Semantic Mates: Intuitive Geometric Constraints for Efficient Assembly Specifications

Fabian Wildgrube*, Alexander Perzylo*, Markus Rickert*, Alois Knoll†

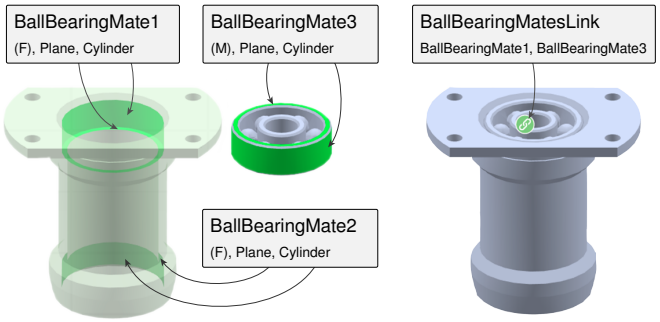
***Abstract*— In this paper, we enhance our knowledge-based and constraint-based approach of robot programming with the concept of Semantic Mates. They describe intended mechanical connections between parts of an assembly. This allows deriving appropriate assembly poses from the type of connection and the geometric properties of the involved parts. The paper presents an ontology-based representation of Semantic Mates that is used to augment object models with additional information regarding their potential use in an assembly. Such semantically annotated object models can be used in our instruction framework to program a robot to perform assembly tasks through simple drag-and-drop operations in a graphical user interface.**

We conducted a user study with 21 participants in order to evaluate the efficiency and usability of the Semantic Mates concept based on a use-case from the domain of mechanical assembly. Across different experience levels in robotics, the participants achieved a significantly faster workflow and improved perceived usability compared to the manual specification of constraint-based assembly operations.

I. Introduction

Although use of robot-based automation is increasing, many small and medium-sized enterprises (SMEs) face a number of challenges when it comes to programming industrial robots for their purposes. This is mainly due to small lot production, in which the efficient instruction of robots is crucial for their financial viability. Many use-cases require large lot sizes to amortize the lengthy and thus costly workflow of state-of-the-art robot programming solutions. However, many SMEs produce bespoke products for their customers in small quantities. This issue does not solely affect SMEs. In some larger companies, production at least partially follows the trend of mass customization. Moreover, creating new and adapting existing programs during changeover requires experienced operators that need to be extensively trained. These conditions severely hinder the integration of robots in such manufacturing processes [1], [2].

Our previous work on an instruction paradigm for industrial robots addresses common robot programming issues by employing a product-centric approach. It aims to drastically reduce the required expertise in robotics and increase the efficiency of robot programming [3], [4]. In this approach users work with CAD models of mechanical parts and define their assembly poses through common CAD relations, mainly by specifying geometric constraints between vertices, edges, or faces of involved parts. The resulting description of an assembly goal is interpreted by the robot system in order to automatically derive the required robot motions [5]. However, for products comprised of numerous individual parts, the definition of assemblies through a multitude of geometric constraints can still be a time-consuming and cumbersome process.



(a) (b)

Fig. 1: Visualization of (a) two assembly parts that have been annotated with compatible connection types for ball bearing-related Semantic Mates, (b) the established part connection based on matching Semantic Mate types that allow the bearing to automatically snap to the correct position.

Additionally, geometric constraints require the user to think in spatial terms and may produce geometrically valid constellations of parts that do not adhere to the semantics of real-world part connections. This leaves geometric and semantic error checking to the user, which causes a certain cognitive load. Increased cognitive load has been found to lead to lower user satisfaction [6] and should therefore be avoided.

To address these challenges, we enhanced our assembly specification workflow to support the automatic arrangement of parts in an assembly based on the semantics of their mechanical connection types. By representing part connections through their semantic meaning rather than only related geometric constraints, such mechanical connections can be easily established while their formal requirements are automatically considered. While the expected reuse of mechanical connection models is highest for standardized parts, e.g., screws or ball bearings, our extensible approach may be used for arbitrary geometries. We refer to a geometrically and semantically compatible pair of connecting geometries as *Semantic Mates*. Internally, Semantic Mates are still represented by a set of geometric constraints between edges or faces of two objects that define their relative displacement in the desired assembly configuration.

*Fabian Wildgrube, Alexander Perzylo, and Markus Rickert are with fortiss, An-Institut Technische Universität München, Munich, Germany.

†Alois Knoll is with Technische Universität München, Munich, Germany. Correspondance should be directed to perzylo@fortiss.org

A Semantic Mate abstracts the underlying low-level geometric requirements away from the user and replaces them with a meaningful, human-understandable symbolic connection. At the same time, it formally encodes the technical details of how the represented connection type can be geometrically established (Fig. 1).

This paper introduces a semantic description language based on the Web Ontology Language (OWL)[1] for modeling concepts related to Semantic Mates. Section III shows how these concepts are used to annotate our semantic object models [7] with their related semantic connection types and how the additional knowledge can be used to simplify the specification of assembly tasks for human operators. The design and results of our user study that investigated the efficiency and usability of our novel robot programming approach are explained in Section IV and discussed in Section V.

II. Related Work

Offline programming (OLP) minimizes robot downtime through specialized software that enables users to asynchronously create robot programs without the need of being physically connected to the robot. Due to high costs and complexity of professional OLP software, such as Dassault Systèmes Delmia or Siemens Tecnomatix solutions, offline programming is mainly used in production environments with large lot sizes [8].

More lightweight OLP approaches have been subject of research in the past years. [9] and [10] developed similar approaches to specify weld seams for a welding robot by defining motion paths directly within common CAD software tools. [11] proposed a more generalized approach, where tool poses are defined within a 3D representation of the workcell in order to extract a robot path that can be automatically transformed into robot code. While these approaches simplify the programming process compared to teach pendant-based workflows, they still require the user to have extensive knowledge in the field of robotics. They all rely on the manual selection of sequences of low-level robot tasks.

Recent approaches to simplifying robot task programming rely on the abstraction and composition of low-level robot movements into higher-level skills [12], [13], [14]. These programming solutions are centered around arranging motion or action-related building blocks in graphical interfaces. This dramatically improves the efficiency of instructing robots while reducing the flexibility of the robot system based on the available set of motion and action primitives. Reuse of such programs in different robot cells—other than the one they were explicitly programmed for—is typically not possible. Furthermore, the user still needs to have a basic understanding of the low-level robot motions in order to use the abstract skills effectively.

In contrast to these methods, our instruction paradigm is about describing processes and products, but not the required motions or control structures of robot programs. For executing a manufacturing task, process and product models are interpreted in order to automatically generate corresponding robot programs. A human operator, who is an expert in a particular application domain, shall be enabled to input a process description without the need of thinking about how a robot system would need to behave for performing associated tasks. Once process and product models are created, they can be reused in multiple workcells with different hardware. As long as provided manufacturing resources offer required capabilities, the associated tasks of a manufacturing process can be assigned to compatible resources [15]. In the assembly domain, this includes specifying assemblies with the help of geometric constraints.

Since constraints offer numerous advantages for assembling parts compared to manually aligning them, all major CAD tools have incorporated them into their graphical user interfaces. However, these generic geometric relations require multiple constraints to model complex assemblies and do not encode the semantics of part connections. Some CAD tools, e.g., Autodesk Inventor, Autodesk Fusion360, and SolidWorks, support some form of semantics in their constraint interfaces.

In particular, SolidWorks offers interesting features regarding the intuitive specification of geometric constraints. Its *smart mates* offer built-in drag-and-drop snapping of single faces or edges of two parts based on matching their geometric parameters. *Magnetic mates* let the user define universal connection points for a part that can snap to connection points defined in another part. As these connection points do not have types, there is no way to add any further meaning to them. With *mate references*, SolidWorks offers a way to encode custom connection types for parts, which are then used to snap them together in an assembly. Mate references need to be preconfigured on single parts by giving each mate reference a name and specifying up to three related geometric entities (i.e., vertices, edges, or faces) of the part. For each entity, a constraint type and its alignment are defined.

While this approach is similar to the *Semantic Mates* concept introduced in this paper, SolidWorks' mate references do not encode the actual semantics of connection types. When a mate reference has been activated, it is replaced by its predefined set of geometric constraints, but the information on how these constraints have been created is lost.

III. Semantic Mates

Semantic Mates model the real-world semantics of mechanical connections and enable related software tools to intuitively establish these connections while respecting associated semantic constraints. Vertices, edges, and faces of assembly parts are annotated with specific ontological concepts that encode the functionality and behavior of connectable parts in a machine-understandable format. Our robot instruction framework uses these annotations to automatically derive required assembly poses from the type of connection and the geometries of the involved parts. The calculated poses are then used to parameterize robot operations that

[1] https://www.w3.org/TR/owl2-primer/

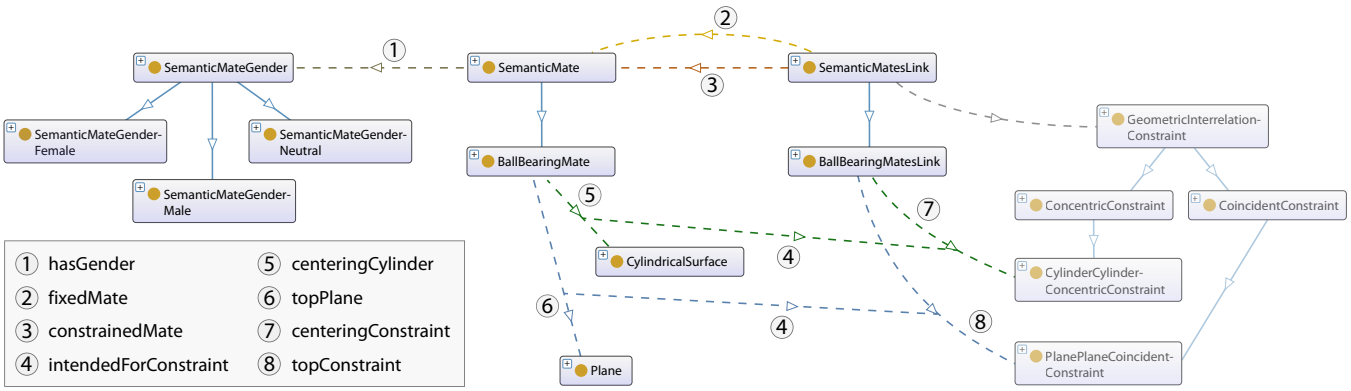


Fig. 2: Visualization of an excerpt of the Semantic Mates ontology. Partially transparent classes are imported from our OntoBREP ontology that defines geometric entities and constraints [7].

are supposed to perform the desired assembly. Thus, the user can create assembly specifications by simply dragging and dropping parts in a graphical user interface. Parts that provide compatible connection types are highlighted and snap into the correct position and orientation as soon as they are dragged close to a target area. A video that shows the workflow of defining an assembly task based on the Semantic Mates concept is available online.[2]

A. Robot instruction framework

The Semantic Mates concept is based on our OntoBREP ontology. OntoBREP's concepts can be used to describe 3D objects in a boundary representation, in which geometries are represented through mathematically meaningful definitions, e.g., cylindrical or planar surfaces. It further defines various geometric constraints that can be applied between different geometric entities. For representing Semantic Mates, OntoBREP has been extended by combining constraints into groups that can capture the higher-level semantics of mechanical connection types between assembly parts.

Our knowledge-based robot instruction approach abstracts away robot specific knowledge from the user by explicitly representing it in ontologies. In the back end, a graph database is used to persistently store the semantic descriptions of processes, manufacturing resources, and objects. They can be read and updated through SPARQL[3] queries from an Angular[4]-based front end that is hosted on a touch-enabled tablet. Fig. 3 gives an overview of the system's architecture.

The individual building blocks of the overall system are explained in detail in our previous publications. Our OntoBREP ontology and the semantic representation of geometric constraints are introduced in [7]. How these object models are used to parameterize our semantic process models is shown in [3]. The framework responsible for solving geometric interrelational constraints and real-time control of a robot that performs an associated assembly is described in [5].

[2] https://youtu.be/o5EiAut3N2c
[3] https://www.w3.org/TR/sparql11-overview/
[4] https://angular.io/

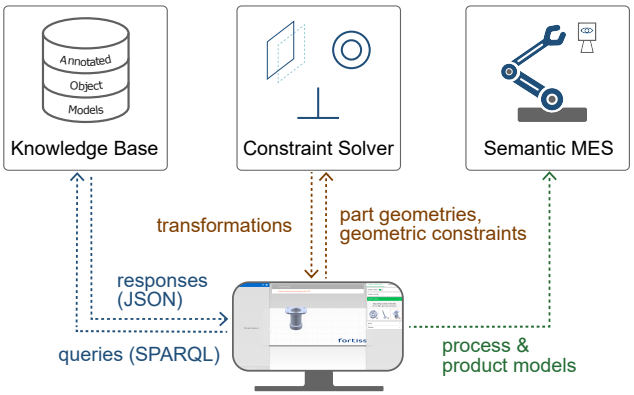


Fig. 3: Overview of the instruction framework. The user interacts with a touch GUI for intuitively defining geometric constraints through Semantic Mate-based drag-and-drop operations. Annotated semantic object models reside in a knowledge base that is accessed by the GUI through SPARQL queries. The established constraints are translated to relative transformations by a constraint solver. The final process description is sent to a manufacturing execution system that interprets the process and product models for performing the assembly task.

In [4], we present the first two iterations of our graphical user interface for interacting with our robot instruction system. It was extended to support the Semantic Mates concept as described in this paper.

B. Semantic Mates ontology

Mechanical connections are established between two physical objects and can be either symmetrical or asymmetrical. They can be geometrically described by individual faces, edges, and vertices on the related parts and constraints that link these geometric entities to each other. The Semantic Mates ontology captures these characteristics by defining three basic concepts and their relations, as visualized in Fig. 2.

A connection type is represented by a *SemanticMatesLink*, which refers to two specific *SemanticMate* individuals when

```
Prefix: sm: <http://www.fortiss.org/kb/semanticmates.owl#>
Prefix: bb: <http://www.fortiss.org/kb/cad/bearing.owl#>

Individual: bb:BallBearingMate1
  Types:
    sm:BallBearingMate
  Facts:
    sm:centeringCylinder bb:CylindricalSurface1,
    sm:hasGender sm:SemanticMateMale,
    sm:topPlane bb:Plane1
```

Fig. 4: Excerpt of OWL description of a *BallBearingMate* individual in Manchester syntax.

```
Prefix: brep: <http://www.fortiss.org/kb/ontobrep.owl#>
Prefix: sm: <http://www.fortiss.org/kb/semanticmates.owl#>

ObjectProperty: sm:topConstraint
  SubPropertyOf:
    sm:ballBearingMateConstraint
  Domain:
    sm:BallBearingMatesLink
  Range:
    brep:PlanePlaneCoincidentConstraint

ObjectProperty: sm:topPlane
  Annotations:
    sm:intendedForConstraint sm:topConstraint,
  SubPropertyOf:
    sm:ballBearingMateFace
```

Fig. 5: Definition of OWL properties *topPlane* and *topConstraint* in Manchester syntax. *topPlane* is used to link required geometries to *BallBearingMate* individuals and *topConstraint* refers to required geometric constraints from a *BallBearingMatesLink* individual. The annotation property *intendedForConstraint* encodes that the *Plane* individual linked through the *topPlane* property shall be used for parameterizing the *PlanePlaneCoincidentConstraint* linked by the *topConstraint* property.

the connection is established. A *SemanticMate* identifies the geometric entities of a particular part that are needed for its specific connection type. The individual geometric entities are categorized through object properties that encode their meaning within the connection type, e.g., *topPlane* or *centeringCylinder*. These object properties are annotated with OWL annotation property *intendedForConstraint* to encode the information of which kind of geometric constraint the referenced geometries shall be used for. A *SemanticMate* defines its *SemanticMateGender* to be either neutral within a symmetric connection or male/female within an asymmetric connection type. Each *SemanticMate* further declares which type of *SemanticMatesLink*, i.e., which type of connection, it belongs to. Two *SemanticMate* individuals of the same type and compatible genders can form a connection through a *SemanticMatesLink* individual. In order to encode all relevant information for calculating target poses, a *SemanticMatesLink* specifies the geometric constraints that are required to be respected. The constraint individuals link particular vertices, edges, or faces of the OntoBREP representation of the two involved parts. Possible types of constraints include parallelism, concentricity, and coincidence.

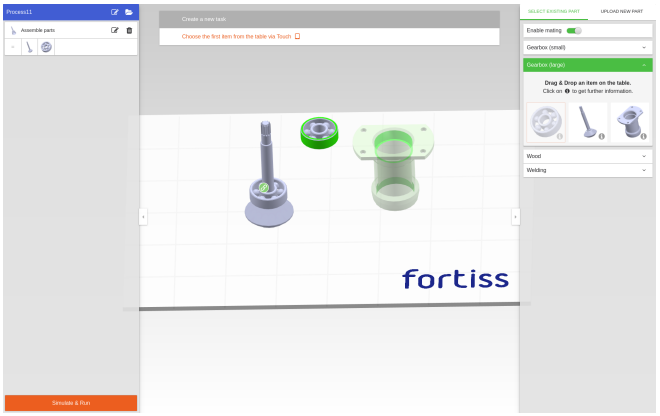


Fig. 6: Screenshot of the robot instruction GUI. New parts are dragged from the part library on the right into the virtual assembly area in the center. Their Semantic Mates are visualized through the highlighting of faces or edges that are referenced from the constraints associated with a particular Semantic Mate type. Previously defined assembly steps are listed on the left.

For each new type of connection that shall be represented, the *SemanticMatesLink* and *SemanticMate* taxonomies have to be extended by adding corresponding subclasses to the existing concepts in the Semantic Mates ontology. *SemanticMate* individuals can then be added to the semantic object models of assembly parts (Fig. 4). *SemanticMatesLink* individuals are created at runtime, when a user assembles parts within our graphical robot instruction framework. Since complex parts might be able to connect to various other parts in different ways, Semantic Mates of the same or different kinds may be asserted within a part's object model.

The OWL class and property structure for a link between a ball bearing and a pipe-like structure, as shown in Fig. 1, is depicted in Fig. 2. *BallBearingMate* individuals link to both a cylindrical and a planar face that are relevant for the specific connection type. Those links are established through object properties *centeringCylinder* and *topPlane*. When two Semantic Mates are connected in the GUI, a *BallBearingMatesLink* individual is created and all relevant constraints between the respective faces are established. The *CylinderCylinderConcentricConstraint* and *PlanePlaneCoincidentConstraint* are linked from the *BallBearingMatesLink* individual via the *centeringConstraint* and *topConstraint* object properties, respectively.

Fig. 4 shows the necessary annotations for endowing a part with a *BallBearingMate* connection type. The specification of OWL property *topConstraint* that links the geometric constraint that needs to be adhered to when establishing a *BallBearingMatesLink* connection type is presented in Fig. 5. Object property *centeringConstraint* is defined accordingly.

C. Robot instruction workflow based on Semantic Mates

We extend our previous implementation, which was based on manually specifying geometric constraints for the description of assembly tasks, with the feature to assemble parts

```
PREFIX sm: <http://www.fortiss.org/kb/semanticmates.owl#>
PREFIX bb: <http://www.fortiss.org/kb/cad/bearing.owl#>

SELECT ?mate ?gender WHERE {
  bb:BallBearing1 sm:hasSemanticMate ?mate .
  ?mate sm:hasGender ?gender .
}
```

Fig. 7: Simplified SPARQL query that retrieves all *SemanticMate*s for the assembly part *BallBearing1* including the mates' genders.

```
PREFIX sm: <http://www.fortiss.org/kb/semanticmates.owl#>
PREFIX bb: <http://www.fortiss.org/kb/cad/bearing.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?geometry ?role ?constraint ?ctype WHERE {
  bb:BallBearingMate1 ?role ?geometry .
  ?role rdfs:subPropertyOf sm:semanticMateFace .
  ?role sm:intendedForConstraint ?constraint .
  ?constraint rdfs:range ?ctype .
}
```

Fig. 8: Simplified SPARQL query that retrieves all geometric entities associated with the given Semantic Mate *BallBearingMate1* including their roles within the Semantic Mates and associated constraint types.

via a simple drag-and-drop mechanism that is enabled by the Semantic Mates concept. In order to integrate Semantic Mates into our framework, we annotate the relevant semantic object models using the ontology concepts introduced in Section III-B. These extended object models are stored in the system's knowledge base.

As soon as the user drags a part from the part library in the GUI (Fig. 6), the SPARQL query from Fig. 7 is used to obtain the information on all provided Semantic Mates of that part, i.e., all *SemanticMate* individuals that are referenced from its semantic object model. For each of its *SemanticMate*s all associated geometric entities are retrieved from the knowledge base using the SPARQL query from Fig. 8. The OWL object properties bound to the *?role* and *?constraint* variables serve as identifiers for the role of referenced geometries within a connection type. The OWL class bound to variable *?ctype* represents the type of constraint the linked geometry shall be used for. The corresponding geometries of identified Semantic Mates are then highlighted, in order to inform the user that Semantic Mates are available for the part currently being dragged. All objects that have already been placed in the virtual assembly area are checked for matching Semantic Mates, i.e., *SemanticMate* individuals of the same type and with a compatible gender. Relevant subgeometries of these parts are highlighted as well, as can be seen in Fig. 6. Thus, the user is presented with potential target poses for the new part. Once the part is dragged close to a potential target pose, it will snap into place.

The role of specific geometries of a part are resolved through the *intendedForConstraint* annotation property. Once the constraint types are known, the set of geometry pairs and

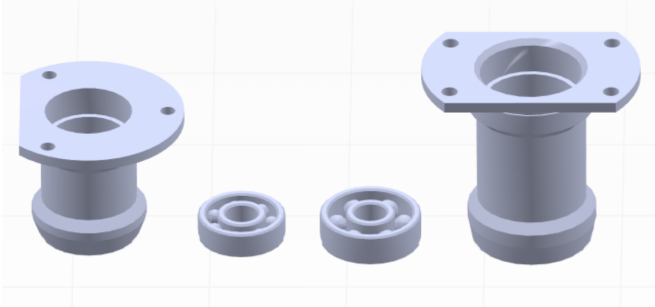


Fig. 9: Two variants of the same ball bearing with different diameters and matching target parts.

their associated constraint types are sent to the framework's constraint solver component via a middleware layer. To calculate transformation matrices that satisfy the constraints, the constraint solver expects one part to be fixed in space and the second part to be free to move to a new position. In our case the part being dragged by the user is the latter one. The constraint solver returns a transformation relative to the fixed part, which is applied to the dragged part, causing it to move to a pose that satisfies all given constraints. From the user's perspective, the dragged part simply snaps into its proper place to form a semantically correct connection with the fixed part.

Dropping the part in this configuration leads to the assertion of a *SemanticMatesLink* individual that links the two matching Semantic Mates to each other. Alternatively, if the user keeps dragging, the part unsnaps and all matching faces are highlighted again. A generated *SemanticMatesLink* individual gets persistently stored in the knowledge base and a robot task is created that represents the robot action needed to assemble the parts and is parameterized with the assembly pose derived from the Semantic Mates. This task is added to the list of process steps and an icon is placed next to the parts to indicate that they have been connected through a *SemanticMatesLink*. Clicking on the icon reveals a context menu through which the connection can be released again. This would cause the parts to separate and the link as well as the robot task associated with the link to be deleted.

Parts that are connected through a *SemanticMatesLink* can be picked up and handled as one, and used for further assembly operations.

D. Automatic selection of Semantic Mate variants

Given an assembly part library in which different variants of a part are stored, the Semantic Mate concept can be extended to enable the automatic selection of a particular variant that fits the Semantic Mate of the target part. For instance, the outer and inner cylindrical surfaces of the ball bearing in Fig. 1 have specific radii. Fig. 9 shows two explicit variants of the same type of ball bearing that differ in their radii. These surface parameters are represented in the parts' OntoBREP models and can be queried by the robot system. By grouping all available variants of the bearing together into a higher-level part entry in the part library, all of them

```
PREFIX sm: <http://www.fortiss.org/kb/semanticmates.owl#>
PREFIX bb: <http://www.fortiss.org/kb/cad/bearing.owl#>
PREFIX mp: <http://www.fortiss.org/kb/cad/mech-pipe.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX brep: <http://www.fortiss.org/kb/ontobrep.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?partBB WHERE {
  ?partMP sm:hasSemanticMate mp:BallBearingMate1 .
  mp:BallBearingMate1 ?role ?geometryMP .
  ?partBB sm:hasSemanticMate ?bearing .
  ?bearing ?role ?geometryBB .
  ?role rdfs:subPropertyOf sm:semanticMateFace .
  ?geometryBB brep:radius ?radiusBB .
  ?geometryMP brep:radius ?radiusMP .
  FILTER (
    xsd:double(?radiusBB) = xsd:double(?radiusMP)
        && ?partBB != ?partMP
  )
}
```

Fig. 10: Simplified SPARQL query that identifies a particular variant of a ball bearing that fits the geometric properties of a target part's Semantic Mate (*BallBearingMate1*).

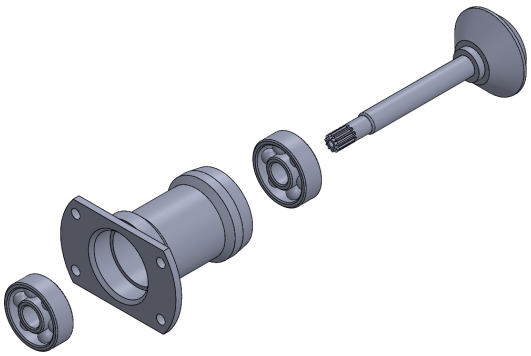


Fig. 11: Exploded view of assembly that the user study is based upon. It involves the assembly of four parts in three steps.

can be automatically considered when dragging the multi-variant bearing from the library. The highlighting of potential locations on the target part is then based on the union of compatible variants of the bearings' Semantic Mates. When the bearing is dropped onto a specific Semantic Mate location on the target part, the matching variant is automatically identified (Fig. 10) and added to the assembly specification.

IV. User Experience Evaluation

A user study with 21 participants (aged between 21 and 40 years, 6 females, 15 males) was conducted using a crossover-repeated measures (within-subjects) design to compare the usability of constraint-based assembly specifications to our Semantic Mates-based approach. Users had to program the same gearbox assembly task (as introduced by [4] and depicted in Fig. 11) twice by interacting with a real robot workcell (Fig. 12). Quantitative GUI interaction data was recorded for both conditions, **C1** (without Semantic Mates) and **C2** (with Semantic Mates). After each trial, participants were asked to take a questionnaire to capture their immediate reactions to the interaction. For this, the standardized

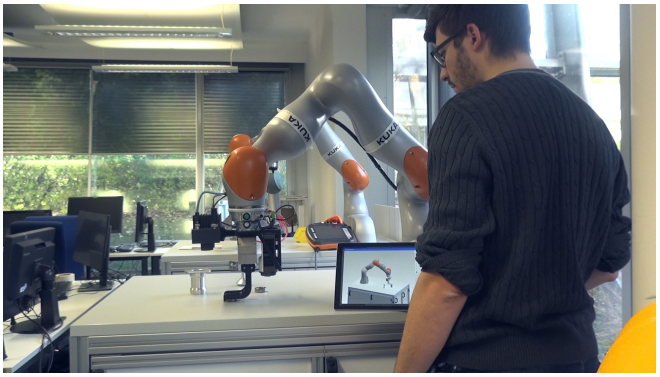


Fig. 12: Cognitive robot workcell used in our user study. The workcell features an industrial robot arm with a mounted multi-tool comprised of a parallel gripper, a 3D camera for object recognition, and an RGB projector for giving feedback to the operator. The robot is instructed by interacting with a GUI that is hosted on a touch-enabled tablet.

User Experience Questionnaire (UEQ)[5] was selected, which measures six different dimensions of usability. The UEQ has been used by academia and industry in multiple domains to evaluate user experiences for a wide variety of applications. The results of these evaluations show its consistency and validity [16]. Additionally, results of the UEQ can be interpreted more easily than a custom questionnaire due to the large number of comparable UEQ data sets.

Usability has been shown to be linked to the quality of users' mental models of an application [17]. Assembling objects in various ways is a concept that is omnipresent in our daily lives. Object manipulation follows the laws of physics, for which humans develop a very intuitive understanding [18]. In contrast to this, geometric constraints are less intuitive mathematical concepts. They may represent spatial configurations of assembly parts that can not exist or persist in the real world due to collisions or the influence of gravity. As a result, greater mental effort is required for using them to describe assemblies, which reduces user satisfaction and increases error rates [17]. Representing part connections through their semantic meaning rather than related geometric constraints addresses these issues and leads to the following hypotheses:

- H1 Users are significantly faster in C2.
- H2 Users produce significantly fewer errors in C2.
- H3 Usability of C2 is significantly higher compared to C1.

The results concerning these hypotheses are presented in the following sections.

A. Semantic Mates are more efficient (H1)

In condition C2, using Semantic Mates for assembly specifications, all participants were able to complete the given task, whereas in condition C1 only two-thirds of the participants managed to do so. In this section, only these 14 successful participants are considered. All of them were faster

[5] https://www.ueq-online.org

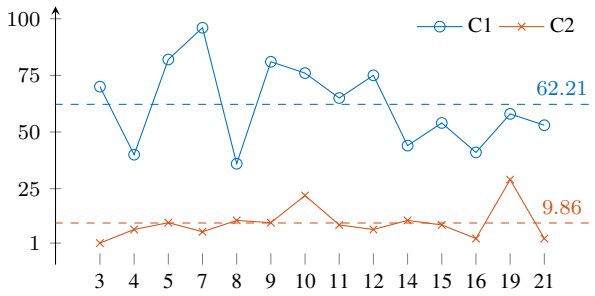


Fig. 13: Number of clicks of 14 participants who completed the task in both conditions.

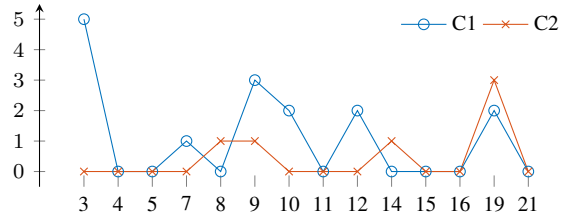


Fig. 14: Number of errors of 14 participants who completed the task in both conditions.

in condition C2. Thus, the means of task completion times differ strongly between the two conditions. In condition C1, completion times ranged between 49.87 s and 401.88 s, with the mean completion time being 302.49 ± 122.14 s. In condition C2, completion times ranged from 27.45 s to 354.88 s with an average of 91.71 ± 81.22 s, making the version with Semantic Mates on average three times faster than the version without them. The average difference in completion time between the two conditions was 210.78 ± 120.31 s. A two-tailed Welch's t-test for paired data shows that the difference between the completion time means is statistically significant at a significance level of 5 % ($p = 1.934e^{-5} < 0.05$).

Exactly half of the 14 participants, who completed the task in both conditions, had no experience in robotics, while the other half claimed their experience to be at beginner or intermediate level. As can be seen in Table I, the participants with robotics expertise were, on average, faster in condition C1 compared to those without experience. Interestingly, in condition C2 the participants without experience were faster, on average, than those with robotics expertise. However, comparing these two groups with two-tailed t-tests shows that there are no significant differences between the two groups in either of these conditions (C1: $p = 0.223 > 0.05$; C2: $p = 0.728 > 0.05$).

B. Users still produce errors (H2)

In order to quantify programming mistakes, we recorded the number of clicks and number of errors (deleting a part, deleting a task, deleting a constraint, or canceling the constraint dialog in condition C1). As depicted in Fig. 13, the number of clicks differed significantly between the two conditions (C1: average of 62.21 ± 18.48 clicks; C2 only 9.86 ± 7.45 clicks on average. A two-tailed t-test shows

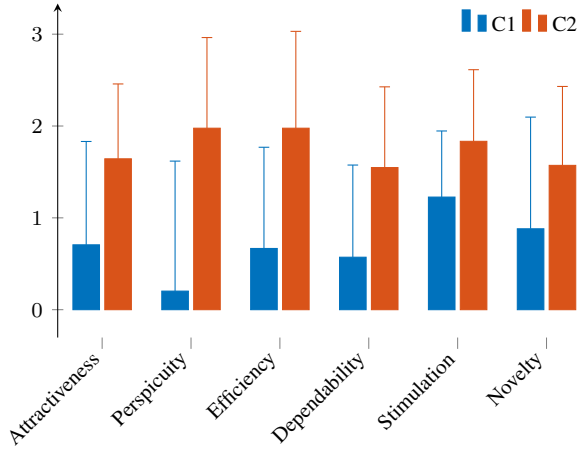


Fig. 15: Means of the UEQ scales in both conditions for all 21 participants.

statistical significance at a significance level of 5 % ($p = 1.863e^{-8} < 0.05$).

However, we found that error rates were similar for both workflows (Fig. 14). While in condition C1 some users had issues with the complexity of geometric constraints, in condition C2 some users struggled with different aspects of the part snapping workflow. A few users dropped the parts either too early or too late. Some participants stated that they did not completely understand the highlighting of specific surfaces on the CAD models and had to get used to it.

C. Semantic Mates improve the system's usability (H3)

The User Experience Questionnaire (UEQ) is evaluated for all 21 participants of the study. The means for the six scales are listed in Table II and visualized in Fig. 15. Condition C2 ranks noticeably higher on all scales. According to Schrepp [19], each scale ranges from -3 to 3. Values between -0.8 to 0.8 indicate a neutral result. Values greater than 0.8 indicate a positive result, while values below -0.8 indicate a negative result. Values greater than 2.0 or less than -2.0 can be interpreted as very positive or very negative, respectively, as they rarely occur [19]. While C1 ranks within the *neutral* interval for all scales except stimulation, C2 ranks in the upper half of the *positive* interval for all scales.

Two-tailed t-tests show that the differences between the conditions are statistically significant at a significance level

TABLE I: Mean task completion times of 14 participants that completed the programming task in both conditions C1 (without Semantic Mates) and C2 (with Semantic Mates). The table also denotes the results of individual subgroups with and without expertise in robotics.

Participants	C1 (s)	C2 (s)	p-value (H1)
7 non-roboticists	344.17 ± 73.21	83.57 ± 29.92	0.00008
7 roboticists	260.81 ± 151.35	99.86 ± 115.08	0.02420
All participants	302.49 ± 122.14	91.71 ± 81.22	0.00002

TABLE II: Means of the six UEQ scales for all 21 participants in both conditions C1 and C2.

Scale	C1	C2	p-value (H3)
Attractiveness	0.71 ± 1.13	1.64 ± 0.82	0.00384
Perspicuity	0.20 ± 1.42	1.98 ± 0.99	0.00004
Efficiency	0.67 ± 1.10	1.98 ± 1.05	0.00032
Dependability	0.57 ± 1.00	1.55 ± 0.88	0.00178
Stimulation	1.23 ± 0.72	1.83 ± 0.78	0.01235
Novelty	0.88 ± 1.22	1.57 ± 0.86	0.04053

of 5 % for all six scales (Table II). The significant differences in usability scores further imply that Semantic Mates and a part-snapping workflow complement users' mental models of assemblies far better than the manual specification of geometric constraints.

Overall, the study suggests that Semantic Mates help to significantly speed up the assembly specification workflow in our robot instruction framework.

V. Conclusion

By developing an ontology that groups geometric constraints into higher-level semantically described part connections, we are able to simplify the instruction of industrial robots regarding the specification of assembly workflows. A user study with 21 participants indicates that working with the Semantic Mates concept enables a significantly faster and more user-friendly interaction.

Assembly parts have to be manually annotated in order to offer Semantic Mates. However, the additional effort only has to be carried out once, while the additional knowledge can be reused and shared for many different assemblies involving the same types of annotated parts.

Our choice of using a knowledge-based representation of Semantic Mates gives us options for further research. It seems very likely that predefined Semantic Mate types can be added to new parts in a (semi-)automatic fashion based on the geometries that they are comprised of. This automatic classification could be carried out through the use of query languages and automatic reasoning techniques. It would enable human operators to use custom parts without the need to annotate them manually.

The combination of rich semantic descriptions of processes, products, and manufacturing resources enables a cognitive robot system to interpret these process and product models in order to automatically generate corresponding manufacturing plans. The integration of Semantic Mates and knowledge-based robot programming into CAD tools would enable CAD designers to check, whether or not their assemblies can actually be manufactured. As a result, product development could be sped up, while at the same time the risk of errors could be reduced.

References

[1] R. D. Schraft and C. Meyer, "The need for an intuitive teaching method for small and medium enterprises," in *Proceedings of the International Symposium on Robotics (ISR)*, Munich, Germany, May 2006, p. 95.

[2] A. Perzylo, M. Rickert, B. Kahl, N. Somani, C. Lehmann, A. Kuss, S. Profanter, A. B. Beck, M. Haage, M. R. Hansen, M. T. Nibe, M. A. Roa, O. Sörnmo, S. G. Robertz, U. Thomas, G. Veiga, E. A. Topp, I. Kessler, and M. Danzer, "SMErobotics: Smart robots for flexible manufacturing," *IEEE Robotics & Automation Magazine*, Jan. 2019.

[3] A. Perzylo, N. Somani, S. Profanter, I. Kessler, M. Rickert, and A. Knoll, "Intuitive instruction of industrial robots: Semantic process descriptions for small lot production," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Republic of Korea, Oct. 2016, pp. 2293–2300.

[4] M. Kraft and M. Rickert, "How to teach your robot in 5 minutes: Applying ux paradigms to human-robot-interaction," in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Lisbon, Portugal, Aug. 2017, pp. 942–949.

[5] N. Somani, M. Rickert, A. Gaschler, C. Cai, A. Perzylo, and A. Knoll, "Task level robot programming using prioritized non-linear inequality constraints," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Republic of Korea, Oct. 2016, pp. 430–437.

[6] P. Schmutz, S. Heinz, Y. Métrailler, and K. Opwis, "Cognitive load in eCommerce applications: Measurement and effects on user satisfaction," *Advances in Human-Computer Interaction*, vol. 2009, pp. 3:1–3:9, Jan. 2009.

[7] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sep. 2015, pp. 4197–4203.

[8] Z. Pan, J. Polden, N. Larkin, S. V. Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87–94, 2012.

[9] J. N. Pires, T. Godinho, and P. Ferreira, "CAD interface for automatic robot welding programming," *Industrial Robot*, vol. 31, no. 1, pp. 71–76, 2004.

[10] L. A. Ferreira, Y. L. Figueira, I. F. Iglesias, and M. Á. Souto, "Offline CAD-based robot programming and welding parametrization of a flexible and adaptive robotic cell using enriched CAD/CAM system for shipbuilding," *Procedia Manufacturing*, vol. 11, pp. 215–223, 2017.

[11] P. Neto, J. N. Pires, and A. P. Moreira, "CAD-based off-line robot programming," in *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Singapore, 2010, pp. 516–521.

[12] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, 2016.

[13] M. Stenmark, M. Haage, and E. A. Topp, "Simplified programming of re-usable skills on a safe industrial robot: Prototype and evaluation," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Vienna, Austria, 2017, pp. 463–472.

[14] F. Steinmetz, A. Wollschläger, and R. Weitschat, "Razer – A human-robot interface for visual task-level programming and intuitive skill parametrization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1362–1369, Jul. 2018.

[15] A. Perzylo, J. Grothoff, L. Lucio, M. Weser, S. Malakuti, P. Venet, V. Aravantinos, and T. Deppe, "Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective," in *Proceedings of the IFAC Conference on Manufacturing Modelling, Management, and Control (MIM)*, Berlin, Germany, Aug. 2019.

[16] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Construction of a benchmark for the user experience questionnaire (UEQ)," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, pp. 40–44, Jun. 2017.

[17] D. A. Norman, "Some observations on mental models," in *Human-computer Interaction: A multidisciplinary approach*, R. M. Baecker and W. A. S. Buxton, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987, pp. 241–244.

[18] J. Fischer, J. G. Mikhael, J. B. Tenenbaum, and N. Kanwisher, "Functional neuroanatomy of intuitive physical inference," *Proceedings of the National Academy of Sciences*, vol. 113, no. 34, pp. E5072–E5081, 2016.

[19] M. Schrepp, "User experience questionnaire handbook," Feb. 2019. [Online]. Available: https://www.ueq-online.org/Material/Handbook.pdf