CIRP Journal of Manufacturing Science and Technology 52 (2024) 366–385

Contents lists available at ScienceDirect

CIRP Journal of Manufacturing Science and Technology

journal homepage: www.elsevier.com/locate/cirpj

Knowledge transfer in Digital Twins: The methodology to develop Cognitive Digital Twins

Rosario Davide D'Amico [a,*], Arkopaul Sarkar [b], Mohamed Hedi Karray [b], Sri Addepalli [a], John Ahmet Erkoyuncu [a]

[a] CDEM, Cranfield University, Cranfield, United Kingdom
[b] LGP-UTTOP, Université de Technologie de Tarbes, France

ARTICLE INFO

Keywords:
Basic Formal Ontology (BFO)
Common Core Ontology (CCO)
Connected Digital Twins
Digital Twin
Industrial Ontology Foundry (IOF)
Knowledge Graph
Semantics
Top-Level Ontology

ABSTRACT

In the realm of Digital Twins (DTs), industry experts have emphasised the pivotal concept of the Federation of Twins, envisioning seamless collaboration across sectors driven by shared semantics. In response to this challenge, the Cognitive Digital Twin (CDT) integrates the DT framework with formal semantics, specifically ontologies. This paper introduces a comprehensive five-step methodology for CDT development. Furthermore, it becomes possible to incorporate human expertise into the DT ecosystem by adopting an ontological approach. The CDT enhances DT services with advanced reasoning capabilities, leading to a profound semantic enrichment of the data. The presented methodology has been validated using a use case where the CDT is employed to detect malfunctions, significantly reducing manual intervention. This paper advocates for the adoption of CDTs, which represent a harmonious fusion of formal semantics and human expertise, enhancing system efficiency and operational performance.

1. Introduction

The current industrial direction is oriented towards enhancing productivity across production and management systems. Businesses are reconsidering their production tools, automation processes, and fostering collaboration within the supply chain, engineering, and sales [54]. In some industries, this transformative shift is seen as the beginning of a cognitive manufacturing era [11]. Within this context, cognitive manufacturing systems are emerging with the capability to autonomously recognise, assess, plan, predict, optimise, react, adapt, and enhance their operations. They exhibit decision-making and action-taking abilities akin to real-world cognitive behaviour, especially in dynamic and changing environments [21].

From an industry perspective, the pressing need is to establish more interconnected and collaborative frameworks. Notably, a survey conducted with industrial experts regarding the future requirements of Digital Twins (DTs) has underlined the significance of the "federation of twins" or the "network of twins" concept [9,17]. To clarify, this paradigm centres on the development of DTs that facilitate interoperability, allowing them to seamlessly function within and across sectors. Moreover, it offers the advantage of a shared common semantics between the twins. Therefore, the federation of twins' paradigm becomes a priority to tackle complex challenges and optimise the operations in large-scale systems, such as smart cities, national digital twin programs,[1] and complex engineering assets [9].

To elucidate further, the concept of "federation of twins" is distinct from "federated learning", which primarily concerns privacy preservation and collaborative machine learning [56]. In contrast, the federation of twins focuses on the interconnection, interoperability, and shared semantics between individual digital twins, enabling them to work collectively and efficiently in diverse domains and applications. This clarification should help avoid any potential confusion between the two concepts.

There has been an increase in the use of ontologies as a framework for managing knowledge and data in DTs. The systematic literature review conducted by Correia et al. Correia et al., [15] reports that the adoption of industrial standards and ontologies enhances interoperability and provides the reference architecture to develop DTs [15]. Ontologies are a tool used to represent knowledge through an "*explicit, formal specification of a shared conceptualisation*" [27]. In ISO

* Corresponding author.
E-mail address: r.damico@cranfield.ac.uk (R.D. D'Amico).
[1] https://digitaltwinhub.co.uk/ndtp/

https://doi.org/10.1016/j.cirpj.2024.06.007
Received 3 October 2023; Received in revised form 13 June 2024; Accepted 17 June 2024
Available online 2 July 2024
1755-5817/© 2024 The Author(s). This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

21838:2021, ontology is defined as a "*collection of terms, relational expressions and associated natural-language definitions together with one or more formal theories designed to capture the intended interpretations of these definitions*". They provide a structured and organised way to model and represent knowledge, allowing for better understanding, communication, and information management [52].

In the context of knowledge management, ontologies serve as a framework for organising and categorising knowledge assets within an organisation. They help to define and standardise the vocabulary and terminology used in a particular domain, enabling effective knowledge-sharing and retrieval [5]. By explicitly specifying the conceptualisation of a domain, ontologies facilitate the identification, classification, and retrieval of relevant information [29]. Preliminary evidence from a case study in Singapore demonstrated that a DT using semantic technologies (ontologies) and knowledge graphs improves the planning of smart cities and the design of future energy systems [3]. The association between these two paradigms (DTs and ontologies) has led to an evolution of the concept of DT where additional components have been added to the standard DT approach (see Section 2).

Cognition and semantics are closely related, as cognition refers to the mental processes and activities involved in acquiring, processing, and using information. In contrast, semantics is the study of meaning in language and how it is encoded and understood [31]. In the Semantic Web context, concerns about its cognitive plausibility exist. Some propose using conceptual spaces, a knowledge representation framework aligned with human cognition, to improve Semantic Web data utility [44]. Additionally, it is suggested that the Semantic Web should consider human cognition, including conceptual representations and reasoning mechanisms. This involves incorporating conceptual spaces rooted in cognitive semantics and emphasising factors like similarity and prototype effects in categorisation as a more realistic approach to knowledge representation [25]. An ontological approach might satisfy this need. Recent research shows that a cognitive or semantic layer has been implemented in the DT architecture. Such DT frameworks with cognitive/semantic aspects are:

- The universal digital twin [3] – a DT which includes semantic web technologies and dynamic knowledge graphs;
- The next-generation digital twin [13] – a framework where DTs are connected through a knowledge graph enhancing knowledge sharing;
- The cognitive twin [38] – a DT that uses ontology for managing interrelationships between models and virtual entities;
- The cognitwin [1] – a third stage of evolution of the DT concept implementing interconnection between twins, and expert knowledge and historical data into the DT;
- The cognitive digital twin [4] – a DT that includes elements of cognitive science, machine learning, and artificial intelligence harnessing implicit knowledge derived from the operational experience of existing manufacturing systems, facilitates the transfer of more effective decision-making and control methods, resulting in enhanced overall performance across the system.

Most of the architectures presented in the literature include an ontology as the methodology to semantically enrich the knowledge into the DT. The concept that attracted most authors is the *cognitive twin*, which recently evolved into the *cognitive digital twin* (CDT). These CDTs include a graph database to manage the data, and a series of ontologies including a top-level ontology (TLO) [ISO 23247–1] to facilitate the semantic interoperability between the CDTs, enabling the creation of the "federation of twins". The CDT is defined as "*a digital representation of a physical system that is augmented with certain cognitive capabilities and support to execute autonomous activities; comprises a set of semantically interlinked digital models related to different lifecycle phases of the physical system including its subsystems and components; and evolves continuously with the physical system across the entire life-cycle*" [60].

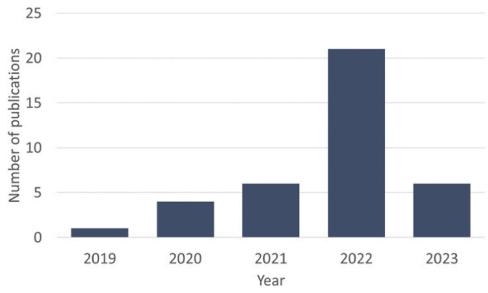


Fig. 1. Number of publications regarding CDTs.

The above-mentioned papers highlight several benefits of CDTs, including interoperability, portability, and the incorporation of human knowledge for reasoning. These characteristics are crucial in facilitating the creation of a federation of twins' paradigm. Despite these advantages, few publications offer a comprehensive methodology for constructing CDTs, indicating a significant gap in the existing literature. There is a clear demand for conceptual frameworks that can effectively guide the development of CDTs. Addressing the need for seamless knowledge sharing, transfer, and scalability within and across multiple domains in CDTs is also essential [4]. This underscores not only the need for structured approaches and best practices in creating CDTs but also the necessity for these systems to be interoperable and flexible, capable of adapting to various environments and applications. This situation highlights the growing interest in CDTs and their potential applications while simultaneously identifying crucial areas for further research and development.

This paper aims to address the above-mentioned gaps by presenting a methodology for developing and validating a CDT architecture by creating the CDT of a laboratory setup. A Festo sorting station constitutes the laboratory setup. Festo is a German industrial automation and control company. They specialise in developing and manufacturing pneumatic and electrical control systems, automation technology, and educational equipment. Festo's products and solutions are widely used in various industries, including manufacturing, automation, and education, to enhance efficiency and productivity. The novelty of this research lies in the proposed methodology, which incorporates human expertise into the DT domain. Furthermore, the methodology is tool-agnostic, based on system engineering approaches, and aligns well with manufacturing systems.

The research questions this paper aims to answer are the following:

1. Can an ontology approach be used to develop a DT?
2. Can human expertise be transferred to the CDT?
3. Can a CDT support a root cause analysis for the malfunction of a system?

The remainder of the paper is structured as follows. Section 2 presents the literature review on CDTs, highlighting the state of the art and the current gaps in addressing them. Section 3 presents the methodology. Section 4 presents the application of the architecture using the Festo sorting station. Section 5 presents the discussion; Section 6 presents the conclusion and future works.

2. Literature review

The number of publications on CDTs is growing, as reported in Fig. 1. The result comes from a search of the "cognitive digital twin" string on the Scopus database on the 22nd of June, 2023. A total of 38 publications, constituting 19 conference papers and 19 journal articles that directly support this research, have been identified.

Key breakthroughs inspire the CDT concept in cognitive science, machine learning, and artificial intelligence. According to Neisser's

Table 2–1
Definitions of Cognitive Digital Twins in the literature.

Ref.	Concept	Definition
Fernández et al. [22]	Associative Cognitive Digital Twin (AC-DT)	*'A contextual augmented description that explicitly includes the relevant associated relationships, connections or information channels of the considered entity with the other entities, for the considered application and purpose'*
Eirinakis et al. [20]	Enhanced Cognitive Twin (ECT)	*'Introduce advanced cognitive capabilities to the DT artefact that enable supporting decisions, with the end goal to enable DTs to react to inner or outer stimuli. The ECT can be deployed at different hierarchical levels of the production process,* i.e. *at sensor-, machine-, process-, employee- or even factory-level, aggregated to allow both horizontal and vertical interplay'*
Lu et al. [38]	Cognitive Twin	*'Digital Twins with augmented semantic capabilities for identifying the dynamics of virtual model evolution, promoting the understanding of interrelationships between virtual models and enhancing the decision-making based on DT'*
Abburu et al. [1]	COGNITWIN - Cognitive Digital Twin	*'An extension of Hybrid Digital Twins incorporating cognitive features that enable sensing complex and unpredicted behaviour and reason about dynamic strategies for process optimisation, leading to a system that continuously evolve its own digital structure as well as its behaviour'*
Intizar Ali et al. [33]	Cognitive Digital Twin	*'An extension of existing digital twins with additional capabilities of communication, analytics, and intelligence in three layers: i) access, ii) analytics, and iii) cognition'*
Al Faruque et al.,[4]	Cognitive Digital Twin	*'Digital Twins endowed with the other elements of cognition such as perception, attention, memory, reasoning, problem-solving, etc'*
Zheng et al. [62]	Cognitive Digital Twin	*'A digital representation of a physical system that is augmented with certain cognitive capabilities and support to execute autonomous activities; comprises a set of semantically interlinked digital models related to different lifecycle phases of the physical system including its subsystems and components; and evolves continuously with the physical system across the entire lifecycle'*
ElMaraghy, ElMaraghy [21]	Cognitive Digital Twin	*'A digital representation, augmentation, and intelligent companion of its physical Twin as a whole including subsystems and across all of its life cycles and evaluation phases'*

classic definition [51], cognition involves '*all the processes by which the sensory input is transformed, reduced, elaborated, stored, recovered and used*'. The fundamental characteristics of cognition are attention (selective focus), perception (forming practical precepts from raw sensory data), memory (encoding and retrieval of knowledge), reasoning (drawing inferences from observations, beliefs, and models), learning (from experiences, observations, and teachers), and problem-solving (achieving goals) [4]. A list of existing definitions of cognitive digital twins and related concepts is reported in Table 2–1.

The literature identifies several gaps, research challenges, and open research questions related to the concept of cognitive digital twins. These include:

- Lack of consensus on the definition and scope of cognitive digital twins: There is no widespread consensus on the definition and scope of cognitive digital twins, leading to ambiguity and confusion [62].
- Integration of heterogeneous digital twin models: Integrating digital twin models from different stakeholders with different protocols and standards is a challenging task due to the heterogeneity of data structures [62].
- Semantic integration of digital twin models: Semantic technologies such as ontology and knowledge graphs are promising solutions for integrating heterogeneous digital twin models, but further research is needed to explore their full potential [37;62].
- Enabling technologies for cognitive digital twins: Research is needed to explore and develop enabling technologies such as knowledge graphs, artificial intelligence, and machine learning techniques to enhance the cognitive capabilities of digital twins [37,62].
- Lifecycle representation of cognitive digital twins: Cognitive digital twins should be able to represent the entire lifecycle of a system, including its subsystems and components, and evolve continuously with the real system. Research is needed to develop frameworks and architectures to support this comprehensive representation [62].
- Decision-making capabilities of cognitive digital twins: Research is needed to explore how cognitive digital twins can support decision-making processes, including real-time data analysis, learning, reasoning, problem-solving, and autonomous decision-making [62].
- Interoperability and connectivity of cognitive digital twins: Research is needed to address the interoperability and connectivity challenges of integrating multiple cognitive digital twins of different systems and domains, enabling seamless communication and collaboration [62].
- Validation and verification of cognitive digital twins: Research is needed to develop methodologies and techniques for validating and verifying the accuracy, reliability, and performance of cognitive digital twins in real-world manufacturing settings [59,62].

Some relevant next steps highlighted in the literature include:

- The need for novel conceptual frameworks and algorithmic breakthroughs to develop cognitive digital twins. The facilitation of knowledge sharing and transfer among digital twin models and the scalability of knowledge sharing across multiple domains in cognitive digital twins [4].
- Enhancing the cognitive capability of industrial systems by combining ontology-based systems and MBSE methods [34].
- Developing methodologies and tools to enable knowledge capturing considering different expert/user-oriented formalisation stages. Improving the effectiveness and efficiency of the ontology-based engineering system and enhancing knowledge capturing and utilisation [6].
- Exploring collaborative behaviours such as value co-creation in the context of tangible and intangible value. Additionally, determining members' contributions in a collaborative venture. These future works aim to enhance further the understanding and implementation of the concept of cognitive digital twins [2].
- Identify and extend ontology-driven services to enhance digital consistency, interoperability, and reusability in engineering activities. There is a lack of research on integrating ontology as a semantic core for industrial system design [32].
- Identify and extend ontology-driven services to enhance digital consistency, interoperability, and reusability in engineering activities [33].
- Generalising knowledge sharing across multiple domains and enabling scalable knowledge sharing in digital twin models [42].
- Exploration of knowledge graph-based analysis in the context of digital twins. There is a lack of tools and methodologies for integrating knowledge graphs into industrial settings [40].
- The development of a top-level ontology-based digital twin to improve human-machine interoperability [16], enhancing the CDT with optimisation algorithms for decision-making [20], and expanding research in the aircraft manufacturing domain to cover additional assembly stations and address system complexity [61].
- The lack of a solid methodological framework for the implementation of DTs in manufacturing and the absence of a systematic approach in the scientific literature that applies the concepts of CDTs in a complex system of systems setup, such as agile supply chain networks [34,46,8]

These gaps, challenges, next steps, and open research questions

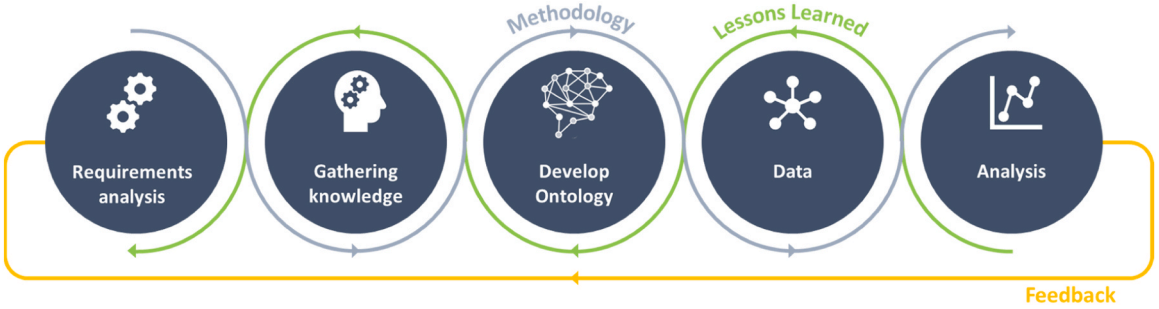


Fig. 2. Methodology to develop CDT.

highlight the need for further research and development to fully realise the potential of cognitive digital twins in various industrial domains. There are limited publications where a methodology to develop CDT is presented. Kalaboukas et al. presented a methodology composed of four steps: scope definition, system modelling, design and operation, and rollout. However, the application of this methodology has a focus on the supply chain. The cognitive capabilities are characterised by a semantic representation of data without any detailed explanation or any use of ontology [34]. Most CDT publications focus more on describing the elements and the architecture rather than a general methodology to develop the CDT. This paper aims to provide a methodology to build a CDT architecture that addresses most of the aforementioned research gaps. Moreover, the methodology has been validated using a laboratory-based experiment.

3. Methodology

In this section, a comprehensive and structured methodology for the development of CDTs is presented. The primary objective of this methodology is to provide a well-defined, step-by-step procedure that encompasses all essential aspects involved in the process of CDT creation. What sets this methodology apart from the existing literature is its emphasis on the procedural dimension rather than delving into architectural or tool-related details. The methodology proposed is the culmination of iterative work, lessons learnt, and research experience gained throughout this project.

This methodology consists of five key steps, each playing a crucial role in the coherent and successful development of CDTs. What characterises a step is the achievement of a complete and tangible result at its conclusion, which then becomes instrumental for the subsequent step. Adhering to this criterion, we have delineated a process comprising five distinct steps. These five steps are not isolated entities but rather intricately interconnected, demonstrating a significant interdependency that ensures a cohesive and effective CDT development process.

The methodology's structure is underpinned by theoretical principles from systems engineering and interdisciplinary insights, ensuring a robust and holistic framework (inspired by the one proposed and discussed in [37]. Each step was designed to align with the goals of CDT development, focusing on integrating human expertise, data interoperability, and facilitating informed decision-making in complex asset management. Consequently, this methodology is tailored for CDT with a strong ontology component and is influenced by the recent methodology for developing ontologies, Linked Open Terms (LOT), proposed by Poveda-Villalón et al. Poveda-Villalón et al. [43]. This approach ensures that the methodology addresses current challenges in CDT development and lays a foundation for future advancements in the field.

A visual representation of the methodology is reported in Fig. 2. The presented methodology provides a structured framework for developing CDTs and emphasises the procedural aspect, ensuring clarity and coherence throughout the CDT creation journey. By following these defined steps, researchers and practitioners can effectively harness the potential of CDTs across various domains and applications.

In particular, the five steps are defined as follows:

1. *Requirements analysis*. The initial step involves a comprehensive analysis of the requirements. This includes gaining a deep understanding of the system of interest, dissecting its structural and functional components, and defining the scope and purpose of the intended CDT.
2. *Gathering knowledge*. This step entails the systematic collection of a wide spectrum of essential information. This information encompasses an array of critical elements, such as underlying facts, assumptions, rules, behavioural patterns, periodic observations, and invaluable human expertise related to the system of interest. The knowledge collected in this step serves as the foundation and needs to be encoded into an ontology in the subsequent phase.
3. *Develop ontology*. The development of the ontology begins with the formulation of Competency Questions (CQs) that precisely define the ontology's purpose. This step proceeds with the creation of the ontology itself, which serves as a formalised structural description. It encapsulates critical aspects like functions, behaviours, processes, and state descriptions of the system.
4. *Data*. In this step, we address the data aspects of the CDT. This includes selecting the appropriate data strategy, which encompasses choosing the suitable graph database technology capable of importing both the ontology and data to the graph database. Additionally, it involves defining the query strategy, middleware selection, and establishing the information and knowledge transfer strategy to and from the CDT.
5. *Analysis*. The final step revolves around conducting analytical and simulation activities essential to fulfil the intended purpose of the CDT. As defined by [55], this element closely relates to the service component of the Digital Twin. The outcomes of this step are intricately tied to the specific application and purpose for which the CDT is being developed. The CDT will then provide feedback, enhancing decision-making and operations.

3.1. Requirements analysis

This step consists of understanding the components and boundaries of the system of interest. In particular, the classification of the components, qualitative formalisation of the parthood and connection relations between components (mereotopology). Mereotopology combines mereology, the theory of parts and wholes, and topology, the study of spatial properties and relations that remain constant under continuous transformations. In essence, mereotopology explores the relationships between parts and wholes and their spatial properties and connections. Mereotopology helps define what constitutes a 'part' of a system and how these parts are spatially and functionally related to each other and to the 'whole' system. This is crucial in systems engineering and design, where understanding the boundaries and interfaces between components is key to effective system integration. Mereotopology provides a framework for modelling and understanding complex systems. It helps

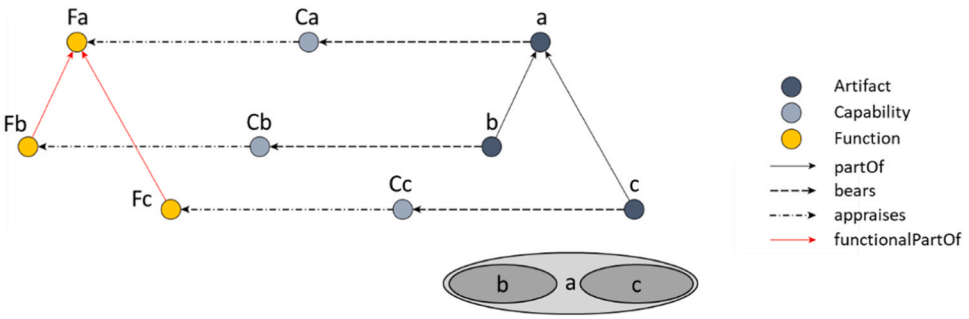


Fig. 3. Mereotopology diagram that shows the relations between artefacts, their functions, and their capabilities.

in breaking down a complex system into more manageable parts, understanding the relationships and dependencies between these parts, and how changes in one part might affect the whole system.

In the context of requirement analysis, mereotopology can aid in understanding how different system components interact and depend on each other. This understanding is essential for identifying and defining system requirements, especially interoperability, scalability, and modularity. Mereotopology plays a crucial role in the requirements analysis, as it is fundamental for the ontology creation by providing a framework for representing and reasoning over part-whole relations (mereology) and spatial relationships (topology). Mereotopology deals with the concepts of parthood and location, which are fundamental for modelling complex relationships between entities in the world. By incorporating mereotopological relations into ontologies, we can capture the spatial and structural aspects of the modelled domain. This allows for a more accurate and comprehensive representation of knowledge, enabling better reasoning and inference capabilities within the ontology [36]. Furthermore, the relation between functions and capabilities of each component, modelling the system's behaviour, the flow of material, energy, and control in the system should all be included in the system boundary.

Capturing the parthood and changes in different situations happening over time in a system can be managed with the Mereotopology theory. Salustri proposes using design mereotopology (DMT) in product modelling, highlighting its potential for improved understanding and manipulation of product knowledge [47]. Forrest presents a mereotopological system based on interior parthood without relying on classical mereology or the Axiom of Choice [23]. Hahmann and Grunninger provide an overview of different axiomatisations and representations of mereotopologies, emphasising the need for practical extensions and integration into comprehensive ontologies [30]. Galton introduces discrete mereotopology, which considers entities built from atomic elements, and discusses its applications in qualitative spatial reasoning [24]. These papers demonstrate the relevance and potential of mereotopology in capturing parthood and changes in various contexts, including product modelling, theoretical frameworks, and practical applications. Casati and Varzi discuss the interplay between mereology, topology, and the theory of spatial location in understanding the structures of spatial representation. Moreover, they presented two strategies representing the combination of mereology and topology and the definition of mereotopology [14].

Mereotopology is pivotal in ontological analysis [58]. This theory allows for the qualitative formalisation of two essential concepts: part-whole relationships (i.e., one entity being a part of another) and connectivity (i.e., the relationship between entities being connected). The current obstacle in applying mereotopology-based theory in engineering design is the need to account for how a product is perceived in the real world [19]. The mereotopology theory is well explained in the literature [57] and has also been associated with the formal ontology domain [49]. It must be noted that the explanation of the underlying mereotopology theory is out of the scope of this paper. However, the application of the mereotopology theory through the use case falls within the scope of this paper and is presented in Section 4.1.

The representation of the mereology structure of a system is reported in Fig. 3. The figure represents a system composed of three artefacts (a, b, and c) where two components *b* and *c* are part of a super component *a*. The Venn diagram at the bottom shows how the two subcomponents belong to a. In this paper, the authors follow the concept of functions and capabilities defined by Merrell et al. Merrell et al., [39]. It should be noted that other schools of thought also exist and slightly differ from what will be described below [12]. A capability is defined *as a disposition in whose realisation some organism or group of organisms has/had an interest* [39]. A function is a capability that explicitly highlights the reason why an entity exists (teleological argument). All designed artefacts have functions because that is the definition of the word artefact. They are entities which were designed to perform a particular process. Moreover, that means that they were designed to have a function which is realised in that process [39].

Also, to verify the system's mechanical design, the Axiomatic Design (AD) theory has been used [53]. The AD theory is used to break down the system. This theory is applicable across various types of systems, encompassing machines, complex systems, software systems, organisational structures, and hybrid systems combining hardware and software elements. Systems are represented using a system architecture framework, which includes the {Functional Requirements}/{Design Parameters}/{Process Variables} hierarchies, a 'junction-module' diagram, and a 'flow diagram'.

The 'flow diagram' within the system architecture succinctly illustrates the system's design, the interconnections among its modules, and the control sequence in operating systems. This flow diagram serves a multitude of purposes, such as aiding in system design, construction, operation, modification, and maintenance. Furthermore, it is a valuable tool for facilitating distributed design and operation of systems, diagnosing system failures, and maintaining archival documentation [53].

With this approach, at each level of decomposition, if specific processes achieve the functions of each component, then the parent component's function is also effectively realised [48]. Axiom (A. 1) sets this limitation by employing the predicate *functionalPartOf*, which acts as a specific form of parthood relation among functions. This axiom helps formalise the concept of systems and their functional relationships within a more extensive system or context.

$$\forall s, s' FunctionalPartOf(s', s) \rightarrow System(s) \wedge System(s') \wedge \exists f, f' \ (bears(s', f') \wedge bears(s, f) \wedge decompose(f', f)) \quad \text{(A. 1)}$$

$\because$ (A.1) For any two entities s and s', if s' is a functional part of s, then both s and s' are systems, and there exist functions f and f' such that s bears f, s' bears f, and f can be decomposed into f.

Sarkar et al. Sarkar et al., [48] have restructured the independence axiom from AD theory, focusing on how the associated components' capabilities back functions at various decomposition levels. From the interplay between functions and the abilities of system components, two

valid configurations can be discerned: the summation junction and the control junction, both considered types of modulo-junction. As originally outlined in AD theory, these configurations align with the diagonal and triangular design matrices.

The predicates *SummationJunction* (A. 2) and *ControlJunction* (A. 3) are established between a component and its two sub-components. Specifically, within the *ControlJunction* predicate, the component occupying the second position exercises control over the component in the third position.

$$\forall c, c', c''\, SummationJunction(c, c', c'') \leftrightarrow \forall p, f \quad (bears(c, f) \wedge realizes(p, f) \rightarrow \exists f', f'', p', p''(bears(c', f') \wedge bears(c'', f'') \wedge decompose(f'', f) \wedge decompose(f'', f) \wedge realizes(p', f') \wedge realizes(p'', f'')) \tag{A. 2}$$

$$\forall c, c', c''\, ControlJunction(c, c', c'') \leftrightarrow \forall p, f \quad (bears(c, f) \wedge realize(p, f) \rightarrow \exists p', f', p'', f''(bears(c', f') \wedge bears(c'', f'') \wedge (realizes(p', f') \rightarrow realizes(p'', f'')))) \tag{A. 3}$$

˙.˙ (A.2) For all configurations *c*, *c'*, and *c''*, the 'SummationJunction' relation holds if and only if, for all processes *p* and functions *f*, if *c* bears *f* and *p* realizes *f*, then there exist functions *f'* and *f''* and processes *p'* and *p''* such that *c'* bears *f'*, *c''* bears *f''*, *f'* decomposes *f'*, *f''* decomposes *f'*, *p'* realizes *f'*, and *p''* realizes *f''*.

˙.˙ (A.3) For all configurations *c*, *c'*, and *c''*, the 'Control Junction' relation is valid if and only if, for all processes *p* and functions *f*, when *c* bears *f* and *p* realizes *f*, there exist processes *p'* and *p''*, as well as functions *f'* and *f''*, such that *c'* bears *f'*, *c''* bears *f''*, and if *p'* realizes *f'*, then *p''* realizes *f''*.

The elements of the AD theory serve as essential guides throughout the stages of system design, development, servicing, and troubleshooting. Furthermore, they offer a valuable means of documenting the system design effectively. When inspecting the flow diagram, it becomes possible to identify vulnerable design connections by recognising the interconnected components within the system's design. This insight can help pinpoint areas of potential weakness in the overall system design. Moreover, the theory should offer clear instructions for identifying the components that need to be examined when the system fails to execute specific functions [53].

In this step of the methodology, the applicability criteria of methods that include mereotopology, mechanical, and system engineering analysis should be aligned with the project's scope and objectives, considering the complexity and scale of the project.

3.2. Gathering knowledge

This section aims to understand the system's behaviour, model the processes that occur within a specific system and then integrate that knowledge into the CDT. This process involves several key steps:

- *Defining Processes within the System*: The first step is to identify and describe the system's various activities, actions, or transformations. This is essential because systems can be pretty complex, and clearly understanding what is happening is crucial for effective modelling.
- *Transferring Knowledge into the CDT*: CDT refers to a digital model of a real-world system that incorporates data, knowledge, and algorithms to simulate and understand how that system behaves. So, part of the process is taking the knowledge about the system's processes and incorporating it into this digital twin for analysis and simulation.
- *Using First Order Logic (FOL) Axioms for Definitions*: To make these processes more precise and formal, they are expressed using First Order Logic (FOL) axioms. FOL is a mathematical language that provides a structured way to represent knowledge. Each process is defined in terms of logical statements, making it clear, unambiguous, and suitable for computer-based reasoning.
- *Thematic Roles - "Instrument" and "Patient"*[2]: To better understand and model the processes, it is essential to distinguish between the various entities or elements involved. Here, two thematic roles are introduced:
  - o *Instrument*: This role represents the entity or element that directly causes or initiates a process. It is the actor or agent responsible for carrying out the action.
  - o *Patient*: In contrast, the patient is the entity that is impacted or affected by the processes. It is the entity undergoing a change as a result of the process. These roles help define the roles of different entities within the system clearly.
  - o *Gathering and Storing Human Expertise*: Often, domain experts possess valuable knowledge about the system. They may have insights that are not necessarily expressed in a formalised way. During this phase, the aim is to collect this knowledge from domain experts and store it for later use.
  - o *Conversion of Human Expertise to Formal Logical Statements*: Human expertise, especially from experts, tends to be in natural language or an informal format. This knowledge is converted into formal logical statements to make it compatible with the formal model. This step ensures that the gathered knowledge can be seamlessly integrated into the model, maintaining consistency and compatibility with the rest of the formal representation.

This approach is all about systematically understanding, modelling, and simulating a specific system. It emphasises the need for formalisation to achieve precision, introduces thematic roles to clarify entity roles in processes, and stresses the importance of gathering, storing, and formalising human expertise from domain experts for a comprehensive analysis of the system.

In this step of the methodology, where the methods focus on systematically understanding, modelling, and simulating a specific system, the applicability criteria are centred around the thoroughness and precision of the information-gathering process. This includes emphasising the need for formalisation to ensure accuracy in the representation of the system. The method calls for clearly defined thematic roles to elucidate the functions of various entities within the processes being modelled. Furthermore, a significant emphasis is placed on collecting, storing, and formalising human expertise, particularly from domain experts, to ensure a comprehensive and insightful analysis of the system. This approach necessitates a meticulous and systematic process of knowledge acquisition, prioritising depth and quality of information to facilitate a robust and accurate modelling and simulation of the system. The criteria, therefore, revolve around the precision of data, clarity in role definition, and the depth of expert insights incorporated into the system analysis.

3.3. Develop ontology

The information gathered in the previous two steps (subsections 3.1 and 3.2) is encoded and used in this phase for the development of the ontology. In general, an ontology is a set of logical statements. A guide that will help the user to formalise what has been found in the previous steps in a set of consistent logical statements has been presented below. The result will then be converted into a file that can be used consecutively to enhance the collected data's quality and perform more advanced analytics. The methodology to develop an ontology takes inspiration from the Linked Open Terms (LOT) methodology published by Poveda-Villalón et al. Poveda-Villalón et al., [43]. It is composed of 4 main steps: (i) ontology requirements specification, (ii) ontology implementation, (iii) ontology publication, and (iv) ontology maintenance. This methodology emphasises the importance of ontology construction for sharing a common understanding of information structure

[2] https://semantic-annotation.uvt.nl/LIRICS_semroles.htm#LIRICS_semrole

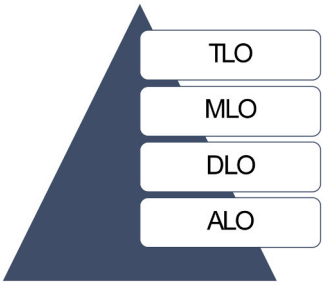


Fig. 4. Levels of abstraction in ontology theory.

and enabling query retrieval, highlighting its role in facilitating ontology reuse, knowledge representation, and information sharing in various domains.

The ontology development process is initiated by identifying its requirements, a pivotal step achieved by formulating competency questions (CQs). These CQs are carefully crafted queries that encapsulate the specific types of information and knowledge the ontology is meant to handle. They are derived from practical scenarios, such as the analysis phase in a CDT, which dictate what kind of information the ontology should effectively manage [28].

In conjunction with the formulation of CQs, it is essential to articulate the answers the ontology should provide when these questions are posed. This step ensures that the ontology is constructed with a clear purpose and that it delivers the necessary information and knowledge when called upon.

Following this, a rigorous verification process is carried out by querying the ontology with these CQs to validate that it aligns with the defined requirements. This verification process systematically compiles a glossary of essential terminology, serving as the foundational framework upon which the ontology is constructed. This terminology helps organise key concepts and relationships within the ontology and informs the development of a hierarchical taxonomy, ensuring that the knowledge representation is structured and efficient for its intended application. In essence, this comprehensive approach ensures that the ontology is tailored to represent, manage, and provide relevant knowledge for its specific domain of application effectively.

In the context of CDT, ontology development holds a pivotal role. One key aspect of this development is the careful selection of upper-level ontologies. The different types of ontologies divided by level of abstraction are reported in Fig. 4. These upper-level ontologies, often referred to as top-level ontologies (TLOs), serve as the foundational building blocks for defining more specific domain ontologies within the CDT framework. TLOs like Basic Formal Ontology (BFO), Suggested Upper Merged Ontology (SUMO), and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) are generic frameworks that describe high-level concepts or universals. They play a critical role in ensuring interoperability between domain ontologies, facilitating seamless integration and the reuse of knowledge. In the context of CDTs, the choice of TLOs and mid-level ontologies (MLOs) is guided by the nature of the system and the knowledge required for analysis, enabling the development of domain-level ontologies (DLO) and application-level ontologies (ALO) tailored to the specific context of a CDT, considering factors such as functional composition and processes [7].

In addition, the design of ontologies within the CDT framework is guided by a set of eight fundamental principles, with a particular emphasis on domain ontologies [7]. These principles encompass essential aspects such as realism, perspectivalism, fallibilism, adequatism, the principle of reuse, the balance between utility and realism, open-ended ontology design, and addressing low-hanging fruit. Adhering to these principles is of paramount importance to ensure the development of ontologies that not only address immediate utility but also accommodate revisions and updates considering the evolving scientific knowledge and technological advancements, which are crucial in the dynamic context of CDTs.

Moreover, in the context of CDTs, it is imperative to avoid developing new ontologies from scratch, as doing so could hinder interoperability. Instead, a comprehensive review and assessment of existing ontology resources related to the CDT's domain should be conducted. Ontologies are intended to facilitate communication between information resources across various domains and disciplines, aligning with the holistic nature of CDTs.

Furthermore, within the CDT framework, designing a domain ontology is just the initial step in an ongoing process. This process involves continuous maintenance, evaluation, updates, and alignment with neighbouring ontologies, considering advancements in scientific knowledge and our evolving understanding of ontology and its associated logical and computational technology. These principles underscore the importance of creating ontologies in a manner that allows for expansion and amendment over time. The best practice guidelines within this framework are intended to achieve this dynamic and adaptable ontology development approach.

In the process of developing domain ontologies within CDTs, a preliminary phase involves classifying fundamental universal concepts and relationships. A recommended approach is to begin by identifying overarching terms commonly found in foundational literature. Subsequently, the process entails systematically representing increasingly intricate entities within the domain. The five general steps to follow

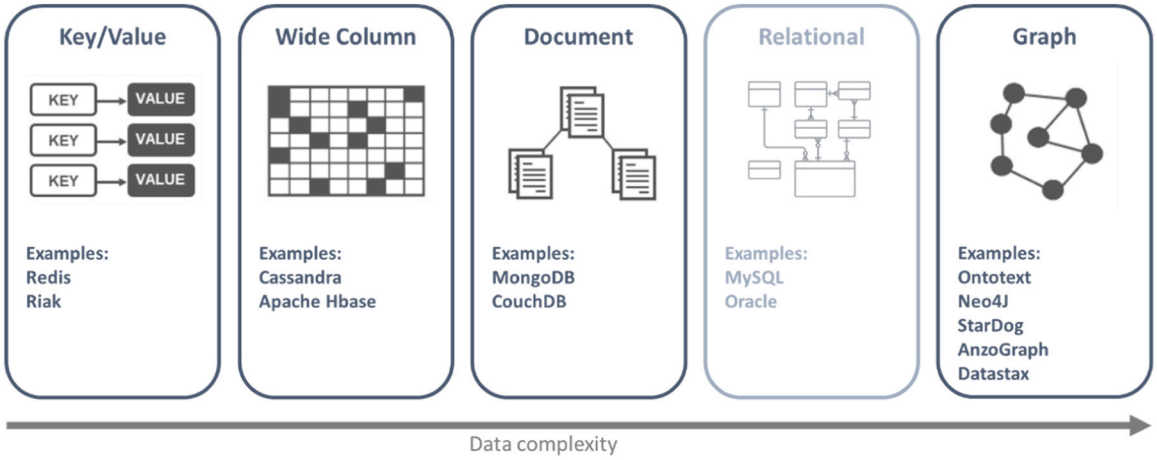


Fig. 5. Database approaches based on data complexity (Inspired from An Introduction to Graph Data Stores and Applicable Use Cases, Sumit Pal, Gartner Research, Published: 24 January 2019, https://www.gartner.com/en/documents/3899263 (accessed 30/06/2023)).

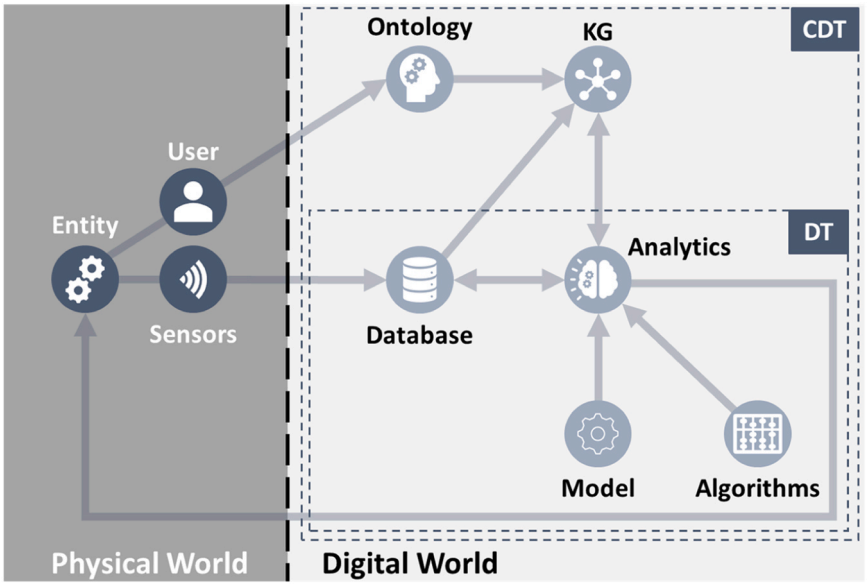


Fig. 6. Relation between the components of the CDT.

when designing a DLO and ALO are [7]:

1. Delimit the ontology's subject matter.
2. Collect data: identify the general terminology used in literature and other ontologies.
3. Sort these terms in a hierarchy.
4. Regulate the outcome to ensure (i) logical, philosophical, and scientific coherence; (ii) coherence and compatibility with neighbouring ontologies; and (iii) clarity, mainly through the development of human-readable definitions.
5. Formalise the outcome in a computer-usable language such that it can be implemented in some computable framework.

3.4. Data

In this phase, a data management strategy is selected to ascertain how to store the data, which kind of databases need to be used, how to send the data into those databases, and where to store those databases (locally or in the cloud). Moreover, this step includes modelling the data, which consists of transforming raw data into the knowledge graph using the ontology vocabulary developed in the previous step (Section 3.3).

The ontology maps the human expertise. It provides a formal way to express the knowledge of the system of interest and establishes common semantics over the system of systems environment through life (using a TLO approach). One of the elements of ontology is the map that represents the relationship between objects, usually in the form of a graph. From the data management point of view, data might be entering the system in different formats and managed with different strategies depending on the complexity of the data. Various types of databases that handle different levels of data complexity have been reported in Fig. 5, with the non-relational database systems highlighted in dark blue [45].

Data complexity is assessed in terms of size, structure, variety, and abstraction [17]. It is important to note that graph databases may not always be the most effective solution due to this approach's complexity and overhead. The optimal solution could be using a combination of databases, with each approach being better suited for handling specific areas of data complexity. Typically, a relational database is the preferable option when dealing with entities, whereas a graph database is better suited for examining the relationships between entities. Further information on these approaches can be found in the literature, such as Bechberger and Perryman [10].

The graph database is the perfect fit if the user wants to shape the data based on an ontology. Furthermore, some advantages of graph databases are [45]:

- Flexible data modelling: Graph databases allow for flexible and dynamic data modelling that can easily handle complex, interconnected data.
- High performance for complex queries: Graph databases are designed for complex queries, which can be executed quickly, even on large datasets.
- Scalability: Graph databases can easily scale horizontally by adding more nodes to the cluster.
- Better handling of unstructured data: Graph databases are good at handling unstructured data like social media data or IoT data.

Applications where graph databases are more efficient than relational databases are [10]:

- Fraud or anomaly detection
- Knowledge graph
- Network/asset monitoring
- Identity and access management
- Social networks
- Machine learning (pathfinding, inferred properties, etc.)

In this methodology step, the applicability criteria revolve around selecting an appropriate data management strategy, which is crucial for determining storage methods, database types, transmission techniques, and whether data should be stored locally or in the cloud. This phase also includes data modelling, transforming raw data into a knowledge graph using the ontology vocabulary developed in an earlier phase. The complexity of the data, measured by size, structure, variety, and abstraction, dictates the choice of database. While graph databases offer flexibility, high performance for complex queries, scalability, and adeptness with unstructured data, they are not universally optimal due to their complexity and overhead. A blend of database types is sometimes more effective, with relational databases favoured for entity management and graph databases for analysing relationships. This decision-making process is also shaped by the data's nature and application areas, with graph databases excelling in fields such as fraud detection and network monitoring. Thus, this step involves a nuanced assessment of data characteristics and system requirements to identify the most suitable data management and modelling approach.

3.5. Analysis

In this phase, the purpose that led to the development of the CDT is converted into a validation strategy. Fig. 6 illustrates a conceptual framework that connects the physical and digital realms. It is partitioned into two principal sections: "Physical World" and "Digital World", demarcated by a dotted line. On the left, under "Physical World", there is an "Entity" linked to a "User" and "Sensors". The entity and sensors have connections leading to a "Database" in the "Digital World" segment, delineated by dotted lines, suggesting data transfer or communication pathways. Within the "Digital World", several components are interlinked:

- "Ontology" is linked to "KG", denoting Knowledge Graph, and to "CDT," standing for Cognitive Digital Twin.
- "Database" is connected to "Analytics", which in turn links to both "Model" and "DT", representing Digital Twin.
- "Model" has connections to "Algorithms".

This configuration indicates a system wherein sensors gather data from the physical environment, which is then stored in a database. This information is analysed and utilised to develop models and algorithms that underpin the DT. The CDT appears to be a more sophisticated iteration that integrates a KG and ontology for enhanced data analysis and decision-making processes. The intent is to clarify the roles of various components in the process of developing a CDT and how it differs from a standard DT.

In the "Physical World", the tangible entity is to be mirrored digitally, which could be an asset, a system, or an organisation. The users and sensors serve as interfaces, collecting data and insights to be transmitted into the "Digital World".

Within the "Digital World", the data gathered by the sensors is held within a database structure. Simultaneously, the insights collected from user and entity interactions are organised within an ontology. The ontology presents knowledge of different models about the physical entity, including ontology for the behaviour, processes, and interactions. This ontology encapsulates the knowledge and shapes the KG, into which the data is assimilated.

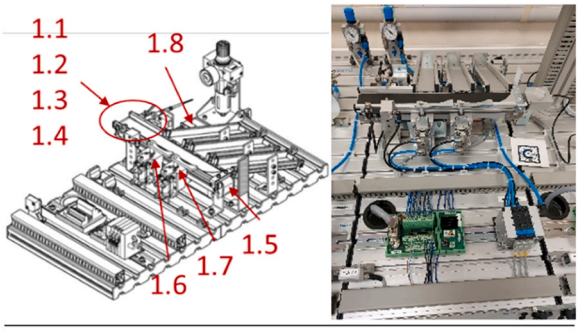
The "Analytics" component illustrates the core of the CDT's capabilities. Herein, the CDT analyses data, executes algorithms, and conducts analytics within the KG. This process entails comparing the digital model to its physical counterpart as represented in the CDT and supplying feedback directly to the entity (to refine operations) or to the user (to support decision-making).

This procedure is reported in Fig. 6, which also visually differentiates the CDT from the DT. The CDT is represented as an enhanced system because it is capable of creating a digital representation of a physical entity and comprehending and using the knowledge concerning the entity to enhance operations and support decision-making. This is contrasted with a DT, which may not possess the same level of cognitive functionality and might typically function primarily as a digital duplicate, lacking advanced analytical and decision-support capabilities.

In this step of the methodology, the applicability criteria revolve around converting the development purpose of the CDT into a validation strategy. This involves linking an entity in the physical world, such as an asset or system, with users and sensors to gather data. This data is then transferred to the digital realm, where it is stored in a database and integrated into an ontology and KG. The CDT's role is to compare the digital model with its physical counterpart and provide feedback for enhancing operations and decision-making, highlighting its advanced analytical and decision-support capabilities over the DT.

4. The use case: Festo sorting station

In this section, the methodology described in Section 3 is validated using the Festo Sorting Station module, a part of the Modular Production System (MPS). This asset has been chosen as a testbed because it is a good example of a material handling system that can represent multiple real-world applications, such as warehouses, baggage handling systems, etc. Additionally, the asset is reprogrammable according to the application of interest and can also be used to import the human experience into the CDT. This example represents an attempt to validate the methodology mentioned above. The focus was on developing the architecture, the graph database, and the import of human expertise. The CDT worked well for the characteristics assessed and responded well in detecting the malfunctioning component. However, some aspects need further investigation and are highlighted in the next section (Section 5) under the limitations paragraph.



	Component	Action
1.1	Diffuse sensor	Part confirmation
1.2	Diffuse colour sensor	Part identification - colour
1.3	Inductive proximity	Part identification - material
1.4	Gate actuator	Part block and release
1.5	Conveyor belt motor	Part moving
1.6	Pushing actuator 1	Sorting 1
1.7	Pushing actuator 2	Sorting 2
1.8	Retro reflective sensor	Stocking/end of cycle confirmation

Fig. 7. Festo Sorting Station, schema, components, actions.

4.1. Requirements analysis

Initially, the asset has been studied and reverse engineered. Classification of the components, mereotopology, relations between artefacts, functions, and capabilities have been carried out. The classification of the components of interest and their actions is reported in Fig. 7. The working procedure for the Festo Sorting Station is described below:

1. Upon entry of the workpiece into the sensing module, the data is transmitted to the CDT. The CDT then pairs the workpiece ID and the spatial area with the provided timestamp.
2. The sensing module determines the workpiece's colour.
3. Once the colour is identified, a command is sent to the gate actuator to release the workpiece.
4. As the gate actuator opens, the colour findings from the sensing phase and the corresponding timestamp are conveyed to the CDT. The CDT then links the workpiece ID, spatial area, and colour data with the provided timestamp.
5. The primary actuator is engaged if the workpiece appears red, directing the piece to the initial collection point. Conversely, a metallic workpiece triggers the secondary actuator, directing it towards the second collection point. When the workpiece is black, no actuator is engaged. Instead, a physical barrier guides the black workpiece via the conveyor system to the third collection point. For such cases, data points for spatial zones are matched with the appropriate workpiece ID in the CDT.
6. Whenever the retro-reflective sensor identifies a workpiece, this data is relayed to the CDT. The CDT then pairs the timestamp and spatial area with the corresponding workpiece ID.

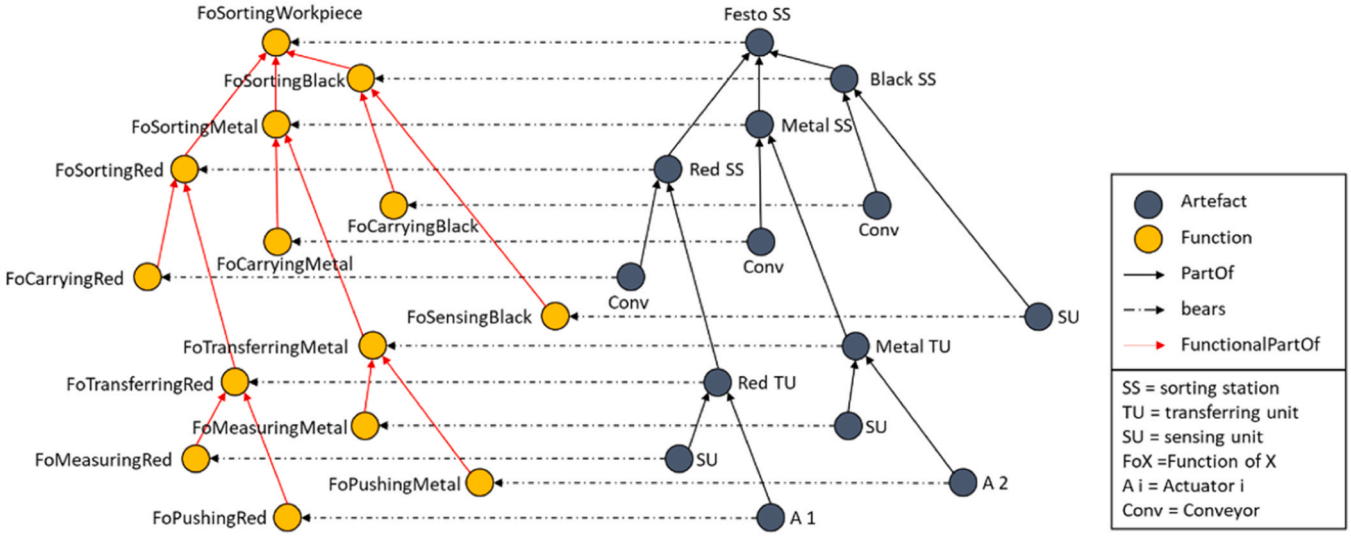


Fig. 8. Mereotopological representation of the system along with functions.

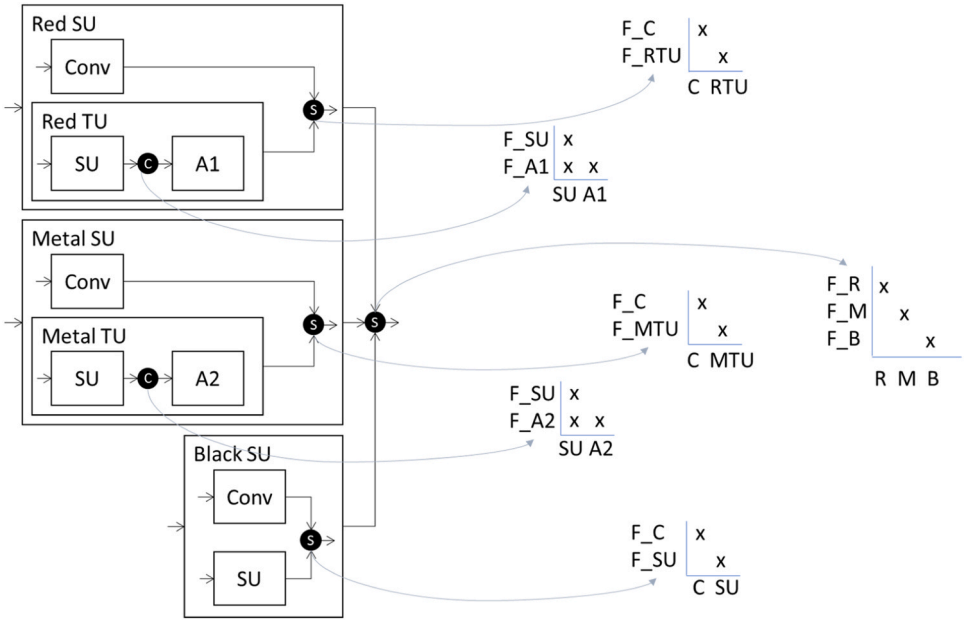


Fig. 9. Modulo junction diagram from AD theory.

Considering the list of components reported in Fig. 7, the Festo sorting station system has been then divided into three subsystems: the red sorting station (Red SS), the metal sorting station (Metal SS), and the black sorting station (Black SS). The Red SS includes the conveyor, the red transferring unit (Red TU), together with the sensing unit (SU) and the first actuator (A1). Similarly, the Metal SS also includes the conveyor and the Metal TU, with the associated SU and the second actuator (A2). While the Black SS contains the conveyor and the SU.

Regarding functions and capabilities, as per 3.1, the functions *are capabilities that specifically highlight the reason why an entity exists*. Therefore, only the functions need to be specified for this application; this paper does not report the connections between the components' capabilities and their functions. The artefact compositions and functional composition of the system are presented in Fig. 8.

The AD theory has been applied to verify the mechanical design and delineate how the functions are realised at every level of the artefacts' decomposition. As represented in the artefacts' decomposition, the sorting station comprises a "SummationJunction" made up of the 'Black SS', 'Red SS', and 'Metal SS'. The first two units function as a "SummationJunction" combining a section of the 'conveyor belt' with a 'transferring unit'. Furthermore, the 'transferring unit' operates as a "ControlJunction", where the 'actuators' are governed by the 'sensing unit'. It is important to note that the final slope lacks an actuator, as reported in Fig. 8. The AD theory allows building matrices and verifies the relations between artefacts and their functions for each uncoupled (SummationJunction (A. 2)) and decoupled (ControlJunction (A. 3)) case in the system. The theory is explained by Suh [53] and is not included in this paper for brevity. The resulting matrices are reported in Fig. 9.

4.2. Gathering knowledge

This section presents all the relevant processes and knowledge to detect malfunctions.

4.2.1. Processes

Whenever a certain kind of process occurs, distinct entity types are involved, each performing specific roles. As anticipated in Section 3.2, to differentiate between the system and the materials it processes, we classify two key thematic roles: the "instrument" (serving as the direct cause) and the "patient" (which is affected). In this context, the material in question is the workpiece that the Festo sorting station classifies based on its colour.

The Festo sorting station consists of three main processes. The workpiece functions as the patient in all these processes, though its interaction varies. For instance, as outlined in one of the subsequent axioms, sensors operate to discern colour through a process termed "ActOfMeasuringColor" (A. 4). At the culmination of this process, the colour data of the workpiece is rendered and sent to the system. The processes "ActOfCarrying" (A. 5) and "ActOfActuating" (A. 6) both target the relocation of the workpiece. While "ActOfCarrying" alters the workpiece's spatial position as it remains stationary on a moving conveyor belt, "ActOfActuating" relocates the workpiece from a conveyor site to a collection point, which serves as its terminal storage.

Based on the functional diagram of the Festo Sorting Station provided in Fig. 8, there exist three distinct "ActOfMeasuringColor" subtypes: "ActOfMeasuringBlack", "ActOfMeasuringRed", and "ActOfMeasuringMetal". These are contingent on the colour variations they identify and the associated functions they execute, e.g., realizes (ActOfMeasuringRed, FunctionOfMeasuringRed). There are also three distinct "ActOfCarrying" types, each corresponding to the terminal locations of the workpiece. However, the black workpiece, transported directly to 'collection point 3' by the conveyor, does not require an actuation process. As such, only two "ActOfActuating" subtypes exist: "ActOfActuatingRed" and "ActOfActuatingMetal".

For analytical purposes, understanding which component malfunctions and the structure of these processes is essential. These operations are executed by higher-tier components, which inherently incorporate these processes. For example, the "RedTransferringUnit" functions as the instrument for "ActOfTransferringRed", which inherently consists of processes like "ActOfMeasuringRed" and "ActOfActuatingRed".

$$\begin{aligned}&\forall p, fActOfMeasuringColor(p) \wedge realizes(p,f) \leftrightarrow ActOfMeasurement(p)\\&\quad \wedge \exists s \quad (Sensor(s) \wedge isInstrumentOf(s,p)) \wedge \exists w(Workpiece(w)\\&\quad \wedge isPatientOf(w,p)) \wedge \forall t(occursOn(p,t) \rightarrow \exists c, ci(Color(c)\\&\quad \wedge ColorName(ci) \wedge hasOutput(p,ci) \wedge isMeasurementOf(ci,c)\\&\quad \wedge \exists te(hasLastInstant(t,te) \wedge inheresIn(c,w,te))\end{aligned} \tag{A. 4}$$

$$\begin{aligned}&\forall p, fActOfCarrying(p) \wedge realizes(p,f) \leftrightarrow ActofLocationChange(p)\\&\quad \wedge \exists c \quad (Conveyor(c) \wedge inInstrumentOf(c,p)) \wedge \exists w(Workpiece(w)\\&\quad \wedge isPatientOf(w,p))\\&\quad \wedge \forall t \quad (occursOn(p,t) \rightarrow \exists l, s1, s2, tb, te \quad (locatedIn(w,l,t)\\&\quad \wedge partOf(l,c) \wedge occupiesSpatialRegion(l,s1,tb)\\&\quad \wedge occupiesSpatialRegion(l,s2,te) \wedge hasFirstInstant(t,tb)\\&\quad \wedge hasLastInstant(t,te) \wedge (s1 \neq s2))\end{aligned} \tag{A. 5}$$

$$\begin{aligned}&\forall p, fActOfActuating(p) \wedge realizes(p,f) \leftrightarrow ActofLocationChange(p)\\&\quad \wedge \exists a \quad (Actuator(a) \wedge inInstrumentOf(a,p)) \wedge \wedge \exists w(Workpiece(w)\\&\quad \wedge isPatientOf(w,p))\\&\quad \wedge \forall t \quad (occursOn(p,t) \rightarrow \exists l, l', s1, s2, tb, te, n, c \quad (Slope(n) \wedge Conveyor(c)\\&\quad \wedge partOf(l',s) \wedge partOf(l,c) \wedge occupiesSpatialRegion(l,s1,tb)\\&\quad \wedge occupiesSpatialRegion(l,s1,tb) \wedge hasFirstInstant(t,tb)\\&\quad \wedge hasLastInstant(t,te)))\end{aligned} \tag{A. 6}$$

⋱ (A.4) If an action of measuring colour p results in the realisation of function f, it is the same as an action of measurement p. Additionally, there must be a sensor s that acts as an instrument for p, and there must be a workpiece w that is the patient of p. Furthermore, for all instances of time t when p occurs, there must be a colour c and a corresponding colour name ci. These colour measurements ci are linked to actual colours c and associated with the workpiece w at specific time instances te.

⋱ (A.5) When an action of carrying p results in the realisation of function f, it is the same as an action of location change p. There must be a conveyor c that p uses as an instrument, and there must be a workpiece w that is the patient of p. Moreover, for all instances of time t when p occurs, there must be a location l, two distinct spatial regions $s1$ and $s2$, time intervals tb and te, and a part l of the conveyor c. The workpiece w must be located in l at time t, and l must occupy spatial regions $s1$ and $s2$ respectively during time intervals tb and te.

⋱ (A.6) If an action of actuating p results in the realisation of function f, it is the same as an action of location change p. There must be an actuator a that acts as an instrument for p, and there must be a workpiece w that is the patient of p. Furthermore, for all instances of time t when p occurs, there must be a location l, another location l', two distinct spatial regions $s1$ and $s2$, time intervals tb and te, a slope n, a conveyor c, and a part l of the conveyor. The locations l and l' must be part of the conveyor c and must occupy spatial regions $s1$ and $s2$ respectively during time intervals tb and te. The slope n and the conveyor c must be part of the scenario.

4.2.2. Malfunction

If a component fails in a process where it serves as the instrument, resulting in an inability to achieve its designated function, its state is malfunctioning. The repercussions of a component's malfunction can be evident both at the individual component and broader system levels. For example, if Actuator 1 fails to activate, the red workpiece might mistakenly be directed to collection point 3. Similarly, a sensing unit failure could misdirect a Metal workpiece into collection point 1. One can employ a series of rules to pinpoint a system-level error down to a specific component, which align with the system's modulo-junction structure showcased in Fig. 9.

We introduce the predicate "malfunctionedFor". This predicate establishes a relationship between a component, the specific instance of a process where the error transpired, and the workpiece the component unsuccessfully managed. This means that the process, where the workpiece was the patient and the component functioned as the instrument, did not deliver the component's intended function.

To illustrate the application of this methodology, four rules geared towards identifying errors when sorting red workpieces have been outlined. Note that different error types could necessitate their distinct rule sets.

$$\begin{aligned}&malfunctionedFor(c,p,w) \wedge RedSortingUnit(c) \wedge ActOfSortingRed(p)\\&\quad \wedge Workpiece(w) \wedge Conveyor(c') \wedge ActOfCarryingRed(p')\\&\quad \wedge isPatientOf(w,p') \wedge isInstrumentOf(c',p')\\&\quad \wedge \neg realizes(p','FunctionOfCarryingRed') \rightarrow malfunctionedFor(c',p',w)\end{aligned} \tag{A. 7}$$

$$\begin{aligned}&malfunctionedFor(c,p,w) \wedge RedSortingUnit(c) \wedge ActOfSortingRed(p)\\&\quad \wedge Workpiece(w) \wedge RedTransferringUnit(c') \wedge ActOfTransferringRed(p')\\&\quad \wedge isPatientOf(w,p') \wedge isInstrumentOf(c',p') \wedge Conveyor(c')\\&\quad \wedge ActOfCarryingRed(p') \wedge isPatientOf(w,p') \wedge isInstrumentOf(c',p')\\&\quad \wedge realizes(p','FunctionOfCarryingRed') \rightarrow malfunctionedFor(c',p',w)\end{aligned} \tag{A. 8}$$

$$malfunctionedFor(c,p,w) \wedge RedTransferringUnit(c) \wedge ActOfTransferringRed(p) \wedge Workpiece(w) \wedge SensingUnit(c') \wedge ActOfMeasuringRed(p') \wedge isPatientOf(w,p') \wedge isInstrumentOf(c',p') \wedge \neg realizes(p','FunctionOfMeasuringRed') \rightarrow malfunctionedFor(c',p',w) \quad \text{(A. 9)}$$

$$malfunctionedFor(c,p,w) \wedge RedTransferringUnit(c) \wedge ActOfTransferringRed(p) \wedge Workpiece(w) \wedge Actuator(c') \wedge ActOfActuatingRed(p') \wedge isPatientOf(w,p') \wedge isInstrumentOf(c',p') \wedge \neg realizes(p','FunctionOfPushingRed') \wedge SensingUnit(c'') \wedge ActOfMeasuringRed(p'') \wedge isPatientOf(w,p'') \wedge isInstrumentOf(c'',p'') \wedge realizes(p'','FunctionOfMeasuringRed') \rightarrow malfunctionedFor(c',p',w) \quad \text{(A. 10)}$$

∵ (A.7) If a sorting unit *c* malfunctions for an action of sorting red *p* concerning a workpiece *w*, and *c* is a red sorting unit, and *p* is an action of sorting red related to *w*, and there exists another conveyor *c'* for carrying red *p'* related to *w*, and *p'* is an action of carrying red related to *c'*, and *c'* is an instrument for *p'*, and *p'* does not realize the function of carrying red, then the malfunction also applies to *c'* for the action of carrying red *p'* concerning *w*.

∵ (A.8) If a sorting unit *c* malfunctions for an action of sorting red *p* concerning a workpiece *w*, and *c* is a red sorting unit, and *p* is an action of sorting red related to *w*. There exists a red transferring unit c' for transferring red p' related to w, and p' is an action of transferring red related to c', and c' is an instrument for p', and there exists a conveyor c'' for carrying red p'' related to w, and p'' is an action of carrying red related to c'', and p' realizes the function of carrying red. The malfunction also applies to *c'* for the action of transferring red *p'* concerning *w*.

∵ (A.9) If a sorting unit *c* malfunctions for an action of sorting red *p* concerning a workpiece *w*, and *c* is a red transferring unit, and *p* is an action of transferring red related to *w*, and there exists a sensing unit *c'* for measuring red *p'* related to *w*, and *p'* is an action of measuring red related to *c'*, and *c'* is an instrument for *p'*, and *p'* does not realize the function of measuring red, then the malfunction also applies to *c'* for the action of measuring red *p'* concerning *w*.

∵ (A.10) If a sorting unit *c* malfunctions for an action of sorting red *p* concerning a workpiece *w*, and *c* is a red transferring unit, and *p* is an action of transferring red related to *w*. There exists an actuator c' for actuating red p' related to w, and p' is an action of actuating red related to c', and c' is an instrument for p', and p' does not realize the function of pushing red. There exists a sensing unit *c''* for measuring red *p''* related to *w*, and *p''* is an action of measuring red related to *c''*, and *c''* is an instrument for *p''*, and *p''* realizes the function of measuring red, then the malfunction also applies to *c'* for the action of actuating red *p'* and to *c''* for the action of measuring red *p''* concerning *w*.

Rule 1 (A. 7) states that if there is an identified failure with the RedSortingUnit (a preliminary fact signifying a red workpiece sorting error), the fault is ascribed to the Conveyor when the related event, ActOfCarryingRed, does not execute properly. The negation of "realizes" indicates that the desired outcome depicted in the right part of (A. 5) is not met, meaning the required output state is not achieved.

If Rule 1 (A. 7) is not activated, implying that the Conveyor is functioning as expected, then Rule 2 (A. 8) attributes the malfunction to the RedTransferringUnit. If the RedTransferringUnit is at fault, Rule 3 (A. 9) and Rule 4 (A. 10) are employed to determine if the SensingUnit or RedTransferringUnit, respectively, are the root causes of the malfunction.

As the SensingUnit is the control mechanism for the RedTransferringUnit, it is not necessary to carry on with the rules and verify if the Actuator also malfunctions (as per (A. 4)). Conversely, the core malfunction can only be linked to the Actuator if the ActOfActuatingRed does not execute (meaning it does not fulfil the Actuator's function), while the prior process, ActOfMeasuringRed, works as expected (meaning it achieves the SensingUnit's function).

4.3. Develop ontology

The first step to creating an application ontology, as per 3.3, is to delimit the subject or formalise the ontology's purpose. To do that, a set of CQs have been created to formalise the pieces of information that the CDT can manage. The CQs created for this use case are the following:

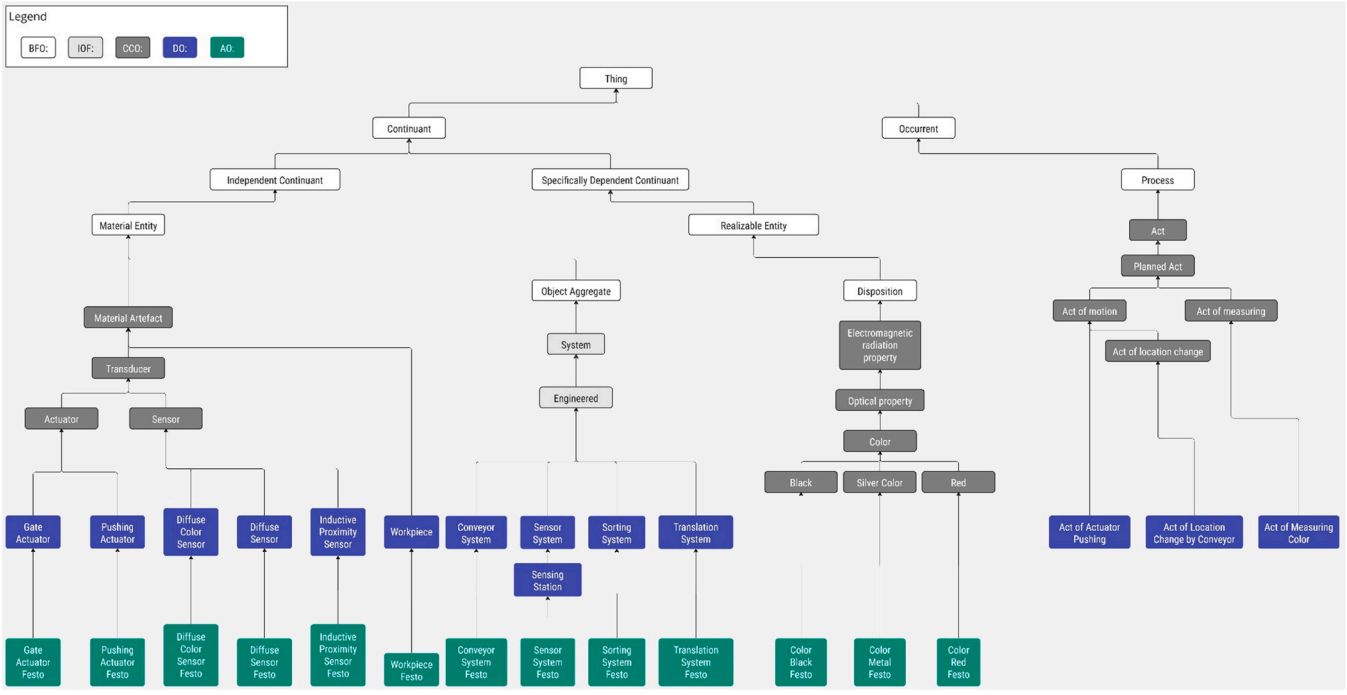


Fig. 10. Overall structure of the ontology.

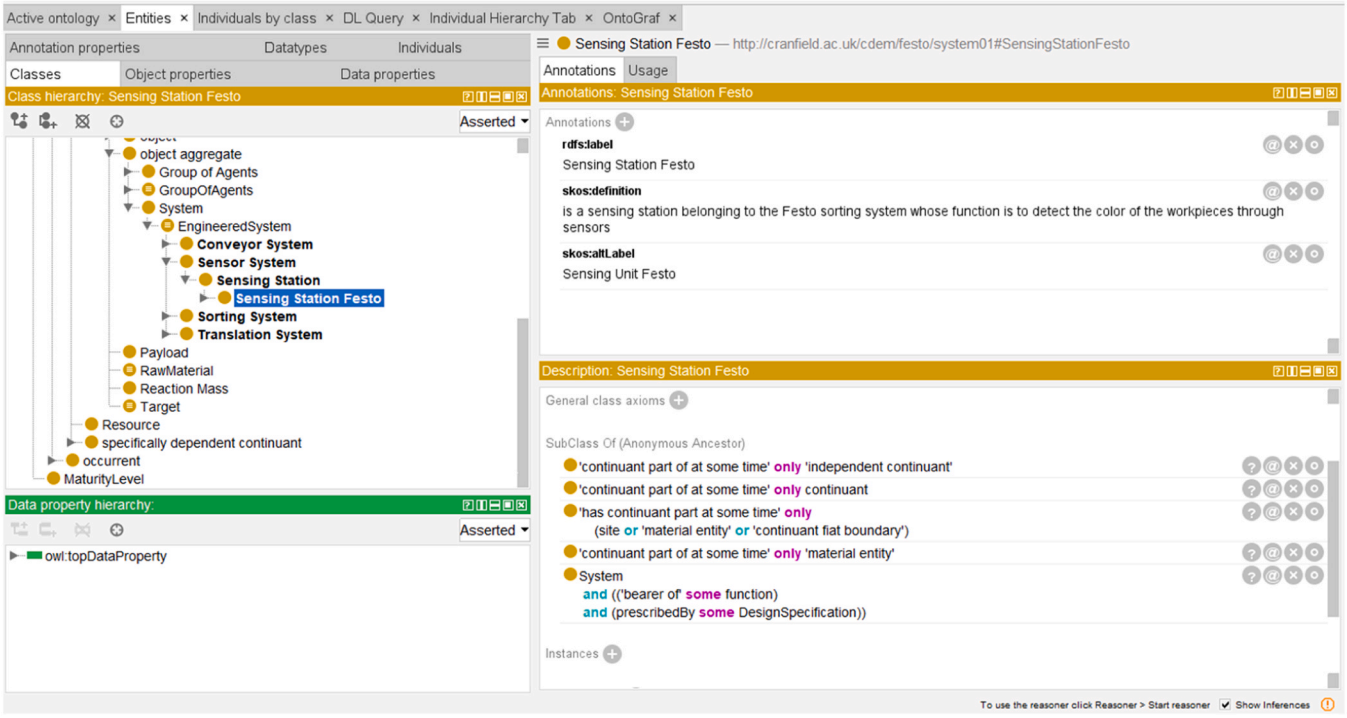


Fig. 11. Screenshot of the Protégé tool focusing on the class Sensing Station Festo.

1. What are the components of the Festo sorting station?
2. How long does the conveyor take to carry the workpiece X?
3. What is the time that the workpiece X entered the sensing unit?
4. What is the colour of workpiece X sensed by the sensing unit?
5. How many workpieces are at the collection point X at time T?
6. Is the workpiece X sensed by the sensing unit within the time interval T?

The next step is to develop the terminology based on the terms used in the CQs and the terms used in the existing ontologies. Regarding the latter, existing ontologies are TLOs and MLOs widely recognised among the scientific and industrial community. The authors, following careful research, have chosen the ontology model for the Festo sorting system by extending the predicates from the Common Core Ontology (CCO) (available at https://www.cubrc.org, accessed on 01/08/2023) and IOF-Core [35]. These ontologies are grounded in a TLO known as the Basic Formal Ontology (BFO) (published by Smith et al. Smith et al., [50]). Given BFO's recognition as an ISO standard (ISO 21838–2:2021), employing it ensures that the application ontology is both universal and interoperable.

A few examples of terminologies inherited from those TLOs and MLOs are terms such as "bears," "realizes," "participatesIn," "OccurrentPartOf," "occursOn," "hasFirstInstance," and "hasLastInstance" and are derived from BFO. Terms such as "hasOutput," "ActOfLocationChange," and "ActOfMeasurement" are borrowed from CCO. The term "System" is sourced from IOF-Core. The authors ensure a rigorous and comprehensive approach to structuring and detailing the Festo sorting system's ontology model by aligning with these foundational ontologies.

The next step is to create the taxonomy, starting from the identified TLOs and the MLOs. The ontology is created using Protégé (https://protegewiki.stanford.edu/wiki/Main_Page, accessed on 01/08/2023). This tool allows the importation of existing ontologies (TLO and MLOs) and creates the new DLO and ALO, extending the previous ones and exporting them into a file that formalises the outcome in a computer-

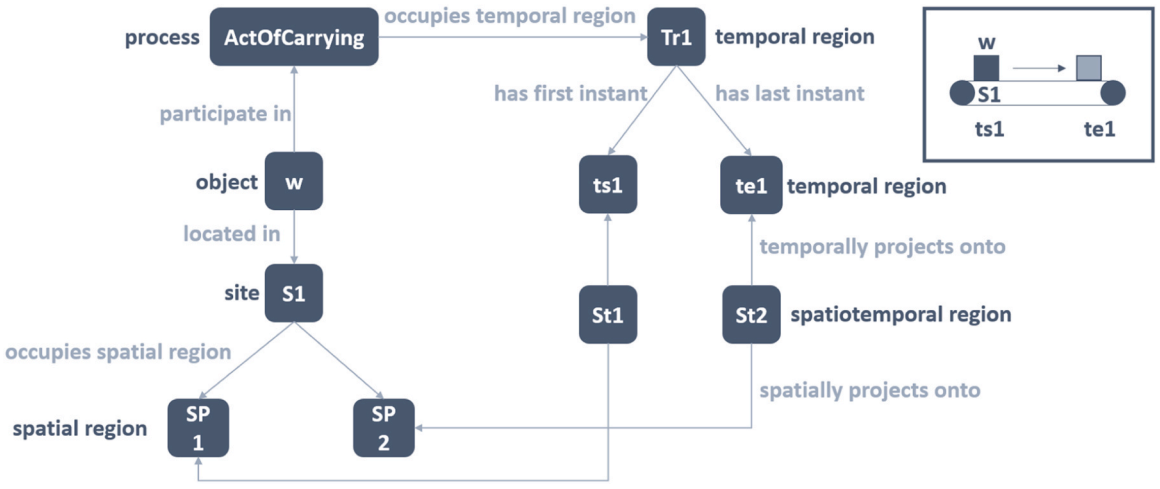


Fig. 12. Diagram that represents the process ActOfCarrying using BFO properties.

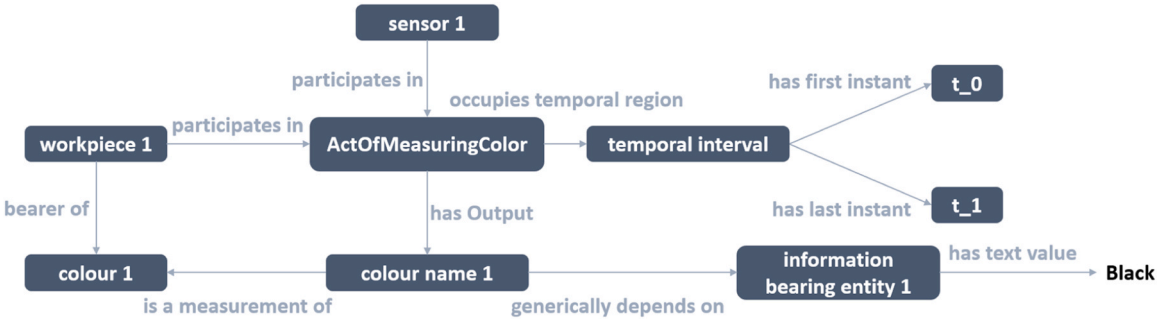


Fig. 13. Diagram that represents the process ActOfMeasuringColor using BFO and CCO properties.

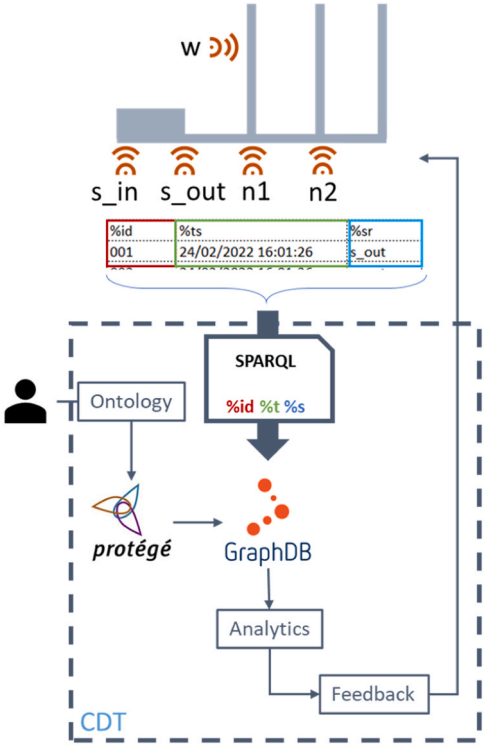


Fig. 14. Data flow in the CDT.

Table 4–1
Colour detection from sensors' output combination. (DS = Diffuse sensor, DCS = Diffuse colour sensor, IPS = Inductive proximity sensor).

	DS	DCS	IPS
Black	1	0	0
Red	1	1	0
Metal	1	1	1

```
INSERT{
    ?WP bfo:BFO_0000108 _:b1; #exists at ?t
        bfo:BFO_0000082 ?site. #located in at all times ?site
    ?site bfo:BFO_0000210 ?s. # site Occupies Spatial Region
    _:b1 a  bfo:BFO_0000203; #?t a temporal instant
         cco:designated_by _:b2. #ice_timeValue
    _:b2 a  cco:TemporalInstantIdentifier;
         ro:RO_0010001/cco:has_datetime_value   ?time.
    ?st  a  bfo:BFO_0000011; #?st1 a spatiotemporal region
         bfo:BFO_0000153 _:b1; #st1 temporally projects onto t
         bfo:BFO_0000216 ?s. # st1 spatially project onto sp
}
 WHERE{
     BIND(IRI(CONCAT(STR(fes:WFMPS), STR(?WPID))) as ?WP)
     BIND(%id as ?WPID)
     BIND(IRI(CONCAT(STR(fer:site_A), STR(?WPID))) as ?site)
     BIND(IRI(CONCAT(STR(fer:S), %sr)) as ?s)
     BIND(%tr as ?time)
     BIND(IRI(CONCAT(STR(fer:s_out),STR(?time)))  as ?st)
 }
```

Fig. 15. Example of SPARQL query to send data point to the KG.

usable language. Moreover, it is possible to reuse and create data and object properties and annotations about the new class using Protégé .

Starting from BFO, CCO, and IOF-Core, a DLO and ALO have been created as reported in Fig. 10. In the diagram, DLO represents the domain-level ontology or rather the ontology that can be applied in the material handling systems domain, while ALO represents the application ontology or rather the ontology developed for this specific Festo sorting station module. Each class has been defined mainly using two methodologies for the definitions, i.e., the Aristotelian and the non-root terms one [7]. An example is reported in Fig. 11.

The processes can be expressed using the BFO properties. An example is the process ActOfCarrying (A. 5), as reported in Fig. 12. The process represents the workpiece moving in the conveyor belt at a specific time range. The properties in light blue are taken directly from BFO, considering the domain and range for each property, whereas the classes in dark blue are the parent classes belonging to BFO, to reflect the domain and range of each property. Another example is the process ActOfMeasuringColor (A. 4), reported in Fig. 13, represented using the properties and classes from BFO and CCO. In particular, the connection between the literal (string: "Black") and the instance ("colour 1") is made through the information-bearing entity class of CCO.

4.4. Data

The outcome of this phase is a file in the Web Ontology Language (OWL) format. The OWL file created will then be imported into the data management tool together with all the other TLO and MLOs used. The data is collected and managed within the same data management tool. The data management tool adopted for this project is GraphDB by Ontotext, an open-source graph database software that allows the importation of multiple ontologies and the receipt of data through an HTTP connection.

SPARQL is the query language used within this environment. Data has been included in SPARQL queries and used to send information about the structure and functions of the system and its components, the related processes, and the reasoning for the malfunctions, as reported in Fig. 14.

The sensing unit is composed of three sensors: i) a diffuse sensor (DS), ii) a diffuse colour sensor (DCS), and iii) an inductive proximity sensor (IPS). The combination of those sensors' output determines the

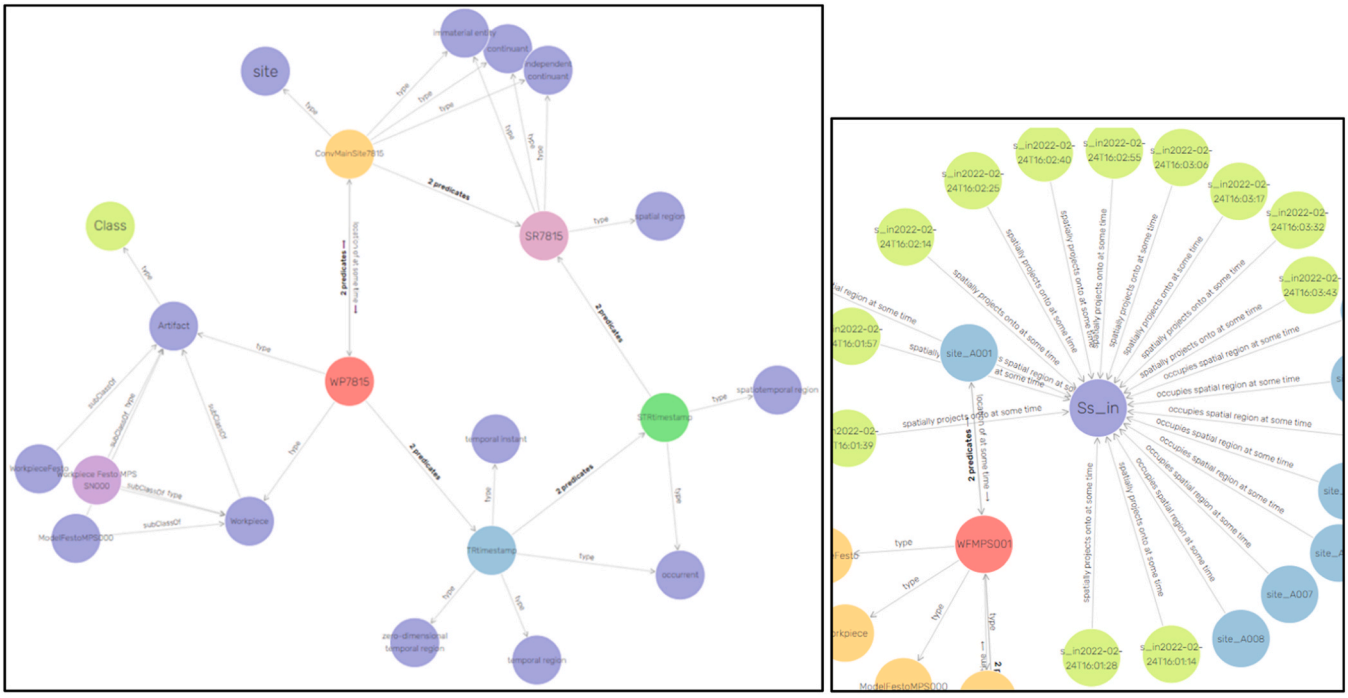


Fig. 16. Snapshot of the KG populated. Part of the ontology (left), and data population containing data related to workpiece ID001 in space s_in (right).

workpiece's colour, as reported in Table 4–1.

An example of the query to import the data into the KG is reported in Fig. 15. The numbers from the sensors are going to be placed where the red underlined placeholders are. This query is enclosed in a script sent via an HTTP transmission to the CDT. Data is transferred into the software (GraphDB) shaped as a knowledge graph. The extraction of the KG is reported in Fig. 16. The ontology imported in the KG shapes the data.

```
SELECT DISTINCT ?c1 ?c1
WHERE {
    ?c fes:hasSubsystemOf* ?c1.
    ?c1 rdfs:label ?c1.
    FILTER(?c = fes:SOUFSS000)
}
```

	c1
1	"Sorting Unit Festo Sorting Station SN000"
2	"Translation Unit Festo Sorting Station SN000"
3	"Sensing Unit Festo Sorting Station SN000"
4	"Conveyor Festo Sorting Station SN000"
5	"Pushing Actuator Festo Sorting Station SN000"
6	"Diffuse Color Sensor Festo Sorting Station SN000"
7	"Diffuse Sensor Festo Sorting Station SN000"
8	"Inductive Proximity Sensor Festo Sorting Station SN000"
9	"Gate Actuator Festo Sorting Station SN000"

All the instances are then appended to the relative object imported from the ontology, as reported in Fig. 16 (right).

In this paper, the only kind of database used is the KG. Other kinds of databases (Fig. 5), such as relational databases, can be used in association with the KG, but are not considered for this project, as they are not needed in this particular case.

4.5. Analysis

4.5.1. Competency questions

One of the methods to verify the operation of the CDT, especially the ontology, is to query the CQs to the CDT. The 6 CQs are converted into SPARQL queries and launched into the GraphDB tool. The 6 CQs are listed below, for brevity, only the first and the last ones include the SPARQL query.

1. What are the components of the Festo sorting station?

2. How long does the conveyor take to carry the workpiece X?

	w	p	time1	time2
1	:WFMPS001	_:node26516	"2022-02-24T16:01:26"	"2022-02-24T16:01:29"

3. What is the time that the workpiece X entered the sensing unit?

	w	time	s
1	:WFMPS001	"2022-02-24T16:01:14"	http://cranfield.ac.uk/cdem/festo/Ss_in

4. What is the colour of workpiece X sensed by the sensing unit?

	w	s	color
1	:WFMPS001	http://cranfield.ac.uk/cdem/festo/Ss_out	"red"

5. How many workpieces are at collection point X at time T?

	result
1	"12"^^xsd:integer

6. Is workpiece X sensed by the sensing unit within the time interval T?

```
ASK {
    ?p  a
fes:ActOfMeasuringColor;
            bfo:BFO_0000057 ?w;
#process has participant
```

Run

keyboard shortcuts

Query took 0.1s, moments ago.

YES

.

4.5.2. Detection of malfunction

A test scenario has been presented to showcase the reasoning process for identifying a malfunctioned component in the event of a system error. In this scenario, a discrepancy is observed at collection point 1, where the actual number of workpieces present (CQ 5) is fewer than the expected quantity. Consequently, the comparison script flags a system-level error, as reported in Fig. 17. Specifically, the workpiece with ID 004 is identified as one of the missing items from the initial collection point.

In order to perform this comparison script, the malfunction function has been created (1). The malfunction function has three parameters: the component (instrument) c, the process p, and the workpiece (patient) w.

$$malfunctionedFor(c, p, w) \tag{1}$$

The function with three parameters cannot be managed in OWL. Hence, the reification has been applied to adopt this function. This procedure decomposes the function into three functions with two parameters each: *malfunctionedComponent*, *malfunctioningProcess*, and *mafunctionedFor*. The schema (reported in Fig. 18) for the component, the process, and the workpiece has been created, reporting the values found in the result of the query in Fig. 17.

	RedSortingUnit	ActOfSortingRed	workpiece	malfunction
1	_:node26827	_:node26826	:WFMPS004	_:node26828

Fig. 17. Malfunctioning of the red sorting unit.

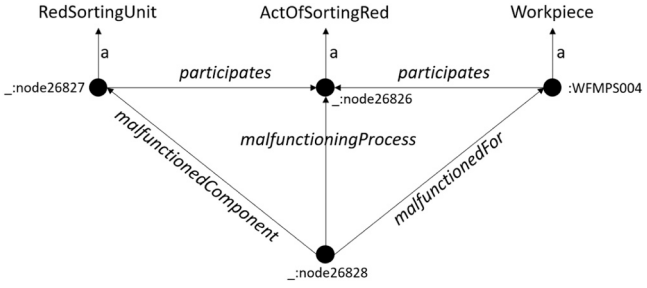


Fig. 18. Schema for the reification.

A component malfunctions if the process in which the component plays the role of the instrument fails to realise its corresponding function. An error/malfunction at the system level can be narrowed down to a component by applying the rules (A. 7) to (A. 10) in order to find the process in which a particular workpiece was a patient and the component an instrument did not realise the function of the component.

The rules are contained in the ontology in the form of Semantic Web Rule Language (SWRL), inserted in Protégé after completing the ontology. SWRL is a rule language for the Semantic Web that extends the OWL. It allows users to specify rules in the form of logical expressions that can be used to make inferences about the data in an OWL ontology. SWRL rules are written using a combination of OWL constructs and a rule syntax that allows the creation of conditional expressions, using constructs such as "if-then" statements, "and" and "or" operators, and mathematical comparisons. SWRL rules can be used to automate reasoning tasks and to infer new knowledge from existing data.

In this instance, upon applying the rules ((A. 7) to (A. 10)) to the KG, rule (A. 8) is activated. This occurs because the process *ActOfCarryingRed* successfully fulfils the Conveyor's function, as detailed by the state description in Fig. 19 (specifically, Workpiece 004 is transported from *s_out* to *w*). As a result of rule (A. 8) being triggered, the assertion is made that the *RedTransferringUnit* is experiencing a malfunction. The *RedTransferringUnit* is the instrument of some *ActOfTransferringRed*, for which some *ActOfMeasuringRed*, *ActOfCarryingRed*, and *ActOfActuatingRed* are occurrent parts.

Following this assertion, only rule (A. 9) is triggered. This is due to the discovery that the output measurement indicates 'black' instead of 'red', a discrepancy that violates (A. 6), specialised for *ActOfActuatingRed*. This violation leads to the non-realisation of the SensingUnit's function.

Rule (A. 9) will assert the *SensingUnit* as the malfunctioned unit as reported in Fig. 20.

The succession of the rule applied to the Red Sorting Unit component is reported in Fig. 21.

This kind of feedback from the analytics section goes back to the user and aims to support the decision-making process at this stage.

5. Discussion

This paper presented a five-step methodology to develop CDTs. The methodology consists of understanding the composition of the system of interest, the definition of the processes and knowledge base around the system, developing the ontology, defining the data strategy, and performing the analytics according to the purpose of the CDT. It provides practical guidance, promotes a holistic perspective, offers flexibility, and encourages interdisciplinary collaboration. Moreover, through a formal ontological approach, the successful transfer of human expertise into the CDT has been achieved. To demonstrate and validate this, this paper has used the detection of the malfunctioned component in the case of a system-level error as a validation of the CDT. Furthermore, as per their intrinsic properties, the use of ontology, particularly the TLO

	Workpiece	Time	Location	Site
1	:WFMPS004	"2022-02-24T16:01:57"	http://cranfield.ac.uk/cdem/festo/Ss_in	http://cranfield.ac.uk/cdem/festo/site_A004
2	:WFMPS004	"2022-02-24T16:02:12"	http://cranfield.ac.uk/cdem/festo/Ss_out	http://cranfield.ac.uk/cdem/festo/site_A004
3	:WFMPS004	"2022-02-24T16:02:16"	http://cranfield.ac.uk/cdem/festo/Sw	http://cranfield.ac.uk/cdem/festo/site_C004

Fig. 19. State description of workpiece ID 004.

	RedSensingUnit	ActOfMeasuringRed	Workpiece	Malfunction
1	_:node26846	_:node26847	:WFMPS004	_:node26848

Fig. 20. Malfunctioning for the Sensing Unit.

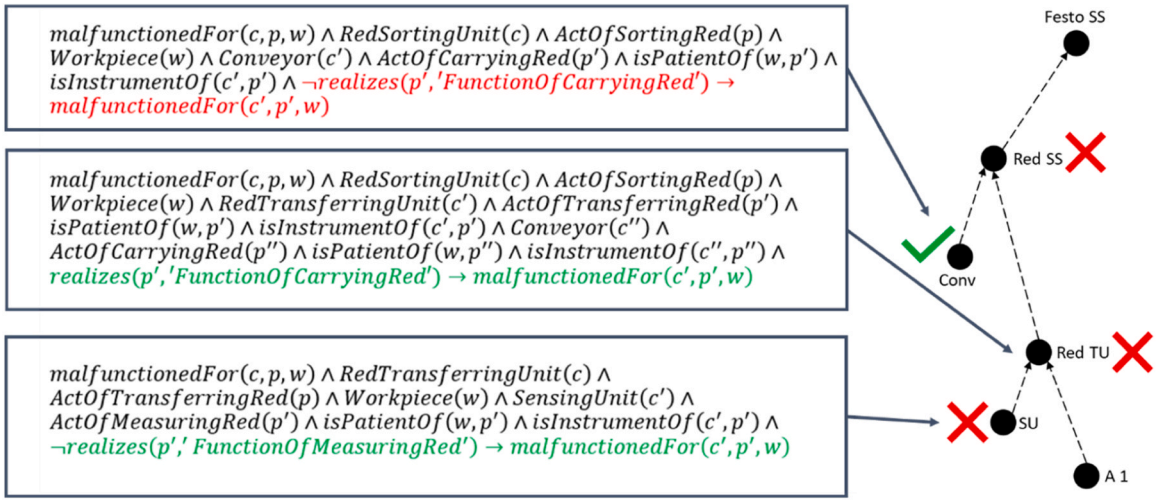


Fig. 21. Visual application of the rules into the system composition.

approach, promotes reusability, interoperability, and interactions among the next-generation CDTs.

This work demonstrates the capability of the proposed methodology to develop a CDT and transfer human expertise into it. The results presented in this paper answer the research questions posed in the introduction section.

1) *Can an ontology approach be used to develop a DT?* Yes, an ontology approach can be used and bring benefits for the DT. As seen in previous sections, a new paradigm has been created that brings semantic technologies into the DT, which is the CDT. Ontologies are formal representations of knowledge that define concepts and relationships. Combining these two can offer several benefits in creating and managing DTs:

- *Semantic Interoperability*: Ontologies provide a standardised way to define concepts, relationships, and properties within a domain. Using ontologies can ensure that different DT components (e.g., sensors, models, and simulations) share a common understanding of the domain, enabling semantic interoperability between different systems and tools.
- *Data Integration*: Digital Twins often involve data from various sources and formats. Ontologies integrate heterogeneous data by providing a common vocabulary and structure, allowing data to be transformed and interpreted consistently across the DT's lifecycle.
- *Knowledge Sharing and Reuse*: Ontologies capture domain knowledge in a structured manner. Different stakeholders can easily share and reuse this knowledge in the development, operation, and maintenance of the DT.
- *Reasoning and Inference*: Ontologies enable automated reasoning and inference, which can be valuable for detecting anomalies, predicting behaviour, and optimising the performance of the DT based on the defined domain knowledge.
- *Adaptability*: As the physical entity or system evolves, the ontology-based DT can be updated to reflect these changes more efficiently, ensuring that the virtual representation remains synchronised with the real-world counterpart.
- *Lifecycle Management*: Ontologies can provide a foundation for managing the entire lifecycle of a DT, from its initial design and development to deployment, operation, and eventual decommissioning.
- *Collaboration*: An ontology-based approach encourages collaboration among experts from different disciplines by offering a shared framework for understanding and discussing the DT's domain.

However, while ontologies offer numerous benefits, they also require careful design and maintenance to ensure they accurately represent the domain and remain aligned with any changes in the real-world system, see Section 3.3. There are some drawbacks to this approach as well. Effective ontology development relies on domain experts who understand the intricacies of the specific field. The availability of such experts can be a limiting factor. In some cases, using ontologies to represent knowledge may be considered over-engineering, especially for simple systems, those with limited knowledge representation needs, or those systems that are siloed and do not require interaction with other systems.

2) *Can human expertise be transferred to the CDT?* Yes, it is possible to transfer human expertise into the CDT via the adoption of a formal ontology approach. Throughout the paper, this knowledge transfer concept, from humans to the CDT has been validated using the Festo system as an example. The human understanding of the system has been converted from natural language into FOL rules, from (A. 7) to (A. 10) and then converted into SPARQL queries able to run into the KG, as reported in Section 4.5. This Festo example has been selected for its simplicity, and through this, the authors could demonstrate the procedure to convert human expertise into CDT knowledge.

In general, to import existing human expertise into the CDT, the following procedure is recommended:

1. A formal ontology approach must be selected and followed, especially with the definition of components, functions, properties, processes, etc.
2. The natural language knowledge should be converted into FOL axioms, following the properties defined in the previous point.
3. The FOL axioms can then be converted into SWRL and loaded into the ontology file using a tool similar to Protégé .

4. Upload the ontology file containing the SWRL rules into KG software and perform the reasoning within the KG.

In this way, human expertise is imported into the CDT, reasoning can be performed into the KG, and new knowledge can be inferred from existing knowledge. The answer outlines a structured procedure for transferring human expertise to a CDT using formal ontology approaches. However, it does not address the practical challenges and complexities associated with each step of the process. For example, the conversion of natural language knowledge to FOL axioms can be challenging, especially when dealing with ambiguous human language. The answer also mentions using tools like Protégé for ontology editing and SWRL rules. While these tools are valuable, it is important to acknowledge that their effective use requires a learning curve and expertise. Additionally, tool compatibility and versioning issues may arise.

3) *Can a CDT improve the process of detection of a failing component?* Yes, this paper has demonstrated that the CDT is able to detect which component is malfunctioning, starting from a general system error. Having imported human expertise translated into rules, the CDT has the capability to autonomously go through these rules and detect the malfunctioning components, like human thought, to the CDT. Without these rules, the CDT would have launched the system-level error, and the human operator should have done the whole procedure manually to understand which component was malfunctioning. This technology is valuable for industries where timely and accurate fault detection is critical for system performance and safety.

Although the answer mentions the translation of human expertise into rules, it does not address the potential complexity of creating and managing these rules, especially in large and intricate systems. Rule maintenance can be a challenging task. Moreover, the answer highlights the importance of translating human expertise into rules for the CDT to operate effectively. The accuracy and comprehensiveness of this translation process can significantly impact the CDT's detection capabilities.

Moreover, another critical limitation arises from the difficulties associated with dynamically integrating expertise knowledge into a CDT. While the dynamic adaptiveness of data is a fundamental characteristic of standard DTs and, by extension, of CDTs [18], this aspect of integrating expertise knowledge presents a complex challenge and necessitates further research and study [26,3,41]. It involves the initial translation of expertise into rules and the ongoing adaptation and updating of these rules to keep pace with evolving expert knowledge and system changes.

The methodology presented in this paper differs from the other methodologies in the literature as this one does not focus on the architecture but rather on the procedure to develop CDTs. Traditional methodologies in the field often prioritise the architectural aspects of CDTs. These methodologies tend to delve deeply into the design and structure of the CDT, including its components, data flows, and system architecture. In contrast, the methodology presented in the paper emphasises the procedural aspects of CDT development. It concentrates on the step-by-step processes, methodologies, and best practices for creating CDTs, considering how to effectively define, model, and operationalise them. Rather than diving immediately into technical architecture, it guides practitioners through the process of conceptualisation, knowledge integration, and operationalisation of CDTs within a specific domain. Moreover, the methodology presented is tool-agnostic and can be adapted to various applications within the industry domain. The case study used for the validation of the methodology serves as a proof-of-concept and does not fully cover all the aspects explained in the methodology. Therefore, additional challenges, out of the scope of this paper, need to be considered in the implementation on a complex engineering asset. The methodology has been applied to a real-world application (Section 4) and can be used as guidance to develop new CDTs.

The use of ontologies for developing CDTs introduces limitations such as the complexity of ontology creation and maintenance, requiring domain expertise for accurate representation; challenges in scaling up ontologies for larger and evolving systems; potential gaps in formalising all forms of human expertise; the need for ongoing maintenance and alignment as the real-world system changes; potential difficulties in achieving standardised interoperability and integration across systems; technical implementation requirements and potential data integration challenges; limitations in addressing exceptionally complex scenarios; and dependence on the accuracy of human expertise translated into rules, impacting CDT performance if initial knowledge is flawed or incomplete.

Table 5–1
Advantages and disadvantages of the CDT approach using TLO enabling federation of twins.

Advantages	Disadvantages
Enhanced reasoning capabilities	High complexity
Reduction of human intervention and better support for decision-making	Resource intensive
Improved interoperability and knowledge sharing	Expertise requirement
Improved predictive analytics supported by a federated approach	Security and privacy concerns

Further limitations, specifically on the use case in relation to the automation and autonomy of the CDT are presented below. The focus of this project was more on the validation of the semantic capability of the DT, such as the ontology, the KG, and the capability to transfer human expertise into the CDT. The automation aspect considers the capability of the CDT to make queries systematically. It can be improved using additional sensors and scripts with control checks. The second aspect to consider is the autonomy of the CDT. It looks at the capability of autonomously running the rules, inferring relationships, and supporting decision-making. This aspect can be improved over time by working more with graph algorithms on the KG. The implementation of these aspects has to be considered in the design phase of the CDT, or as an iterative implementation optimising the operation of the asset.

Based on the content of this article, the advantages and disadvantages of the CDT approach using TLO enabling the federation of twins are reported in Table 5–1. The advantages include enhanced reasoning capabilities, as a deterministic reasoning system is included on top of the possible probabilistic models (mostly based on Machine Learning); reduction of human intervention and better support for decision-making, since CDTs provide a digital framework to formalise human expertise, reducing human intervention throughout the lifecycle and enhancing support in the decision-making process, aiming for an autonomous decision-making process; improved interoperability and knowledge sharing, as CDTs use formal semantics and ontologies to enable better interoperability between different systems and across various domains. This shared semantic framework facilitates more effective knowledge transfer and integration compared to the more isolated traditional DTs; and improved predictive analytics supported by a federated approach. By incorporating cognitive capabilities, CDTs can more accurately predict equipment failures and optimise maintenance schedules. This leads to reduced downtime and maintenance costs, as CDTs can foresee and mitigate potential issues before they occur. This can also be supported by the network of CDTs connected in the federation of twins.

In the validation phase presented in the article, the CDT applied to the Festo Sorting Station showed significant improvements in two main areas: i) the detection of malfunctions, where the CDT was used to detect malfunctions in the sorting station, significantly reducing the need for skilled human manual intervention, and ii) semantic enrichment, where the CDT enhanced the DT services by providing advanced reasoning capabilities, leading to a profound semantic enrichment of the data.

While CDTs offer significant advantages, they also come with notable disadvantages. These include increased complexity, as the integration of formal semantics and ontologies, along with rule-based deterministic models, makes CDTs more complex to design and implement compared to traditional DTs; resource intensiveness, as integrating data from

multiple sources and ensuring interoperability between different systems can be complex and resource-intensive; expertise requirements, as developing and managing CDTs requires expertise in various aspects of semantics, graph technologies, and domain-specific knowledge, making it difficult to find and retain skilled professionals; and security and privacy concerns, as the extensive data usage and integration in CDTs raise significant security concerns. Protecting sensitive data from breaches and ensuring compliance with privacy regulations is crucial. Advanced systems like CDTs can be more vulnerable to cyber-attacks due to their complexity and reliance on interconnected data sources.

6. Conclusion

The paper outlines a methodology comprising five steps for creating CDTs leveraging ontologies to enhance their development. These ontologies offer benefits like standardised understanding, integrated data, knowledge sharing, reasoning, adaptability, lifecycle management, and collaboration. The paper demonstrates the transfer of human expertise into CDTs through formal ontological approaches, exemplified by converting human understanding into rules and integrating them into ontologies for reasoning in the KG. Additionally, the paper showcases how CDTs, enriched with human expertise, can autonomously detect failing components when system errors occur, reducing the need for manual intervention and improving overall performance in complex systems. The fundamental benefit of DTs is their ability to predict and simulate physical states. Since CDTs encapsulate all the attributes of DTs, they retain this predictive and rehearsal capacity and enhance it through their cognitive dimension.

Integrating human expertise into the digital domain is set to bolster the predictive analytics capabilities of DTs significantly. Our approach augments the traditional reliance on physical models embedded within DTs. By incorporating the insights and experience of experts and operators familiar with the asset, we provide a more nuanced understanding of its behaviour. Hence, this methodology improves DTs' predictive precision and self-sufficiency, enabling them to act more effectively and autonomously in foreseeing and managing potential issues within complex systems.

Future research in this field is needed to address several critical aspects. The CDT concept requires further elucidation, necessitating the adoption of a unified definition to mitigate inconsistencies. Additional applications are essential to validate various dimensions of this paradigm thoroughly. Conducting case studies across diverse domains holds significant promise for assessing the practical implementation, advantages, and challenges of ontology-based CDTs, offering valuable insights for real-world adoption. Further exploration is warranted into mechanisms that ensure the precision and comprehensiveness of human expertise translated into ontologies, thereby fortifying the reliability of CDTs. Advancing reasoning and inference techniques to manage intricate scenarios marked by complexity and uncertainty would substantially bolster the capabilities of CDTs in handling intricate systems. Another promising avenue is delving into mechanisms to enhance the synergy between human experts and CDTs, enabling continual ontology refinement and knowledge transfer. Efforts to automate the translation of human expertise into formal axioms and rules could streamline the process and minimise dependence on manual conversion. Seamless integration of ontologies with existing systems, databases, and data formats is pivotal for the pragmatic implementation of ontology-based CDTs. Research on real-time ontology updates, aligning with changes in the physical system, holds immense potential for enhancing the adaptability and relevance of CDTs over time. Formulating standards and protocols for ontologies within the context of CDTs is vital to augment interoperability between different systems and stakeholders. The authors do not claim that BFO is the best TLO existing today. In fact, this methodology can be followed by selecting another TLO. The authors believe that a choice must be made to make some progress in this field.

In conclusion, this research highlights the significant advancements and potential of CDTs in enhancing the capabilities of traditional DTs. The integration of formal semantics within CDTs offers profound improvements in reasoning, interoperability, and operational efficiency. However, these advancements come with increased complexity, higher costs, and security and privacy concerns. While this study underscores the benefits and drawbacks of CDTs, it also opens the door for further exploration. Future research could focus on a detailed performance comparison between CDTs and standard DTs, using a well-defined set of metrics. Such comparative studies would provide deeper insights into the practical advantages and limitations of CDTs, guiding their implementation in various industrial domains..

Table 6–1
Main takeaways from this article.

#	Main Takeaway
1	Introduces a 5-step methodology for developing Cognitive Digital Twins (CDTs) to enhance Digital Twin (DT) services with advanced reasoning capabilities and semantic enrichment.
2	Highlights the concept of the "Federation of Twins" for seamless collaboration and shared semantics across sectors, emphasising interoperability and system efficiency.
3	Emphasises understanding system components, boundaries, and relations using mereotopology to ensure effective system integration and interoperability.
4	Focuses on systematically collecting and formalising knowledge, including human expertise, into the CDT using thematic roles and formal logical statements.
5	Details a structured approach to developing ontology for CDT, ensuring interoperability and reusability by leveraging existing upper-level and mid-level ontologies.
6	Discusses selecting appropriate data management strategies and database technologies based on data complexity, emphasising the use of graph databases for semantic data modelling.
7	Explains the validation strategy for CDT development, comparing digital models with physical entities to enhance decision-making and operational efficiency.
8	Validates the methodology using a practical example, demonstrating the application of CDT in detecting malfunctions and reducing manual interventions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the Centre of Digital Engineering and Manufacturing (CDEM) at Cranfield University (UK) and by the Production Engineering Laboratory LGP-UTTOP, University of Technology of Tarbes (France). The authors would also like to acknowledge Babcock International for supporting this work.

References

[1] Abburu S, Berre AJ, Jacoby M, Roman D, Stojanovic L, Stojanovic N. COGNITWIN - hybrid and cognitive digital twins for the process industry. Proc - 2020 IEEE Int Conf Eng, Technol Innov, ICE/ITMC 2020 2020. https://doi.org/10.1109/ICE/ITMC49519.2020.9198403.

[2] Adu-Kankam KO, Camarinha-Matos LM. A Framework for the Integration of IoT Components into the Household Digital Twins for Energy Communities (IFIP) IFIP Adv Inf Commun Technol 2022;665:197–216. https://doi.org/10.1007/978-3-031-18872-5_12/COVER.

[3] Akroyd J, Mosbach S, Bhave A, Kraft M. Universal digital twin-A dynamic knowledge graph. Data-Centr Eng 2021;2(4)):14. https://doi.org/10.1017/dce.2021.10.

[4] Al Faruque MA, Muthirayan D, Yu SY, Khargonekar PP. Cognitive Digital Twin for Manufacturing Systems. *DATE, 2021-February* Proc -Des, Autom Test Eur 2021: 440–5. https://doi.org/10.23919/DATE51398.2021.9474166.

[5] Almeida MB, Barbosa RR. Ontologies in knowledge management support: A case study. J Am Soc Inf Sci Technol 2009;60(10):2032–47. https://doi.org/10.1002/asi.21120.

[6] Arista R, Zheng X, Lu J, Mas F. An Ontology-based Engineering system to support aircraft manufacturing system design. J Manuf Syst 2023;68:270–88. https://doi.org/10.1016/J.JMSY.2023.02.012.

[7] Arp, R., Smith, B., & Spear, A.D. (2015). Building ontologies with Basic Formal Ontology. The MIT Press.

[8] Ashraf M, Eltawil A, Ali I. Time-To-Recovery Prediction in a Disrupted Three-Echelon Supply Chain Using LSTM. IFAC-Pap 2022;55(10):1319–24. https://doi.org/10.1016/J.IFACOL.2022.09.573.

[9] Baek MS. Digital Twin Federation and Data Validation Method. APCC 2022 - 27th Asia-Pac Conf Commun: Creat Innov Commun Technol Post-Pandemic Era 2022: 445–6. https://doi.org/10.1109/APCC55198.2022.9943622.

[10] Bechberger, D., & Perryman, J. (2020). Graph databases in action. Manning Publications.

[11] Bonnaud, S., Didier, C., & Kohler, A. (2019). Industry 4.0 and Cognitive Manufacturing Architecture Patterns, Use Cases and IBM Solutions.

[12] Borgo, S., Sanfilippo, E.M., & Terkaj, W. (2021, September). Capabilities, Capacities, and Functionalities of Resources in Industrial Engineering. *FOMI2021: 11th International Workshop on Formal Ontologies Meet Industry*.

[13] Boschert S, Heinrich C, Rosen R. Next Generation Digital Twin. Proc TMCE 2018 2018.

[14] Casati, R., & Varzi, A.C. (2003). Parts and Places: The Structures of Spatial Representation. In MIT press: Vol. MIT Press. The MIT Press. https://doi.org/10.7551/MITPRESS/5253.001.0001.

[15] Correia JB, Abel M, Becker K. Data management in digital twins: a systematic literature review. Knowl Inf Syst 2023;65(8):3165–96. https://doi.org/10.1007/S10115-023-01870-1.

[16] D'Amico RD, Addepalli P, Erkoyuncu JA. Is a Top Level Ontology Based Digital Twin the Solution to Human-Machine Interoperability? SSRN Electron J 2021. https://doi.org/10.2139/SSRN.3945058.

[17] D'Amico RD, Addepalli S, Erkoyuncu JA. Industrial Insights on Digital Twins in Manufacturing: Application Landscape, Current Practices, and Future Needs. *7*(3), 126 Big Data Cogn Comput 2023 2023;Vol. 7:*126*. https://doi.org/10.3390/BDCC7030126.

[18] D'Amico RD, Erkoyuncu JA, Addepalli S, Penver S. Cognitive digital twin: An approach to improve the maintenance management. CIRP J Manuf Sci Technol 2022;38:613–30. https://doi.org/10.1016/J.CIRPJ.2022.06.004.

[19] Demoly F, Matsokis A, Kiritsis D. A mereotopological product relationship description approach for assembly oriented design. Robot Comput-Integr Manuf 2012;28(6):681–93. https://doi.org/10.1016/J.RCIM.2012.03.003.

[20] Eirinakis P, Kalaboukas K, Lounis S, Mourtos I, Rozanec JM, Stojanovic N, Zois G. Enhancing Cognition for Digital Twins. *Proceedings - 2020*. IEEE Int Conf Eng, Technol Innov, ICE/ITMC 2020 2020. https://doi.org/10.1109/ICE/ITMC49519.2020.9198492.

[21] ElMaraghy H, ElMaraghy W. Adaptive Cognitive Manufacturing System (ACMS) – a new paradigm. Int J Prod Res 2022;60(24):7436–49. https://doi.org/10.1080/00207543.2022.2078248.

[22] Fernández F, Sánchez Á, Vélez JF, Moreno AB. Symbiotic Autonomous Systems with Consciousness Using Digital Twins. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11487*. LNCS 2019:23–32. https://doi.org/10.1007/978-3-030-19651-6_3/COVER.

[23] Forrest P. Mereotopology without mereology. J Philos Log 2010;39(3):229–54. https://doi.org/10.1007/S10992-010-9130-X/METRICS.

[24] Galton A. Discrete Mereotopology. Mereology Sci 2014;371:293–321. https://doi.org/10.1007/978-3-319-05356-1_11/COVER.

[25] Goddard C, Wierzbicka A. Semantics and cognition. Wiley Interdiscip Rev: Cogn Sci 2011;2(2):125–35. https://doi.org/10.1002/WCS.101.

[26] Graessler I, Poehler A. Integration of a digital twin as human representation in a scheduling procedure of a cyber-physical production system. *IEEE Int Conf Ind Eng Eng Manag*, *2017-Dec* 2017:289–93. https://doi.org/10.1109/IEEM.2017.8289898.

[27] Gruber TR. Toward principles for the design of ontologies used for knowledge sharing. Int J Hum-Comput Stud 1995;43:907–28.

[28] Grüninger, M., & Fox, M.S. (1995). The Role of Competency Questions in Enterprise Engineering. Benchmarking — Theory and Practice, 22–31. https://doi.org/10.1007/978–0-387–34847-6_3.

[29] Guarino, N., Oberle, D., & Staab, S.. (2009). What is an Ontology? In Handbook on Ontologies, International Handbooks on Information Systems. Springer Berlin Heidelberg. https://doi.org/10.1007/978–3-540–92673-3.

[30] Hahmann, T., & Grüninger, M. (2012). Region-Based Theories of Space: Mereotopology and Beyond. Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions, 1–62. https://doi.org/10.4018/978–1-61692–868-1.CH001.

[31] Hsiao, Y.E. (2003). Semantics and Cognition: An Introduction. In LANGUAGE AND LINGUISTICS (Vol. 4).

[32] Hu X, Arista R, Lentes J, Lu J, Zheng X, Sorvari J, Ubis F, Kiritsis D. Ontology-centric industrial requirements validation for aircraft assembly system design. IFAC-Pap 2022;55(10):3016–21. https://doi.org/10.1016/J.IFACOL.2022.10.191.

[33] Intizar Ali M, Patel P, Breslin JG, Harik R, Sheth A. Cognitive Digital Twins for Smart Manufacturing. IEEE Intell Syst 2021;36(2):96–100. https://doi.org/10.1109/MIS.2021.3062437.

[34] Kalaboukas K, Rožanec J, Košmerlj A, Kiritsis D, Arampatzis G. Implementation of cognitive digital twins in connected and agile supply networks-an operational model. Appl Sci (Switz) 2021;11(9). https://doi.org/10.3390/app11094103.

[35] Karray MH, Otte N, Kiritsis D, Ameri F, Kulvatunyou B, Will C, Arista R, Rai R, Smith B. The Industrial Ontologies Foundry (IOF) perspectives. *Industrial Ontology Foundry (IOF) - Achieving Data Interoperability Workshop*. Int Conf Inter Enterp Syst Appl 2021.

[36] Keet CM, Fernández-Reyes FC, Morales-González A. Representing mereotopological relations in OWL ontologies with OntoPartS. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7295*. LNCS 2012:240–54. https://doi.org/10.1007/978-3-642-30284-8_23/COVER.

[37] Lu J, Yang Z, Zheng X, Wang J, Kiritsis D. Exploring the concept of Cognitive Digital Twin from model-based systems engineering perspective. Int J Adv Manuf Technol 2022;121(9–10):5835–54. https://doi.org/10.1007/S00170-022-09610-5.

[38] Lu J, Zheng X, Gharaei A, Kalaboukas K, Kiritsis D. Cognitive Twins for Supporting Decision-Makings of Internet of Things Systems. Lect Notes Mech Eng 2020: 105–15.

[39] Merrell, E.C., Limbaugh, D.G., Koch Ac, P.M., & Smith, B. (2022). Capabilities. https://doi.org/https://philarchive.org/rec/MERC-14.

[40] Meyers B, Van Noten J, Lietaert P, Tielemans B, Hristov H, Maes D, Gadeyne K. Knowledge Graphs in Digital Twins for Manufacturing - Lessons Learned from an Industrial Case at Atlas Copco Airpower. IFAC-Pap 2022;55(10):13–8. https://doi.org/10.1016/J.IFACOL.2022.09.361.

[41] Mohammadi N, Vimal A, Taylor JE. Knowledge Discovery in Smart City Digital Twins. *2020-January*, 1656–1664 Proc Annu Hawaii Int Conf Syst Sci 2020. https://doi.org/10.24251/HICSS.2020.204.

[42] Mortlock T, Muthirayan D, Yu SY, Khargonekar PP, Abdullah Al Faruque M. Graph Learning for Cognitive Digital Twins in Manufacturing Systems. IEEE Trans Emerg Top Comput 2022;10(1):34–45. https://doi.org/10.1109/TETC.2021.3132251.

[43] Poveda-Villalón M, Fernández-Izquierdo A, Fernández-López M, García-Castro R. LOT: An industrial oriented ontology engineering framework. Eng Appl Artif Intell 2022;111:104755. https://doi.org/10.1016/J.ENGAPPAI.2022.104755.

[44] Raubal M, Adams B. The Semantic Web needs more cognition. Semant Web 2010;1(1–2):69–74. https://doi.org/10.3233/SW-2010-0002.

[45] Robinson, I., Webber, J., & Eifrem, E. (2015). Graph databases: New Opportunities for Connected Data. O'Reilly.

[46] Rožanec JM, Lu J, Rupnik J, Škrjanc M, Mladenić D, Fortuna B, Zheng X, Kiritsis D. Actionable cognitive twins for decision making in manufacturing. Int J Prod Res 2021;60(2):452–78. https://doi.org/10.1080/00207543.2021.2002967.

[47] Salustri FA. Mereotopology for product modelling. A new framework for product modelling based on logic. J Des Res 2002;2(1):0. https://doi.org/10.1504/jdr.2002.009816.

[48] Sarkar A, Šormaz D, Koonce D, Farah S. Developing a Resource-Based Manufacturing Process Capability Ontology. *1407 CCIS*, 293–306 Commun Comput Inf Sci 2020. https://doi.org/10.1007/978-3-030-76307-7_22.

[49] Smith B. Mereotopology: A theory of parts and boundaries. Data Knowl Eng 1996;20(3):287–303. https://doi.org/10.1016/S0169-023X(96)00015-8.

[50] Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg LJ, Eilbeck K, Ireland A, Mungall CJ, Leontis N, Rocca-Serra P, Ruttenberg A, Sansone SA, Scheuermann RH, Shah N, Whetzel PL, Lewis S. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol* 2007;Vol. 25(Issue 11):1251–5. https://doi.org/10.1038/nbt1346.

[51] Solso, R.L., MacLin, O.H., & MacLin, M. Kimberly (2008). Cognitive psychology. Pearson/Allyn and Bacon.

[52] Steiner CM, Albert D. Validating domain ontologies: A methodology exemplified for concept maps. Cogent Educ 2017;4(1). https://doi.org/10.1080/2331186X.2016.1263006.

[53] Suh NP. Axiomatic Design Theory for Systems. Res Eng Des 1998:189–209.

[54] Tao F, Qi Q. Make more digital twins. *7775*, *573*(7775) Nature 2021 2019;573:490–1. https://doi.org/10.1038/d41586-019-02849-1.

[55] Tao F, Zhang H, Liu A, Nee AYC. Digital Twin in Industry: State-of-the-Art. IEEE Trans Ind Inform 2019;15(4):2405–15. https://doi.org/10.1109/TII.2018.2873186.

[56] Treleaven P, Smietanka M, Pithadia H. Federated Learning: The Pioneering Distributed Machine Learning and Privacy-Preserving Data Technology. Computer 2022;55(4):20–9. https://doi.org/10.1109/MC.2021.3052390.

[57] Varzi AC. Parts, wholes, and part-whole relations: The prospects of mereotopology. Data Knowl Eng 1996;20(3):259–86. https://doi.org/10.1016/S0169-023X(96)00017-1.

[58] Varzi, A.C. (1998). Basic Problems of Mereotopology. In Formal Ontology in Information Systems. IOS Press, 29–38.

[59] Yitmen I, Alizadehsalehi S, Akıner İ, Akıner ME. An adapted model of cognitive digital twins for building lifecycle management. Appl Sci (Switz) 2021;11(9). https://doi.org/10.3390/app11094276.

[60] Zheng X, Hu X, Arista R, Lu J, Sorvari J, Lentes J, Ubis F, Kiritsis D. A semantic-driven tradespace framework to accelerate aircraft manufacturing system design. J Intell Manuf 2022. https://doi.org/10.1007/s10845-022-02043-7.

[61] Zheng X, Hu X, Arista R, Lu J, Sorvari J, Lentes J, Ubis F, Kiritsis D. A semantic-driven tradespace framework to accelerate aircraft manufacturing system design. J Intell Manuf 2022;1:24. https://doi.org/10.1007/S10845-022-02043-7/FIGURES/19.

[62] Zheng X, Lu J, Kiritsis D. The emergence of cognitive digital twin: vision, challenges and opportunities. Int J Prod Res 2022;60(24):7610–32. https://doi.org/10.1080/00207543.2021.2014591.



https://dspace.org/

Knowledge transfer in Digital Twins: the methodology to develop Cognitive Digital Twins

D'Amico, Rosario Davide; Sarkar, Arkopaul; Karray, Mohamed Hedi; Addepalli, Sri; Erkoyuncu, John Ahmet

2024-09-01
Elsevier
https://doi.org/10.1016/j.cirpj.2024.06.007
D'Amico RD, Sarkar A, Karray MH, et al., (2024) Knowledge transfer in Digital Twins: the methodology to develop Cognitive Digital Twins. CIRP Journal of Manufacturing Science and Technology, Volume 52, September 2024, pp.366-385

Downloaded from a DSpace repository. https://dspace.org/