Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-026-02862-y

KG4CUT: an ontology to facilitate cutting tool selection and interoperability

Beatriz Olarte Martinez[2] · **Leonardo Piano**[1] · **Salvatore Mario Carta**[1] · **Marco Manolo Manca**[1] · **Enrico Motta**[3] · **Livio Pompianu**[1] · **Giuseppina Salluzzi**[2] · **Massimiliano Annoni**[2]

Received: 31 August 2025 / Accepted: 26 March 2026

Abstract

Selecting cutting tools for milling is a critical and complex task that directly affects product quality, cost, and operational efficiency. The growing diversity of tools and vendor-specific catalogues makes this process especially challenging, particularly for less experienced operators. In this paper, we present `KG4CUT`, an application ontology aligned with W3C Semantic Web standards and FAIR principles, designed to standardize and integrate cutting tool information across providers. To demonstrate its practical utility, we populated a knowledge graph using an automated pipeline that extracts structured data from real-world PDF catalogues. This graph serves as both a proof of concept and a functional basis for intelligent tool recommendation and cutting parameter retrieval, based on material properties, operation types, and geometric constraints. Evaluation with domain experts showed improved retrieval efficiency and reduced selection errors. `KG4CUT` thus supports the digitalization of machining knowledge and enables faster, more accurate process planning in industrial settings.

Keywords Cutting tool selection · Ontology · Industry 4.0

Introduction

Chip removal processes and machine tools are a fundamental pillar of the European economy (Labucay, 2022). In economic terms, its production has recovered from the Covid-19 crisis and new sustainable processes are increasing the orders for machine tools in the European landscape. Despite these promising trends, the industry faces challenges, particularly the shortage of skilled labor (Akyazi et al., 2020). Cutting tools are a fundamental resource in machining, where the correct and efficient selection of cutting parameters plays a decisive role in a twofold manner, enhancing product quality and optimizing manufacturing costs. They are estimated as 8% of the total manufacturing costs in the manufacturing phase of the product life cycle (Zhou et al., 2017), making it a critical factor for competitiveness and sustainability (Ji et al., 2018). Selecting the appropriate tool and its cutting parameters for a machining process is not trivial, as numerous factors, including material properties, tool geometry,

---

✉ Leonardo Piano
leonardo.piano@unica.it

Beatriz Olarte Martinez
beatriz.olarte@polimi.it

Salvatore Mario Carta
salvatore@unica.it

Marco Manolo Manca
marcom.manca@unica.it

Enrico Motta
enrico.motta@open.ac.uk

Livio Pompianu
livio.pompianu@unica.it

Giuseppina Salluzzi
giuseppina.salluzzi@mail.polimi.it

Massimiliano Annoni
massimiliano.annoni@polimi.it

[1] Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, Cagliari 09124, Italy

[2] Department of Mechanical Engineering, Polytechnic of Milan, Via G. La Masa 1, Milan 20156, Italy

[3] Knowledge Media Institute, The Open University, Walton Hall, Berrill Building, Milton Key MK7 6AA, United Kingdom

workpiece geometry, machine-tool capabilities, machining conditions, and customer requirements, influence it.

Machining quality is strongly affected by the geometric characteristics of the cutting tool, such as the number of flutes, tool length, rake angle, and helix angle. Tools with a higher number of flutes provide greater stiffness and allow more cutting edges to engage simultaneously, leading to improved surface finish and higher achievable feed rates. Conversely, longer tools tend to increase vibration, negatively affecting surface quality. A positive rake angle reduces cutting forces and temperature, thereby improving surface finish, while a higher helix angle enhances chip evacuation and reduces chatter, resulting in better surface finish and dimensional accuracy. In addition to tool geometry, external factors such as coolant application and tool wear play a significant role in machining quality. The use of coolant lowers cutting temperatures and provides lubrication, improving chip evacuation, surface finish, dimensional accuracy, and tool life. The interplay of these numerous factors makes cutting tool selection a critical and time-consuming process for manufacturing companies.Also incorrect decisions can lead to inefficiencies, increased costs, and reduced product quality (Ji et al., 2018), and expert knowledge is a fundamental part of selecting good parameters.

Additionally, advances in manufacturing have led to processes that are more diverse, automated, and tailored to specific needs. The demand for intricate and specialized parts has risen, and businesses are transitioning from traditional mass production to smaller, highly varied product batches. As a result, chip-removal processes and, more specifically, cutting-tool manufacturers have expanded the range of cutting tools available in different catalogues. This variety complicates decision-making, especially for less-experienced operators, and makes cross-vendor comparisons difficult and time-consuming (Brenner et al., 2017). These challenges highlight the need for advanced solutions to manage and effectively utilize domain-specific knowledge.

In industry, traditional relational databases are commonly employed to manage structured manufacturing data. While effective for basic storage and retrieval, these systems lack the expressiveness needed to formally capture domain-specific semantics and complex interrelationships, such as those between tool geometry, material compatibility, and cutting strategies. Furthermore, schema rigidity and inconsistency across systems limit interoperability, reuse, and integration of data from diverse sources. Accordingly, semantic technologies, specifically ontologies, offer significant advantages in data management, improving the organization and shareability of knowledge among humans and machines. The widely accepted definition of an ontology, originally proposed by Gruber (1993), can be simplified as a structured vocabulary that represents a specific domain of knowledge and its relationships.

To facilitate cutting tool selection and improve semantic interoperability across vendors, we propose `KG4CUT`, an application ontology that formalizes cutting tool data, usage parameters, and application contexts (e.g., materials and operations). `KG4CUT` is designed in accordance with Semantic Web standards and adheres to the FAIR principles (Wilkinson et al., 2016). To demonstrate its practical applicability, we populate a knowledge graph from real-world vendor PDF catalogues using the `KG4CUT` ontology as a schema. This graph enables intelligent querying and tool recommendation based on geometric features, material types, and machining operations. Beyond representing tool properties for selection, `KG4CUT` can also serve as a semantic layer for CAM systems, offering semantically grounded recommendations that integrate cutting tools, parameters, and machining strategies. Tool and parameter selection represents a critical and knowledge-intensive step in CAM workflows. It constrains subsequent stages from operation definition to toolpath generation and NC code production, which remains largely manual and vendor-specific. By formalizing this knowledge in `KG4CUT`, users can apply it to existing CAM operations to support consistent parameterization, reducing semantic inconsistencies and limiting error propagation throughout the planning process.

We further validate the approach through an expert evaluation involving realistic machining scenarios, showing that `KG4CUT` improves retrieval efficiency and reduces selection errors. By formalizing machining knowledge in a reusable and interoperable form, `KG4CUT` supports smarter, more digitized process planning in industrial environments.

The proposed ontology is expected to benefit stakeholders across various roles in the machining industry. Process planning engineers, particularly junior staff, will benefit from fast and reliable retrieval of optimal cutting parameters from multiple vendors, reducing cross-referencing errors and streamlining approval workflows. Supply chain personnel will gain immediate access to vendor catalog data, including direct order links, thereby minimizing miscommunication and reducing the need for back-and-forth clarification efforts. Business and operations managers will obtain a clearer understanding of tooling usage and purchasing patterns, enabling the adoption of lean manufacturing strategies such as just-in-time procurement and optimized inventory management.

The rest of the paper is organized as follows: Section Related works discusses related works and reviews the State-Of-The-Art of manufacturing ontologies. Section Methodology presents the methodology followed for the ontology development. Section KG4CUT: Overview introduces `KG4CUT`, our application ontology tailored for the solid carbide end-mill domain. Section Ontology evaluation outlines the experimental setup and validation. Section Discussion discusses the findings and the limitations of our work. Section Conclusions, finally, concludes the paper.

Related works

In this Section, we review relevant literature in two main areas. First, in Section Review of industrial ontologies, we examine general industrial ontologies proposed for manufacturing and production systems. Then, in Section Ontological approaches for cutting tool selection, we focus on works that specifically address applications related to cutting tools.

Review of industrial ontologies

In recent years, the need for standardized ontological representations of the manufacturing environment has become increasingly important in the context of Industry 4.0 (Kumar et al., 2019). Several studies have emphasized the role of a shared vocabulary for machines, processes, and production resources in enabling interoperability and supporting automated decision-making (Sapel et al., 2024).

Among the foundational contributions, the MAnufacturing's Semantics ONtology (MASON) (Lemaignan et al., 2006) stands out as a pioneering proposal for a manufacturing upper ontology aimed at enabling interoperability across heterogeneous systems. MASON formalizes core concepts of the manufacturing domain: entities, operations, and resources into a shared semantic framework using OWL. The authors demonstrate its applicability through two implementations: an expert system for automated cost estimation and a multiagent simulation framework where OWL ontologies serve as both a communication vocabulary and a cognitive model for intelligent agents. However, its design intentionally leaves detailed configuration knowledge and rule-based reasoning to domain-specific extensions, limiting its direct applicability to fine-grained cutting tool selection.

Similarly, Ameri and Dutta (2006) introduced the *Manufacturing Service Description Language*, an OWL-based upper ontology designed to formalize the description of manufacturing services in virtual enterprise environments. It models key concepts, including capabilities, processes, and contextual attributes (e.g., materials, tolerances, experience), thereby supporting semantic matchmaking across service providers. MSDL adopts a modular structure that enables domain-specific extensions, but its high-level abstraction and limited coverage of tool-level details constrain its applicability to low-level manufacturing scenarios.

Another notable contribution is the *Machine-Tool Model (MTM)* (Kjellberg et al., 2009), an ontology centered on the machine tool as a crucial element of manufacturing systems. Their approach emphasizes the importance of understanding machine-tool functionality for process planning and cutting-tool selection.

Building on these foundations, the Semantically Integrated Manufacturing Planning Model (SIMPM) (Šormaz, 2019) advances the conceptualization of manufacturing process planning with a novel upper-level ontology that explicitly models the fundamental constraints of variety, time, and aggregation.

Foundational ontologies have also influenced manufa cturing-specific models. Borgo and Leitão (2004) demonstrated the applicability of foundational ontologies in this domain. The ADACOR ontology (Leitao et al., 2005), developed on top of DOLCE (Masolo et al., 2003) , exemplifies this approach by supporting a holonic manufacturing platform and autonomous intelligent manufacturing systems. Similar to Borgo et al, Ali et al. (2019), construct a manufacturing ontology aligned with multiple foundational ontologies, including BFO [1], CCO [2], and CHAMP [3], thereby enhancing semantic consistency and interoperability. Closer to the scope of this work, the Manufacturing Resource Capability Ontology (MaRCO) (Järvenpää et al., 2019) focuses on capability matchmaking with an emphasis on assembly operations, primarily targeting machine and assembly reconfiguration scenarios. Although MaRCO provides a formal and flexible capability framework, it does not adequately address chip removal and cutting-tool processes, which are central to our domain of interest. Similar limitations apply to other capability-oriented ontologies for manufacturing resources (Sarkar et al., 2020).

The InVor ontology (Rehage & Gausemeier, 2015) offers a more domain-specific contribution by modeling machine tools and their configuration-derived capabilities to support the allocation of alternative CNC machines for given NC programs. While InVor provides detailed representations of machine components, its focus is on machine allocation rather than on cutting tools, machining parameters, or workpiece transformations. In contrast, our work specifically targets the semantic modeling of milling cutting tools and their role in machining processes. It captures fine-grained, process-oriented knowledge unique to solid carbide end mills, including detailed geometric properties (e.g., flute number, helix angle, corner radius following ISO 13399), material compatibility (ISO 15608), cutting parameter ranges, functional roles (e.g., rougher, finisher, chamfering tool), and milling strategies. These aspects are not explicitly represented in other machining ontologies such as MASON, MaRCO, or InVor. Most importantly, `KG4CUT` introduces functional and process-oriented modeling pat-

---

[1] http://purl.obolibrary.org/obo/bfo.owl

[2] https://github.com/CommonCoreOntology/CommonCoreOntologies

[3] https://www.jneilotte.com/category/industry/

terns that are absent from prior work. Milling activities are modeled as explicit transformations, in which a combination of cutting tool, cutting parameters, and raw geometry yields a machined feature. This modeling choice allows machining operations to be described in terms of their functional effect on the workpiece, rather than as static taxonomic relations.

From a brodaer perspective, although not an ontology itself, the Process Specification Language (PSL) (Schlenoff et al., 2000), based on the Knowledge Interchange Format (KIF) [4], plays a key role in facilitating the exchange of process information across heterogeneous systems. While PSL enables interoperability among applications, it is not designed for ontological modeling of machining knowledge.

Finally, even if scheduling and planning are not the core focus of our ontology, they are integral to manufacturing operations. Two relevant ontologies in this area are the Generic Task Ontology for Scheduling Applications [1], which conceptualizes milling tasks without an OWL-based formalization, and the Scheduling Reference Ontology (Vegetti & Henning, 2022), which aims to standardize scheduling activities through semantic interoperability. Both have influenced our modeling choices, particularly in framing manufacturing activity as a task. Similarly, Qin and Lu (2025) propose a Knowledge Graph-enhanced Multi-Agent Reinforcement Learning approach for adaptive scheduling, demonstrating the potential of knowledge graphs for semantic interoperability in dynamic manufacturing environments, though without addressing cutting tool modeling.

Ontological approaches for cutting tool selection

Although significant advances have been made in the study of tool selection over the years, to the best of our knowledge, there are very few studies based on an ontological approach. In this context, Duan et al. (2021) has developed a recommendation framework based on a metal-cutting tool knowledge graph. In particular, their approach recommends tools that exploit a revised version of the PageRank algorithm, using graph node connectivity as a ranking measure. Zhou et al. (2017) developed an ontology-based framework for improving cutting tool process configuration and consequently reducing carbon emissions.

Xia et al. (2024) presents an ontology-based framework for integrating multi-source, heterogeneous data within cutting tool lifecycle management. The authors propose an ontology model integrated with the System Modeling Language (SysML) to provide a standardized, semantically consistent description of materials, information, processes, and their interactions, ultimately leveraging ontology queries and logical reasoning to develop an assisted decision-making application for tool-selection scenarios.

Table 1 summarizes the reviewed manufacturing ontologies in a synthetic table, highlighting the major differences with our proposal. Despite progress in both foundational and domain-specific manufacturing ontologies, significant limitations persist when applied to the cutting-tool selection domain. Upper-level ontologies such as MASON, MSDL, or ADACOR provide general representations of manufacturing entities and services but lack the in-depth granularity required to model the detailed geometric, technical, and functional properties and the operational aspects of cutting tools. The focus of those ontologies remains on the machine tool, ignoring the constraint that an inappropriate cutting tool or an ideal cutting tool can impose on the final workpiece. Specifically, tools are often treated generically, without accounting for distinctions such as tool shape, cutting parameters, or compatibility with different materials and tasks. This situation limits their practical utility in real-world environments, where tool selection is critical to quality, time, and cost. Our knowledge model addresses the complexity arising from the ad hoc adoption of ISO standards, such as ISO 13399 (2018), and ISO 14649 (2020), particularly regarding tool naming conventions used by different manufacturers. At the same time, it supports one of the most challenging tasks in machining: selecting appropriate cutting tools and parameters for the specific milling operation and workpiece material. The resulting semantic model is designed to be accessible and usable by both expert and non-expert users. To the best of our knowledge, `KG4CUT` is the first publicly available, FAIR-compliant OWL ontology that offers a detailed and extensible model for cutting tool representation and application, supporting real-world decision-making use cases.

Methodology

This section presents the methodological approach adopted in this work. Section Ontology development provides an overview of the general methodology, while the subsequent sections describe how each of its main phases was carried out. Specifically, Section Ontology requirements specification covers the Requirements Specification phase, Section Ontology implementation focuses on Ontology Implementation, Section Ontology publication discusses Ontology Publication, and finally, Section Section Ontology maintenance addresses Ontology Maintenance.

Ontology development

In this work, we adopted the Linked Open Terms (LOT) methodology (Poveda-Villalón et al., 2022) to guide the development of our ontology. LOT promotes a structured, iterative approach to ontology engineering, emphasizing alignment between ontology artifacts and software develop-

---

[4] ttps://logic.stanford.edu/kif/kif.html

Table 1 Comparison of representative manufacturing ontologies. Level: A = Application, D = Domain, U = Upper, F = Foundational. Partial reuse indicates that no implementation is available, but the ontology can be reconstructed from the reference paper

Ontology	Scope	Level	Formalism	Availability/Reuse
MASON	General manufacturing	D/U	OWL	Yes / No
SIMPM	Process planning	D/U	OWL	Partial / Yes
MSDL	Service description for virtual enterprises	D	OWL DL	Partial / No
MTM	Machine tool modeling	A	UML	No / No
ADACOR	Manufacturing control architecture	D	F-Logic, OWL	Partial / Yes
PSL	Process info integration across systems	F	KIF (Interlingua)	Partial / No
MaRCO	Assembly resource capabilities	D/U	OWL	Yes / No
InVor	Machine tool selection	A	UML-based OWL	Partial / No
`KG4CUT`	Cutting tool selection, parameters, CAD	A	OWL 2	Yes / Yes

ment practices and the reuse of existing semantic resources where possible.

The LOT methodology is organized into four main phases:

1. **Requirements Specification:** analysis of the domain, use cases, and actors, leading to the identification of competency questions (CQs) that the ontology must support.
2. **Ontology Implementation:** conceptual modeling of classes and relationships, ontology encoding in OWL, reuse of existing ontologies, and validation through reasoning and verification of competency questions.
3. **Ontology Publication:** release of the ontology along with documentation and persistent identifiers to ensure long-term accessibility and reuse.
4. **Ontology Maintenance:** ongoing updates based on user feedback and evolving requirements.

LOT also defines three key roles in the development process: *ontology developers*, who are experts in formal modeling; *domain experts*, who provide knowledge of the application domain; and *ontology users*, who interact with the final artifact.

In our case, these roles often overlapped: several team members contributed both domain expertise and ontology modeling skills. This convergence facilitated efficient communication and rapid iteration throughout the development process.

Ontology requirements specification

The requirements specification phase of `KG4CUT` (Knowledge Graph for Cutting Tools) was guided by a concrete industrial need: efficiently recommending suitable cutting tools for specific milling operations, including the retrieval of recommended parameters and the identification of compatible target materials and geometrical features. In particular, we target scenarios in which both experienced and less experienced users must select appropriate tools from extensive, heterogeneous vendor catalogues. The goal is to reduce manual effort and decision errors by providing structured, machine-interpretable knowledge.

To formalize this knowledge, we adopted a competency-question-driven approach (Gruninger, 1995). Competency questions (CQs) are natural-language queries that specify the information the ontology should provide. They help define its scope and structure and serve as a validation mechanism throughout the development cycle.

The identified CQs cover key aspects such as tool properties, material compatibility, process parameters, machining capabilities, and the types of geometrical features that tools can produce. These questions were iteratively refined in collaboration with domain experts and reflect real-world decision-making contexts.

CQs and requirements were gathered through an interview-based phase, during which ontology developers asked domain experts natural language questions such as: *What are the main modules involved?*, *Which elements are critical in the selection process?*, and *How are these elements related?* The experts' responses were initially captured and organized using informal conceptual models, which facilitated the identification of key concepts, relationships, and attributes within the domain. These preliminary models were then systematically translated into ontology triples, from which the categorization, taxonomy, and interrelationships were derived. To ensure consistency across different manufacturers, terminology was carefully mapped and standardized, resulting in uniform class naming throughout the ontology.

Table 2 presents the grouped CQs that informed the ontology design.

Ontology implementation

The ontology was developed iteratively over six conceptual and technical sprints, following an agile-inspired approach

Table 2 Grouped competency questions and their focus

Grouped Competency Questions (GCQ)	
GCQ1	**Physical properties of a cutting tool:** • What are the characteristics of a mill? • What is the type of shank associated with this mill?
GCQ2	**Materials that can be machined by a cutting tool:** • What materials can be machined by a certain mill?
GCQ3	**Cutting parameters of a cutting tool:** • Which cutting parameters would you suggest in roughing/finishing?
GCQ4	**Operations/capability of a mill:** • What are the operations that can be made with this mill?
GCQ5	**Final geometry of the workpiece:** • Which mills are good for machining a pocket? • Which mills are good to machine a groove/slot?

(Abdelghany et al., 2019). A sprint, a concept of Scrum or Agile project management methodologies (Fowler et al., 2001), is a scheduled interval during which a defined set of tasks must be completed. Each sprint focused on refining specific modules of the ontology, with close collaboration between domain experts and ontology engineers.

The initial phase involved conceptualizing key entities and relationships based on expert knowledge, industry standards, and vendor catalogues. This included the formalization of cutting tools, their properties, and usage contexts. Subsequent iterations expanded the model to include material classification (based on ISO/TR 15608), recommended cutting parameters, and machining tasks.

Modeling activities were first carried out using graphical tools to ensure shared understanding, then translated into OWL using `Protégé`[5]. Reasoning was performed using the HermiT reasoner[6] to ensure logical consistency. The ontology was processed programmatically using Python libraries such as `rdflib`, enabling integration with downstream data pipelines.

Each sprint concluded with validation activities, including reasoning checks and competency-question testing, to ensure that the ontology remained aligned with its intended use cases and retained semantic coherence across iterations.

Although no existing ontology fully met the requirements of our domain, we prioritized reusing relevant models and design patterns to promote interoperability and alignment with Semantic Web best practices.

Specifically, we integrated selected elements from the InVor ontology (Rehage & Gausemeier, 2015), which provides a structured representation of machine tools and their components. InVor was originally modeled in UML, and we adapted it to OWL, incorporating relevant classes such as `MachineTool` and `ToolHolder` into our schema. For part-whole relationships and compositional modeling, we adopted the `componency` pattern from the Ontology Design Patterns (ODP) library (Gangemi & Presutti, 2009). Additionally, to enhance metadata expressiveness and facilitate linking with upper-level ontologies, we reused widely adopted vocabularies such as FOAF[7] and Dublin Core[8].

We distinguish between two types of reuse:

- **Hard reuse**, where external ontologies or modules are directly imported and used without modification (e.g., FOAF, componency).
- **Soft reuse**, where only specific classes or properties are selectively adapted or aligned (e.g., from InVor).

This reuse strategy supports the long-term maintainability of `KG4CUT` while ensuring semantic compatibility with related ontologies in the manufacturing and engineering domains.

Ontology publication

The ontology was published as open source on the software development platform GitHub, which facilitates continuous integration, version control, maintenance, peer review, and static web hosting. These capabilities support both collaborative development and long-term sustainability of the ontology. An initial version of the ontology's webpage was created to enable the request for a persistent identifier via w3id.org[9], which ensures a stable, long-lasting reference to the ontology[10]) was assigned to the project.

---

[5] https://protege.stanford.edu/

[6] http://www.hermit-reasoner.com/

[7] http://xmlns.com/foaf/spec/

[8] https://www.dublincore.org/

[9] https://w3id.org/

[10] https://w3id.org/kg4cut

To enhance accessibility and usability, the ontology webpage was further enriched using Widoco[11], an ontology documentation tool that automatically generates comprehensive documentation from OWL files. Widoco is integrated with WebVOWL, which provides an interactive visual representation of the ontology, thereby improving clarity and supporting both technical and non-technical stakeholders in understanding its structure.

Ontology maintenance

Ontology maintenance refers to the ongoing activities required to ensure the ontology remains accurate, relevant, and aligned with evolving domain knowledge and usage needs. In accordance with the LOT methodology, this phase involves identifying and managing changes to the ontology over time, including bug fixes, extensions, deprecations, and updates to align with external vocabularies or standards.

To support maintainability, we have adopted the following strategies:

- Versioning: We adopt semantic versioning to maintain a clear versioning scheme, allowing users to track changes and ensure backward compatibility where possible.
- Issue tracking: We utilize GitHub as a public issue tracker to collect feedback, report issues, and suggest improvements, thereby fostering community involvement.
- Documentation updates: All updates to the ontology are reflected in its human-readable documentation, which is versioned in parallel with the ontology file.
- Publishing updated versions: New versions are published using persistent URIs and are archived to ensure long-term availability and transparency.

These practices ensure that the ontology can evolve in a controlled and transparent manner, enabling its continued reuse and integration within broader semantic ecosystems.

KG4CUT: Overview

`KG4CUT` is an application ontology tailored to the solid-carbide end-mill domain. It serves as a unified framework to represent cutting tools, their parameters, suitable materials, recommended operations, and geometric machining features. The ontology reuses and aligns with established standards and vocabularies, including ISO 13399 (for tool parameters) and ISO 15608 (for material classification).

Ontology architecture

The ontology follows a polyhierarchical modular design where the major classes and relationships are illustrated in Figure 1. The ontology builds on established ontology design patterns, reusing the odp:Object class as a foundational anchor point for mapping to any of the upper-level ontologies summarised in Table 1.

To improve clarity and facilitate the presentation of the ontology in this paper, we introduce a conceptual modularization of `KG4CUT`. This division serves as a narrative device to guide the reader through the ontology's main components. Additionally, to preserve the paper's readability, we report the formal axioms defining the ontology in Section A. References to individual axioms throughout the text are denoted using the format `[A<N>]`.

Cutting tool core

At the heart of the ontology lies the class CuttingTool, representing any tool used in machining. This is specialized into MillingTool, which is further subdivided into SolidEndTool and ModularTool, to differentiate between the former mills made of just one piece of hard metal and the latter composed of at least two parts.

Each tool is associated with a unique identifier (ordering number), a manufacturer, and a set of recommended usage parameters.

Manufacturers are represented through the class CuttingToolManufacturer, modeled as a subclass of foaf:Organization and equivalent to org:Organization. Tools and manufacturers are connected via the object properties :manufacturesCuttingTool and its inverse :hasCuttingToolManufacturer.

All physical properties relevant for tool selection have been included as datatype properties in KG4CUT. Geometric characteristics include: :cutterDiameter for tool diameter, :hasLength for overall length, :hasNumberofFlutes and :hasFlutesHelixAngles for the number and angles of flutes, and :hasCornerRadius and :hasBCH for corner radius dimensions. Ramp angles are represented by :maxRampAngleDegree and :minRampAngleDegree. Tool-holder compatibility is specified with :shankType, while surface treatment is given by :hasCoating. Lubrication requirements are represented by :hasUseofCoolant. To support tool procurement from vendor catalogues, additional properties have been included: :hasCatalogueNumber, :hasCataloguePage, and :orderingNumber, enabling users to locate and order the corresponding commercial tool models.

To represent usage conditions, we define the class CuttingParameters, which includes datatype properties such as :hasAxialDepthOfCut, :hasRadialDepthOfCut, :feedRate, :hasFeedPerTooth, and three variants of cutting speed: :hasRecommendedCuttingSpeed, :hasMinimumCuttingSpeed, and :hasMaximumCuttingSpeed, accommodating both range-based and

---

[11] https://github.com/dgarijo/Widoco/

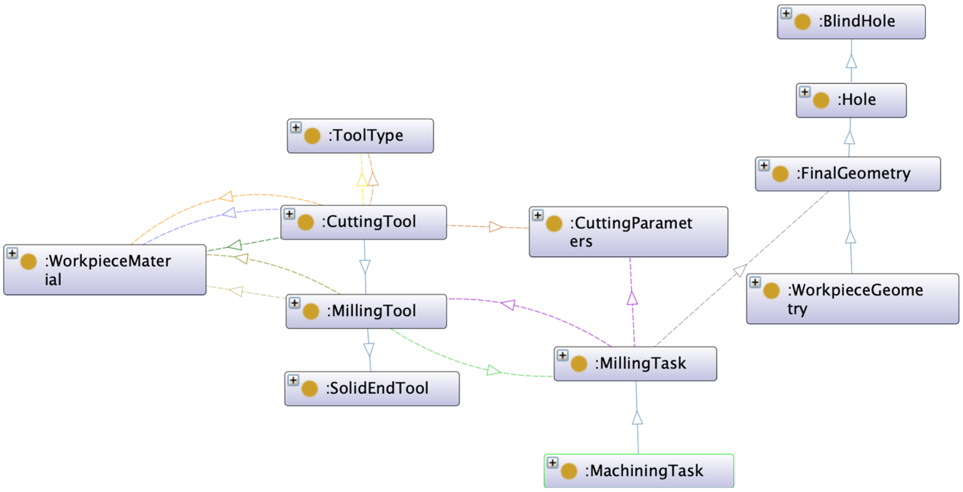


Fig. 1 `KG4CUT` compact schema: main classes and relationships

point-based specifications commonly employed by different vendors.

Additionally, the tools are linked to a ToolHolder, formalized through the MachineToolComponent class [A4].

Tool functional categories

To reflect practical classifications in industry, `KG4CUT` introduces the class ToolType, capturing a tool's functional role, such as Chamfering, Finisher, or Rougher. Since tools may simultaneously belong to multiple categories, we chose to represent functional types as distinct classes rather than as properties.

The property :hasToolType establishes a link between a CuttingTool and its functional role(s). This modeling choice promotes semantic clarity and enables reasoning over the alignment between tool capabilities and machining tasks. Axioms [A9–A11] formalize the alignment between tool types and specific MillingTask instances. Specifically, Axiom 9 states that all chamfering processes involve a chamfer milling task. However, :Finisher and :Rougher can be applied to any different MillingTasks [A10-A11]. This modeling supports multifunctionality and tool reuse under controlled parameter settings.

Materials and machine conditions

The class WorkingPieceMaterial models workpiece materials, organized according to ISO 15608 groupings. Each MillingTool is associated with one or more compatible materials [A5] and suited machining tasks [A6], such as SlotMilling, ChamferMilling, and 3DProfiling.

To support tool selection logic, we distinguish between preferred and optional applications via the properties :hasPreferredOperationType and :hasOptionalOperationType.

We defined four subclasses under the SolidEndTool class: BallNose, Chamfered, Flat-end, and Radiused, with the recognition that this hierarchy can be further extended to accommodate additional tool types such as micro milling or taper tools.

Tool and machine components

The ontology incorporates machine-level concepts from the InVor ontology (Rehage & Gausemeier, 2015). The class MachineTool, subclass of odp:Object, is linked to its components through odp:hasPart. The class MachineToolComponent generalizes elements like Toolholder, Axis, MachineToolTable, Control, Magazine, and MainDriver, contributing to axiom [A6].

Milling tasks and output geometries

Machining operations are modeled using the class MillingTask, which defines a taxonomy of milling operations, including EndMilling, SlotMilling, and ChamferMilling.

In addition to providing a taxonomy of milling operations, the MillingTask class supports a more flexible, process-oriented representation of machining activities, thereby overcoming the limitations of rigid, static structures often found in ontology-based models. By leveraging the object properties :hasInput and :hasOutput, it enables the explicit modeling of tasks as transformations from specific inputs to defined outputs. In our case, each task takes as input a MillingTool, a set of CuttingParameters, and a RawGeometry, and produces as output a FinalGeometry. These semantic relationships are formally encoded in Axioms 11-14 A and implemented in the ontology. The ontology defines two geometry classes: RawGeometry (pre-machining) and FinalGeometry (post-machining), both subclasses of WorkpieceGeometry. Output geometries are further specialized into features such as Pocket, Shoulder, Hole, and Chamfer. These are intentionally modeled in alignment with industrial terminology, rather than relying on formal topological models.

Geometric features are linked to suitable tasks via isPrimaryMachinedBy. Examples include:

- 3DProfile → 3DProfiling [A16]
- Chamfer → ChamferMilling [A17]
- BlindHole → Ramping [A18]

The ontology also encodes more complex machining scenarios that require multi-step operations. For instance, producing a HoleWithTolerance typically involves both Drilling and a subsequent finishing process such as BoringMilling or HoleMilling, to satisfy precision requirements [A19].

In addition, certain concepts are defined using domain-specific heuristics. A notable case is LongHoleWithTolerance, modeled according to the widely adopted industrial rule of thumb whereby a hole is considered "long" if its depth exceeds twice its diameter [A20]. Although such heuristics lack formal topological grounding, they reflect established manufacturing practices and enhance the ontology's applicability in real-world decision-making contexts.

Manufacturing strategies

To reflect the sequence and complexity of real manufacturing processes, the ontology distinguishes among similar geometries produced at different stages. For instance:

- PocketCreation models the initial formation of a cavity, typically performed using combinations of Drilling, CircularMilling, PlungeMilling, or CircularRamping [A22].
- PocketWidening represents the enlargement of a pre-existing pocket, best achieved through BoringMilling, CircularRamping, or CircularMilling [A23].

The ontology encodes similar associations for planar and slot features. For instance, a Plane is best machined using FaceMilling [A22], while features such as ShallowSlot, DeepSlot, or NarrowSlot are linked to suitable strategies like PlungeMilling, PeckMilling, or LinearRamping, depending on accessibility and tolerance requirements [A26-A30].

Ontology evaluation

To assess the quality, completeness, and practical utility of the proposed ontology, we conducted a multi-dimensional evaluation covering structural consistency, competency question coverage, FAIR compliance, and expert-based validation.

Coherence and consistency

As described in the methodology, each sprint concluded with a validation step to ensure the ontology's coherence and consistency. We used the HermiT reasoner (Glimm et al., 2014), a standard tool for checking OWL ontologies, which performs tableau-based reasoning over Description Logic (DL) axioms to detect inconsistencies, infer hierarchies, and verify constraints.

In addition, we evaluated the ontology using the OOPS! (Ontology Pitfall Scanner) tool (Poveda-Villalón et al., 2014), which automatically detects common modeling pitfalls. The analysis reported no issues, confirming the structural soundness and quality of the ontology.

Completeness

To assess the adequacy and completeness of the `KG4CUT` ontology, we verified that all Competency Questions (CQs) defined in Table 2 could be translated into SPARQL queries to ensure that the ontology fully captured the requirements. We began by querying the physical properties of a cutting tool, which are summarized in Listing 1. These queries can be extended as the ontology evolves, since it includes a growing set of datatype properties that describe tool characteristics in more detail.

```
# What are the characteristics of a
    mill?
SELECT ?t ?l ?d ?f
WHERE {
  KG4CUT:Sandvik rdf:type owl:
      NamedIndividual .
  KG4CUT:Sandvik rdf:type KG4CUT:
      CuttingToolManufacturer .
  KG4CUT}:Sandvik KG4CUT:
      manufacturesCuttingTool ?t .
  ?t KG4CUT:hasLength ?l ;
     KG4CUT:hasDiameter ?d ;
     KG4CUT:hasNumberOfFlutes ?f .
}

# What is the type of shank associated
    with this mill?
SELECT ?t ?shankType
WHERE {
  ?t KG4CUT:shankType ?shankType .
}
```

Code Listing 1 SPARQL queries for milling physical properties.

Listing 2 gathers the SPARQL queries to obtain not only the preferred material, but also for the alternative ones:

```
# What materials can be machined by a
    certain mill?
SELECT ?mat1 ?mat2
WHERE {
  KG4CUT:H1TE4CH0200R005HAM KG4CUT:
      preferredMat ?mat1 .
  KG4CUT:H1TE4CH0200R005HAM KG4CUT:
      altMat ?mat2 .
}
```

Code Listing 2 SPARQL query related to preferred and alternative machinable materials.

Snippet 3 presents the SPARQL query used to retrieve cutting parameters, with roughing shown as an example. However, the same query structure can be applied to finishing operations and further refined by adding constraints as needed.

```
# Which cutting parameters would you
    suggest in roughing?
SELECT ?t ?params ?vc ?fz ?sv ?cd
WHERE {
  ?t KG4CUT:prefToolType "rougher" .
  ?t KG4CUT:hasCuttingParameters ?
      params .
  ?params KG4CUT:maxCuttingSpeed ?vc ;
          KG4CUT:feedRate ?fz ;
          KG4CUT:startingCuttingSpeed ?
              sv ;
          KG4CUT:cuttingDepth ?cd .
}
```

Code Listing 3 SPARQL query: recommended cutting parameters for roughing (Note: replacing "rougher" by "finisher") the second part of the CQ is obtained.

In listing 4, we queried information about one of the individuals added to the ontology, to verify whether the data retrieval fulfilled our intended objectives.

```
#What are the operations that can be
    made with this mill?
SELECT ?toolType ?op
WHERE {
  :H1TE4CH0200R005HAM :prefToolType ?
      toolType .
  :H1TE4CH0200R005HAM KG4CUT:
      hasPreferredOperationType ?op .
}
```

Code Listing 4 SPARQL query: operations and capabilities of a mill.

The linkage between the geometry and the milling tool was implemented in the final sprint using the `:hasInput` and `:hasOutput` properties. These enable the retrieval of cutting tools capable of producing the required geometry, as illustrated in snippet 5.

```
# Which mills are good for machining a
    pocket?
SELECT ?t ?mt
WHERE {
  ?mt KG4CUT:hasOutput KG4CUT:Pocket .
  ?mt KG4CUT:hasInput ?t .
}

# Which mills are good to machine a
    groove/slot?
SELECT ?t ?mt
WHERE {
  ?mt KG4CUT:hasOutput KG4CUT:Slot .
  ?mt KG4CUT:hasInput ?t .
}
```

Code Listing 5 SPARQL queries: final geometry of the workpiece.

Ontology fairness

Once the ontology was finalised and published, we evaluated its compliance with the FAIR principles (Findability, Accessibility, Interoperability, and Reusability). For this purpose, we used the FAIR-Checker tool[12] (Gaignard et al., 2023), which assesses the FAIRness of digital resources based on Semantic Web standards and metadata quality.

The evaluation yielded a FAIRness score of 75%, indicating good alignment with the principles summarised in Table 3. The tool, using the FAIRification Framework, provided detailed feedback on areas of strength, such as the use of persistent URIs and machine-readable metadata, and on opportunities for improvement, including enhanced discoverability through richer metadata linking. Figure 2 provides a visual representation of the FAIR score distribution across the four FAIR dimensions. The chart shows strong performance in Accessibility and Interoperability, both above 80%, indicating that the ontology is well exposed and aligns with external standards. Findability also scores relatively well, whereas the Reusability dimension is the weakest, indicating a need to improve provenance metadata to facilitate future reuse.

Application of KG4CUT to the cutting tool selection

To demonstrate the utility of the developed ontology, we present a case study focused on the cutting tool selection process. We used real-world catalogues from two manufacturers (Sandvik Coromant and Kennametal) and populated the `KG4CUT` ontology to build a queryable knowledge graph.

---

[12] https://fair-checker.france-bioinformatique.fr

Table 3 FAIR scoring summary table

FAIR Principle	Criterion	Score (2)
Findable	Unique Identifier: URL is reachable and unique.	2/2
	Persistent Identifier: at least one identifiers.org namespace present.	1/2
	Structured Metadata: access-policy property present.	2/2
	Shared Vocabularies: some ontology terms recognised in major registries.	1/2
Accessible	Open Resolution Protocol: resource accessible via HTTP.	2/2
	Access Rights Specified: rights/ licence properties included.	2/2
Interoperable	Machine-Readable Format: access-policy property present.	2/2
	Use of Shared Ontologies: partial use of recognised ontology terms.	1/2
	External Links: at least three external domains referenced.	2/2
Reusable	License Included: licence property from accepted vocabularies present.	2/2
	Community Standards: partial use of ontology terms in major registries.	1/2
	Provenance Metadata: no provenance properties found.	0/2

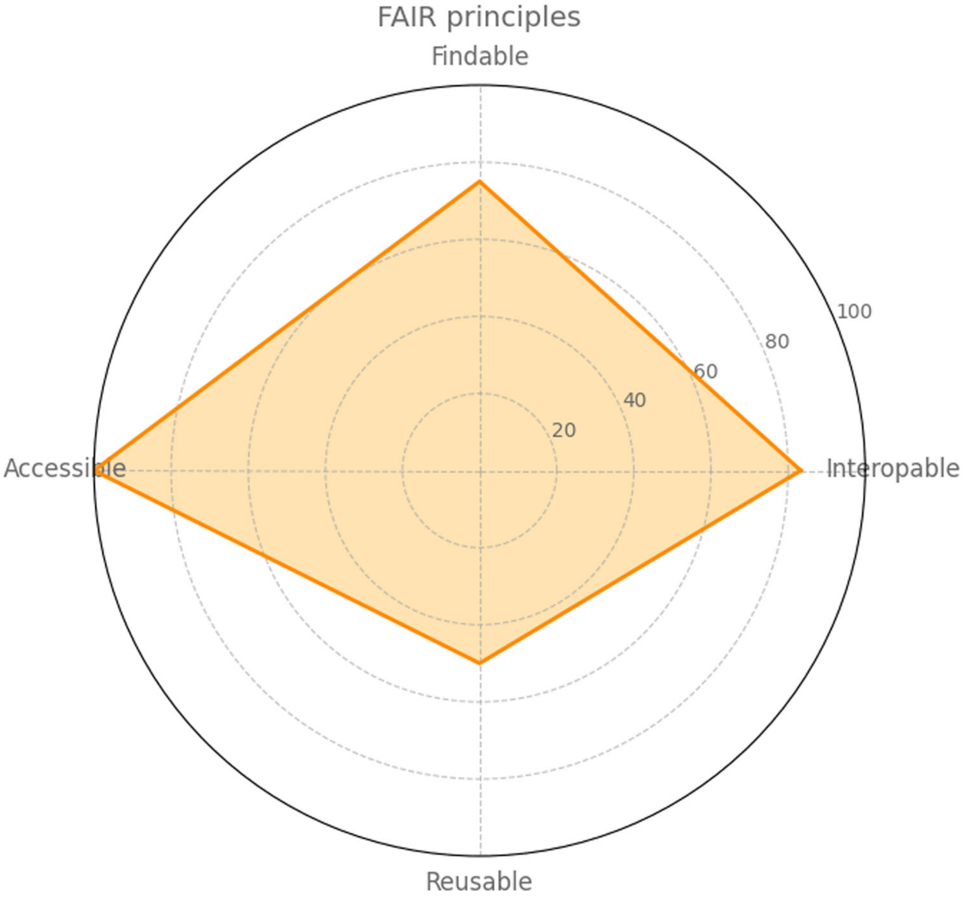
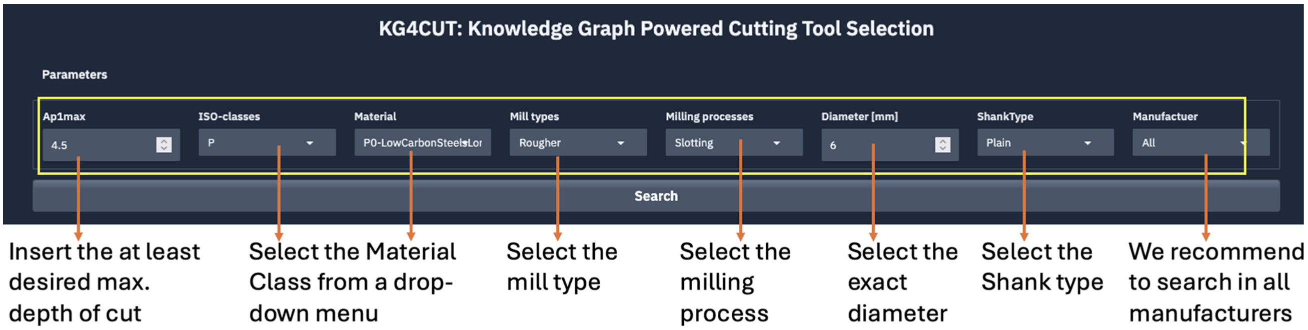


Fig. 2 Spider-chart that assesses the alignment of the ontology to the four FAIR dimensions: Findable, Accessible, Interopable and Reusable

Data extraction and transformation

The population process comprises two main steps: (1) extraction of structured data from PDF catalogues, and (2) mapping of the extracted data to RDF triples based on the ontology schema.

Cutting tool catalogues are typically available as unstructured PDF documents that combine textual descriptions with tabular data. To extract relevant information, we developed a catalogue-specific pipeline that includes layout-aware parsing and data cleaning procedures. The parsing process



KG4CUT: Knowledge Graph Powered Cutting Tool Selection

Parameters

Ap1max: 4.5
ISO-classes: P
Material: P0-LowCarbonSteelsLo
Mill types: Rougher
Milling processes: Slotting
Diameter [mm]: 6
ShankType: Plain
Manufactuer: All

Insert the at least desired max. depth of cut

Select the Material Class from a drop-down menu

Select the mill type

Select the milling process

Select the exact diameter

Select the Shank type

We recommend to search in all manufacturers

Fig. 3 Excerpt of the web application's user interface powered by `KG4CUT`, used during the evaluation study

was implemented using the `pdfplumber` Python library[13], which supports the programmatic extraction of tables as structured lists or data frames. However, due to the high variability in catalogue layouts, we also defined custom, catalogue-specific rules to improve the accuracy of table detection and processing.

The extracted content, mainly tool specifications and cutting parameters, was converted into an intermediate JSON format. In the second step, this JSON data was systematically mapped to RDF triples. Each cutting tool instance was represented as an individual of the `KG4CUT`:CuttingTool class, with associated data and object properties derived from the catalogue. The mapping also captured references to materials, geometric features, and operational parameters. All literals were typed using XML Schema Datatypes, and nested structures (e.g., cutting parameters) were translated into linked individuals of the corresponding ontology classes.

While the PDFs used as source catalogues are natively digital and do not involve OCR errors, data uncertainty can arise from structural variations in table layouts, which may lead to incorrect mapping of numerical values to their corresponding keys. To mitigate such issues, after populating the knowledge graph we performed systematic validation by executing representative SPARQL queries and inspecting the resulting triples. This process allowed us to identify and correct any inconsistencies, such as repeated properties or values not present in the original catalogue, ensuring the reliability and semantic integrity of the RDF graph.

The result is a semantically enriched RDF graph aligned with the `KG4CUT` ontology, ready for use in querying and reasoning.

Domain experts validation

To validate and assess the practical utility of the proposed ontology and its corresponding knowledge graph, we developed a web-based application that enables users to efficiently search for cutting tools using predefined input constraints. Figure 3 shows a portion of the application's user interface, which was employed during the user study.

To simulate a realistic machining scenario where an additional geometric feature must be produced on a workpiece (see Figure 4), we conducted a structured survey. Participants with at least intermediate expertise in machining were asked to identify and select the most appropriate cutting tools for the task, based on the system-provided constraints.

We defined two use cases related to a common practice of remanufacturing a workpiece in batches. In Case 1, shown in 4b, the experts needed to determine the optimal cutting tool for machining the missing, highlighted blue feature in the workpiece. For Case 2, in 4c, the new feature highlighted in blue is a cavity. For simplicity, we present the final workpiece, whose dimensions are shown in 4a, which is necessary to determine the cutting tool capable of performing both tasks. The users were asked to complete this task first using traditional vendor catalogues and subsequently using the web application described earlier, which is powered by the `KG4CUT` ontology.

This comparative approach allowed us to evaluate the usability and effectiveness of the ontology-based tool in supporting decision-making in realistic industrial contexts.

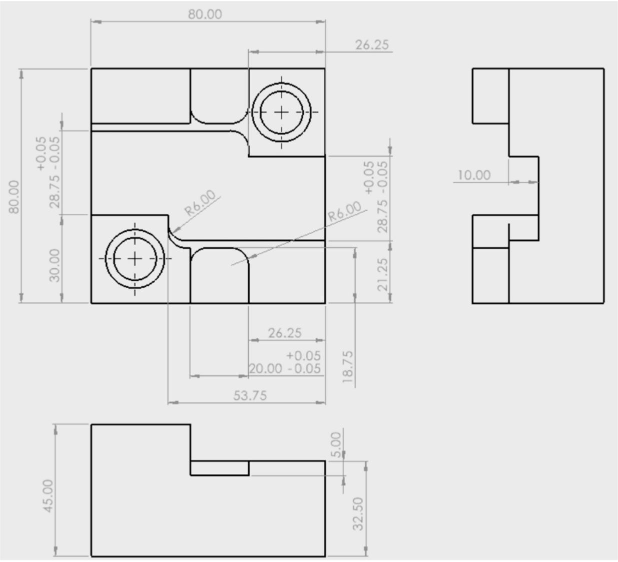
Our survey was distributed to anonymous participants in the manufacturing departments of institutions and companies in two European countries. Respondents included students who had completed a module on cutting tools, professors specializing in machine tools, and industry experts trained in chip-removal processes and fixtures, ensuring medium to high levels of expertise. In total, 21 individuals with diverse backgrounds but substantial knowledge of the machine tool industry, particularly cutting tools, participated. The evaluation followed a heuristic approach (Nielsen & Molich, 1990), relying on expert judgment to identify potential issues.

Expert survey results

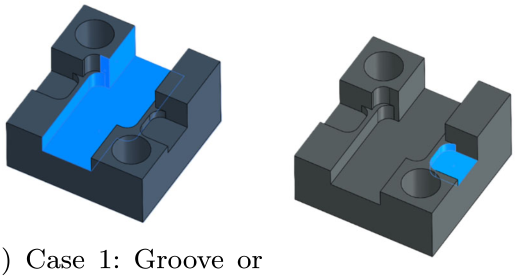
The expert survey was designed to evaluate the practical utility of our ontology-driven approach across three key dimensions: (i) its effectiveness in assisting users in select-

---

[13] https://github.com/jsvine/pdfplumber



(a) 2D Drawing of the dimensions to be achieved.



(b) Case 1: Groove or open slot to be manufactured

(c) Case 2:New Cavity to be machined

Fig. 4 Workpiece used for the expert validation

ing suitable cutting tools, (ii) its impact on reducing the time required for the selection process, and (iii) its ability to facilitate interoperability across different tool vendors.

Validation of the selected cutting tools

A key observation was that participants selected different tools for the same tasks, reflecting the complexity and subjectivity involved in manual tool selection.

To assess the validity of the selected cutting tools, we categorized the responses into three groups:

- **Good**: Tools that could fully machine the geometrical feature and meet the required dimensions and tolerances.
- **Partially Effective**: Tools that successfully performed aspects of the geometrical feature but were not yet able to complete the entire task.
- **Bad**: Tools that were entirely inappropriate for the task.

Experienced professionals and academics with substantial expertise in the field conducted the evaluation.

Figure 5 compares the quality of selected cutting tools across two machining scenarios (Case 1 and Case 2), using either traditional vendor catalogues or the `KG4CUT`-powered application.

Where it is possible to see that KG4CUT increased effective outcomes in Case 1 from 66.7% to 81.0% and reduced bad outcomes from 33.3% to 19.0%, corresponding to a 42.9% relative error reduction in the selection of the tools. In Case 2, KG4CUT increased the proportion of "Good" outcomes from 14.3% to 19.0%, representing a 33% relative improvement in high-quality outcomes without increasing error rates.

Results suggest that semantic guidance helps users consistently avoid unsuitable tools.

While the majority of `KG4CUT` selections were classified as Partially Effective, this does not indicate a failure of the ontology; rather, it reflects the practical constraints of the use case. In many real-world scenarios, a single best-fit tool may not be easily identifiable due to trade-offs in dimensions, tolerances, or available vendor information.

`KG4CUT` aims to narrow the search space and prevent clearly unsuitable choices, offering a list of semantically relevant candidates. In doing so, it increases the likelihood of a correct or acceptable selection, while still requiring user expertise for final validation.

Tool selection and ease of use

When asked which method facilitated the selection process, 16 out of 21 participants preferred `KG4CUT`. The remaining five found both approaches equally effective. Participants highlighted the system's intuitive interface, reduced effort, and speed of use, especially when compared to navigating multiple catalogues manually.

Figure 6 reports the average completion time for each case.

KG4CUT significantly reduced task completion time compared to the catalogue-based approach. In Case 1, mean completion time decreased from $\approx$ 22 minutes to 8 minutes, corresponding to a 63.6% reduction . In Case 2, mean time decreased from $\approx$ 8 minutes to 5.5 minutes, representing a $\approx$ 30.0% reduction.

Users completed tasks significantly faster using `KG4CUT`, confirming the platform's potential to streamline decision-making.

Several participants also emphasized that the interface required minimal learning effort and became intuitive after only a few interactions.

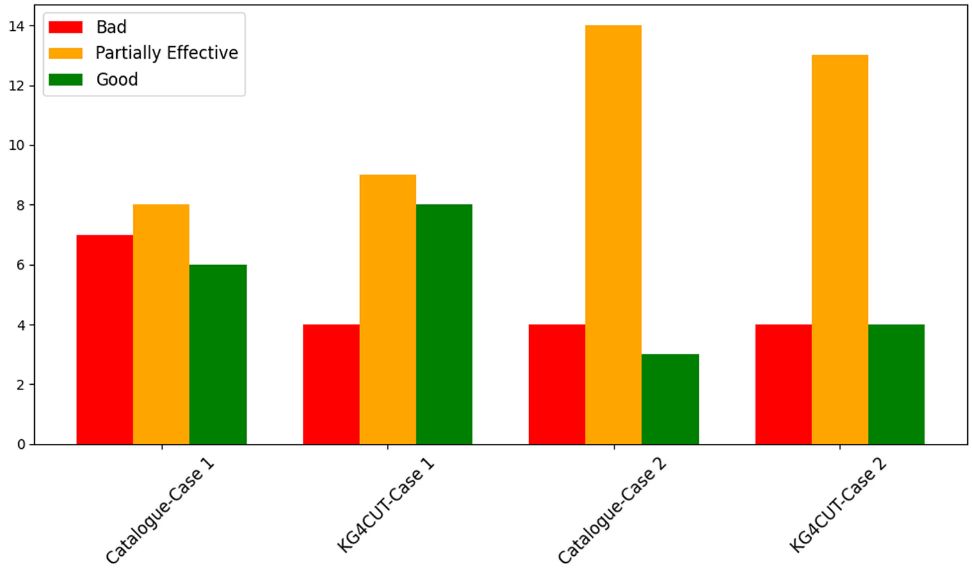


Fig. 5 Bar chart comparing number of *Good*, *Partially Effective* and *Bad* tools retrieved by `KG4CUT` and human manual selection

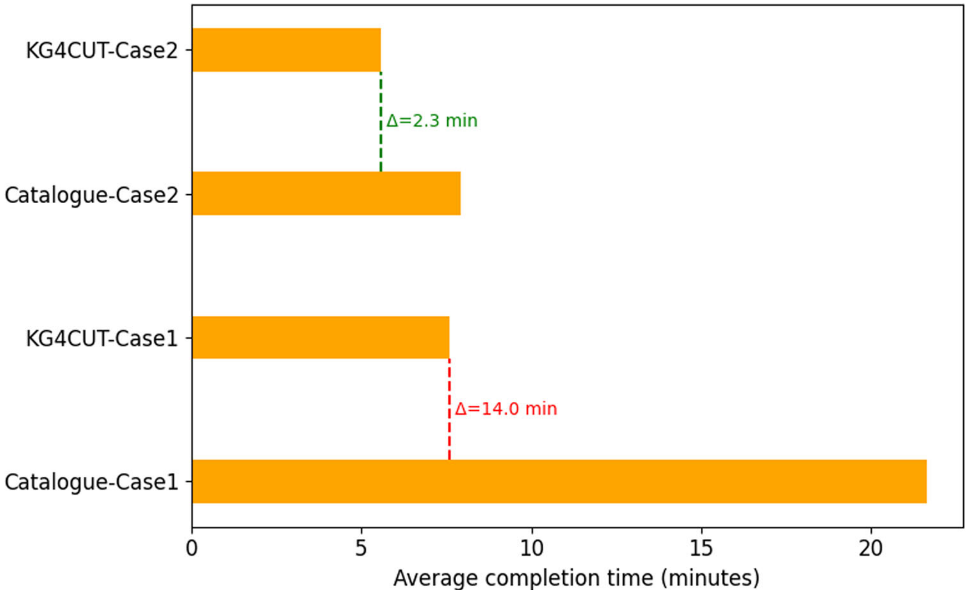


Fig. 6 Comparison in terms of average completion time of the defined use cases between `KG4CUT` and manual selection from catalogues

Handling of vendor catalogues

Managing data from different tool vendors is a known challenge. While participants reported varying levels of difficulty when using traditional catalogues, all agreed that `KG4CUT` significantly improved catalogue interoperability. This capability was seen as one of the most valuable features of the platform. {Figure 7 shows that, when using catalogues, users tend to consistently select tools from the same vendor. In Case 1, for example, 90.5% of users selected Kennametal. In contrast, when using `KG4CUT`, 57.1% of users selected it, thereby increasing vendor interoperability and resulting in a more balanced distribution of vendor choices. Overall, `KG4CUT` increased the diversity of vendors considered by 33.4% in Case 1 and 14.3% in Case 2, compared to catalogue-based selection. This consistent positive response underscores one of the most valuable contributions of `KG4CUT`: reducing the cognitive and practical burden of comparing tools across fragmented vendor systems. demonstrating its role in broadening user choices beyond the offering of a single vendor.

Support and user interface feedback

In the survey, users were asked to evaluate the support provided by KG4CUT on a scale from 1 to 5. The system received a high average score of 4.3, indicating strong satisfaction with its guidance and responsiveness. Participants also provided actionable suggestions for improvement. Several requested additional visual aids, such as tool images,

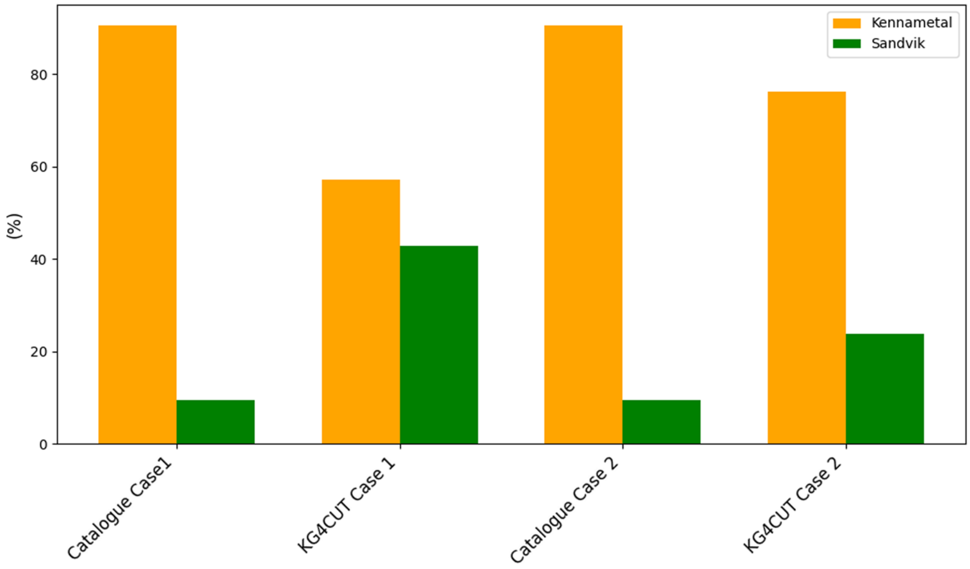


Fig. 7 Bar plot showing the percentage of selected tools belonging to Sandvik and Kennametal for each use case. Each bar represents the proportion of users choosing a particular manufacturer across the different test cases

graphical explanations of parameters, and a more engaging interface. One participant noted that the interface was too plain and could benefit from more color and visual elements, while still appreciating the overall design. Others suggested clarifications on key parameters (e.g., *Is D the effective cutting diameter?*) and the inclusion of catalogue links, as well as tool productivity and price ratings in the results. We acknowledge that the current user interface is basic; however, the main objective was to assess whether experts could recognise the value of KG4CUT even in this minimal interface. As a future step, we plan to apply human-computer interaction principles to create a more intuitive and visually appealing interface.

Adoption interest and educational potential

Participants were also asked whether they would be interested in a version of `KG4CUT` for their company or academic institution. Fifteen of twenty one expressed interest in adopting the platform for industrial use or classroom settings. The feedback suggests that `KG4CUT` is perceived not only as a research prototype, but also as a promising tool with practical potential.

Discussion

Cutting tool selection remains a highly complex task, often dependent on the operator's experience and constrained by the fragmented structure of vendor data. The reduction in the number of professionals (machinists) working in chip-removal processes is increasing hourly rates and delaying production; new approaches are seeking to retain knowledge using AI. While artificial intelligence has long been applied to manufacturing problems, recent interest has shifted toward semantic technologies and ontologies as mechanisms for explicit knowledge representation and interoperability. Unlike data-driven approaches, ontology-based systems emphasize structured reasoning, transparency, and standards compliance qualities that are particularly relevant in safety-critical and engineering domains. Despite extensive prior work on manufacturing ontologies, our analysis revealed the absence of a comprehensive, standards-aligned semantic model dedicated to milling cutting tools. `KG4CUT` addresses this gap by providing a unified, machine-interpretable representation that integrates cutting tools, machining operations, materials, and geometric features in a single semantic framework. The ontology has been created and evaluated by experts in the milling and chip-removal process, thereby enhancing its reliability in the domain modelled and its usability. Beyond filling a structural gap, the conducted expert study enables several broader insights into the nature of cutting tool selection and the role of semantic technologies in this domain. The variability observed in expert selections indicates that multiple acceptable solutions often coexist, shaped by trade-offs among geometric constraints, tolerances, material properties, and catalogue availability. This finding challenges approaches that frame tool selection as the search for a single optimal solution. Instead, `KG4CUT` demonstrates that practical support is best achieved by identifying feasible solution spaces and eliminating incompatible options, rather than prescribing a definitive choice. By externalizing tacit machining knowledge into a formal semantic structure, `KG4CUT` reduces the need for extensive hands-on experience when selecting cutting tools. Rather than replacing expert judgment, the system helps less-experienced users make informed decisions while supporting expert workflows. This insight is particularly relevant in light of current workforce challenges in manufacturing, where knowledge retention and transfer are becoming critical concerns. The ability to compare tools from different manufacturers indicates that interoperability challenges are mainly related to differences in how tool parameters are defined and interpreted. By representing vendor catalogues within a shared semantic model, `KG4CUT` enables consistent reasoning across tools that are otherwise difficult to compare. These findings aligns `KG4CUT` with the emerging paradigms of Trustworthy and Explainable AI (XAI) within the Industry 5.0 framework. While data-driven 'black-box' models often yield recommendations lacking transparent logic, an ontology-based approach ensures that the selection process remains inherently interpretable. As argued by Zhou et al. (2026), Knowledge Graphs are pivotal in Industry 5.0 as they provide a structured reasoning layer that fosters human-centric collaboration and trust. By explicitly representing the relationships between tool geometries, materials, and machining constraints, `KG4CUT` allows operators to understand the why behind a specific suggestion. This transparency is fundamental for safety-critical manufacturing environments, where explainability is not just a technical feature but a requirement for building trust between human experts and autonomous systems.

Overall, `KG4CUT` illustrates how formal knowledge representation can meaningfully support complex engineering decisions without attempting to replace human expertise. By framing tool selection as a semantic reasoning problem grounded in constraints and interoperability, the proposed approach provides a strong foundation for future integration within digital manufacturing and Industry 5.0 ecosystems.

Limitations

The main limitation of this study is the current scope of the knowledge graph, which is limited to solid carbide milling tools. Although the `KG4CUT` ontology is designed to be extensible, the initial implementation focuses on a subset of the machining domain to ensure data consistency and manageable catalogue parsing complexity. In particular, we excluded modular tools, whose catalogue structures and parameter definitions vary considerably and would require other custom parsing logic. Moreover, the ontology does not yet cover the full range of geometric features or machining strategies that may be relevant in more advanced or niche use cases. Instead, we focused on the most commonly encountered features and operations, as reflected in industrial best practices and expert input. This prioritization allowed us to validate the core modeling framework without overcomplicating the initial implementation. We have covered the most important and commonly used properties of cutting tools, except wear, which is difficult to handle because it depends on cutting conditions and varies. Unfortunately, ontologies are static, which limits their ability to represent this condition. Furthermore, the reasoning framework of `KG4CUT` in its current form is primarily centered on symbolic logic and taxonomic constraints. While effective for categorization, this approach does not yet incorporate physical knowledge models that capture the dynamic phenomena of the machining process, such as cutting forces, thermal distributions, or vibration patterns. Moving beyond purely symbolic reasoning toward the integration of physics-based insights would be a key step to support the high-fidelity simulations and real-time optimizations required by next-generation digital twin environments.

Finally, while the expert evaluation confirms the utility of the system, the sample size remains small, but very significant to the studied field as we have obtained feedback from experts. Broader validation across additional user groups and industrial settings would strengthen generalizability and help identify edge cases or usability barriers not observed in the current study.

Despite these limitations, `KG4CUT` provides a strong foundation for future expansion and offers a formal, machine-interpretable representation of cutting tool knowledge that has traditionally been under-structured and difficult to reuse across platforms and vendors.

Future work

The future work is directly motivated by the limitations observed in the current version of `KG4CUT`. A first direction involves the development of a new ontology version with improved FAIR compliance, particularly regarding provenance. This version will also include additional datatype properties for cutting tools, such as measures to estimate optimal tool types for fine finishing and expected tool wear, as well as the inclusion of cutting tool model URLs. Furthermore, additional machining features will be considered to enrich the ontology's expressiveness. A second line of work concerns the user interface, which, while currently aesthetically designed, could be made more user-friendly to improve acceptance and usability. Moreover, integrating `KG4CUT` with cyber–physical production systems would enable a closed feedback loop in which real-time machining and cutting-tool data collected from the shop floor enriches the ontology with empirical operational knowledge. This integration provides a strategic opportunity to explore the synergy between semantic representations and Physics-Informed Machine Learning (PIML). PIML facilitates the integration of physical laws to direct model training (Guo et al., 2026; Leng et al., 2025), thereby augmenting accuracy, interpretability, and robustness, especially in engineering contexts where data can be challenging to compile. By embedding physical knowledge such as thermal distributions or cutting forces, into the reasoning engine, the framework would evolve toward a simulation-integrated digital twin that respects fundamental machining laws as done in Yang et al. (2025). Another prominent future work direction is the integration of Generative AI. By leveraging the structured semantic foundation of `KG4CUT`, Large Language Models can be empowered through Retrieval-Augmented Generation to provide intuitive, natural-language interfaces for tool selection (Leng et al., 2026). Specifically, the KG can act as a "grounding" mechanism to mitigate hallucinations in AI-generated recommendations, ensuring that tool suggestions remain compliant with strict engineering constraints.

Finally, a key line of work focuses on integration with CAM and PLM systems. This includes securing collaboration with a company (e.g., a cutting-tool manufacturer or machining firm) to test, refine, and populate the knowledge graph with real-world company data. Such collaboration would allow the expansion of tool types and the mapping of `KG4CUT` concepts to CAM operations and PLM process objects, including STEP-NC entities, vendor-specific CAM schemas, and tool libraries exposed via company APIs. The current KG4CUT version advances the Technology Readiness Level (TRL) assessment from an initial 1-2 to 4, reflecting validated prototype demonstrations in a relevant environment. These future integrations can further progress to TRL 5-6 via systematic industrial validation.

Conclusions

The well-known phrase *"Knowledge is power"* is particularly pertinent in manufacturing, where optimizing time, cost, and quality is essential. This study addressed the inherent complexity of machining processes, an area increasingly pressured by stricter sustainability and quality requirements, rapid production demands, and the ongoing erosion of skilled labor due to workforce turnover.

Inspired by the successful adoption of ontologies in the Architecture, Engineering, and Construction (AEC) domain, and recognizing the lack of a comprehensive solution in the machining sector, we developed a domain-specific ontology using the Linked Open Terms (LOT) methodology. The resulting ontology, `KG4CUT`, adheres to the FAIR principles as it is Findable, Accessible, Interoperable, and Reusable, and is publicly available. Our approach integrates `KG4CUT` into a knowledge graph, forming the foundation of a web application that delivers structured information from two established cutting tool vendors. This platform enables more informed decision-making through a unified and user-friendly interface.

We evaluated the system using a survey based on two realistic manufacturing scenarios and collected feedback from domain experts. The results confirmed both the usability and value of the tool, with particular appreciation for the structured data presentation and for suggestions to enhance the interface and guidance features in future iterations.

This work marks a progression from *Technology Readiness Level (TRL) 1-2* to *TRL 4*, demonstrating a functional prototype validated by domain experts in conditions comparable to a laboratory environment. The incorporation of expert feedback sets a solid foundation for future development toward *TRL 5*, with potential extensions including broader vendor integration, improved UI/UX, application to adjacent manufacturing domains and CAM integration.

In summary, this research introduces a novel, ontology-driven approach to supporting complex industrial decision-making. By formalizing and structuring domain knowledge, we take a decisive step toward more intelligent, interoperable, and scalable manufacturing systems.

Logical Axioms (Summary)

This Appendix lists the ontological logical axioms adopted in this work, in support of the theoretical assumptions and formalizations developed in the main body of the paper.

Axiom 1 Every CuttingTool has some CuttingParameters.

$$\forall x, (CuttingTool(x) \rightarrow \exists y, (hasCuttingParameters(x, y) \wedge CuttingParameters(y)))$$

Axiom 2 Every CuttingTool has a CuttingToolManufacturer.

$$\forall x, (CuttingTool(x) \rightarrow \exists y, (hasCuttingToolManufacturer(x, y) \wedge CuttingToolManufacturer(y)))$$

Axiom 3 Every CuttingTool has a unique ordering number.

$$\forall x\ (CuttingTool(x) \rightarrow \exists! y\ (orderingNumber(x, y) \wedge String(y)))$$

Axiom 4 Every CuttingTool is mounted on a ToolHolder.

$$\forall x\ (CuttingTool(x) \rightarrow \exists y\ (isMountedOnToolHolder(x, y) \wedge ToolHolder(y)))$$

Axiom 5 A MillingTool is associated with a machine and workpieceMaterial.

$$\forall x\ (MillingTool(x) \rightarrow \exists y\ (primaryMachines(x, y) \wedge workpieceMaterial(y)))$$

Axiom 6 A MillingTool performs a MillingTask.

$$\forall x\ (MillingTool(x) \rightarrow \exists y\ (performsTask(x, y) \wedge MillingTask(y)))$$

Axiom 7 A MachineTool has at least one MachineToolComponent.

$$\forall x\, (MachineTool(x) \rightarrow \exists y\, (hasPart(x, y) \wedge MachineToolComponent(y)))$$

Axiom 8 A SolidEndTool is valid if its corner radius fits geometry.

$$\forall t, w, r_1, r_2 (SolidEndTool(t) \wedge hasCornerRadius(t, r_1) \wedge WorkpieceGeometry(w) \wedge hasMinCornerRadius(w, r_2) \wedge r_1 < r_2 \rightarrow ValidTool(t))$$

Axiom 9 Chamfering tools perform ChamferMilling.

$$\forall x\, (Chamfering(x) \rightarrow \exists y\, (performsTask(x, y) \wedge ChamferMilling(y)))$$

Axiom 10 Finisher performs a MillingTask.

$$\forall x\, (Finisher(x) \rightarrow \exists y\, (performsTask(x, y) \wedge MillingTask(y)))$$

Axiom 11 Rougher performs a MillingTask.

$$\forall x\, (Rougher(x) \rightarrow \exists y\, (performsTask(x, y) \wedge MillingTask(y)))$$

Axiom 12 MillingTask requires CuttingParameters.

$$\forall x\, (MillingTask(x) \rightarrow \exists y\, (hasInput(x, y) \wedge CuttingParameters(y)))$$

Axiom 13 MillingTask requires a MillingTool.

$$\forall x\, (MillingTask(x) \rightarrow \exists y\, (hasInput(x, y) \wedge MillingTool(y)))$$

Axiom 14 MillingTask requires RawGeometry.

$$\forall x\, (MillingTask(x) \rightarrow \exists y\, (hasInput(x, y) \wedge RawGeometry(y)))$$

Axiom 15 MillingTask outputs FinalGeometry.

$$\forall x\, (MillingTask(x) \rightarrow \exists y\, (hasOutput(x, y) \wedge FinalGeometry(y)))$$

Axiom 16 3DProfile is machined by 3DProfiling.

$$\forall x\, (3DProfile(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge 3DProfiling(y)))$$

Axiom 17 Chamfer is machined by ChamferMilling.

$$\forall x\, (Chamfer(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge ChamferMilling(y)))$$

Axiom 18 BlindHole is machined by Ramping.

$$\forall x\, (BlindHole(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge Ramping(y)))$$

Axiom 19 HoleWithoutTolerance is machined by DrillingTask.

$$\forall x\, (HoleWithoutTolerance(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge DrillingTask(y)))$$

Axiom 20 HoleWithTolerance is machined by Drilling + Boring/HoleMilling.

$$\forall x\, (HoleWithTolerance(x) \rightarrow (\exists y\, (isPrimaryMachinedBy(x, y) \wedge DrillingTask(y)) \wedge \exists z\, (isPrimaryMachinedBy(x, z) \wedge (BoringMilling(z) \vee HoleMilling(z)))))$$

Axiom 21 LongHoleWithTolerance is machined by Ramping.

$$\forall x\, (LongHoleWithTolerance(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge Ramping(y)))$$

Axiom 22 Plane is machined by FaceMilling.

$$\forall x\, (Plane(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge FaceMilling(y)))$$

Axiom 23 PocketCreation can involve various milling tasks.

$$\forall x\, (PocketCreation(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge (CircularRamping(y) \vee (CircularMilling(y) \wedge DrillingTask(y)) \vee (DrillingTask(y) \wedge PlungeMilling(y)))))$$

Axiom 24 PocketWidening uses Boring, Circular or CircularRamping.

$$\forall x\, (PocketWidening(x) \rightarrow \exists y\, (isPrimaryMachinedBy(x, y) \wedge (BoringMilling(y) \vee CircularMilling(y) \vee CircularRamping(y))))$$

Axiom 25 Shoulder is machined by FaceMilling or SideMilling.

$$\forall x\,(Shoulder(x) \rightarrow \exists y\,(isPrimaryMachinedBy(x, y) \\ \wedge (FaceMilling(y) \vee SideMilling(y))))$$

Axiom 26 CloseSlot is machined by EndMilling or Plunge Milling.

$$\forall x\,(CloseSlot(x) \rightarrow \exists y\,(isPrimaryMachinedBy(x, y) \\ \wedge (EndMilling(y) \vee PlungeMilling(y))))$$

Axiom 27 LongSlot is machined by LinearRamping or PeckMilling.

$$\forall x\,(LongSlot(x) \rightarrow \exists y\,(isPrimaryMachinedBy(x, y) \\ \wedge (LinearRamping(y) \vee PeckMilling(y))))$$

Axiom 28 Deep and Open Slot requires both Face and Side Milling.

$$\forall x\,(DeepSlot(x) \wedge OpenSlot(x) \\ \rightarrow \exists y\,(isPrimaryMachinedBy(x, y) \\ \wedge FaceMilling(y) \wedge SideMilling(y)))$$

Axiom 29 NarrowSlot is machined by LinearRamping or PeckMilling.

$$\forall x\,(NarrowSlot(x) \rightarrow \exists y\,(isPrimaryMachinedBy(x, y) \\ \wedge (LinearRamping(y) \\ \vee PeckMilling(y))))$$

Axiom 30 ShallowSlot is machined by EndMilling or Plunge Milling.

$$\forall x\,(ShallowSlot(x) \rightarrow \exists y\,(isPrimaryMachinedBy(x, y) \\ \wedge (EndMilling(y) \\ \vee PlungeMilling(y))))$$

Acknowledgements This work was partially supported by the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5 – Call for Tender No. 3277, published on December 30, 2021, by the Italian Ministry of University and Research (MUR), and funded by the European Union – NextGenerationEU. Project Code: ECS0000038, Project Title: e.INS – Ecosystem of Innovation for Next Generation Sardinia, CUP: F53C22000430001, Grant Assignment Decree No. 1056 adopted on June 23, 2022, by MUR.

We acknowledge additional financial support from the project FIATLUCS, funded under the NRRP RAISE Liguria initiative, Spoke 01, CUP: F23C24000240006.

This work also benefited from funding provided under Ministerial Decree No. 351 of April 9, 2022, based on the NRRP – funded by the European Union – NextGenerationEU – Mission 4 "Education and Research", Component 1 "Enhancement of the offer of educational services: from nurseries to universities", Investment 4.1.

Furthermore, this research was carried out in the context of the DIGIMAN4.0 project (DIGItal MANufacturing Technologies for Zero-defect Industry 4.0 Production, http://www.digiman4-0.mek.dtu.dk/), a European Training Network supported by the Horizon 2020 Framework Programme for Research and Innovation (Project ID: 814225).

Funding Open access funding provided by Università degli Studi di Cagliari within the CRUI-CARE Agreement.

Data Availability The developed ontology is publicly available at the link https://w3id.org/kg4cut

Declarations

Conflicts of Interest / Competing Interests The authors declare that they have no conflict of interest.

Ethical Approval Not applicable.

References

A generic task ontology for scheduling applications. In: Proceedings of the International Conference on Artificial Intelligence (ICAI'2001) (2001)

Abdelghany, A. S., Darwish, N. R., & Hefni, H. A. (2019). An agile methodology for ontology development. *International Journal of Intelligent Engineering and Systems, 12*(2), 170–181. https://doi.org/10.22266/ijies2019.0430.17

Akyazi, T., Goti, A., Oyarbide-Zubillaga, A., Alberdi, E., Carballedo, R., Ibeas, R., & Garcia-Bringas, P. (2020). Skills requirements for the european machine tool sector emerging from its digitalization. *Metals, 10*(12), 1665. https://doi.org/10.3390/met10121665

Ali, M. M., Rai, R., Otte, J. N., & Smith, B. (2019). A product life cycle ontology for additive manufacturing. *Computers in Industry, 105*, 191–203. https://doi.org/10.1016/j.compind.2018.12.00

Ameri, F., & Dutta, D. (2006). An upper ontology for manufacturing service description. In: International design engineering technical conferences and computers and information in engineering conference, pp 651–661, https://doi.org/10.1115/DETC2006-99600.

Borgo, S., & Leitão, P. (2004). The role of foundational ontologies in manufacturing domain applications. *Lecture Notes in Computer Science, 3290*, 670–688. https://doi.org/10.1007/978-3-540-30468-5_43

Brenner, D., Kleinert, F., Imiela, J., & Westkämper, E. (2017). Life cycle management of cutting tools: comprehensive acquisition and aggregation of tool life data. *Procedia Cirp, 61*, 311–316. https://doi.org/10.1016/j.procir.2016.11.168

Duan, Y., Hou, L., & Leng, S. (2021). A novel cutting tool selection approach based on a metal cutting process knowledge graph. *The International Journal of Advanced Manufacturing Technology, 112*(11), 3201–3214. https://doi.org/10.1007/s00170-021-06606-5

Fowler, M., Highsmith, J., et al. (2001). The agile manifesto. *Software development, 9*(8), 28–35.

Gaignard, A., Rosnet, T., Lamotte, D., & Frédéric and Lefort, Vincent and Devignes, Marie-Dominique,. (2023). Fair-checker: supporting digital resource findability and reuse with knowledge graphs and semantic web standards. *Journal of Biomedical Semantics, 14*(1), 7. https://doi.org/10.1186/s13326-023-00289-5

Gangemi, A., & Presutti, V. (2009). Ontology Design Patterns. In: Handbook on Ontologies. Springer Berlin Heidelberg, p 221–243, https://doi.org/10.1007/978-3-540-92673-3_10.

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., & Wang, Z. (2014). Hermit: an owl 2 reasoner. *Journal of automated reasoning, 53*(3), 245–269. https://doi.org/10.1007/s10817-014-9305-1

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition, 5*(2), 199–220. https://doi.org/10.1006/knac.1993.1008

Gruninger, M. (1995). Methodology for the design and evaluation of ontologies. In: Proc. IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing.

Guo, M., Gong, D., Huang, Z., Guo, W., Jiang, X., & Lv, D. (2026). Physics-informed deep adaptive denoising and multi-scale feature fusion method for tool wear prediction. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-026-02837-z

Järvenpää, E., Siltala, N., Hylli, O., & Lanz, M. (2019). The development of an ontology for describing the capabilities of manufacturing resources. *Journal of Intelligent Manufacturing, 30*(2), 959–978. https://doi.org/10.1007/s10845-018-1427-6

Ji, W., Wang, L., Haghighi, A., Givehchi, M., & Liu, X. (2018). An enriched machining feature based approach to cutting tool selection. *International Journal of Computer Integrated Manufacturing, 31*, 1–10. https://doi.org/10.1080/0951192X.2017.1356472

Kjellberg, T., von Euler-Chelpin, A., Hedlind, M., Lundgren, M., Sivard, G., & Chen, D. (2009). The machine tool model-a core part of the digital factory. *CIRP annals, 58*(1), 425–428. https://doi.org/10.1016/j.cirp.2009.03.035

Kumar, V. R. S., Khamis, A., Fiorini, S., Carbonera, J. L., Alarcos, A. O., Habib, M., Goncalves, P., Li, H., & Olszewska, J. I. (2019). Ontologies for industry 4.0. *The Knowledge Engineering Review, 34*, Article e17. https://doi.org/10.1017/S0269888919000109

Labucay, I. (2022). Is there a smart sustainability transition in manufacturing? tracking externalities in machine tools over three decades. *Sustainability, 14*(2), 838. https://doi.org/10.3390/su14020838

Leitao, P., Colombo, A., & Restivo, F. (2005). Adacor: a collaborative production automation and control architecture. *IEEE Intelligent Systems, 20*(1), 58–66. https://doi.org/10.1109/MIS.2005.2

Lemaignan, S., Siadat, A., Dantan, J. Y., & Semenenko, A. (2006). MASON: A proposal for an ontology of manufacturing domain. *Proceedings - DIS 2006: IEEE Workshop on Distributed Intelligent Systems - Collective Intelligence and Its Applications, 2006*, 195–200. https://doi.org/10.1109/DIS.2006.48

Leng, J., Zuo, K., Xu, C., Zhou, X., Zheng, S., Kang, J., Liu, Q., Chen, X., Shen, W., Wang, L., et al. (2025). Physics-informed machine learning in intelligent manufacturing: a review. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-025-02641-1

Leng, J., Zheng, K., Li, R., Chen, C., Wang, B., Liu, Q., Chen, X., & Shen, W. (2026). Aigc-empowered smart manufacturing: Prospects and challenges. *Robotics and Computer-Integrated Manufacturing, 97*, Article 103076. https://doi.org/10.1016/j.rcim.2025.103076

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., & Oltramari, A. (2003). WonderWeb Deliverable D18. *Communities, 2003*, 343.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In: Proceedings of the Eighth ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '90). ACM, pp 249–256, https://doi.org/10.1145/97243.97281.

Poveda-Villalón, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2014). Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS), 10*(2), 7–34. https://doi.org/10.4018/ijswis.2014040102

Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., & García-Castro, R. (2022). Lot: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*. https://doi.org/10.1016/j.engappai.2022.104755

Qin, Z., & Lu, Y. (2025). Knowledge graph-enhanced multi-agent reinforcement learning for adaptive scheduling in smart manufacturing. *Journal of Intelligent Manufacturing, 36*(8), 5943–5966. https://doi.org/10.1007/s10845-024-02494-0

Rehage, G., & Gausemeier, J. (2015). Ontology-based determination of alternative CNC machines for a flexible resource allocation. *Procedia CIRP, 31*, 47–52. https://doi.org/10.1016/j.procir.2015.03.054

Sapel, P., Molinas Comet, L., Dimitriadis, I., Hopmann, C., & Decker, S. (2024). A review and classification of manufacturing ontologies. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-024-02425-z

Sarkar, A., Šormaz, D., Koonce, D., & Farah, S. (2020). Developing a resource-based manufacturing process capability ontology. In: International Conference of Production Research–Americas, Springer, pp 293–306, https://doi.org/10.1007/978-3-030-76307-7_22.

Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., & Lee, J. (2000). The process specification language (PSL) overview and version 1.0 specification. US Department of Commerce, National Institute of Standards and Technology ...

Šormaz, D., & Sarkar, A. (2019). SIMPM – Upper-level ontology for manufacturing process plan network generation. *Robotics and Computer-Integrated Manufacturing*, **55**(April 2018):183–198. https://doi.org/10.1016/j.rcim.2018.04.002

Vegetti, M., & Henning, G. (2022). Ontology network to support the integration of planning and scheduling activities in batch process industries. *Journal of Industrial Information Integration*. https://doi.org/10.1016/j.jii.2021.100254

Wilkinson, M. D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., others (2016) The fair guiding principles for scientific data management and stewardship. *Scientific data*3(1):1–9. https://doi.org/10.1038/sdata.2016.18

Xia, M., Zhao, X., & Hu, X. (2024). An ontology-based multi-source heterogeneous data integration framework for cutting tool lifecycle management*. In: 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), pp 1040–1045, https://doi.org/10.1109/CASE59546.2024.10711596.

Yang, S. M., Choe, J.-H., Kim, J., Kim, D. M., & Park, H. W. (2025). Intelligent digital twin process for the end-milling simulation combined with a multi-physical model. *Journal of Intelligent Manufacturing*, pp 1–18. https://doi.org/10.1007/s10845-025-02686-2

Zhou, G., Lu, Q., Xiao, Z., Zhou, C., Yuan, S., & Zhang, C. (2017). Ontology-based cutting tool configuration considering carbon emissions. *International Journal of Precision Engineering and Manufacturing,18*, 1641–1657. https://doi.org/10.1007/s12541-017-0193-2

Zhou, L., Chen, L., Zhu, T., Su, X., Dong, Y., Wen, L., Liu, H., Lu, X., Leng, J., Liu, Q., et al. (2026). Towards trustworthy ai in industry 5.0: Ante-hoc interpretability with deep learning. *Computers & Industrial Engineering,* 111805. https://doi.org/10.1016/j.cie.2026.111805