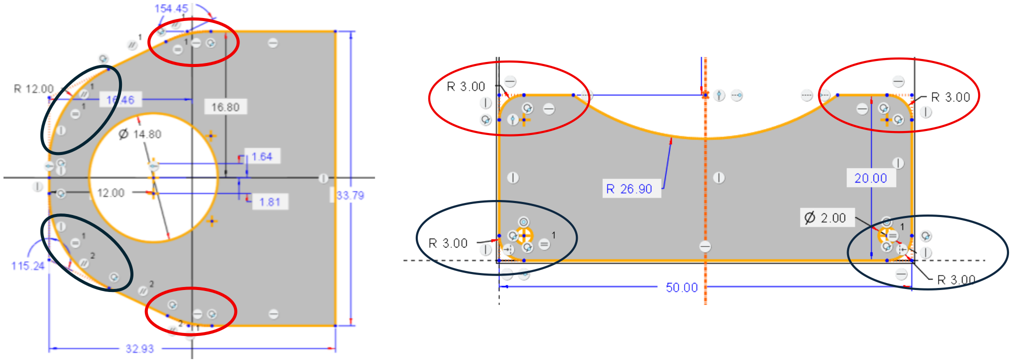
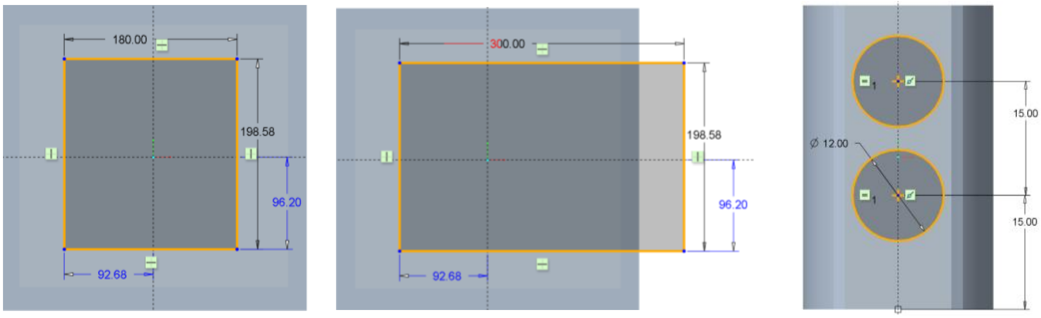


Fig. 2: Circles indicate equivalent sets of constraints and their associated entities that appear around an axis, forming constraining patterns that indicate symmetry design intent.

When constraining patterns that imply symmetry are 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.

Inner loops Design Intent

During sketching, face boundaries are described by outer or inner loops. The corresponding intention regularities, at the sketch level, are SIR_FaceInnerLoop and SIR_FaceOuterLoop. SIR_FaceInnerLoop captures the intention to create a sketch that defines an internal loop to a pre-existing face, while SIR_FaceOuterLoop corresponds to a sketch that defines an outer loop to a pre-existing face.



(a) (b) (c)

Fig. 3: (a) An internal loop that defines a 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 (Fig. 3(a)), none of the designer set limits to establish the design intent and avoid regeneration failures after setting wrong parametric values (Fig. 3(b)). The recognition of this intention regularity is related to parent-child relationships. The sketch analysis indicates common patterns in the geometry of the outer and inner loop profiles in MCAD designs, e.g., loops of identical geometry (Fig. 3(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 (Fig. 3(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.

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 triplet enhances MCAD educational practices by supporting a learning-by-activity paradigm: students learn to associate specific sets of constraints with implicit design intents and to understand the underlying semantics of constraining schemas. Finally, the triplet 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 model modification.

References:

[1] 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.

[2] 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

[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] 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

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

[6] 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

[7] 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.

[8] 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.

[9] 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

[10] 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

[11] 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