Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-023-02213-1

An ontology for defining and characterizing demonstration environments

Wei Nie[1] · Katharina De Vita[1,2] · Tariq Masood[1,3]

Received: 13 October 2022 / Accepted: 5 September 2023

Abstract

Demonstration Environments (DEs) are essential tools for testing and demonstrating new technologies, products, and services, and reducing uncertainties and risks in the innovation process. However, the terminology used to describe these environments is inconsistent, leading to heterogeneity in defining and characterizing them. This makes it difficult to establish a universal understanding of DEs and to differentiate between the different types of DEs, including testbeds, pilot-plants, and living labs. Moreover, existing literature lacks a holistic view of DEs, with studies focusing on specific types of DEs and not offering an integrated perspective on their characteristics and applicability in different contexts. This study proposes an ontology for knowledge representation related to DEs to address this gap. Using an ontology learning approach analyzing 3621 peer-reviewed journal articles, we develop a standardized framework for defining and characterizing DEs, providing a holistic view of these environments. The resulting ontology allows innovation managers and practitioners to select appropriate DEs for achieving their innovation goals, based on the characteristics and capabilities of the specific type of DE. The contributions of this study are significant in advancing the understanding and application of DEs in innovation processes. The proposed ontology provides a standardized approach for defining and characterizing DEs, reducing inconsistencies in terminology and establishing a common understanding of these environments. This enables innovation managers and practitioners to select appropriate DEs for their specific innovation goals, facilitating more efficient and effective innovation processes. Overall, this study provides a valuable resource for researchers, practitioners, and policymakers interested in the effective use of DEs in innovation.

Keywords Demonstration environments · Ontology · Testbed · Living lab · Pilot plant · Innovation management

Introduction

Confronted with increasingly complex challenges in a fast-changing environment, organizations have to continuously innovate to remain competitive. During the innovation processes, confidence is built through the use of Demonstration Environments (DEs) that offer space for prototyping, testing, and verification of new products, services and systems. DEs, including testbeds, pilot-plants, and living labs, play a crucial role in ensuring that innovative outcomes can effectively fulfill their intended functions (Högman & Johannesson, 2013). These environments are supported and utilized by a diverse set of stakeholders, including but not limited to, universities, government organizations, and private businesses. Accurate differentiation of DEs is essential for selecting and utilizing the most appropriate environment based on their distinct capabilities and characteristics, to achieve specific

✉ Katharina De Vita
K.DeVita@greenwich.ac.uk

Wei Nie
wn230@cam.ac.uk

Tariq Masood
tariq.masood@strath.ac.uk

[1] Department of Engineering, Institute for Manufacturing, University of Cambridge, Cambridge CB3 0FS, UK

[2] Greenwich Business School, Old Royal Naval College, University of Greenwich, Park Row, Greenwich SE10 9LS, UK

[3] Department of Design, Manufacturing and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK

Published online: 10 October 2023

demonstration outcomes. These outcomes may include cost reduction, improved user experience, and more.

Despite the significance of DEs in practice as well as the growing scholarly literature, there remain notable gaps and challenges in the existing studies pertaining to their differentiation. The need for further research in this area can be attributed to two key factors: (1) the heterogeneity of DE domain knowledge, and (2) the lack of a holistic view of different types of DEs. Consequently, this paper seeks to address the identified gaps by offering a differentiated understanding of DEs, fostering increased knowledge sharing across application fields, and resolving the existing heterogeneity in terminology.

Aligned with the works of Costa et al. (2016), Järvenpää et al. (2019), Montero Jiménez et al. (2021), Chen et al. (2021), the literature gaps can be closed by creating a DE ontology that explicitly structures the entities of a domain and their relationships. This ontology would explicitly structure the domain entities and their relationships, facilitating a comprehensive characterization of DEs that encompasses various types, distinctive features, applications, and involved components, such as stakeholders and facilities (Turchet et al., 2020). Furthermore, the ontology would enable reasoning based on semantic rules, a crucial capability not yet explored in any previous study related to DEs. Not only has no previous study created an ontology for DEs, but this holds for both a range of different DEs and specific DE types, such as testbeds.

This paper aims to address the research question, "How can Demonstration Environments be defined and characterized?" To achieve this, we propose the Demonstration Environments Ontology (DEO) as a representation of domain knowledge related to DEs. The DEO is publicly available at http://purl.archive.org/demonstration-environments-ontology/documentation. Following the METHONTOLOGY methodological framework (Fernández-López et al., 1997), a systematic collection of peer-reviewed English journal articles that specifically focused on Demonstration Environments (DEs) was carried out to develop the ontology. To establish a semantic framework, word embeddings and knowledge extraction through four algorithms to extract domain words, identify concepts and properties, build ontology hierarchy and identify concept-property relations were utilized in the analysis (Mahmoud et al., 2018; Albukhitan & Helmy, 2016). In line with Montero Jiménez et al. (2021) we then manually populated the ontology in Protégé. Indeed, a key contribution of this research is the provision of a systematic framework, realized through the development of the ontology, which fosters a common understanding of DEs and standardizes terminologies used in both literature and practice.

Following a comprehensive review of existing innovation-related ontologies ("Literature review") and "Methodology" sections of this paper delves into the ontology development methodology and deep-learning steps employed. "Demonstration environments ontology" section presents the main components and structures of the DEO, establishing a robust and standardized framework for understanding DEs. In "Application of DEO" section, the applicability of the DEO is effectively demonstrated through the examination of two use cases. The key findings are discussed in "Discussion" section, affirming the fulfilment of the ontology requirements. The paper concludes with an outline of future research directions in "Conclusion" section.

Literature review

Demonstration environments

DEs can be used in different contexts and for diverse purposes in various stages of the innovation process (Ballon et al., 2005). For example, testbeds are built in an isolated laboratory environment and are commonly seen in the early stage of design and development such as proof-of-concept evaluations (Papadopoulos et al., 2017). Pilot-plants mirror real-life manufacturing plants and they are usually used before commercialization to demonstrate the feasibility of technologies (Hellsmark et al., 2016). Living labs closely resemble the context of the product or service in the real-life environment and involve users as co-creators of innovation outcomes (Almirall et al., 2012). However, current literature lacks a holistic view of different types of DEs. Numerous studies on DEs such as testbeds (Abuarqoub et al., 2012; Kim et al., 2020), pilot-plants (Deiana et al., 2017; Carrera et al., 2022) and living labs (Leminen et al., 2015; Voytenko et al., 2016; De Vita & De Vita, 2021) have shown how these DEs can be used in innovation processes realizing different objectives. While these studies add important insights about a particular DE type, they fail to depict the diverse landscape of DEs and do not offer an integrated perspective on the characteristics and applicability of such environments in different application contexts. Only a few studies have provided holistic insights into the diverse landscape of DEs. The first effort towards the discussion and differentiation of different DEs is reported in Ballon et al. (2005). The authors characterize and benchmark five innovation platforms for broadband innovation, including the prototyping platform, testbed, field trial, living lab, market pilot and societal pilot. However, a limitation of the developed framework is that the authors adopted a general focus on those open and joint platforms only. More importantly, the developed framework is structured as a typology, this prevents the knowledge of DEs from being expanded and joining the ecosystem formed around the innovation. This is important for enriching the domain knowledge of DEs in the future. For example, integrating the domain knowledge of DEs with other innovation-related concepts for better innovation management. Similarly, Schuurman and Tõnurist (2017) carry out a comparison between innovation labs and living labs. They explore similarities and differences between the two DEs and propose a model for further integration of them. The clear limitation is that the study only considers two similar environments. Therefore, current efforts to provide a systematic overview of DEs are limited. This gap leads to the fact that different DEs are not defined in a way such that they can be compared and benchmarked against each other. This made the differentiation between DEs challenging since there is no universal standard for a systematic DE description, annotation and classification. Moreover, the diversity of actors involved in the DEs further adds to the complexity of understanding DEs and how they can be leveraged effectively.

Although recognizing the value of DEs in manufacturing, scanty literature has focused on differentiating DEs for better implementation and application of the DEs in manufacturing. Firstly, there is an inconsistency in the use of terminologies that represent different types of DEs, leading to the heterogeneity of DE domain knowledge. For example, both 'Living Lab' and 'Innovation Lab' have been used to describe DEs that are based on the concept of open innovation and co-creation involving end-users in the process of innovation (Bergvall-Kareborn & Stahlbrost, 2009; Leminen et al., 2012; Whicher & Crick, 2019; Zurbriggen & Lago, 2019; Bulkeley et al., 2019; Fecher et al., 2020; Greve et al., 2020, 2021). Nevertheless, the two terminologies are not always used consistently in extant literature. Some researchers believe the two terms represent different environments. Schuurman and Tõnurist (2017) argue that living labs and innovation labs are two distinctly different environments. Osorio et al. (2019) also regard the two as different environments. However, other researchers treat innovation labs and living labs as interchangeable terms (Criado et al., 2021). Therefore, current studies on DEs do not share a common understanding of the use of terminologies to represent different DEs. This inconsistency in the use of terminologies indicates an information heterogeneity in defining and characterizing DEs. Such heterogeneity makes it challenging to establish a universal understanding to differentiate DEs in DE-related studies.

Ontology

Ontology is an explicit specification of a conceptualization which refers to an abstract model of some phenomenon that is formed by relevant concepts (Studer et al., 1998). It describes concepts and relationships among them, it can be defined as a 4-tuple: $\sigma =< C, R, I, A >$,where C is a set of classes (or concepts), R is a set of taxonomic and non-taxonomic relations, I is a set of individuals and A is a set of axioms (Gruber, 1993). Ontology-based semantic technologies are about the activities, theories and principles governing the domain, they can assist in managing, standardizing and integrating heterogeneous information (Fernández-López et al., 1997; El Bassiti & Ajhoun, 2014). In the following parts, we reviewed (1) ontology engineering and its methodologies; (2) ontology learning (OL); and (3) ontology validation approaches.

Ontology engineering methodologies

Ontology engineering investigates a series of principles, methods and tools for activities like knowledge representation, ontology design, standardization, reuse and knowledge sharing (Mizoguchi & Ikeda, 1997). An ontology engineering methodology provides a set of guidelines and activities for ontology development. Several ontology engineering methodologies have been proposed to date. While some methodologies for developing ontologies were proposed at the outset, others emerged as a result of the experience and insights gained during the development of ontologies for various projects.

Based on the experience of developing an ontology in the domain of chemicals, the METHONTOLOGY methodological framework was proposed for developing an ontology for a domain from scratch (Fernández-López et al., 1997). This framework introduces detailed development activities and techniques, including (1) the specification which identifies the audience, scope, use scenarios and requirements; (2) the conceptualization where the domain knowledge will be structured in a conceptual model in terms of the domain vocabulary; (3) the formalization of the ontology from the conceptual model; (4) the integration with existing ontologies; (5) the implementation of ontology with an appropriate environment; and (6) the maintenance of the implemented ontology. Orthogonal to the development activities, there are three support activities to be accomplished during the lifetime of the ontology: knowledge acquisition, documentation and evaluation of the ontology developed (Fernández-López et al., 1997). This methodological framework has been tested through developing ontologies for different domains like music (Turchet et al., 2020), biology (Ayadi et al., 2019b) and manufacturing (Catalano et al., 2009; Talhi et al., 2019).

Ontology Development 101 is another well-known domain ontology development methodology developed by Noy and McGuinness (2001). Also deriving from the experience, the authors provided a detailed example of constructing a wine ontology from scratch. It includes seven stages: (1) Determine the domain and scope of the ontology; (2) Consider reusing existing ontologies; (3) Enumerate important terms in the ontology; (4) Define the classes and class hierarchy; (5) Define the property of classes; (6) Define the facets of the slots; and (7) Create instances for classes. Even though the method includes detailed ontology construction steps, it does not consider the ontology development life cycle.

Besides, editing tools are required for ontology engineering, for example, the Protégé editor (Porzel and Malaka, 2000) which is one of the most prevalent ontology editors, NeOn Toolkit[1] which is suitable for heavy-weight projects, and SWOOP (Kalyanpur et al., 2006) which is a small and simple ontology editor.

Ontology learning

An ontology can be constructed manually by domain experts and knowledge engineers; or cooperatively where the construction system is supervised by the expert; or (semi-) automatically where there is limited human intervention in the process of construction (Al-Aswadi et al., 2019). Among these methods, a semi-automatic approach is preferred. It not only enables discovering ontological knowledge from large datasets at a faster pace and minimal search efforts but also alleviates the risks of having human-introduced biases and inconsistencies when acquiring the knowledge (Zhou, 2007).

Ontology construction through learning is a semi-automatic approach to creating a new ontology with minimal human effort. Ontology learning is essentially the process of extracting high-level concepts, relations, and sometimes axioms from information in order to construct an ontology. Many established techniques have contributed to the progress in ontology learning, for example, natural language processing, data mining and machine learning (Wong et al., 2012). In recent years, there has been a significant move towards the use of deep learning techniques for ontology construction (Al-Aswadi et al., 2019). Albukhitan and Helmy (2016) introduce a framework that used word embedding models such as Continuous Bag of Words (CBOW) and Skip-gram (SKIP-G) (Mikolov et al., 2013) to extract the taxonomic relation extraction from Arabic text data. Similarly, Mahmoud et al. (2018) proposes an OL methodology that combines word embeddings using Word2Vec (Mikolov et al., 2013) with Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) (Zhang et al., 1996) algorithm for text big data extraction.

Ontology validation approaches

Ontology validation is important for checking ontology quality. Ayadi et al. (2019b) summarize and evaluate common validation approaches:

1. **Task-based approach**: This approach applies ontology to certain tasks and evaluates performance (Porzel & Malaka, 2004). However, it does not assess the ontology structure and tends to ignore the shortcomings in ontology conceptualization.
2. **Automated consistency checking**: This approach uses a Description Logic reasoner like HermiT to check the internal consistency of an ontology, for example, if the ontology contains any contradictory information (Turhan, 2011). However, this approach ignores the background knowledge of ontology.
3. **Gold standard checking**: This approach compares the constructed ontology with a benchmark ontology by measuring the conceptual and lexical similarities (Maedche & Staab, 2002). One limitation is that there might be a lack of domain ontology for benchmarking.
4. **Criteria-based approach**: This approach evaluates the ontology with a set of predefined criteria (Table 1) and it combines both quantitative and qualitative methods of validation (Vrandečić 2009). Nevertheless, some of the criteria validation can be highly dependent on subjective judgement.
5. **Data-driven validation**: This approach compares the ontology with the domain data to be covered by the ontology, but the correctness and clarity cannot be verified with this approach (Brewster et al., 2004).
6. **Expert validation**: This approach evaluates the ontology against a set of predefined criteria, requirements and standards by expert judgements, but the validation result is subjective and depends on the capability of the experts (Lozano-Tello & Gomez-Perez, 2004).

   Indeed, the ontology validation approach should be selected based on the validation purposes, ontology application of which an ontology is to be used, and aspects that an ontology is to be validated or evaluated (Walisadeera et al., 2016).

Innovation-related ontologies

Over the last decades, ontologies gained popularity as a supporting tool for innovation management modelling. The scope of these innovation-related ontologies is wide-ranging, varying from a focused area on innovation idea selection (Bullinger, 2008), to the entire innovation management life-cycle in general (Westerski et al., 2010). These ontologies also differ in their intended purposes and level of abstraction. For example, the OntoGate Ontology (Bullinger, 2008) only covers the early stage of innovation, whereas GI2MO (Westerski et al., 2010) includes the entire innovation management life-cycle. They are also designed for different usages: OntoGate Ontology was developed for idea assessment and selection, while Ontology for Innovation (Greenly, 2012) focuses on ensuring the needs of the idea have been met. Nevertheless, to our knowledge, there is no common

[1] http://neon-toolkit.org/wiki/Main_Page.html.

Table 1 Ontology validation criteria

Criteria-based validation criteria	Definitions (Vrandečić, 2009; Raad & Cruz, 2015)
Accuracy	If the definitions and descriptions in the ontology comply with the domain knowledge
Adaptability	Measures the efficiency of a new tool or unexpected situation can use the ontology
Clarity	If there is any unambiguous description in definitions of concepts and relations
Completeness	Whether the constructed ontology can meet the research objectives and answer the research questions, and whether the DE domain is appropriately represented
Computational efficiency	The working speed of the reasoner
Conciseness	If there is any redundant and irrelevant term
Consistency	If there is any contradictory axiom, and if the logical axioms are satisfiable and consistent

ontology created explicitly for DEs, neither individual type nor as a whole.

Ning et al. (2006) introduce the Semantic Innovation Management system that includes an Iteams ontology to create a semantic web of innovation knowledge. The Iteams ontology aims to manage innovation information and to operate innovation processes across the extended enterprise seamlessly. It specifies five abstract classes: Goals, Actions, Results, Teams and Community. This ontology can be used to facilitate the distributed collection and development of ideas.

OntoGate ontology (Bullinger, 2008) is a formal domain heavyweight ontology for the early stage of innovation. This ontology was developed based on empirical research to assist idea assessment and selection. It covers three main aspects for the idea to be assessed: market, strategy and technology. Through introducing four key elements: Participant, Input, Gate and Output, the OntoGate ontology structures a company's understanding of the innovation process.

The Idea ontology (Riedl et al., 2011) is a lightweight application ontology that supports the innovation life-cycle management in open-innovation scenarios. It provides technical means to represent complex idea evaluations along with various concepts. Unlike the Iteams ontology (Ning et al., 2006), the Idea ontology takes a modular approach which allows it to be complemented by additional concepts like customized evaluation methods.

The Generic Idea and Innovation Management Ontology (Gi2MO) ontology (Westerski et al., 2010) is a domain ontology for idea management systems in innovation. It covers the basic idea management processes including idea generation, idea improvement, idea assessment, idea implementation and idea deployment. This ontology is designed to annotate and describe resources gathered inside idea management facilities.

Zhang et al. (2011) design a multi-layered ontology called the Science and Technology Innovation Concept Knowledge-Base (STICK) ontology to capture important properties of the innovation Eco-system over time. It is comprised of three layers: (1) the basic layer includes innovation concepts and products, which distinguishes the concept of innovation from the materialization of innovation; (2) the extended layer covers innovations, key players, and their dynamic interactions; and (3) an evidence layer contains mainly supporting documents. The STICK ontology focuses on innovations and the individual or organizational members of the innovation communities, and their relationships over time.

Unlike the aforementioned ontologies, Greenly (2012) proposes an ontology to facilitate the matching of needs and innovations. The key class *Innovation* is related to eight classes including the Development stage, Innovator, Embodiment, Improvement, Benefit, Disruption, Problem, and Need.

Zanni-Merk et al. (2013) present a formal ontology for the Theory of Resolution of Inventive Problems (TRIZ) which is a systematic methodology that helps guide the search for inventive solutions to understanding and solving challenging problems. The TRIZ ontology is developed for creativity assistance, it covers three sub-ontologies:(1) the sub-ontology of the TRIZ system model; (2) the substance-field models sub-ontology; and (3) the contradictions and parameters sub-ontology. The TRIZ ontology stresses more on the formulation of problems and guidance to the solution designs, rather than a later stage of the solution validation or demonstration.

Creative Workshop Management Ontology (CWMO) (Gabriel et al., 2019) is created for a different perspective of creativity management. It describes the area of knowledge of creative workshops to assist people involved in the creative workshop in realizing the problem-solving task. CWMO covers the concepts of evaluation strategy (e.g., "quick evaluation"), evaluation techniques (e.g., "multi-criteria evaluation") and evaluation criteria as part of the creative workshop process. However, the knowledge of evaluation is limited because this ontology is only for application in the context of the creative workshops to solve the problem collectively. Besides, there are also a few creativity management-related ontologies available, for example, the ontology for e-brainstorming (Lorenzo et al., 2021),

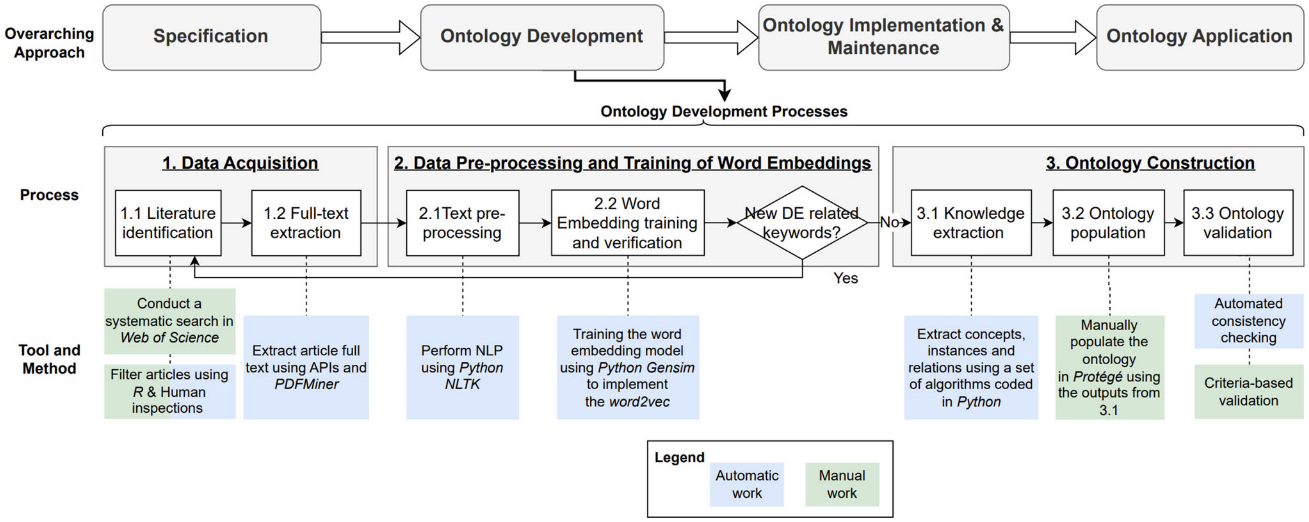

Overarching Approach
Specification
Ontology Development
Ontology Implementation & Maintenance
Ontology Application
Ontology Development Processes
Process
1. Data Acquisition
1.1 Literature identification
1.2 Full-text extraction
2. Data Pre-processing and Training of Word Embeddings
2.1Text pre-processing
2.2 Word Embedding training and verification
New DE related keywords?
No
Yes
3. Ontology Construction
3.1 Knowledge extraction
3.2 Ontology population
3.3 Ontology validation
Tool and Method
Conduct a systematic search in Web of Science
Filter articles using R & Human inspections
Extract article full text using APIs and PDFMiner
Perform NLP using Python NLTK
Training the word embedding model using Python Gensim to implement the word2vec
Extract concepts, instances and relations using a set of algorithms coded in Python
Manually populate the ontology in Protégé using the outputs from 3.1
Automated consistency checking
Criteria-based validation
Legend
Automatic work
Manual work

Fig. 1 Research overarching approach, process, tool and method

linked data source for expert finding (Stankovic, 2010), and engineering knowledge management (Brandt et al., 2008). However, none of them has stressed the demonstration, evaluation or validation of the creative subject.

As described, none of the ontology has explicitly mentioned the demonstration of the idea or innovation during the innovation development. In fact, all these ontologies centered on the innovation or idea itself. Therefore, a new ontology entitled "Demonstration Environment Ontology" (DEO) for the representation of this domain is proposed in this study.

Methodology

The main aim of this study is to provide a formal and semantic representation of DEs as a standardized framework. To accomplish this objective, this study aims to create a domain ontology for DEs. The methodology for designing and developing the DEO followed METHONTOLOGY methodological framework (Fernández-López et al., 1997). Figure 1 details the overarching approach following the METHONTOLOGY framework, ontology development processes and corresponding tools and methods for each process. We have documented our research data in the OSF platform which includes detailed steps, relevant codes and sample data, this is publicly available at https://osf.io/y9nem/.

Specification

Goals, target uses and audience

The main goal of this ontology is to provide a formal domain model for the DEs with definitions and characterizations. There are several intended uses of this ontology:

1. A knowledge representation of the DEs
2. A formal reference model that standardizes the terminologies related to DEs
3. A conceptual model capable of guiding target users in the demonstration process

The target audience of the DEO is represented by all actors and stakeholders that are involved in the innovation life-cycle, including research institutions, incubators & accelerators, startups & enterprises, private companies, governments and so on.

Requirements

The ontology should be designed in a way such that it meets the following requirements (Järvenpää et al., 2019):

1. Providing a structure to define and differentiate DEs by determining aspects that characterize different DEs
2. Ensuring the semantics are clearly defined such that there is no ambiguity in terminology and the meanings in the knowledge representation are clear; redundancy in relevant terminologies has to be captured by this ontology
3. Offering means for searching and querying information and supporting activities such as searching for and selecting suitable DEs and proposing feasible DE designs
4. Allowing the semantic model to be understood and interpreted by different users, applications and systems

Ontology development

Following Fig. 1, this section describes the detailed development processes of the DEO, including (1) Data acquisition;

Table 2 Search String related to DEs (Adapted from Greve & O'Sullivan (2019))

Search String
"test bed" OR "testbed*" OR "living lab*" OR "co-innovation hub" OR "coinnovation hub" OR "co-innovation centre" OR "coinnovation centre" OR "co-innovation facilit*" OR "coinnovation facilit*" OR "co-innovation space" OR "coinnovation space" OR "co-innovation environment" OR "coinnovation environment" OR "co-innovation lab*" OR "coinnovation lab*" OR "cocreation hub" OR "co-creation hub" OR "co-creation centre" OR "cocreation centre" OR "co-creation facilit*" OR "cocreation facilit*" OR "co-creation space" OR "cocreation space" OR "co-creation environment" OR "cocreation environment" OR "co-creation lab*" OR "cocreation lab*" OR "co-creation hub" OR "open innovation environment" OR "open innovation facility*" OR "open innovation hub" OR "open innovation centre*" OR "open innovation space" OR "open innovation lab*" OR "innovation lab*" OR "cooperation lab*" OR "innovation hub*" OR "innovation lab*" OR "pilot* demo*" OR "pilot facilit*" OR "pilot plant*" OR "demonstration plant*" OR "pilot space" OR "pilot lab*" OR "pilot environment" OR "pilot line" OR "scale-up facilit*" OR "scale up facilit*" OR "scale-up environment" OR "scale up environment" OR "scale-up space" OR "scale-up centre" OR "scale up centre" OR "scaleup space" OR "scale-up hub" OR "scale up hub" OR "scale-up lab*" OR "scale up lab*" OR "prototyp* plant*" OR "prototyp* platform*" OR "pilot-scale plant" OR "pilot scale plant" OR "demonstration-scale plant" OR "demonstration scale plant" OR "demonstrat* environment*" OR "demonstrat* facilit*" OR "demonstration cent*" **OR "sandbox*" OR "test bench" OR "test benches" OR "test-bench" OR "test bench*"**

Search terms highlighted in bold were added after the initial search

(2) Data pre-processing and training of word embeddings; and (3) Ontology construction.

Data acquisition

This study discovered relevant ontological knowledge from text data which are among the commonly used data sources for constructing the ontology (Faria et al., 2014; Arguello Casteleiro et al., 2017; Mahmoud et al., 2018; Ayadi et al., 2019b; Al-Aswadi et al., 2019; Deepa & Vigneshwari, 2022). To be more specific, this study used peer-reviewed journal articles for constructing the ontology where data diversity, quality and statistical significance can be achieved (Mahmoud et al., 2018; Ayadi et al., 2019a; Kumar & Starly, 2021).

To prepare the text corpus for ontology construction, a search string that is relevant to DEs (Table 2) was used to locate peer-reviewed English journal articles on the topic of DEs in the Web of Science (WoS) database. Due to its wide coverage and relevance for this study, WoS was used to retrieve articles. As opposed to, for example, Scopus, WoS indexes also the Technology Innovation Management (TIM) Review, which is the journal with the largest number of special issues and articles on living labs to date (Westerlund et al., 2018). Moreover, WoS has already been employed to analyse similar domains, such as open innovation (Dahlander & Gann, 2010), technology business incubation (Mian et al., 2016), living labs (Greve et al., 2021).

The initial search was conducted on 7th June 2020, which produced a total of 4922 unique articles. Obtained articles were screened with a pre-formulated retrieval strategy. The exclusion criteria comprised: (1) search keywords were included as false-positive words. For example, Evans and Karvonen (2014) use 'living lab' as a metaphor to describe a lab with living components; and (2) the research does not discuss DEs. For example, Clark et al. (2018) use 'testbed' as an official name of a program (Hazardous Weather Testbed Spring Forecasting Experiment) without discussing the testbed itself. Detailed screening steps can be found in Appendix A, we also detailed the list of false positive words in Table 4. Following the exclusion of irrelevant results, a total of 3621 articles were finalized as the text corpus for further extraction.

To build up and formulate the text corpus for performing the word embedding, the full text of these articles was extracted using tools such as Elsevier's API, PDFMiner[2] (20191125 version) to obtain the article in plain text format and store them as separate.txt documents for future analysis.

Data pre-processing and training of word embeddings

This study exploited word embeddings to automate the process of extracting ontology components from relevant journal articles.

This process consists of three steps: (1) pre-processing the text acquired; and (2) word embedding training and verification. These steps are supported by NLP techniques and algorithms pre-trained in Python.

(1) Pre-processing the text acquired

The extracted texts are unstructured and need to be transformed into an understandable format before applying the word embedding algorithm. This step was performed with the support of Python's NLTK[3] package, a suite of libraries for NLP. Words in the text were standardized with pre-established norms and were in accordance with the naming rules of WordNet. For example, since the same DE type can have different spellings (e.g. 'testbed' and 'test bed'), they were pre-coded to be consistent to avoid information dilution due to spelling differences. Two tasks were performed in this step:

[2] https://pypi.org/project/pdfminer/.

[3] https://www.nltk.org/.

1. **Tokenization** The long text strings in each document are segmented into words. This task relies on the word tokenizer from NLTK.
2. **Normalization** This process transforms texts into a single canonical form and consists of a series of tasks such as (1) removing punctuations; (2) removing non-English words with the support of WordNet[4] which is a lexical database of English words; (3) removing general stop words like ‘the’ and ‘a’; and (4) recognising words’ grammatical form through part-of-speech (POS) tagging, then removing their inflectional endings and returning the base or dictionary format of the word accordingly through lemmatization (e.g. from ‘demonstrating’ to ‘demonstrate’) (Manning et al., 2008).

(2) Word embedding training and verification

Pre-processed texts were used to develop a language model based on Word2Vec with the support of Python Gensim package.[5] Word2Vec is a neural network model that converts each word in the corpus to a structured vector (Mikolov et al., 2013). It allows words with similar meanings to obtain a similar representation through a word embedding approach (Ni et al., 2022). By capturing subtle semantic relations between words, these models enable identifying concepts and relations from texts. For example, terms related to ‘testbed’ can be found by measuring cosine similarities between the vector ‘testbed’ and other vectors.

CBOW and SKIP-G are two main methods to implement word embeddings using Word2Vec (Mikolov et al., 2013). This study used the SKIP-G model *(min count = 30, window size = 6)*. The objective is to identify candidate ontological components. Therefore, pairs of similar words determined by the model are the focus rather than achieving a high accuracy as it does not result in major differences in the similar words identified.

To verify the trained model, a simple set of keywords was derived for preliminary verification. These keywords were obtained by reviewing full articles during a structured literature review. To further evaluate the model performance in identifying similar keywords, these keywords were grouped such that they were expected to appear together when applying the similarity function:

S_1 = [testbed, modular, heterogeneous, fidelity, federation]

S_2 = [pilot-plant, pilot line, pilot-scale plant, pilot-facility, scale, demonstration, experiment]

S_3 = [living lab, innovation lab, innovation, creative, stakeholder, user, collaboration, participation]

Through checking similar words of the listed words, potential keywords that are not included in the search string may appear, therefore requiring a search string refinement.

In the experiment, two new terms, ‘sandbox’ and ‘testbench’ were identified and added to the search string (Table 2) for conducting a new search to enrich the text corpus with articles related to ‘sandbox’ and ‘testbench’. The new search was conducted on 7th July 2020, where 586 articles were added. The new text corpus consists of 4045 articles (total word count: 10,071,175; unique words: 36,523).

Ontology construction

(1) Knowledge extraction

With the language model trained, several algorithms were introduced to identify ontological components based on model outputs. These algorithms were adapted from similar works done by Mahmoud et al. (2018) and Albukhitan and Helmy (2016). This includes extracting domain words, identifying candidate concepts and properties from the domain words, building the ontology hierarchy and identifying concept-property relations. In this work, the BIRCH algorithm was used because it provides a scalable and efficient clustering algorithm for a large volume of text data.

Algorithm 1 Algorithm for Domain Words Extraction.

```
Input: Seed terms $Seeds$, language model $Model$, Threshold $T$
Output: Domain word list $DomainWordList$
$DomainWordList = Seeds$
for each $seed \in Seeds$ do
    $SimilarTerms$ = findSimilarTerms($seed, Model$)[a]
    for each $term \in SimilarTerms$ do
        if Similarity($seed, term$) $> T$ then[b]
            $DomainWordList = DomainWordList \cup term$
        end if
    end for
end for
$DomainWordList$ = uniqueWords($DomainWordList$)
return $DomainWordList$
```

[a] findSimilarTerms(): find similar terms of the input seed term.

[b] Similarity(): get cosine similarity value between two input terms.

- *Domain words extraction* The first step is to extract domain words (Algorithm 1). Initial seed terms and the pre-trained language model are taken as inputs. Words related to the domain are extracted iteratively based on their relevance to the initial seed terms, represented as $findSimilarTerms(seed, Model)$ in the algorithm. In this work, we chose to measure the relevance between two words with the cosine similarity rather than the Euclidean distance since the magnitude of the word vectors (i.e., word counts) does matter, which follows the method as the previous studies, such as Kulmanov et al.

[4] https://wordnet.princeton.edu/.

[5] https://radimrehurek.com/gensim/models/word2vec.html.

Table 3 Seed terms used to extract domain words

Type	Terms
Keywords related to DE	Testbed, pilot-plant, pilot line, pilot-scale plant, pilot facility, living lab, innovation lab, prototype plant, prototyping platform, sandbox, test-bench
Keywords related to characteristics	Feasibility, vulnerability, viability, virtual, modular, heterogeneous, fidelity, creative
Others	Innovation, stakeholder, user, prototype, collaboration, participation, experiment, demonstration, federation, scale, hardware, software, architecture, tool, facility

Table 4 List of false-positive words

Search keywords	False-positive words
"living lab"	"living labour", "living labor", "live labour", "live labor", "living-labor", "living-labour"
"innovation lab"	"innovation labour", "innovation labor", "innovation-labour", "innovation-labor"
"scale-up environment/facility"	"scaleup environmental", "scale-up environmental", "scale up environmental", "scaleup facilitate", "scale up facilitate", "scale-up facilitate"
"pilot demonstration"	"pilot demonstrate", "pilots demonstrate", "pilot demonstrates", "pilot demonstrates"
"demonstration environment/facility"	"demonstrate environmental", "demonstrates environmental", "demonstrate facilitat", "demonstrates facilitat"

(2020) and He et al. (2006). This was realized through the *Word2Vec.most_similar* method in Python Gensim.[6] A similarity threshold value of 0.5 is employed to determine the inclusion of words that are deemed to be relevant to the domain. This value was set in order to obtain sufficient candidate terms initially. This threshold value can be adjusted according to the average and standard deviation of all the similar terms. Eventually, this algorithm returns a domain word list as the output for the next step. In this study, seed terms used (Table 3) were derived from the search string and full article reviews conducted in a structured literature review. Following Algorithm 1, 2256 unique words were identified to be relevant domain words.

- *Candidate concept and properties identification* The second step is to identify two ontology components: concepts and properties from the domain word list. Algorithm 2 illustrates how domain words are organized to be concepts or properties. This step takes a POS tagger and the domain word list as the input. A POS tagger from the NLTK module will tag each word as a noun, verb, adjective, etc. Noun or adjective words are considered to be concepts, whereas verb words are taken to be properties. In this study, tagging the obtained words from the previous step gave 145 candidate properties and 2033 candidate concepts.
- *Ontology hierarchy building* To build the ontology hierarchy, Balanced Iterative Reducing and Clustering (BIRCH) algorithm[7] (Zhang et al., 1996), which is a hierarchical clustering algorithm that exhibits better time performance for large datasets, was applied. BIRCH has two phases: (1) building cluster feature (CF) tree for initial clustering; and (2) global clustering phase that applies an existing clustering algorithm to further refine

[6] https://radimrehurek.com/gensim/models/word2vec.html.

[7] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html.

the initial clustering (Zhang et al., 1996). This step is supported by Python's Scikit-learn[8] (version 0.23.1) module. In the hierarchy-building process, the vector representation for each identified concept was obtained from the language model and input to the BIRCH function for initial clustering (Algorithm 3). Agglomerative clustering[9] was applied in the second phase for each initial cluster to obtain an ontology hierarchy. This clustering process generated nine CF-tree initially. Applying the agglomerative clustering algorithm produced different word clusters, which provides indications of concept groups because each group can relate to different themes (e.g., word clusters related to testbeds). We also plotted the dendrograms to obtain the word hierarchy that helps to determine the ontology structure.

Algorithm 2 Candidate Concept and Properties Identification.

```
Input: Domain word list $DomainWordList$, $POSTagger$
Output: Concept list $ConceptList$, Property list $PropertyList$
$ConceptList = []$, $PropertyList = []$
for each $term \in DomainWordList$ do
    $tag$ = wordTag($term, POSTagger$)$^a$
    if $tag$ is noun $\vee$ $tag$ is adjective then
        $ConceptList = ConceptList \cup term$
    else
        if $tag$ is verb then
            $PropertyList = PropertyList \cup term$
        end if
    end if
end for
return $ConceptList, PropertyList$
```

^{a}wordTag(): tag the input term using a POS Tagger.

Algorithm 3 Ontology Hierarchy Building.

```
Input: Concept list $ConceptList$, language model $Model$
Output: $OntologyHierarchy$
$ConceptVectors = []$
for each $concept \in ConceptList$ do
    $conceptVector$ = Model($concept$)$^a$
    $ConceptVectors = ConceptVectors \cup conceptVector$
end for
$CfTree$ = birch($ConceptVectors$)
for each $cluster \in CfTree$ do
    $OntologyHierarchy$ = AgglomerativeClustering ($cluster$)
end for
return $OntologyHierarchy$
```

^{a}Model(): obtain the word vectors which will be used as inputs in BIRCH.

- *Concept-property relation identification* Each concept has properties or attributes that describe it in an ontology. Therefore, the final step matches candidate properties with most related candidate concepts to provide inferences for generating axioms. This relation is determined by measuring the similarity between word vectors of concepts and properties from the pre-trained language model (Algorithm 4).

Algorithm 4 Concept-Property Relation Identification.

```
Input: Concept list $ConceptList$, Property list $PropertyList$, language model $Model$
Output: $property - concept$ pairs
$SimilarityValue = []$
for each $prop \in PropertyList$ do
    $SimilarityValue = SimilarityValue \cup$ <Similarity($prop, con$), $prop, con$ >,
end for
Sort $SimilarityValue$ by descending order of Similarity($prop, con$)
return first 5 tuples $\in SimilarityValue$
```

(2) Ontology population

Ontologies are often represented with Ontology Web Language (OWL) following the World Wide Web Consortium (W3C) standards (Lu et al., 2019; Montero Jiménez et al., 2021).In this study, OWL was chosen to model the DEO. Based on the hierarchical structure and concept-property relations obtained in the previous step, we manually populated the ontology in an editor. The ontology was coded using the open-source Protégé editor (version 5.5.0) (Porzel and Malaka, 2000). To respect the recommendation about reusing concepts from other ontologies, we used the Linked Open Vocabulary[10] search engine and enriched the DEO with concepts from FOAF and SKOS. The full URI of the below-mentioned namespace prefixes can be seen in Table 5.

(3) Ontology validation

The constructed ontology aims to define and conceptualize DEs. Therefore, validation approaches that aim to assess the correctness and knowledge coverage of the developed ontology were deemed relevant and appropriate for this research. Therefore, both automated consistency checking and criteria-based validation approaches were selected due to their relevance and availability in this case. Task-based and data-driven approaches are not as relevant, moreover, this ontology was built from scratch since there is no benchmark ontology available. Similarly, expert validation is not

[8] https://scikit-learn.org/.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html. We used the default value of the "metric" parameter, that is "ward", for calculating the linkage between instances.

[10] http://lov.okfn.org/dataset/lov/.

applicable at the point when this research was conducted. These approaches were thus not considered.

The validation started with automated consistency checking. This is conducted to ensure that the logical axioms are consistent and there is no contradictory fact. To achieve this, the description logic reasoner HermiT[11] (version 1.4.3.456), a reasoning plugin in Protégé 5 was used. HermiT determines the consistency of the ontology and identifies subsumption relationships between classes (Shearer et al., 2008). The reasoner was used to debug the ontology along the ontology construction. The final results made by the reasoner revealed that no contradicting information was found in the DEO ontology. We also applied the OntOlogy Pitfall Scanner[12] during the ontology development, which provided great evidence of the design error-free of the DEO.

The criteria-based validation was also applied to evaluate a set of constructs iteratively during the ontology construction (Vrandečić 2009). Two criteria were excluded from this validation. First, the criteria of ontology adaptability was not assessed in this work as it was not relevant to the research question for which the DEO is developed. Second, external expert validation is needed to ensure the criteria of clarity. While our ontology includes well-annotated *rdfs:comment*[13] to effectively convey intended meanings, we acknowledge that external validation could offer greater clarity. We further detail this as a limitation in "Limitations and future research directions" section and explore opportunities for updates to the DEO. While we followed established approaches, external ontology validation may provide new avenues for future research and discovery.

The criteria-based validation of DEO is as follows:

1. **Accuracy** To ensure the descriptions in DEO agree with the domain knowledge, the DEO was cross-validated against several journal articles that discuss the conceptual ideas of DEs, as well as DE papers from different application domains. These articles were selected from SLR and conceptual articles with the highest citations in the domain. Validation results have provided evidence of correctness.
2. **Completeness** This research has successfully identified, and classified various named DEs and characterized them accordingly ("Demonstration environments ontology" section). It can thus be concluded that the completeness of DEO was achieved as the research objectives detailed in "Methodology" have been met.
3. **Computation Efficiency** The operation time of the HermiT reasoner was usually in seconds. This indicates that inferences can be done in a reasonable time and thus computation efficiency was achieved.
4. **Conciseness** No concepts in the same class were found greatly overlapping with each other (except for the equivalent class), so it was concluded that there were no redundant terms in the DEO and conciseness was met.
5. **Consistency** To ensure the logical axioms are satisfiable and consistent, it has to be possible to find a situation when all the axioms are true and there are no contradictions within the axioms. The output results of HermiT reasoner along the ontology development stages have verified this satisfaction and consistency.

To conclude, ontology validation has proven the validity of DEO from various aspects.

Ontology implementation, maintenance and application

New types of DEs can constantly appear in the literature or practices, creating the need for ontology maintenance. The ontology development is accomplished in GitHub as a public online repository. For traceability and transparency of the ontology development, the issue tracking system in GitHub is utilized as a communication channel for the maintenance as well as the future updates of the DEO by the public (Diehl et al., 2016; Turchet et al., 2020). This opens the opportunity for the community to validate and contribute to the DEO. The GitHub repository is freely available at https://purl.archive.org/demonstration-environments-ontology which is synced with our OSF project. For future use, this repository will be able to track the previous version of the DEO, while the latest version of DEO will always be accessible.

In addition, the application of the DEO is also demonstrated in this study with a real example through ontology instantiation, which is detailed in "Application of DEO" section.

Demonstration environments ontology

This section describes and highlights the building blocks of DEO that define and conceptualize DEs. Figure 2 provides an overview of the structure of the DEO. Figure 3 shows an example of the classes, object properties, axioms asserted and annotations of classes described in DEO.

Classes

This ontology consists of five key classes:

- *DEO:DemonstrationEnvironment*
- *DEO:Component*

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

[12] https://oops.linkeddata.es/.

[13] https://www.w3.org/TR/rdf-schema/.

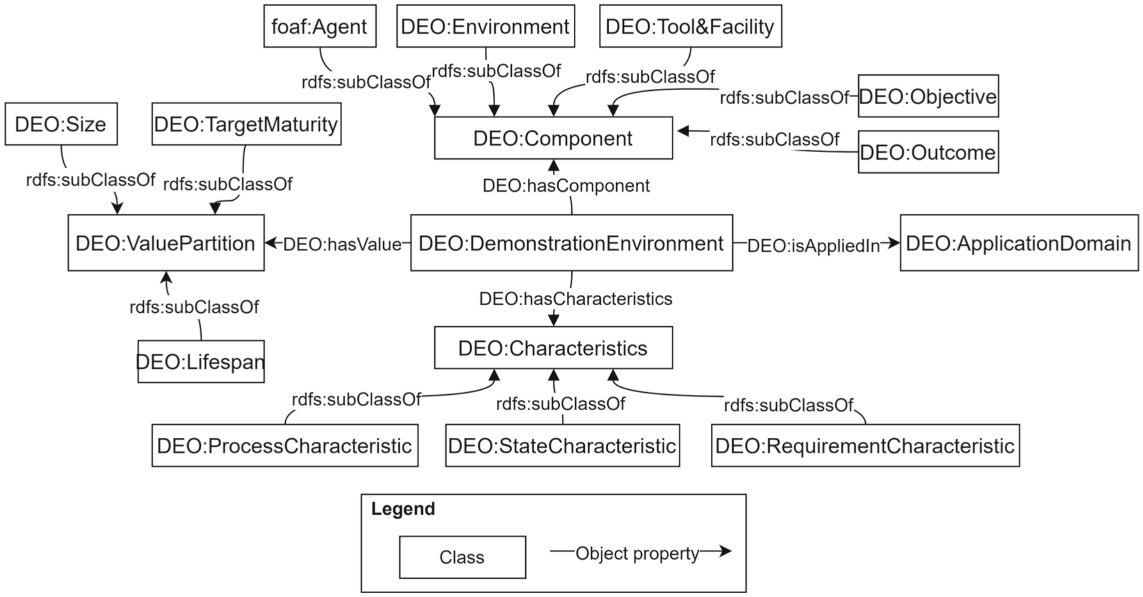
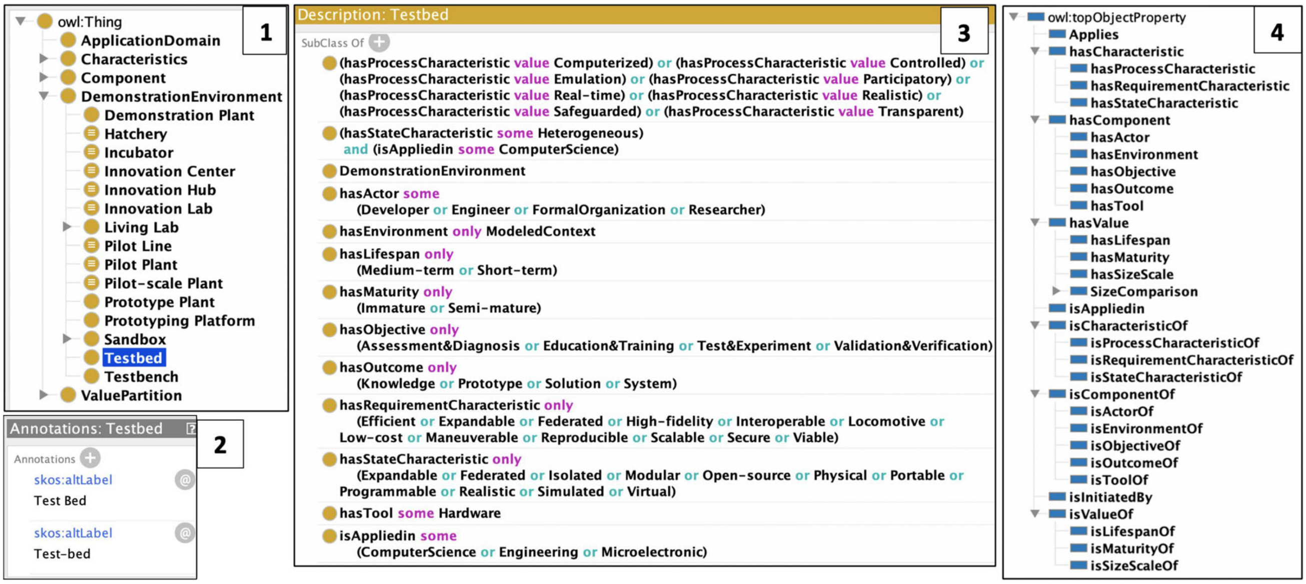


Fig. 2 Overview of the structure of Demonstration Environments Ontology (DEO)



skos:altLabel
Test Bed

skos:altLabel
Test-bed

```
(hasProcessCharacteristic value Computerized) or (hasProcessCharacteristic value Controlled) or
(hasProcessCharacteristic value Emulation) or (hasProcessCharacteristic value Participatory) or
(hasProcessCharacteristic value Real-time) or (hasProcessCharacteristic value Realistic) or
(hasProcessCharacteristic value Safeguarded) or (hasProcessCharacteristic value Transparent)
(hasStateCharacteristic some Heterogeneous)
 and (isAppliedin some ComputerScience)
DemonstrationEnvironment
hasActor some
    (Developer or Engineer or FormalOrganization or Researcher)
hasEnvironment only ModeledContext
hasLifespan only
    (Medium-term or Short-term)
hasMaturity only
    (Immature or Semi-mature)
hasObjective only
    (Assessment&Diagnosis or Education&Training or Test&Experiment or Validation&Verification)
hasOutcome only
    (Knowledge or Prototype or Solution or System)
hasRequirementCharacteristic only
    (Efficient or Expandable or Federated or High-fidelity or Interoperable or Locomotive or
Low-cost or Maneuverable or Reproducible or Scalable or Secure or Viable)
hasStateCharacteristic only
    (Expandable or Federated or Isolated or Modular or Open-source or Physical or Portable or
Programmable or Realistic or Simulated or Virtual)
hasTool some Hardware
isAppliedin some
    (ComputerScience or Engineering or Microelectronic)
```

Fig. 3 Structure of DEO represented in the Protégé ontology editor software. (1) Class hierarchy (2) Annotation for class 'Testbed' (3) Class relationships and axioms of 'Testbed' (4) Object properties

- *DEO:Characteristic*
- *DEO:ApplicationDomain*
- *DEO:ValuePartition*

DEO:DemonstrationEnvironment is the class for named DEs. The four classes *DEO:Component*, *DEO:Characteristic*, *DEO:ApplicationDomain* and *DEO:ValuePartition* further define and characterize these named DEs.

Demonstration environment class

This class defines 15 named DEs as concepts, with 9 of them are unique DE types (Fig. 4). Those grouped in the grey circle are regarded as equivalent classes. For example, *DEO:Pilot-scalePlant*, *DEO:PilotPlant* and *DEO:PilotLine* are considered interchangeable. Similarly, *DEO:InnovationLab* has four other different names representing the same concept.

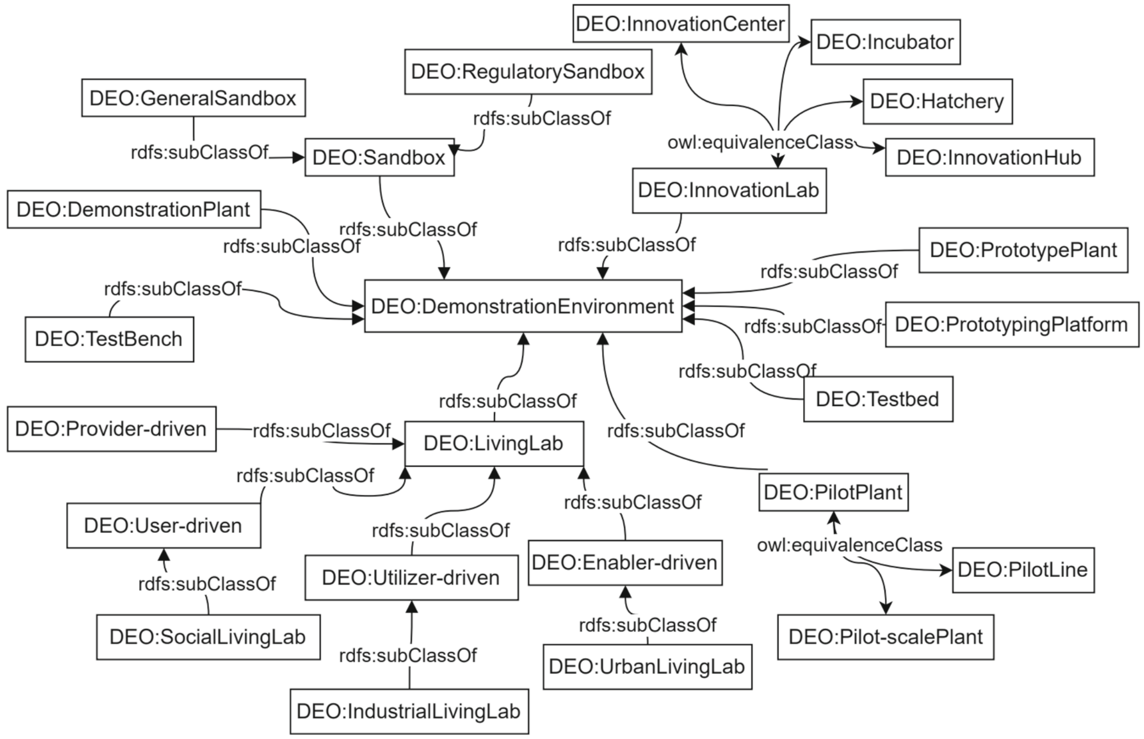


Fig. 4 *DEO:DemonstrationEnvironment* class structure

Some named DEs can be further classified based on the utility: *DEO:Sandbox* is divided into *DEO:GeneralSandbox* and *DEO:RegulatorySandbox*; *DEO:LivingLab* is categorized into four subcategories based on who initiates the LL.

Component class

The component class is defined to describe the network surrounding different types of DEs and is further split into five major subclasses (Fig. 5, only expands this class up to 2nd degree of the subclass):

1. **DEO:Objective** This class defines a set of objectives that a DE application aims to fulfil. These objectives may include tasks like conducting assessments and diagnoses (*DEO:Assessment&Diagnosis*) to detect potentially suspicious behaviors.
2. **foaf:Agent** This class refers to actors that are involved in exploiting the DEs. For example, human beings or organizational institutions that design, develop, implement or utilize a DE. Here the well-known ontology FOAF (friend of a friend)[14] was reused according to Noy and McGuinness (2001). Once FOAF is connected to the DEO, it defines the *foaf:Agent* as things (e.g., person, group, software or physical artefact) that do stuff. Two of its subclasses *foaf:Person* and *foaf:Organization* were also included to further specify the type of agents that initiate the use, support or management of a DE.
3. **DEO:Environment** This class represents a specific DE that is working in. Based on the degree of realism, the environment is classified into (1) *DEO:ModeledContext* which usually refers to the laboratory or simulated program; (2) *DEO:Real-lifeContext* such as the campus or the city which is consistent with the final application environment; and (3) *DEO:Semi-realisticContext* which lies in between the formers (e.g. a partially simulated environment placed in a real-life context like the scaled-down plant).
4. **DEO:Outcome** This class defines a set of tangible outcomes such as a product or a prototype, or intangible outcomes such as a service or a concept of the demonstration.
5. **DEO:Tool&Facility** This class pertains to the tools and facilities employed by a DE to facilitate its operations and processes. These tools and facilities might encompass software applications or machinery.

Application domain class

The *DEO:ApplicationDomain* includes a set of application domains of DEs that are most commonly identified in the text corpus. For example, living labs are commonly applied

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

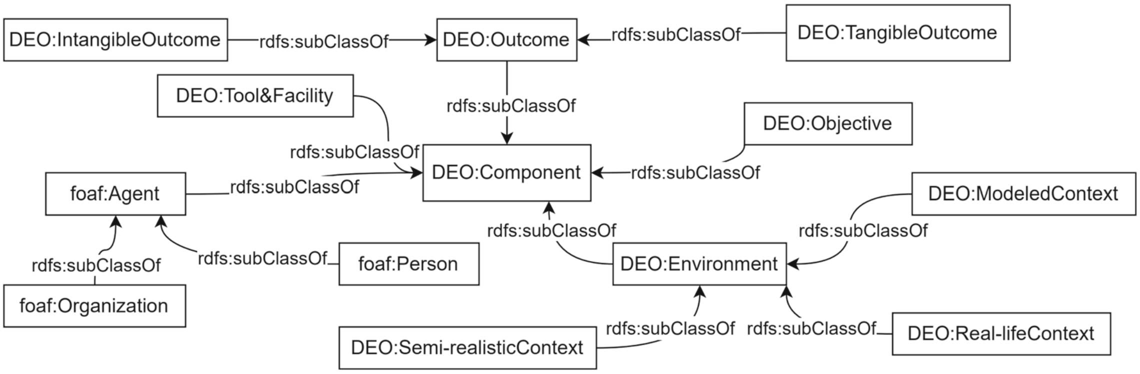


Fig. 5 *DEO:Component* class structure (excerpt)

in the domain of public affairs. The domains included are not exhaustive but representative to describe the most common areas of DE applications. This class includes defined instances such as *Engineering* and *Biochemistry*.

Characteristic class

This class includes a set of terms that are used to describe the general characteristics that are shared within a DE class. These characteristics can belong to subclass *DEO:Process* Characteristic that characterizes the process of applying DE, or subclass *DEO:StateCharacteristic* which describes the inherent features of DEs, or subclass *DEO:Requirement* Characteristic which includes a set of characteristics that need to be possessed by a DE. In addition, an instance of a characteristic can belong to multiple subclasses.

Value partition class

A value partition refers to a pattern that represents the value properties of DEs. This class defines partitions for three values: *DEO:SizeScale*, *DEO:Lifespan* and *DEO:Target* Maturity of the DEs. *DEO:SizeScale* class is used to describe the size of the DEs. This class is further split into two subclasses: *DEO:Large-scale* and *DEO:Small-scale*. The former one includes scales (defined as instances) such as *DEO:Production-scale* and *DEO:Demonstration-Scale*, which are often used to represent DEs that are applied in the last stage of demonstration. In addition, *DEO:Lifespan* describes the period during which a DE is functional, and *DEO:TargetMaturity* indicates if a DE deals with rather mature (market-ready) or nascent technologies.

Application of DEO

This section aims to illustrate the applicability of DEO by first presenting a real example for ontology instantiation, and then demonstrating how SPARQL[15] can be used to search defined knowledge in the DEO.

The MoteLab: a wireless sensor network testbed example

To test the performance of the DEO and better present its structure, a real example of the testbed called MoteLab - a wireless sensor network testbed developed by Werner-Allen et al. (2005) is presented. As shown in Fig. 6, the graph firstly introduces *DEO:MoteLab* as a specific example of a testbed. By exploiting the DEO, a semantic network to describe and characterize this DE is set up.

1. **Application domains and the value measurement** Wireless sensor network has emerged to be a new area of research in the field of Computer Science. In this case, the MoteLab is used in the application domain of *DEO:ComputerScience*. MoteLab has been largely deployed at a building in Harvard University, so its size scale is *DEO:Large-scale*. Moreover, Werner-Allen et al. (2005) mention the next phase of this testbed, as such, the maturity of the MoteLab is still *DEO:Semi-mature*.
2. **Components involved in the MoteLab** Researchers, software developers and organizations such as Intel Research and Havard University were involved in the life-cycle of MoteLab development. All these instances are pointed to *DEO:MoteLab* through the *DEO:hasActor* predicate. Moreover, the FOAF ontology then provides the human name values of *foaf:Person* instances through *foaf:name*. A set of hardware devices and software components such as PHP-coded web server and DBLogger are defined as instances of *DEO:Hardware&Software* that are linked to *DEO:MoteLab* through *DEO:hasTool* relationship in the DEO. *DEO:hasObjective* is connected with *DEO:Experimentation* and *DEO:Teaching* to show

[15] https://www.w3.org/TR/rdf-sparql-query/.

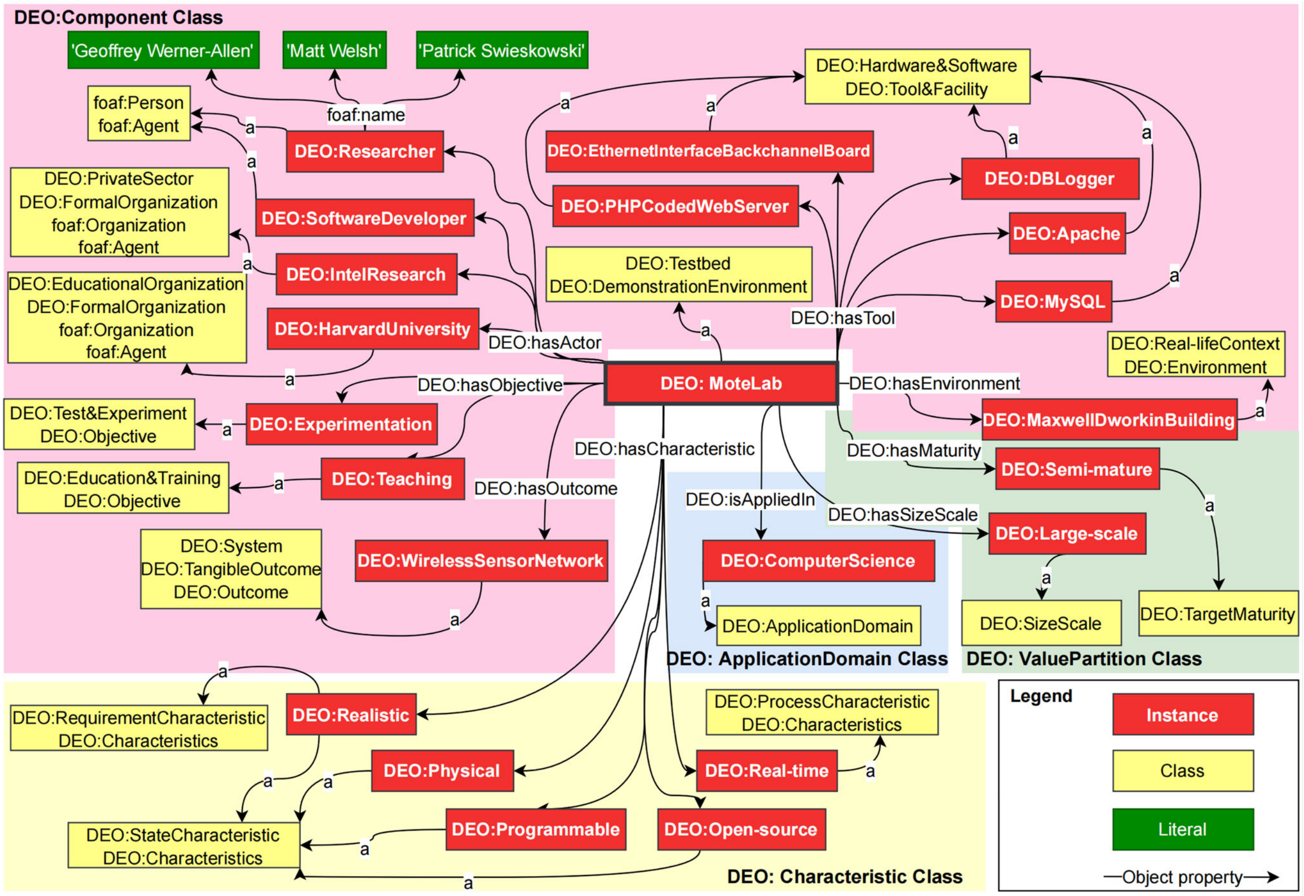


Fig. 6 Using the MoteLab example to instantiate the DEO

the objectives of developing this testbed. The corresponding outcome is hence a *DEO:WirelessSensorNetwork*, it is defined to be an instance of the class *DEO:System* which is a subclass of *DEO:TangibleOutcome*. Besides the MoteLab has been physically deployed in *DEO:* MaxwellDworkingBuilding which is a real-life environment.

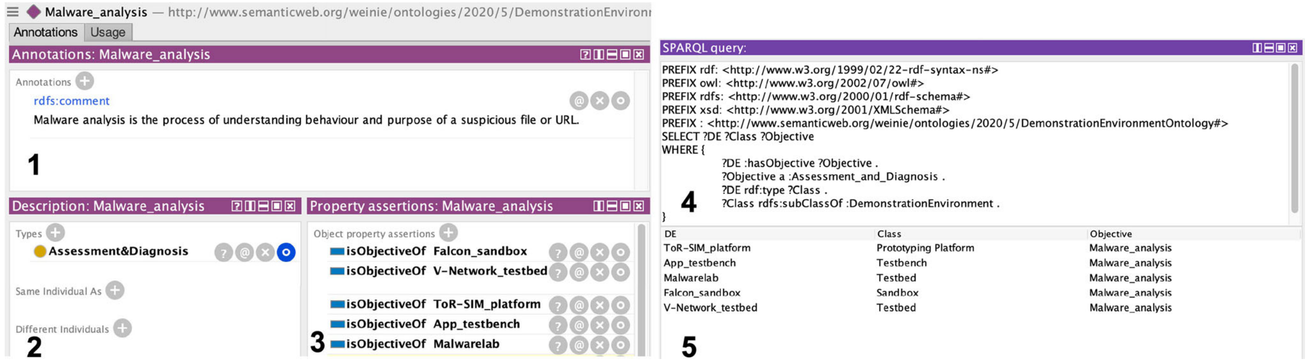
3. **Characteristics of the MoteLab** *DEO:hasCharacteristic* links with several adjectives to describe the characteristics of MoteLab. As described in the original article (Werner-Allen et al., 2005), the MoteLab was designed in a way that enables users' real-time access through several manners to perform real-time data analysis. As such, *DEO:Real-time* is defined to be an instance of the *DEO:ProcessCharacteristic* class. The article also mentioned the requirement when designing the MoteLab: the testbed has to be designed for realistic wireless sensor networks. Therefore, we instantiate *DEO:Realistic* within the *DEO:RequirementCharacteristic*. Besides the MoteLab has been designed to be open-source, automatically programmable and it is a physical thing that involves several devices. *Open-source*, *Programmable* and *Physical* are thus instances of *DEO:StateCharacteristic* that describe the states of the MoteLab.

SPARQL-based knowledge searching

This section explains how DEO can be used for knowledge searching through querying with SPARQL, which is a query language that supports the retrieval and manipulation of data stored in RDF[16] format. The DEO has established a conceptual framework for DEs. Through proper instantiation of classes and concepts, the DEO can be further used as a dictionary to search for defined knowledge given search strings coded in SPARQL syntax. To illustrate, this part presents a use case of searching for specific DEs that support the objective of assessment and diagnosis. This use case is selected because one typical user story could be a practitioner looking for suitable DEs to satisfy the innovation objective.

The full user story is described as follows: An IT maintenance staff is looking for DEs that support malware analysis. Figure 7 (top) shows the instantiation of DEO for

[16] https://www.w3.org/RDF/.


Malware_analysis — http://www.semanticweb.org/weinie/ontologies/2020/5/DemonstrationEnvironm
Annotations
Usage
Annotations: Malware_analysis
Annotations
rdfs:comment
1
Description: Malware_analysis
Types
Assessment&Diagnosis
Same Individual As
Different Individuals
2
Property assertions: Malware_analysis
Object property assertions
3
SPARQL query:
4
5

Malware analysis is the process of understanding behaviour and purpose of a suspicious file or URL.

isObjectiveOf Falcon_sandbox
isObjectiveOf V-Network_testbed
isObjectiveOf ToR-SIM_platform
isObjectiveOf App_testbench
isObjectiveOf Malwarelab

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/weinie/ontologies/2020/5/DemonstrationEnvironmentOntology#>
SELECT ?DE ?Class ?Objective
WHERE {
        ?DE :hasObjective ?Objective .
        ?Objective a :Assessment_and_Diagnosis .
        ?DE rdf:type ?Class .
        ?Class rdfs:subClassOf :DemonstrationEnvironment .
}
```

DE	Class	Objective
ToR-SIM_platform	Prototyping Platform	Malware_analysis
App_testbench	Testbench	Malware_analysis
Malwarelab	Testbed	Malware_analysis
Falcon_sandbox	Sandbox	Malware_analysis
V-Network_testbed	Testbed	Malware_analysis

Fig. 7 (Left) A screenshot of instance 'Malware_analysis' from Protégé (1) definition of 'Malware_analysis', (2) concept that 'Malware_analysis' belongs to, (3) asserted object relations between 'Malware_analysis' and *DEO: DemonstrationEnvironment* instances; (Right) Interface of corresponding SPARQL query, (4) query, and (5) query results display

the given use case: *Malware_analysis* is an instance (specific task) of *DEO:Assessment&Diagnosis*. Several real examples (instances) of DEs that can meet this objective have been created and corresponding object properties have been defined (For example, *App_testbench* which is an instance of *DEO:Testbench*) (Liang et al., 2014; Ahmad et al., 2016; Caranica et al., 2019).

Protégé provides the SPARQL Query tab as a development environment for creating SPARQL queries with the built ontology (Fig. 7 bottom). This use case presents searching instances of *DEO:DemonstrationEnvironment* that provide assessment and diagnosis of products, and look for which specific task they can perform and what DE type they are of.

It has been demonstrated how the DEO can be instantiated and used as a dictionary for querying defined knowledge. The next step of this user story could be searching for information on any of the DEs in the query result. This can be done by a similar instantiation step shown in "The MoteLab: a wireless sensor network testbed example" section and then applying SPARQL again to search the information.

Discussion

A multi-dimensional framework for conceptualising DEs was developed, with nine types of DEs being differentiated based on the four aspects which are domains to be applied, the network components involved, their characteristics and the value attributes they possess. The DEO provides a representative conceptualization of DEs with certain statistical significance. DEO has been designed and developed in a way such that it meets the four objectives as stated in "Methodology" section:

1. **Providing a structure to define and differentiate DEs by determining aspects that characterize different DEs** The structure of DEO, where the class of DE is connected with four other classes (Fig. 2), provides a standardized framework for defining DEs of different categories. In this regard, a DE can be described and characterized by its components, its characteristics, the contexts it is being applied, and dimensional values like its size. Such a standardized framework enables DEs of different types to be viewed and compared together, spurring future research to have integrated views of different DE types rather than focusing on a DE solely.
2. **Ensuring the semantics are clearly defined such that there is no ambiguity in terminology and the meanings in the knowledge representation are clear; redundancy in relevant terminologies have to be captured by this ontology** As described in Ontology Validation ("Ontology construction" section), the clarity of DEO has been examined to ensure that there is no ambiguity in defined vocabularies and relations. In addition, ontology has captured and documented redundant terminologies that have been used to describe the same thing. Differences in spelling have also been annotated if exist. This includes 'living lab', 'living laboratory' and 'living-lab' which all represent the same demonstration environment. Differences in wording have been captured as well. For example, 'pilot plant' and 'pilot-scale plant' are indeed the same with a certain statistical significance level. This is said because the language model trained in this study has indicated that these two terminologies can be considered equivalent given the text corpus we have. Annotating these terminologies enables the heterogeneity in terminologies to be reduced in future DE studies, or at least this heterogeneity could be traced.
3. **Offering means for searching and querying information, and supporting activities such as searching for and selecting suitable DEs, and proposing feasible DE designs** "SPARQL-based knowledge searching" sec-

tion presents a use case of how DEO can be employed for knowledge searching using SPARQL. This use case demonstrates the applicability of DEO to be used for querying and searching defined information. The capability of performing querying enables DEO to be deployed as a tool for practitioners like decision-makers to identify suitable DEs given certain conditions, for example, the innovation goals they seek to achieve. Practitioners like LL facilitators can also use insights gained from querying the ontology to design and prepare the DEs. For example, the developed ontology specifies the network components surrounding the DEs, such as who is involved and what tools to use. Instantiating the DEO with real-life LL examples would help to satisfy this need.

4. **Allowing the semantic model to be understood and interpreted by different users, applications and systems** Since DEO is built upon formal OWL-based specifications, it ensures the information is interoperable across a wide variety of tools and platforms like Protégé by different users.

In addition to meeting the objectives, it is noteworthy that even though the DEO has defined nine categories of DEs, the classification and characterization presented should be considered as general-typical rather than absolute. There is no absolute independence when applying different DEs. For example, the typical user-centred and real-life context features from the living labs can be incorporated into a testbed for large-scale demonstration, one instance is the smart city testbed for carrying out a controlled IoT experimentation at the city level (Sanchez et al., 2014). Further instantiations of the DEO would enable it to capture both the generality and granularity of the DEs at an instance level.

Conclusion

The evidence for the need for a standardized framework for defining and characterizing DEs is compelling due to the two gaps outlined above, namely (1) the heterogeneity of DE domain knowledge, and (2) the lack of a holistic view of different types of DEs. Therefore, this paper offers a systematic and quantitative method of realizing such a framework. This paper describes the DEO in OWL for defining and characterizing nine different DE types. The creation of this ontology is motivated by the heterogeneous terminologies and inconsistent interpretations of different DEs in existing studies. Instead of relying on a limited number of experts, this study attempts to incorporate as much diverse knowledge as possible to expand the pool of collective intelligence by applying word embedding on relevant peer-reviewed journal articles to construct the ontology. Moreover, the structure and content of the DEO have been evaluated based on automatic consistency checking, criteria-based validation and cross-validation with a set of relevant journal articles. The result of ontology validation indicates that the DEO is consistent, effective and follows good practice.

Contributions

This study makes two key contributions to the innovation management literature. First, the DEO provides standardization of terminologies related to DEs, which solves heterogeneity in the description of various names of DEs across domains. Second, the DEO also offers a common understanding of DEs through a systematic framework, which enables different DEs to be benchmarked together with the established dimensions: domains to be applied, the network components involved, their characteristics and the value attributes they possess.

This study also has practical implications for innovation managers and practitioners. Defining and characterizing DEs is essential for practitioners and researchers working in different fields, as it provides a common understanding of the essential features of DEs. This shared understanding enables effective benchmarking of different DEs across established dimensions, which is crucial for assessing the effectiveness of new technologies, products, or services in a consistent and reliable manner.

Moreover, distinguishing between living labs, pilot plants, and testbeds and characterizing these DEs is vital for ensuring that practitioners use the right type of testing environment for the specific technology, product, or service being developed. This is critical for increasing the accuracy and reliability of testing results and ensuring that the final product or service meets the required standards and satisfies the needs of the end users. By using the appropriate DE, practitioners can test new technologies, products, or services in a realistic environment, with real users, in real-life scenarios (in the case of living labs), or in a controlled environment (in the case of pilot plants and testbeds).

Ultimately, a better understanding of DEs and the appropriate use of different types of testing environments can help to accelerate the development and deployment of new technologies, products, and services, and promote innovation in various fields.

Limitations and future research directions

This study has some limitations and thus we could point out potential avenues for the development of future research. First, although we collected the data from peer-reviewed journal articles to ensure the data diversity, quality and statistical significance, some subjective judgements from domain experts can help in refining and validating the DEO further. Nevertheless, our ontology is available in the GitHub repository, enabling open contributions from the community through pull requests to suggest contributions to the DEO. Second, the DEO has been constructed in a way such that it entirely focuses on DEs and only includes necessary innovation-related classes (e.g., *DEO:Outcome*, *DEO:Tool&Facility*) to define and characterize the DEs (*DEO:DemonstrationEnvironments*). The DEO can be further expanded to cover more components in innovation management through integrating with other innovation-related ontologies, such as those reviewed in "inlinkInnovation-related ontologiesInnovationspsrelatedOntologies" section. Third, the Word2Vec model used in this study may fail to capture creative and insightful DE knowledge of the few as Schnabel et al. (2015) find that the word frequency information is contained in the word embeddings. A more sophisticated algorithm for improving the word embedding model performance is needed. Future research can be aimed at exploring algorithms for improving the word embedding model performance with low-frequency words. Finally, since the DEO first provides a standardized framework with formal and semantic representation for DEs, it serves to be an important foundation for exploring DEs in the future. Future work direction should be directed to disseminate the DEO to the industry and academia for testing its performance with real-life use cases and inputs to further improve the model usability and enrich the contents in the field of DEs.

Acknowledgements This study started whilst all authors were affiliated with the Institute for Manufacturing, University of Cambridge. We would like to thank Dr Eoin O'Sullivan for his valuable support and guidance.

Data availability The list of input journal articles analysed during the study and the Demonstration Environments Ontology developed are available in the OSF repository: https://osf.io/y9nem/. The ontology can also be accessed via the link http://purl.org/deo.

Declarations

Competing interests 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.

Appendix

Appendix A: Literature screening process

The process of screening papers to exclude irrelevant ones is semi-automatic. First, we developed an R script to automatically identify papers where the search keywords were included as false-positive words that have structural errors. For example, when searching "demonstration facility" using the search term "demonstrat* facilit*", papers using the term "demonstrate facilitate" can also be included in the search results. These papers are irrelevant and hence excluded. Second, the search keywords can be used as metaphors in the papers and therefore these papers were considered to be irrelevant as well. This step requires manual inspection. Similarly, human efforts are needed to check if the paper discusses the DEs in the main text. For example, we specifically examined if the description and implementation of the DEs have been mentioned in the paper.

Appendix B: Namespace prefixes

Table 5 lists out the full URI of all the namespace prefixes that have been used in the Demonstration Environments Ontology.

Table 5 Namespace prefixes

Prefix	URI
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
foaf	http://xmlns.com/foaf/0.1/
skos	http://www.w3.org/2004/02/skos/
DEO	http://purl.org/deo

References

Abuarqoub, A., Al-Fayez, F., Alsboui, T., Hammoudeh, M., & Nisbet, A. (2012). Simulation issues in wireless sensor networks: a survey. In *The sixth international conference on sensor technologies and applications (SENSORCOMM 2012)* (pp. 222–228).

Ahmad, M. A., Woodhead, S., & Gan, D. (2016). The v-network testbed for malware analysis. In *2016 international conference on advanced communication control and computing technologies (ICACCCT)* (pp. 629–635). https://doi.org/10.1109/ICACCCT.2016.7831716

Al-Aswadi, F. N., Chan, H. Y., & Gan, K. H. (2019). Automatic ontology construction from text: A review from shallow to deep learning trend. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-019-09782-9

Albukhitan, S. & Helmy, T. (2016). Arabic ontology learning from unstructured text. In *2016 IEEE/WIC/ACM international conference on web intelligence (WI)* (pp. 492–496). IEEE. https://doi.org/10.1109/WI.2016.0082

Almirall, E., Lee, M., & Wareham, J. (2012). Mapping living labs in the landscape of innovation methodologies. *Technology Innovation Management Review, 2*, 12–18. https://doi.org/10.22215/timreview/603

Arguello Casteleiro, M., Maseda Fernandez, D., Demetriou, G., Read, W., Fernandez Prieto, M. J., Des Diz, J., Nenadic, G., Keane, J., & Stevens, R. (2017). A case study on sepsis using pubmed and deep learning for ontology learning. *Studies in Health Technology and Informatics, 235*, 516–520. https://doi.org/10.3233/978-1-61499-753-5-516

Ayadi, A., Samet, A., de Beuvron, F., & Zanni-Merk, C. (2019). Ontology population with deep learning-based NLP: A case study on the Biomolecular Network Ontology. *Procedia Computer Science, 159*, 572–581. https://doi.org/10.1016/j.procs.2019.09.212

Ayadi, A., Zanni-Merk, C., de Beuvron, F., Thompson, D. B., & Krichen, S. (2019). BNO—An ontology for understanding the transittability of complex biomolecular networks. *Journal of Web Semantics, 57*, 100495. https://doi.org/10.1016/j.websem.2019.01.002

Ballon, P., Pierson, J., & Delaere, S. (2005). Test and experimentation platforms for broadband innovation: Examining European practice. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1331557

Bergvall-Kareborn, B., & Stahlbrost, A. (2009). Living lab: An open and citizen-centric approach for innovation. *International Journal of Innovation and Regional Development, 1*(4), 356–370. https://doi.org/10.1504/IJIRD.2009.022727

Brandt, S. C., Morbach, J., Miatidis, M., Theißen, M., Jarke, M., & Marquardt, W. (2008). An ontology-based approach to knowledge management in design processes. *Computers & Chemical Engineering, 32*(1), 320–342. https://doi.org/10.1016/j.compchemeng.2007.04.013

Brewster, C., Alani, H., Dasmahapatra, S., & Wilks, Y. (2004). Data driven ontology evaluation. In *Proceedings of the fourth international conference on language resources and evaluation (LREC'04)*. European Language Resources Association (ELRA).

Bulkeley, H., Marvin, S., Palgan, Y. V., McCormick, K., Breitfuss-Loidl, M., Mai, L., von Wirth, T., & Frantzeskaki, N. (2019). Urban living laboratories: Conducting the experimental city? *European Urban and Regional Studies, 26*(4), 317–335. https://doi.org/10.1177/0969776418787222

Bullinger, A. C. (2008). *Innovation and ontologies: Structuring the early stages of innovation management*. Springer. https://doi.org/10.1007/978-3-8349-9920-7

Caranica, A., Vulpe, A., Parvu, M. E., Draghicescu, D., Fratu, O., & Lupan, T. (2019). ToR-SIM—A mobile malware analysis platform. In *2019 international conference on speech technology and human-computer dialogue (SpeD)* (pp. 1–8). https://doi.org/10.1109/SPED.2019.8906638

Carrera, J., Carbó, O., Doñate, S., Suárez-Ojeda, M. E., & Pérez, J. (2022). Increasing the energy production in an urban wastewater treatment plant using a high-rate activated sludge: Pilot plant demonstration and energy balance. *Journal of Cleaner Production, 354*, 131734. https://doi.org/10.1016/j.jclepro.2022.131734

Catalano, C. E., Camossi, E., Ferrandes, R., Cheutet, V., & Sevilmis, N. (2009). A product design ontology for enhancing shape processing in design workflows. *Journal of Intelligent Manufacturing, 20*(5), 553–567. https://doi.org/10.1007/s10845-008-0151-z

Chen, Y., Yu, C., Liu, X., Xi, T., Xu, G., Sun, Y., Zhu, F., & Shen, B. (2021). Pclion: An ontology for data standardization and sharing of prostate cancer associated lifestyles. *International Journal of Medical Informatics, 145*, 104332. https://doi.org/10.1016/j.ijmedinf.2020.104332

Clark, A. J., Jirak, I. L., Dembek, S. R., Creager, G. J., Kong, F., Thomas, K. W., Knopfmeier, K. H., Gallo, B. T., Melick, C. J., Xue, M., Brewster, K. A., Jung, Y., Kennedy, A., Dong, X., Markel, J., Gilmore, M., Romine, G. S., Fossell, K. R., Sobash, R. A., ... D. A. (2018). The community leveraged unified ensemble (clue) in the 2016 NOAA/hazardous weather testbed spring forecasting experiment. *Bulletin of the American Meteorological Society, 99*(7), 1433–1448. https://doi.org/10.1175/BAMS-D-16-0309.1

Costa, R., Lima, C., Sarraipa, J., & Jardim-Gonçalves, R. (2016). Facilitating knowledge sharing and reuse in building and construction domain: An ontology-based approach. *Journal of Intelligent Manufacturing, 27*(1), 263–282. https://doi.org/10.1007/s10845-013-0856-5

Criado, J. I., Dias, T. F., Sano, H., Rojas-Martín, F., Silvan, A., & Filho, A. I. (2021). Public innovation and living labs in action: A comparative analysis in post-new public management contexts. *International Journal of Public Administration, 44*(6), 451–464. https://doi.org/10.1080/01900692.2020.1729181

Dahlander, L., & Gann, D. M. (2010). How open is innovation? *Research Policy, 39*(6), 699–709. https://doi.org/10.1016/j.respol.2010.01.013

De Vita, K., & De Vita, R. (2021). Expect the unexpected: Investigating co-creation projects in a living lab. *Technology Innovation Management Review, 11*, 6–20. https://doi.org/10.22215/timreview/1461

Deepa, R., & Vigneshwari, S. (2022). An effective automated ontology construction based on the agriculture domain. *ETRI Journal, 44*(4), 573–587. https://doi.org/10.4218/etrij.2020-0439

Deiana, P., Bassano, C., Calì, G., Miraglia, P., & Maggio, E. (2017). Co2 capture and amine solvent regeneration in Sotacarbo pilot plant. *Fuel, 207*, 663–670. https://doi.org/10.1016/j.fuel.2017.05.066

Diehl, A. D., Meehan, T. F., Bradford, Y. M., Brush, M. H., Dahdul, W. M., Dougall, D. S., He, Y., Osumi-Sutherland, D., Ruttenberg, A., Sarntivijai, S., et al. (2016). The cell ontology 2016: Enhanced content, modularization, and ontology interoperability. *Journal of Biomedical Semantics, 7*, 1–10. https://doi.org/10.1186/s13326-016-0088-7

El Bassiti, L., & Ajhoun, R. (2014). Semantic representation of innovation, generic ontology for idea management. *Journal of Advanced Management Science, 2*(2), 128–134. https://doi.org/10.12720/joams.2.2.128-134

Evans, J., & Karvonen, A. (2014). 'Give me a laboratory and i will lower your carbon footprint!'—-Urban laboratories and the governance of low-carbon futures. *International Journal of Urban and Regional Research, 38*(2), 413–430. https://doi.org/10.1111/1468-2427.12077

Faria, C., Serra, I., & Girardi, R. (2014). A domain-independent process for automatic ontology population from text. *Science of Computer*

Programming, 95, 26–43. https://doi.org/10.1016/j.scico.2013.12.005

Fecher, F., Winding, J., Hutter, K., & Füller, J. (2020). Innovation labs from a participants' perspective. *Journal of Business Research, 110*, 567–576. https://doi.org/10.1016/j.jbusres.2018.05.039

Fernández-López, M., Gomez-Perez, A., & Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. In *1997 AAAI spring symposium*.

Gabriel, A., Chavez, B. P., & Monticolo, D. (2019). Methodology to design ontologies from organizational models: Application to creativity workshops. *AI EDAM, 33*(2), 148–159. https://doi.org/10.1017/S0890060419000088

Greenly, W. (2012). *Ontology for innovation*. Retrieved from http://www.lexicater.co.uk/vocabularies/innovation/ns.html

Greve, K., De Vita, R., Leminen, S., & Westerlund, M. (2021). Living labs: From niche to mainstream innovation management. *Sustainability, 13*(2), 791. https://doi.org/10.3390/su13020791

Greve, K., Leminen, S., De Vita, R., & Westerlund, M. (2020). Unveiling the diversity of scholarly debate on living labs: A bibliometric approach. *International Journal of Innovation Management, 24*(08), 2040003. https://doi.org/10.1142/S1363919620400034

Greve, K., & O'Sullivan, E. (2019). Demonstration environments for emerging technologies: Insights from a living lab. In *ISPIM conference proceedings. The international society for professional innovation management (ISPIM)* (pp. 1–13).

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

He, T., Zhang, X., & Ye, X. (2006). An approach to automatically constructing domain ontology. In *Proceedings of the 20th pacific asia conference on language, information and computation* (pp. 150–157).

Hellsmark, H., Frishammar, J., Söderholm, P., & Ylinenpää, H. (2016). The role of pilot and demonstration plants in technology development and innovation policy. *Research Policy, 45*(9), 1743–1761. https://doi.org/10.1016/j.respol.2016.05.005

Högman, U., & Johannesson, H. (2013). Applying stage-gate processes to technology development-experience from six hardware-oriented companies. *Journal of Engineering and Technology Management, 30*(3), 264–287. https://doi.org/10.1016/j.jengtecman.2013.05.002

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

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B. C., & Hendler, J. (2006). Swoop: A web ontology editing browser. *Journal of Web Semantics, 4*(2), 144–153. https://doi.org/10.1016/j.websem.2005.10.001

Kim, D.-Y., Park, J.-W., Baek, S., Park, K.-B., Kim, H.-R., Park, J.-I., Kim, H.-S., Kim, B.-B., Oh, H.-Y., Namgung, K., & Baek, W. (2020). A modular factory testbed for the rapid reconfiguration of manufacturing systems. *Journal of Intelligent Manufacturing, 31*(3), 661–680. https://doi.org/10.1007/s10845-019-01471-2

Kulmanov, M., Smaili, F. Z., Gao, X., & Hoehndorf, R. (2020). Semantic similarity and machine learning with ontologies. *Briefings in Bioinformatics, 22*(4), bbaa199. https://doi.org/10.1093/bib/bbaa199

Kumar, A., & Starly, B. (2021). "FabNER": Information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01807-x

Leminen, S., Nyström, A.-G., & Westerlund, M. (2015). A typology of creative consumers in living labs. *Journal of Engineering and Technology Management, 37*, 6–20. https://doi.org/10.1016/j.jengtecman.2015.08.008

Leminen, S., Westerlund, M., & Nyström, A.-G. (2012). Living labs as open-innovation networks. *Technology Innovation Management Review, 2*, 6–11.

Liang, S., Du, X., Tan, C. C., & Yu, W. (2014). An effective online scheme for detecting android malware. In *2014 23rd international conference on computer communication and networks (ICCCN)* (pp. 1–8). https://doi.org/10.1109/ICCCN.2014.6911740

Lorenzo, L., Lizarralde, O., Santos, I., & Passant, A. (2021). Structuring e-brainstorming to better support innovation processes. *Proceedings of the International AAAI Conference on Web and Social Media, 5*(2), 20–23. https://doi.org/10.1609/icwsm.v5i2.14205

Lozano-Tello, A., & Gomez-Perez, A. (2004). ONTOMETRIC: A method to choose the appropriate ontology. *Journal of Database Management, 15*(2), 1–18. https://doi.org/10.4018/jdm.2004040101

Lu, Y., Wang, H., & Xu, X. (2019). ManuService ontology: A product data model for service-oriented business interactions in a cloud manufacturing environment. *Journal of Intelligent Manufacturing, 30*(1), 317–334. https://doi.org/10.1007/s10845-016-1250-x

Maedche, A., & Staab, S. (2002). Measuring similarity between ontologies. In A. Gómez-Pérez & V. R. Benjamins (Eds.), *Knowledge engineering and knowledge management: Ontologies and the semantic web* (pp. 251–263). Springer. https://doi.org/10.1007/3-540-45810-7_24

Mahmoud, N., Elbeh, H., & Abdlkader, H. M. (2018). Ontology learning based on word embeddings for text big data extraction. In *2018 14th international computer engineering conference (ICENCO)* (pp. 183–188). IEEE. https://doi.org/10.1109/ICENCO.2018.8636154.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

Mian, S., Lamine, W., & Fayolle, A. (2016). Technology business incubation: An overview of the state of knowledge. *Technovation, 50–51*, 1–12. https://doi.org/10.1016/j.technovation.2016.02.005

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:1301.3781

Mizoguchi, R. & Ikeda, M. (1997). Towards ontology engineering. In *Proceedings of joint pacific asian conference on expert systems: international conference on intelligent systems* (pp. 259–266). Singapore.

Montero Jiménez, J. J., Vingerhoeds, R., Grabot, B., & Schwartz, S. (2021). An ontology model for maintenance strategy selection and assessment. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01855-3

Ni, X., Samet, A., & Cavallucci, D. (2022). Similarity-based approach for inventive design solutions assistance. *Journal of Intelligent Manufacturing, 33*, 1–18. https://doi.org/10.1007/s10845-021-01749-4

Ning, K., O'Sullivan, D., Zhu, Q., & Decker, S. (2006). Semantic innovation management across the extended enterprise. *International Journal of Industrial and Systems Engineering, 1*(1–2), 109–128. https://doi.org/10.1504/IJISE.2006.009052

Noy, N. F. & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. *Stanford Knowledge Systems Laboratory*. Retrieved from http://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Osorio, F., Dupont, L., Camargo, M., Palominos, P., Peña, J. I., & Alfaro, M. (2019). Design and management of innovation laboratories: Toward a performance assessment tool. *Creativity and Innovation Management, 28*(1), 82–100. https://doi.org/10.1111/caim.12301

Papadopoulos, G. Z., Gallais, A., Schreiner, G., Jou, E., & Noel, T. (2017). Thorough IoT testbed characterization: From proof-of-concept to repeatable experimentations. *Computer Networks, 119*, 86–101. https://doi.org/10.1016/j.comnet.2017.03.012

Porzel, R. & Malaka, R. (2004). A task-based approach for ontology evaluation. In *ECAI workshop on ontology learning and population* (pp. 1–6).

Protégé (2000). The Protégé Project. http://protege.stanford.edu.

Raad, J. & Cruz, C. (2015). A survey on ontology evaluation methods. In *Proceedings of the international joint conference on knowledge discovery, knowledge engineering and knowledge management, IC3K 2015* (pp. 179–186). SCITEPRESS—Science and Technology Publications, Lda. https://doi.org/10.5220/0005591001790186

Riedl, C., May, N., Finzen, J., Stathel, S., Kaufman, V., & Krcmar, H. (2011). An idea ontology for innovation management. In Sheth, A., (Ed.), *Semantic services, interoperability and web applications: Emerging concepts* (pp. 303–321). IGI Global. https://doi.org/10.4018/978-1-60960-593-3.ch012

Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., & Pfisterer, D. (2014). Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks, 61*, 217–238. https://doi.org/10.1016/j.bjp.2013.12.020

Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 298–307). Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1036

Schuurman, D., & Tõnurist, P. (2017). Innovation in the public sector: Exploring the characteristics and potential of living labs and innovation labs. *Technology Innovation Management Review, 7*, 7–14.

Shearer, R. D., Motik, B., & Horrocks, I. (2008). Hermit: A highly-efficient owl reasoner. In *Owled* (vol. 432, p. 91).

Stankovic, M. (2010). Open innovation and semantic web: Problem solver search on linked data. In *Proceedings of international semantic web conference (ISWC) 7th–11th Novebmer*.

Studer, R., Benjamins, V., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering, 25*(1), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

Talhi, A., Fortineau, V., Huet, J.-C., & Lamouri, S. (2019). Ontology for cloud manufacturing based Product Lifecycle Management. *Journal of Intelligent Manufacturing, 30*(5), 2171–2192. https://doi.org/10.1007/s10845-017-1376-5

Turchet, L., Antoniazzi, F., Viola, F., Giunchiglia, F., & Fazekas, G. (2020). The internet of musical things ontology. *Journal of Web Semantics, 60*, 100548. https://doi.org/10.1016/j.websem.2020.100548

Turhan, A.-Y. (2011). Description logic reasoning for semantic web ontologies. In *Proceedings of the international conference on web intelligence, mining and semantics, WIMS '11*. Association for Computing Machinery. https://doi.org/10.1145/1988688.1988696

Voytenko, Y., McCormick, K., Evans, J., & Schliwa, G. (2016). Urban living labs for sustainability and low carbon cities in Europe: Towards a research agenda. *Advancing Sustainable Solutions: An Interdisciplinary and Collaborative Research Agenda, 123*, 45–54. https://doi.org/10.1016/j.jclepro.2015.08.053

Vrandečić, D. (2009). *Ontology evaluation* (pp. 293–313). Springer. https://doi.org/10.1007/978-3-540-92673-3_13

Walisadeera, A. I., Ginige, A., & Wikramanayake, G. N. (2016). Ontology evaluation approaches: A case study from agriculture domain. In *Computational science and its applications—ICCSA 2016, July 4–7* (vol. 9789, pp. 318–333). Springer. https://doi.org/10.1007/978-3-319-42089-9_23

Werner-Allen, G., Swieskowski, P., & Welsh, M. (2005). MoteLab: A wireless sensor network testbed. In IPSN 2005. In *Fourth international symposium on information processing in sensor networks* (pp. 483–488). https://doi.org/10.1109/IPSN.2005.1440979

Westerlund, M., Leminen, S., & Rajahonka, M. (2018). A topic modelling analysis of living labs research. *Technology Innovation Management Review, 8*, 40–51. https://doi.org/10.22215/timreview/1170

Westerski, A., Iglesias, C. A., & Rico, F. T. (2010). A model for integration and interlinking of idea management systems. In S. Sánchez-Alonso & I. N. Athanasiadis (Eds.), *Metadata and semantic research* (pp. 183–194). Springer. https://doi.org/10.1007/978-3-642-16552-8_18

Whicher, A., & Crick, T. (2019). Co-design, evaluation and the northern Ireland innovation lab. *Public Money & Management, 39*(4), 290–299. https://doi.org/10.1080/09540962.2019.1592920

Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys*. https://doi.org/10.1145/2333112.2333115

Zanni-Merk, C., de Bertrand de Beuvron, F., Rousselot, F., & Yan, W. (2013). A formal ontology for a generalized inventive design methodology. *Applied Ontology, 8*(4), 231–273. https://doi.org/10.3233/AO-140128

Zhang, P., Qu, Y., & Huang, C. (2011). Designing a multi-layered ontology for the science and technology innovation concept knowledge-base. In *2011 44th Hawaii international conference on system sciences (HICSS 2011)* (pp. 1–10). IEEE Computer Society. https://doi.org/10.1109/HICSS.2011.147

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). Birch: An efficient data clustering method for very large databases. *ACM SIGMOD Record, 25*(2), 103–114. https://doi.org/10.1145/235968.233324

Zhou, L. (2007). Ontology learning: State of the art and open issues. *Information Technology and Management, 8*(3), 241–252. https://doi.org/10.1007/s10799-007-0019-5

Zurbriggen, C., & Lago, M. G. (2019). An experimental evaluation tool for the public innovation lab of the Uruguayan government. *Evidence & Policy, 15*(3), 437–451. https://doi.org/10.1332/174426419X15537488717501