Towards Ontology-based Handling of Uncertainties in Robotic Assembly

Dominik Mittel[0009-0002-3374-7330], Alexander Perzylo[0000-0002-5881-3608], Ingmar Kessler[0000-0002-9945-7144], and Junsheng Ding[0009-0001-1522-4213]

fortiss, Research Institute of the Free State of Bavaria
associated with Technical University of Munich, Munich, Germany
{mittel|perzylo|ikessler|ding}@fortiss.org, https://www.fortiss.org

Abstract. In the context of Industry 4.0, robot systems need to handle new and more complex tasks to produce highly customized products at small lot sizes. Small and medium-sized enterprises, in particular, lack the expert knowledge to parameterize such systems and to take relevant uncertainties into account. Ontologies provide functionality to explicitly encode knowledge using a common vocabulary. In this work, ideas towards the ontology-based representation of uncertainties and associated handling strategies are presented. We define an uncertainty taxonomy and combine it with knowledge about products, manufacturing processes, and resources, following the PPR modeling paradigm. The concept is implemented and tested using a robotic assembly task of an electronic component. As a result, the integration of different types of knowledge enables the automatic adjustment of robot processes based on the consideration of involved uncertainties. This may lead to an easier adaption of robot programs for new products and a more robust operation.

Keywords: Uncertainty · Ontology · Robotics

1 Introduction

In the realm of manufacturing, digitalization is an ongoing endeavor aimed at navigating the intricacies in the implementation, operation, and monitoring of production systems. This encompasses the digital representation of manufacturing specifications, such as process descriptions or models of production systems along with their associated data. Manufacturing environments, such as robotic assembly lines, are complex systems that demand a wealth of expertise from the automation domain for effective use. However, even if this expertise is available, it often remains tacit, residing solely within the minds of human experts.

Through the formal modeling of relevant contextual knowledge, technical systems can gain the capacity to interpret the implications of production, comprehending the dependencies and ramifications of their actions. They acquire the capability to evaluate the plausibility of their actions within a given manufacturing context. As a result, their level of autonomy can be increased. Recognizing the fallibility of technical systems, the explicit handling of uncertainties becomes a necessity. The architecture of these systems must be designed to navigate potential uncertainties across various levels, e.g., within individual devices or at the overarching process level. This entails the formal modeling of uncertainties as well as the development of corresponding handling mechanisms.

In this work, we propose leveraging semantic technologies for the formal representation and interpretation of knowledge regarding uncertainties, particularly within the context of robotic assembly. We investigate the semantic modeling of manufacturing processes, product specifications, and manufacturing resources (so-called PPR models) and explore how these models can be augmented to explicitly address uncertainties and possible handling strategies. The implemented models are tested in a robotic assembly process of an electronic component.

2 Related Work

Uncertainties have been a topic for extensive research. Shneier et al. [12] claim that due to "tight tolerances, difficult orientation requirements or access, and the extensive use of a variety of tools and assistive devices to achieve the join operation, much of assembly has been beyond the abilities of current robotic systems". Furthermore, the authors provide a definition of robust assembly and the requirements for robotic systems, in particular the insensitivity to variations: "An assembly operation is called robust if its performance or execution is substantially insensitive to variations that might occur, for example, in the sizes, shapes, and locations of parts, external loads, or operating conditions, so long as the variations are within the specified tolerances". In other words, they emphasize, that even robust systems have their limitations, within which a robust execution is or should be provided.

More authors describe a set of factors that influence the difficulty in robotic assembly [9,12]:

- Part properties that affect handling difficulty: dimensions, mass, surface texture/properties (e.g., sticky, greasy), fragility, flexibility, etc.
- Difficulty based on part shapes: sphere, cylinder, hexagon, cube, rivet, hex face peg, square face peg, 180° peg, 360° peg[1]
- Necessity for using additional tools or other assistance
- Conditions that affect insertion time: insertion region is accessible/visible, insertion operation is difficult, e.g., due to low clearances between parts
- Fastener type and fastening process time (e.g., clip-in)
- Insertion direction: down-from-top, from-the-side, angled or twisted, up-from-below[1]

This helps on the one hand to evaluate tasks and categorize possible uncertainties. On the other hand, the importance of geometric properties and circumstances is highlighted.

Diab et al. [4] highlight a solution towards an ontological characterization of *what is a failure* and *what concepts are useful to formulate causal explanations of*

[1] in ascending order of difficulty

failure. They further suggest the integration of knowledge of resources and their capabilities. Robots do not provide robust performance on their own, as their failure handling often consists of scripted responses to foreseen complications.

Huckaby and Christensen [5] define a taxonomic framework for task descriptions. This categorization may serve as a foundation for systematically investigating the necessary process steps for an assembly and their respective uncertainty aspects. The authors emphasize the need for an explicit modeling of assembly uncertainties. By capturing this knowledge, the system could be made aware of specific uncertainties and automatically handle them towards a more robust assembly.

3 Concept

3.1 PPR Modeling Paradigm

The objective of the *Product-Process-Resources* (PPR) paradigm [2] is to provide industrial robots and automation systems with intelligence [1] by formally representing knowledge concerning the product under construction, the production process leading to its creation, and the manufacturing resources involved in performing the process steps. The aim is to transform disparate local data to shared, unequivocal semantic knowledge models, employing OWL ontologies to enhance modularity, commonality, and reusability.

We extend the approach to include geometric data extracted from CAD models of products and manufacturing resources using the OntoBREP ontology [10]. This geometric representation is automatically converted from industry-standard STEP files utilizing an in-house developed conversion tool, which internally uses the Open CASCADE library and adheres to the boundary representation (BREP) standard. Consequently, all geometric data resides directly within the same semantic repository as the other models. OntoBREP distinguishes between topological entities organizing features hierarchically, and geometric entities describing these features through specific points, curves, and surfaces. Each entity is endowed with a unique identifier (IRI), facilitating linkage and annotation with additional information. In addition, the geometric models can be visualized in a self-developed, web-based *OntoBREP Viewer*. The Angular-based viewer allows to interact with the geometric models, e.g., to highlight or select different geometric entities. As the geometric representation is loaded from a common semantic repository, the viewer has access to all associated information as well.

A product model pertains to a specific product type, that typically is part of one or more taxonomies (class hierarchies) for product categorization or grouping. This facilitates the establishment of relations or properties for entire product sets. Product information may encompass precise geometry descriptions based on an OntoBREP representation, a bounding box/bounding sphere, material specifications, or mass and handling attributes.

A process model specifies a series of tasks involving various resources, such as machines, robots, and tools, to manufacture a product from input parts. Each task comprises parameters, whose values may be simple attributes or references to other entities, such as involved parts, tool types, or geometric constraints for assembly. OntoBREP representations can be used to define sets of geometric constraints between individual surfaces of two parts, e.g., to describe assembly poses. For instance, two surfaces could be specified to be coincident or concentric.

A resource model represents information regarding hardware or software components within a production environment. Resources may provide skills, defined as specific realizations of functionality [11]. Examples include *CartesianLinearMove*, *GraspGripper*, *ChangeTool*, or *PickAndPlace*. Hardware resources encompass actuators (machines, robots, tools) or sensors (cameras, force-torque sensors), while software components may provide computations or combine skills of other components to provide more complex functionalities. Individual resource models are further combined into environment models describing spatial layouts of components and their topological connections. OntoBREP is employed to encode geometry models of physical resources.

3.2 Uncertainty

Uncertainty in robotics may occur in many different processes and task types. Huckaby and Christensen [5] define a taxonomic framework for task descriptions. Within the scope of this work, we consider the *detect*, *pick-up*, and *insertion* tasks. In the following, these steps are described in detail with associated types of uncertainties. The identified types are stated in parentheses.

The *detection* step is applicable for all objects that should be detected by a perception system. In this case, the camera is mounted on the robot. Therefore, the resources involved are the robot, the camera, and the detection algorithm. As the camera is guided by the robot, the detection relies on the robot position accuracy (measurement error). Even if the camera is not mounted on a robot, the following uncertainties apply: the resolution and detection accuracy (measurement error). The detection algorithm itself introduces an uncertainty, which depends on the chosen implementation. For matching the perception result with a given model, the process can be influenced by object tolerances (tolerances). Lastly, the recognition of an object type is often defined by given categories and a threshold (binary decision).

As a next step, the robot shall *pick-up* a known object. Therefore, different hard- and software components are involved. The robot and the gripper define the relevant hardware. In this task, the robot position (measurement error) affects the uncertainty. Furthermore, the mounting of the gripper (measurement error) and tolerances of the gripper fingers (tolerances) could influence the application behavior. Involved software components could be a grasp planner or a collision checker. For collision checks, and grasp point and stability calculations, the use of lower-complexity models (simplified model) is a common uncertainty parameter. Additionally, the part could slip during pick-up or afterwards (unrecognized change).

For assembling a picked-up object, the *insertion* task is the logical consequence, which depends on many different entities. Often collision checks or other simulations are used, which rely on approximations of shape, size, or behavior of objects (simplified model). The picked-up object is inserted into another object. Both are affected by their geometric tolerances (tolerances). The quality of pose estimation of the object held by the robot depends on the robot accuracy and the mounting of the gripper (measurement error). The target object for the assembly, which is located somewhere else in the workcell, may also suffer from different types of uncertainty. It could be detected by the camera, haptically located by the robot, or be placed in a known fixture (measurement error). The uncertainty evaluation of the insertion task therefore needs to consider the combination of uncertainties pertaining to both involved objects. At the end, the success of a task needs to be assessed, which typically depends on the evaluation of the occurrence of an expected effect (binary decision). For a higher robustness, the robot could use force feedback and force control. There, latency and lag could influence the behavior and therefore the result (temporal uncertainties).

In typical scenarios, a lower overall uncertainty can be assessed, if there is a smaller chain of dependencies between involved entities, as uncertainties increase along the chain. This investigation of different robot tasks does not aim at providing a complete list of all possible uncertainties. Nevertheless, it gives a good estimate on the categorization of uncertainties relevant for mechanical assembly. For this work, the uncertainties are therefore categorized and defined as follows:

- Measurement errors: systematic or random error of sensor/measured data
- Tolerances: tolerances of product, robot, or workcell entities
- Binary decisions
- Unrecognized changes: changes of poses or environment
- Simplified models: errors due to approximations
- Temporal uncertainties: latency, jitter

These categories are transferred into an OWL ontology. The modeled concepts are then used to annotate product and resource models. Based on the augmented models, the uncertainty of a specific task, involving products and resources can be described. The ontology design is introduced in Section 4.

For mitigating uncertainties, coping strategies may be applied to offset or minimize the risk of failure.

3.3 Strategies

The combination of formally represented knowledge on uncertainty in combination with PPR models can increase the robustness during robotic assembly. When relying on the semantic knowledge, parameters for processes and resources may be automatically derived or dynamically adapted, e.g., if a gripper, robot, or tool changes. As a result, the need for hard-coded solutions can be reduced.

The semantic models could also provide possible strategies that can be pursued to counter the described uncertainties, if they are identified to interfere with a successful production run. On the one hand, the process specification and its requirements can be checked beforehand to choose appropriate strategies. On the other hand, failures occurring during execution require dynamic adaption of processes.

In our concept, implemented hardware and software skills are compared with the requirements of a given task. The task requirements are matched with the given uncertainties to decide whether the requirements can be met without further measures or which handling strategy shall be applied. For assembly tasks, the requirements are mostly given by geometric constraints and conditions. Therefore, the requirements can be derived from an analysis of the semantic geometry representation (OntoBREP). Through the interpretation of a process model, the robot is instructed in how to perform a given task. This can be adapted and extended with different strategies to reduce the impact of related uncertainties. Possible strategies may include:

– Choose more accurate detection algorithms
– Align and center objects with the tool, e.g., through gripper movements
– Align and measure object poses with the robot, e.g., based on robot movements and force feedback
– Use fixtures to place required parts and reduce their pose uncertainty

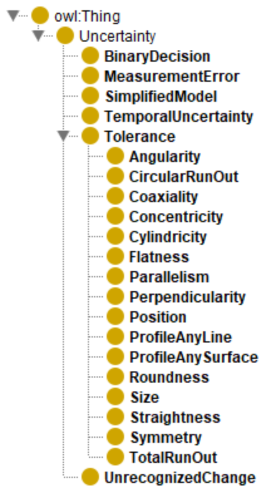
Additionally, geometric structures could be used to define further strategies. Hardware designs of fixtures or gripper fingers and the resulting functionalities could be directly linked to specific types of surfaces or other geometric entities. With V-shaped grippers, for example, certain parts can be automatically centered without the need for an additional step. This approach enables the specification and selection of suitable grasp modes.

To each strategy an "input" value should be assigned, up to which an uncertainty can be handled. The provided action reduces the uncertainty to a specific amount, which is the "output" value and assigned to the involved parts. If, for example, an object has to be centered by the robot, the gripper span and design must be considered as well as the potentially uncertain object location. Due to the pose uncertainty, the object could be in a slightly different position than expected. After the centering step, the uncertainty value of the object's pose changes depending on the given conditions, e.g., robot accuracy, gripper assembly, etc.

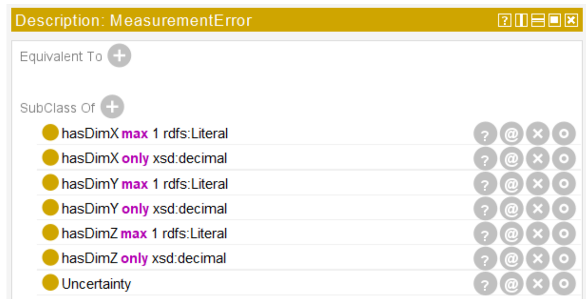
The check that is carried before an execution run aims at assessing under which circumstances an assembly is feasible, nevertheless the execution may still fail. Therefore, the semantic model should provide strategies to react dynamically to failures during task execution. These could involve:

– Retry the failed step with slightly different parameters
– Execute previous steps again
– Try different alignment strategies
– Ask for human input

The strategies are much more dependent on the circumstances of the failure, depend on when the failure occurs, and presuppose that the error can be recognized.



(a) Uncertainty taxonomy.



(b) MeasurementError class.

Fig. 1: Overview of the uncertainty ontology.

The introduced concepts describe aspects of uncertainties and handling strategies for robotic assembly processes and do not claim to be complete. The following sections detail specific implementations for some of them, in particular an implementation for describing the pose uncertainties of specific parts and retrieving uncertainty-related task parameters. The dynamic strategies are not further considered within this work.

4 Uncertainty Ontology

This section briefly describes the current ontology design. The design focuses on its applicability for assembly tasks and the implementation of an exemplary use case, which is described in the subsequent section. The ontology's class taxonomy is based on the identified categories from the previous section. For these, further subclasses could be considered. Fig. 1a depicts the taxonomy of categories as subclasses of the *Uncertainty* class. The *Tolerance* class is augmented with subclasses adhering to ISO 1101, the standard for geometric tolerances [3].

In this work, measurement errors regarding the robot and the camera, the gripper mounting accuracy, and the tolerances of object properties are of primary interest. These have three dimensional values that can vary for each dimension. Different values might be considered for positive and negative deviations. Fig. 1b shows possible data properties of the *MeasurementError* class. Here, we assume symmetrical tolerances.

According to the guide to the expression of uncertainty in measurement [6], the error is defined by the standard uncertainty $u(x_i)$ for each dimension. These values can be evaluated by measurement (Type A) or are given by specification

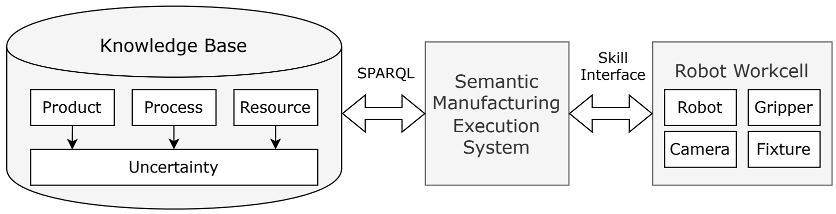


Fig. 2: Overview of the proposed system architecture.

(Type B). If not given, the values could also be derived from experience. For type A, multiple independent measurements are typically carried out. The uncertainty is then calculated using the arithmetic mean and the standard deviation.

Type B is the standard type of error for this work's use case (see Section 5). In the specification, often different conditions exist, which affect the uncertainty values. If only upper and lower limits are specified, the authors assume that the value X_i lies exactly between these boundaries. However, for given boundaries, different distributions could be assumed. In addition to the normal distribution, also rectangular or triangular distributions could be considered, which would affect the calculation of the standard uncertainty $u(x_i)$.

The combined standard uncertainty $u_c(y)$ for a whole system can be derived by the positive square root of the combined variance $u_c^2(y)$ of single standard uncertainties $u(x_i)$:

$$u_c(y) = \sqrt{u_c^2(y)} = \sqrt{\sum_{n=1}^{N} u^2(x_i)} \tag{1}$$

If the distribution is given, the standard uncertainty $u(x_i)$ could be transferred into a probabilistic value. For instance, assuming a normal distribution, three-times the standard uncertainty would result in 99.7 %.

In this work, the standard uncertainty $u(x_i)$ is used to describe the behavior of the components. Relevant values can be combined and compared with process requirements. As a result, an estimate of the uncertainty and a probability of success of a given assembly step could be provided. This enables a robot system to assess *how* the assembly should be carried out and *if* and *what* measures are required to reduce uncertainty.

5 Application

Fig. 2 shows an overview of the proposed system architecture. Following the PPR paradigm, all involved objects and resources, with their skills, are modeled in ontologies. The process models reference objects and resources to describe assembly tasks. The interpretation of the models and the communication between components is coordinated by a semantic Manufacturing Execution System (sMES). The sMES knows about the provided skills of the available re-

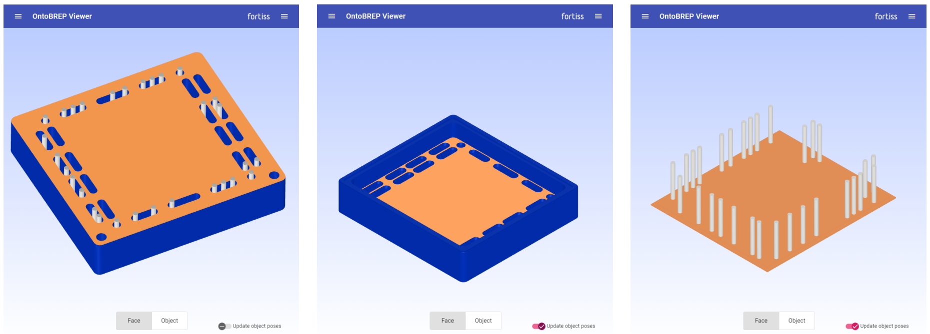


Fig. 3: OntoBREP visualization of full assembly, housing, and electronic chip.

sources, such as the robot (various move skills) or the gripper (open/close gripper and set gripper span skills).

Firstly, an abstract process model is defined. The Knowledge Base (KB) persistently stores the model and maps the defined abstract objects and resources to specific ones that are provided in a model of the production environment. This could also include a comparison of task requirements and resource skills with their uncertainties to evaluate handling strategies and whether or not the abstract process model needs to be adapted. As a result, a specific process model is created, that can be interpreted by the sMES for a particular robot system to perform the contained tasks.

For the implementation of the uncertainty ontology, an assembly process of an electronic component is investigated, which is described more precisely in the following subsections.

5.1 Use Case

The proposed concept has been tested with a real-world use case provided by an industry partner. Due to legal aspects, another comparable use case is presented here, that still allows the reader to follow the involved steps. The simplified electronic component to be assembled is composed of two subcomponents, a housing and an electronic chip. The electronic chip has several pins, which need to be inserted into the housing. The parts should be assembled by a KUKA LBR iiwa robot. In this setup, the electronic chips are provided via a special infeed tray. The housings are provided loosely on the robot table. The pins of the chip protrude beyond the housing, therefore, a special fixture and outfeed tray are required. The fixture is designed to feature a sloped storage area, to fix the housing in three dimensions by exploiting gravity. The assembly and the subcomponents are depicted in Fig. 3, which depicts the parts in the previously mentioned self-developed *OntoBREP Viewer*. The orange surfaces of the housing are highlighted for providing a better visualization of its features.

```
### http://www.fortiss.org/ont/process-electronic#AssemblyTask-1
process-electronic:AssemblyTask-1
        rdf:type           owl:NamedIndividual , task:AssemblyTask , core:Abstract ;
        core:hasActor      process-electronic:Actor-1 ;
        core:hasNext       process-electronic:AssemblyTask-2 ;
        core:hasPickObject  process-electronic:ElectronicChip-1 ;
        core:hasPlaceObject process-electronic:EletronicHousing-1 ;
        core:hasTargetFrame process-electronic:AssemblyTask-1-TargetFrame-1 ;
        core:hasTool       process-electronic:ParallelGripper-1 .
```

Listing 1: Definition of an abstract assembly task in the Turtle syntax.

```
PREFIX cad: <http://www.fortiss.org/ont/ontobrep#>
SELECT ?compound ?surface ?radius ?position WHERE {
    VALUES (?object) {(<http://www.fortiss.org/ont/electronics/perception-1.ttl#Housing-1>)}
    ?object   cad:hasShape      ?geometry .
    ?geometry rdf:type          cad:Compound ;
              cad:contains*     ?compound .
    ?compound rdf:type          cad:Compound ;
              cad:contains      ?solid .
    ?solid    rdf:type          cad:Solid ;
              cad:boundedBy     ?shell .
    ?shell    cad:contains      ?face .
    ?face     cad:representedBy ?surface .
    ?surface  rdf:type          cad:CylindricalSurface ;
              cad:radius        ?radius ;
              cad:position      ?position . }
```

Listing 2: Excerpt of a SPARQL SELECT query to retrieve the dimensions and positions of the holes in the housing.

5.2 Automatic Task Parameterization

Through the specific process model, the required tasks are fully defined. In this example, we focus on the assembly of the two introduced electronic parts. In our design, an *AssemblyTask* is a subclass of a *PickAndPlaceTask*. Listing 1 shows the abstract description of an *AssemblyTask* individual. The specification of the assembly task defines the actor, the tool, the involved parts, the target position, and the next task. These must be mapped to real objects and resources in the workcell. The definition of the *Actor* states necessary skills that the actor should provide. In this case, the required *PickAndPlace* capability is provided by the corresponding skill implementation of the robot. The tool is mapped from an abstract *ParallelGripper* to the specific gripper mounted on the robot. The *PickObject* and the *PlaceObject* are set to the chip and the housing, respectively. The *TargetFrame* defines the assembly pose as a relative transformation between the two parts.

Some task requirements are defined by the geometric properties of the *pickObject* and the *placeObject*. In particular, the clearance between the objects should be inspected. This can be automatically derived via a SPARQL query based on the parts' OntoBREP models. Listing 2 shows an excerpt of a query to retrieve the dimensions of the housing to further calculate the clearance. As

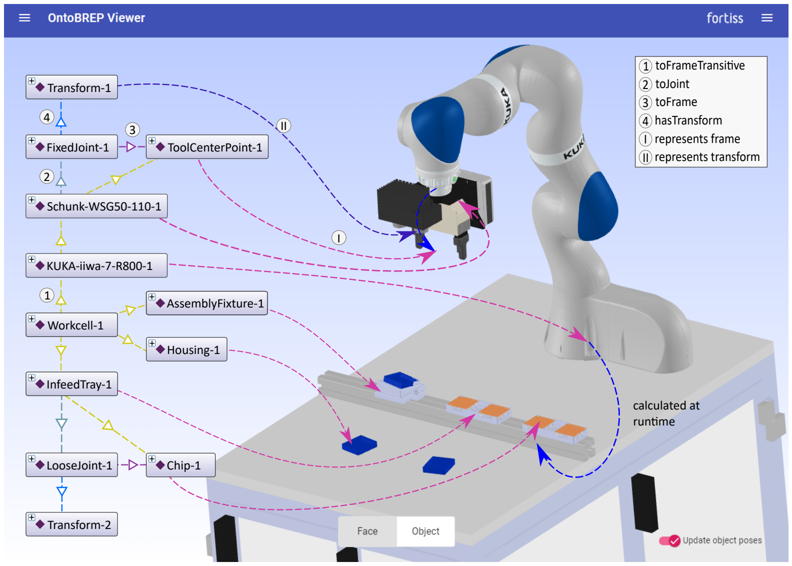


Fig. 4: Modeled products and resources shown in the OntoBREP Viewer (right) and a visualization of the semantic representation of their topology (left).

an example, the clearance of the round pins of the chip and the round holes of the housing are analyzed. The slotted holes of the housing are not considered.

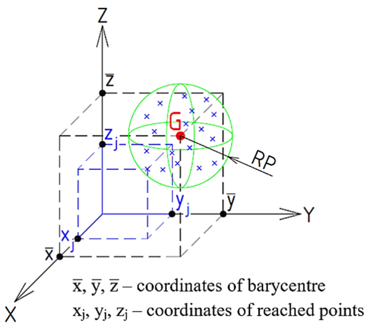
The extracted circular entities state a radius of 1.25 mm for the holes and 1.0 mm for the pins. This results in a total difference of 0.5 mm and a clearance of $\pm$ 0.25 mm. Consequently, the task uncertainty must be compared against 0.25 mm and be smaller than this value to ensure a robust assembly.

5.3 Workcell and Perception

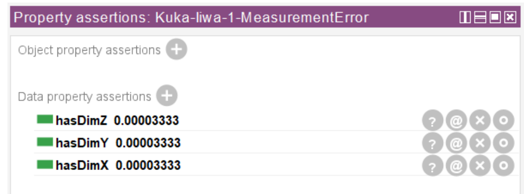
A specific process can only be executed in a particular workcell. In the workcell model, all contained resources and objects are semantically described and linked via a scene graph. Fig. 4 shows all relevant entities in the OntoBREP Viewer. The geometric representation of resources relies on their OntoBREP models. The workcell contains a KUKA LBR iiwa robot, a Schunk parallel gripper, a Roboception camera, as well as an assembly fixture and an infeed tray for the electronic components. The gripper and the camera are connected to the robot flange via a 3D-printed multi-tool adapter.

The camera uncertainty is given by its specification[2]. Not only the camera has an uncertainty value, but also the detection algorithm could introduce additional uncertainties. Here, a template matching algorithm is used. It can detect the orientation of the objects reliably, which is essential for the the given task. Nevertheless, the algorithm is affected by parallax errors, which can be up to a

[2] https://doc.rc-visard.com/v23.10/en/hardware_spec.html



(a) Graphical display of pose repeatability [8], adapted from [7].



(b) Property assertions for the robot's specified standard uncertainty in *meters*.

Fig. 5: Representation of the standard uncertainty of the robot.

standard value of 1 mm, which is already too high for the stated task requirements. Every detection in the workcell can create a perception instance in the ontologies. The perception entries contain the used sensor, algorithm, detected objects, and associated uncertainties.

A robot model may provide properties like its maximum payload, attached tool, or implemented skills, e.g., a *CartesianPtpMoveSkill*. The uncertainty value of the robot is given by its specification[3] and further defined in the ISO 9283 standard [7]. The positioning repeatability is defined as RP_i, which is the radius of the sphere whose center is the barycenter G (see Fig. 5a). The probability of a reference point falling into the described sphere is 99.7 %. That means that 99.7 % of the actually reached end effector poses are within a distance of 0.1 mm of the target pose for the KUKA LBR iiwa. As the uncertainty ontology is designed to work with the standard uncertainty, it is approximated by dividing the repeatability by three. The standard uncertainty $u(robot)$ is therefore $0.0\overline{3}$ mm. (see Fig. 5b).

The standard uncertainty of the gripper $u(gripper)$ is mostly affected by the 3D-printed adapter. In this case, the overall gripper value is set to $0.0\overline{3}$ mm, i.e., 99.7 % are between $\pm$ 0.1 mm. The uncertainty of the 3D-printed adapter has been assigned directly to the gripper for reasons of simplicity in subsequent calculations. Objects picked by the parallel gripper are aligned in the direction the gripper fingers close. The uncertainty value of the picked object is adapted depending on the value of the corresponding axis. The value is calculated with the Pythagorean addition of the involved uncertainties, in this case, the ones from the robot and the gripper. The involved components with their uncertainties could be automatically derived due to the semantic scene graph model.

The gravity-based assembly fixture is designed with a slope and a cutout for the protruding pins. The housing is placed at the calculated center of the fixture. The gravity and the slope let the housing slide in a known stable position. In

[3] https://www.kuka.com/event/media

Fig. 4, a housing is already placed in the fixture. The assembly fixture's pose can be measured by the robot and the gripper. With multiple measurements of the fixture, a standard uncertainty for the fixture $u(fixture)$ of 0.045 mm is determined.

Similarly, the infeed tray is statically attached to the workcell table. This means that the infeed tray is independent of camera detections. The uncertainty of the infeed tray $u(infeedTray)$ is set to 0.045 mm.

These introduced fixtures provide a specific effect: They assign their standard uncertainty value $u(x_i)$ to contained objects. For future use cases and applications, a "threshold" value for the uncertainty handling should be defined. Since the fixture can reduce the uncertainty, e.g., for objects that were previously recognized by the camera, the pose uncertainty of the grasped object within the gripper must be smaller than the "threshold" value of the fixture. The "threshold" could also be calculated automatically from the geometric properties of the fixture's cavity and the size of the housing.

5.4 Process Adaptation as Uncertainty Mitigation

As mentioned before: in our approach, abstract process models provide a hardware-agnostic, declarative specification of assembly tasks. In a deployment step, the abstract properties of these process model are mapped to specific objects and resources in a given workcell. During this deployment, the system may consider relevant uncertainties to adapt the task sequence to mitigate potential issues.

The default assembly process consists of individual steps, like the detection of objects, the pick-up of the electronic chip, and the insertion of the chip in the housing. The task requirements can be derived automatically by a SPARQL query investigating the semantic geometry representation. As determined in Section 5.2, the clearance of chip and housing has a value of $\pm$ 0.25 mm. The uncertainty value of the complete system must be smaller than this reference value. The optimum would be, if three times the combined standard uncertainty $u_c(y)$ is smaller than the clearance in order to reach a level of confidence of at least 99.7 %.

After the visual detection, the pose and the corresponding uncertainty of the objects is known. The uncertainty value $u(detection)$ is calculated from the camera and the detection algorithm. As stated before, this value is too high for the task and prevents the direct insertion of the chip in the loosely supplied housing.

The robotic system must choose a strategy to reduce the uncertainty to an adequate level. One could be to center the housing with the gripper. Due to the rectangular shape, the housing would need to be centered at least twice using orthogonal gripper alignments. But, it cannot be guaranteed, that the object does not move after its release from the gripper or during the second alignment procedure. The protruding pins would prevent a simple assembly anyways. Therefore, the use of the assembly fixture is defined as a necessary mitigation.

At the beginning, the housing has a pose uncertainty value derived from the detection: $u(housing) = u(detection)$. After the placement in the assembly fixture, the uncertainty is reduced: $u(housing) = u(fixture) = 0.045\ mm$. For the evaluation of this strategy, the detection uncertainty needs to be investigated. On the one hand, it needs to be possible to grasp the object with the given uncertainty, which mostly affects the required minimum gripper span and distances to other objects. On the other hand, the fixture must able to handle this uncertainty level, which was previously defined as the "threshold" value of the fixture. Both preconditions are given in this use case.

The electronic chip needs to be picked up by the robot, after the housing is placed in the assembly fixture. The initial uncertainty of the chip depends on the uncertainty value of the infeed tray, which is 0.045 mm: $u(chip) = 0.045\ mm$. When the object is grasped by the robot, the uncertainty changes. One axis is aligned by the gripper, i.e., one axis is reduced to the uncertainty value of the robot and the gripper. The other axes still depend on the robot, the gripper, and the uncertainty of the infeed tray. For the sake of simplicity, only the latter uncertainty is considered, depending on the robot, the gripper, and the infeed tray:

$$u_c(chip) = \sqrt{u_c^2(chip)} = \sqrt{u^2(robot) + u^2(gripper) + u^2(infeedTray)} \quad (2)$$

With the uncertainty of the grasped chip, the overall uncertainty of the task can be calculated. The overall combined uncertainty $u_c(task)$ is the Pythagorean addition of the uncertainties of the two involved objects:

$$u_c(task) = \sqrt{u_c^2(task)} = \sqrt{u^2(housing) + u_c^2(chip)} \quad (3)$$

The calculated value can then be compared against the requirement, which is given by the available clearance between the two parts in the assembly configuration. As a result, a decision can be made that the assembly task can be performed in this particular workcell, if, as a first step, the housing is placed in the assembly fixture.

6 Conclusion

This work presents a concept for extending our knowledge-augmented engineering methodology tailored for robotic assembly, with a specific focus on representing and managing uncertainties. We introduce semantic description languages based on OWL for modeling processes, products, and manufacturing resources, following the PPR modeling paradigm. Additionally, we discuss relevant uncertainties in robotic assembly, exploring how integrating associated knowledge extends existing PPR models. Our experiment demonstrates that by semantically integrating knowledge from various sources encompassing all relevant facets of an automation task, a technical system can gain a deeper understanding of production objectives and contexts. This enhanced understanding enables it to analyze tasks and evaluate their feasibility. Automatic adjustments to process plans can be derived, leveraging hardware-agnostic process specifications across different hardware configurations, while explicitly addressing task-specific uncertainties. Consequently, the autonomy and robustness of robotic automation systems can be enhanced.

Acknowledgments. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952197.

References

1. Björkelund, A., Bruyninckx, H., Malec, J., Nilsson, K., Nugues, P.: Knowledge for intelligent industrial robots. In: Konidaris, G. (ed.) AAAI Technical Report SS-12-02, Designing Intelligent Robots: Reintegrating AI. vol. SS-12-02. AAAI (2012)
2. Cutting-Decelle, A.F., Young, R.I., Michel, J.J., Grangel, R., Le Cardinal, J., Bourey, J.P.: Iso 15531 mandate: a product-process-resource based approach for managing modularity in production management. Concurrent Engineering **15**(2), 217–235 (2007)
3. Deutsche Institut für Normung e.V. (DIN): DIN EN ISO 1101:2017 - Geometrical product specifications (GPS) - Geometrical tolerancing - Tolerances of form, orientation, location and run-out (Sep 2017)
4. Diab, M., Pomarlan, M., Borgo, S., Beßler, D., Rossel, J., Bateman, J., Beetz, M.: FailRecOnt – An Ontology-Based Framework for Failure Interpretation and Recovery in Planning and Execution. In: Proceedings of the Joint Ontology Workshops co-located with the Bolzano Summer of Knowledge (BOSK 2021), Virtual & Bozen-Bolzano, Italy, September 13th to September 17th, 2021. CEUR Workshop Proceedings, CEUR-WS.org (2021)
5. Huckaby, J., Christensen, H.I.: A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics. AAAI Workshop - Technical Report pp. 94–101 (01 2012)
6. International Bureau of Weights and Measures, International Organization for Standardization (eds.): Evaluation of Measurement Data - Guide to the Expression of Uncertainty in Measurement. International Organization for Standardization, Genève, Switzerland, 2008 edn. (2008)
7. ISO/TC 184 Automation systems and integration: ISO 9283:1998 - Manipulating industrial robots - Performance criteria and related test methods (Apr 1998)
8. Kuric, I., Tlach, V., Ságová, Z., Císar, M., Gritsuk, I.: Measurement of industrial robot pose repeatability. MATEC Web of Conferences **244**, 01015 (2018)
9. Norton, A., Saretsky, A., Yanco, H.: Developing Metrics and Evaluation Methods for Assessing AI-Enabled Robots in Manufacturing
10. Perzylo, A., Somani, N., Rickert, M., Knoll, A.: An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions. In: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems. pp. 4197–4203. Hamburg, Germany (Sep 2015)
11. Profanter, S., Perzylo, A., Rickert, M., Knoll, A.: A generic plug & produce system composed of semantic opc ua skills. IEEE Open Journal of the Industrial Electronics Society **2**, 128–141 (Jan 2021)
12. Shneier, M.O., Messina, E.R., Schlenoff, C.I., Proctor, F.M., Kramer, T.R., Falco, J.A.: Measuring and Representing the Performance of Manufacturing Assembly Robots. Tech. Rep. NIST IR 8090, National Institute of Standards and Technology (Dec 2015)