

Collaborative Ontology Development Approach for Data Spaces

Erik Paul Konietzko, Sonika Gogineni, Kai Lindow

Intelligent Integration

Fraunhofer Institute for Production Systems and Design Technology IPK

Berlin, Germany

e-mail: erik.paul.konietzko@ipk.fraunhofer.de

e-mail: sonika.gogineni@ipk.fraunhofer.de

e-mail: kai.lindow@ipk.fraunhofer.de

Abstract — The increasing demand for reliable, secure, and sovereign cross-organizational data exchange has led to the emergence of data spaces. Effective collaboration hinges on shared semantic models, positioning collaborative ontology design as a critical research area. However, existing literature on approaches is limited and often overlooks the combination of essential factors for successful collaboration in engineering contexts, such as processes, stakeholders, and artifacts. This paper introduces the Collaborative Ontology Development approach for data Spaces (CODES approach) considering these factors. The paper concludes with a discussion and proposes the next steps for evaluation.

Keywords—*Ontology alignment; data space; collaborative ontology development approach.*

I. INTRODUCTION

Collaborative ontology development is increasingly relevant in the context of knowledge management and information sharing across diverse domains. Especially the conception of semantic web technologies was based on the advantages of sharing and reusing ontologies across various domains. The collaborative development facilitates the integration and alignment of heterogeneous data from multiple perspectives and stakeholders ensuring that ontologies are representative of a broader range of expertise and use cases [1]. The collaborative approach not only enhances the quality and usability of the ontologies but supports in lifecycle management of the ontologies and its applications, as it has verified inputs from various stakeholders working together as a team [2].

Over the last few years, the concept of data space has gained popularity and support to manage and integrate large, heterogeneous and distributed data sources over company borders [3]. Data spaces are decentralized infrastructure which enable trustworthy, sovereign, and secure data exchange based on common principles and policies [4]. To manage heterogeneous data sources within such data spaces, the use of ontologies is becoming increasingly important, as they have proven to be beneficial for enhancing interoperability and integrating diverse data [4]. Consequently, they have already been incorporated into several initiatives, such as Catena-X or Gaia-X with more initiatives expected to follow [5].

Collaborative ontology development involves various processes, activities, artefacts, roles, and IT tools and technologies. Ontology engineering and management encompasses ontology requirements specification, implementation, evaluation, publication, evolution and maintenance [6][7]. These processes are made up of various activities, artefacts, roles, IT tools, and technologies. All these aspects need to be considered in the context of data space ontology development. There are very few publications which present approaches to deal with ontology engineering for data space involving various internal and external stakeholders [8][9]. These publications, however, do not focus on the aspects of harmonization of data models, of ontologies and of architectures, interactions between the stakeholders, change and dependency management and governance of the ontologies [5][8][10].

This publication aims to present the current state of the art for collaborative ontology development in data spaces, to derive the research challenges (Section II). This is followed by the presentation of the Collaborative Ontology Development for Data Spaces (CODES) approach (Section III). The conclusion is documented in Section IV, and the outlook underscores the need for evaluation of the approach in practical projects. Furthermore, it highlights the importance of focused research and exploration of individual topics to strengthen the foundation of collaborative ontology design.

II. STATE OF THE ART: COLLABORATIVE ONTOLOGY DEVELOPMENT IN ENGINEERING DOMAINS

Semantically aligned ontologies play a pivotal role in the context of data spaces; however, significant research gaps persist across various related topics, both within and beyond data spaces. This section organizes and categorizes key topics essential for collaborative ontology development, emphasizing existing research while identifying areas requiring further investigation. The concluding subsection focuses on outlining research challenges to be addressed to support the progress of collaborative ontology development for data spaces.

A. Collaborative ontology development approaches

The authors of [2] present the evolution of ontology approaches over three generations. Early ontology engineering methodologies, such as [11], [12] emphasized

core activities including requirements analysis, conceptualization, implementation, evaluation, and maintenance. These methodologies assumed that formal domain knowledge specification precedes system development. In contrast, second-generation methodologies adopted a more iterative approach, integrating application-specific requirements into the requirements analysis phase and allowing for incremental releases of ontology versions to accommodate changing needs. A notable characteristic of these approaches is the clear division of responsibilities among domain experts, knowledge engineers, ontology engineers, and users, with engineers driving the process by gathering requirements, implementing them, testing ontologies, and managing their evolution. Examples include Methontology [13] and OnToKnowledge [14][15]. The current third generation employs a participatory approach, emphasizing collaboration among a diverse group of contributors and providing technological support to enable non-experts to engage in ontology development activities beyond requirements of engineering. Several methodologies detail the collaborative engineering process for developing and maintaining ontologies in decentralized scenarios, with DILIGENT [16] and HCOME [17] being the most notable examples. However, these methodologies are limited in terms of concrete case study descriptions and associated technological support. In this context, approaches, such as Ontology Maturing [18], suggest and evaluate tool support based on degree of maturity and phase of ontology usage. Additionally, RapidOWL [19] presents a valuable set of guidelines that can inform the design of collaborative ontology engineering methodologies and help align existing methodologies with broader principles of agile engineering and rapid prototyping. In the Hozo approach an environment for distributed ontology development is described based on types of dependencies between ontologies and resulting change patterns, as well as collaborative implications [20]. This evolution in methodology underscores the ongoing need for adaptable frameworks to support effective ontology development in diverse and dynamic environments. However, these methodologies do not focus on the dynamics and challenges of collaborative ontology engineering specific to data spaces.

The authors of [9] present the AIME methodology for collaborative ontology development in data spaces. They focus on the data space challenges, integration of FAIR principles and various stakeholders. However, the identification and management of change and dependency management are not explicitly detailed. The authors of [21] focus on ontology-based data access for data spaces and do not provide a methodology for collaboratively engineering the ontology. In [22], the authors present agile-based collaborative steps for developing the information model for the international data space initiative. The model includes the following conceptual areas: digital resources, participants, roles, identities, usage of contracts / policies, metadata, and infrastructure processes. However, no explicit approach is defined as a detailed collaborative approach to develop ontologies in data spaces.

B. Identified research challenges

As identified in the preceding subsections, there are several research gaps in literature, which need to be addressed or further explored. The gaps have been grouped into five challenges based on the type of gap:

Challenge 1 (C1) – Aligned knowledge management: Managing knowledge across varied domains poses substantial challenges due to differences in conceptual understanding, workflows, data models, and contextual data interpretation. Effective knowledge integration requires harmonizing disparate perspectives, aligning processes, and coordinating multiple stakeholders [23]. A critical aspect of this effort involves developing unified ontology for the shared data space or establishing effective mechanisms to integrate existing ontologies with newly defined ones.

C2 - Access and governance in data spaces: Data spaces face challenges in access and governance, stemming from ambiguity around who holds the authority to define key elements, their roles within the ecosystem, and the rights they possess [8][9]. Participation is constrained by sector-based eligibility and closed governance, limiting cross-domain expertise and innovation. Opaque decision rights and unclear ownership of ontology elements deter external contributors and create bottlenecks.

C3 - Collaborative models in data spaces: The key challenge is to identify and implement collaborative structures that actively engage diverse participants with varying motivations. This requires a framework that aligns incentives, appreciates the complexities of collaboration, and clarifies the nature of shared responsibilities in defining data spaces and their semantics. There is a need to understand different types of open-source collaboration, as these models significantly impact participation incentives [24]. Commonly, a core team manages development, while other contributors suggest or implement changes — a structure that may limit broader engagement.

C4 - Harmonization of data models, ontologies and architectures: This challenge concerns the establishment of harmonized semantic annotations within the data space and with particular use cases it supports. Hence, there is a need for collaboration formats, mapping standard and operating procedures to integrate and improve interoperability [8].

C5 - Data value: Despite advancements in transaction metadata and data format specifications, data spaces face a critical challenge in the absence of shared, machine-readable semantics for exchanged data. Distribution mechanisms remain fragmented and proprietary [1].

The solution lies in developing methods, processes, and governance frameworks to collaboratively create and maintain shared, extensible ontology and semantic profiles. These must facilitate consistent interpretation and integration of heterogeneous data across data spaces and support incremental adoption.

III. CODES APPROACH: COLLABORATIVE ONTOLOGY DEVELOPMENT FOR DATA SPACES APPROACH

To address the previously identified challenges, this section outlines a structured approach built upon best practices identified in the literature and insights gained from practical experience in developing data spaces [25][26][27]. This methodology integrates established theoretical frameworks with lessons learned from real-world projects, for the development of a novel approach tailored to the complexities of data spaces. Figure 1 outlines the ten steps of CODEs for collaborative ontology development in data spaces. The steps are outlined in detail below:

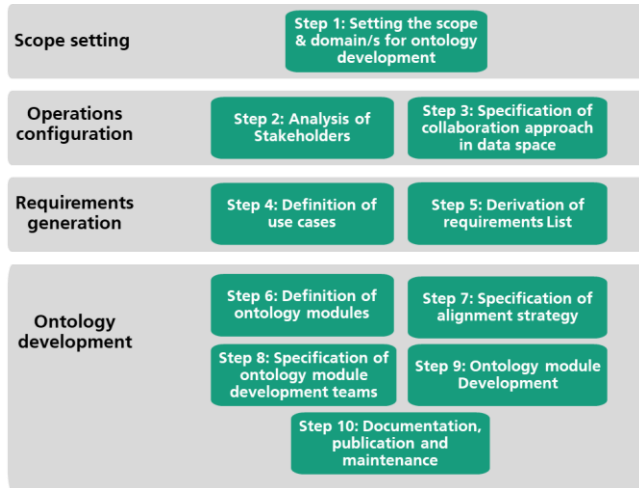


Figure 1. Visualization of CODEs approach steps.

A. Step 1: Setting the scope of ontology to be developed in data space

There are two main parts in this step. The first is to determine the domain/s the data space aims to address. The second is to identify the existing ontologies, semantic artefacts, governance frameworks and standards in the identified domain/s [5][9]. Selecting relevant state of the art for the data space with the existing stakeholders. This includes commonly used data, models, communication and certification mechanisms, infrastructure (protocols, interfaces, etc.) and tooling in the domains and their development environments.

B. Step 2: Stakeholder Analysis

Stakeholder analysis defines roles, responsibilities, and collaboration approaches through two sub-steps: (1) identifying and verifying stakeholder coverage across development areas, and (2) documenting assignments and commitments while acknowledging their evolution over time.

Figure 2 illustrates role distribution in ontology engineering within data spaces. Indirect roles (data space participants, domain experts) contribute content with varying involvement levels. Direct roles focus on ontology development across three levels.

Roles in data space ontology development define general semantics for data space basis or federation services at the

most abstract level, with largely decoupled collaboration to individual dataspace communities.

Federation ontology development roles specify semantics for core services and standards (versioning, interfaces), requiring close collaboration with data space design teams.

Roles for domain ontology development collaborate with domain experts and participants, adapting to domain-specific engagement forms.

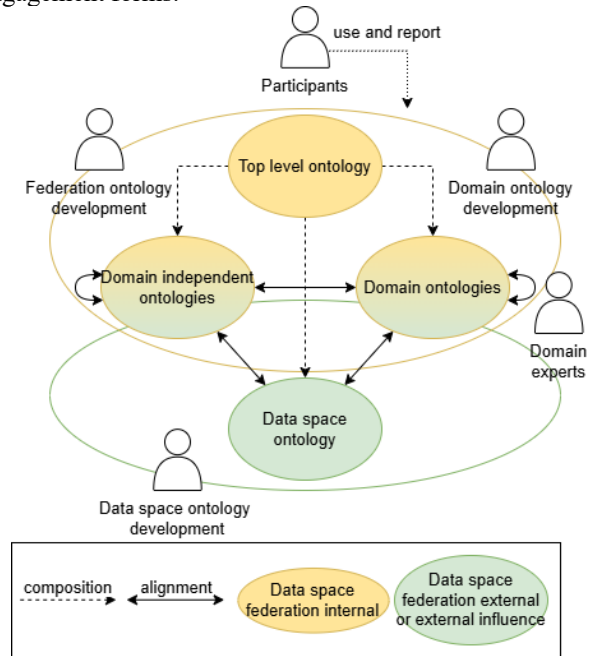


Figure 2. Semantic distribution and roles in data space setting.

Examples of stakeholders can be found in [2][9][28]. The key stakeholders are summarized below, although this list is not exhaustive: Participant/Technology user roles (end-users, Original Equipment Manufacturers (OEMs), suppliers); Domain roles (domain experts, engineers, data stewards, legal/business representatives); Ontology roles (engineers, stewards, maintainers); Collaboration roles (project managers, team leaders, contributors); Federation roles (federators, governance/compliance managers); and external roles (standardization bodies, prospective participants) ensuring continuity, alignment, and adoption.

C. Step 3: Specification of collaboration approach in the data space

While not directly a development step for collaborative ontology development, the selected collaboration model (e.g., open/closed source or consortium-based) plays a critical role in shaping stakeholder participation in ontology development within the data space [24][29]. This step involves the following key aspects:

1) Specification of collaboration scope

Building on the stakeholder documentation from step 2, this involves defining the roles, activities, and responsibilities required for collaboration, along with their timeframes and allocation.

2) *Specification of collaboration roles and cruciality*

Collaboration roles depend on the chosen collaboration model and the data space structure. Key roles may include the core team (comprised of associations, standardization bodies, research institutions, and industry representatives), contributors (data space participants), driving forces, and federators.

3) *Specification of collaboration processes*

Collaboration processes define how and when stakeholders interact. For instance, contributors might propose ideas, while the core team evaluates, develops, integrates, and communicates these contributions. Additionally, the mechanisms for redefining or reassigning roles and processes over time need to be outlined in this step.

4) *Specification of collaboration technology stack*

Identify the tools, platforms, and documentation formats required to support collaboration. This includes tools for ontology specification and processes, such as annotating ontologies with custom metamodel attributes during collaboration.

5) *Specification of communication artifacts and channels*

Define what information will be visible to external parties, how contributors will be informed, and strategies to foster engagement. For example, regular releases of models and specifications can serve as communication artifacts to maintain transparency and encourage participation.

D. *Step 4: Definition of use cases*

This step focuses on identifying and specifying the use cases that will guide activities within the data space. It is a critical step for determining the actions to be performed and the information required for successful implementation [9], [30]. The process consists of the following activities:

1) *Define Use Cases*

Stakeholders should collaboratively define the use cases for the data space. This includes outlining specific scenarios where the data space will be used for offering and consuming various services, ensuring a comprehensive understanding of user needs and operational requirements.

2) *Cluster and prioritize Use Cases*

Domain experts and participants play a pivotal role in clustering the defined use cases based on relevance and interdependence. Prioritization should follow to ensure that the most critical use cases are addressed first, aligning with the overall goals of the data space.

3) *Document Use Cases (Backlog)*

Finally, all defined and prioritized use cases should be thoroughly documented in a backlog. This documentation serves as a reference point for the next sprint of collaborative development process, ensuring alignment and clarity as the project progresses.

4) *Map Roles to Use Cases*

It is essential to map the roles identified in Step 2 to the defined use cases. This mapping should focus on specifying responsibilities in the collaborative process and the expected outcomes for each role, ensuring that everyone understands their contributions to the use cases.

5) *Match Stakeholders with Roles*

Once roles are mapped, individual stakeholders should be matched to these roles based on their expertise and capacity. Open and transparent communication during this phase is vital, allowing stakeholders to express their capabilities and constraints, such as their availability.

E. *Step 5: Derivation of requirements list for ontology development*

In collaborative developments involving diverse expertise, it is essential to translate stakeholder needs into clear requirements. Since data spaces combine content-related and technical components, multiple requirement categories must be considered when designing data space semantics.

Content-related requirements are derived from domain vocabularies, standards, specifications, and data models. Competency questions validate the ontologies by comparing expected and actual query results.

Functional and technological requirements define semantic functionalities and technologies, such as supported languages (e.g., Resource Description Framework (RDF), Web Ontology Language (OWL), JavaScript Object Notation for Linked Data (JSON-LD)), software tools (e.g., for modeling or validation), inference depth, rule usage, validation mechanisms (e.g., Shapes Constraint Language (SHACL)), ontology topologies (e.g., single vs. multiple ontologies), modularization, versioning, alignment strategies, and lifecycle management.

Quality requirements encompass structural, syntactic, and content checks for ontology instances, as well as maintainability, maturity, and usability. Best practices for modeling and overarching quality criteria should be collaboratively defined and applied.

Performance requirements focus on system and semantic performance (e.g., query and reasoning efficiency) and influence decisions on technologies and semantic networks, such as virtualization vs. materialization, distributed reasoning, query batching, networked repositories, and database partitioning.

However, the requirement categories depend on the data spaces and the use cases and are not exhaustive.

Nachabe et al. [9] present interaction modules for requirement identification. The method for identifying and documenting requirements must be defined among stakeholders according to the agreed collaboration form. Extensive approaches, such as data flow analysis [31], offer deeper insights for specifying infrastructure, performance, and functional requirements of the ontology topology and core data space services within holistic development environments [25].

F. *Step 6: Definition of ontology modules*

In data spaces, multiple domains converge with overlapping information requirements, necessitating a modular ontology structure. Decomposing complex domain semantics into manageable subgraphs — referred to as ontology modules [9] — improves maintainability, reusability, access control, and interdisciplinary collaboration, while enabling standardized linkage across heterogeneous

environments [32]. The resulting ontological topology is derived from the requirements established in the preceding step. Three module types are proposed:

Ontology domain modules represent domain-specific subgraphs developed collaboratively by domain experts. They define and refine concepts from existing sources, adapted to the data space context, while managing internal dependencies and alignment with shared vocabularies.

Ontology alignment modules act as bridges between domain ontologies, defining cross-domain dependencies and relationships, such as equivalence or similarity (e.g., via Simple Knowledge Organization System (SKOS) ontology). They support structured queries, interoperability, and governance functions including versioning and dependency management [20].

Ontology metamodules provide generic, administrative models applicable across all modules, defining lifecycle management guidelines, versioning, quality assurance, and modeling standards (e.g., via SHACL shapes). They enforce governance and compatibility across the data space and can elevate domain models to meta-level hierarchies where required (e.g., via Data Catalog Vocabulary (DCAT)).

Together, these three module types simplify ontology development, enhance cross-domain collaboration, and ensure long-term maintainability within the data space.

G. Step 7: Specification of alignment strategy

In this step, alignment strategies and alignment modules are individually defined. An alignment module specifies the links or mismatches between ontologies that need to be addressed. The specification should include documentation of conditions under which alignments were established, the use cases and requirements covered, and the stakeholders involved. Ideally, this information should be incorporated directly into the model itself.

There are several ways to detect and express ontology alignments [33]. The authors of [20] propose strategies for handling dependencies in distributed environments, defining a variety of relationship types. The authors of [32] introduce the concept of "linkset," describing dependencies and alignments between ontologies. As a lightweight approach, alignments are formulated as a triple structure between two ontologies, each as a subject and object. Thus, posing as simple way to specify alignments in a model-wise manner and as easy way to communicate between model domains. These approaches offer additional advantages when embedded in the data space context, such as reduced complexity, enhanced manageability, and extensibility. Expressing links between ontologies as independent, customizable concepts allows for further detailing or restriction, such as using metamodules for dependency management if required.

For each alignment module, the following elements must be specified, mapped, and documented: the dependencies between interfacing concepts and their relating domain ontologies; the involved stakeholders and notification requirements; the targeted ontology requirements; and the applicable circulation criteria, including temporal or event-based triggers and communication channels. Additionally, the collaboration type between domain developments must be

defined (e.g., separate, joint, or partially integrated workflows), alongside the alignment type (e.g., ontology merging, OWL axiom or SKOS mapping, custom translators, or shared vocabularies). Finally, the technical and IT requirements necessary to implement the specified alignment must be established, and detailed alignment specifications documented accordingly.

H. Step 8: Specification of ontology module development teams

Building on the previous steps, it is essential to establish cross-domain teams responsible for developing and managing the different ontology modules. This step should be conducted in parallel with defining the ontology modules and specifying alignments between domains. The collaborative principles guiding each module development team should be outlined to ensure integration into the overall data space collaboration framework while allowing flexibility for team-specific workflows. The following key aspects should be addressed:

1) Assignment of ontology modules

Map ontology modules to their respective requirements and assign them to the appropriate stakeholders or development teams.

2) Specification of requirements

Clearly define the requirements for each ontology module and document them for the development process.

3) Collaboration format

Allow each team to establish its own collaboration format, provided it aligns with the overarching collaboration framework for the data space (Step 3). Teams may choose formats, such as agile workflows, workshops, or distributed version control (e.g., Git-based workflows).

4) Circulation criteria for dependencies and alignments

Define how identified dependencies and alignments will be communicated and managed. Ensure these criteria are consistent with the agreed collaboration approach (Step 3) to facilitate smooth coordination across teams.

I. Step 9: Ontology module development

Once individual ontology modules and their collaboration frameworks have been defined, the modules are developed in parallel. Various procedural models for ontology development are available in the literature, such as the NeOn Methodology [7] and the Methontology Framework [13]. These models provide structured approaches for creating, managing, and iterating ontologies. The choice of method should align with the specific requirements of the domain and the overarching data space framework.

During development, the specified circulation criteria for interactions with alignment and metamodules must be adhered to. These criteria ensure that dependencies and alignments between modules are effectively managed. Alignment modules and linksets can also be used to monitor and control iterative progress across modules, ensuring consistency and interoperability.

New requirements arising during ontology module development should be carefully evaluated. Decisions must be made on whether to include these requirements in the use case backlog for future consideration or to address them

immediately through iterative alignment and updates to the module.

It is important to note that ontology module development is a highly domain-specific and individualized process. The complexity of the domain, the roles of the stakeholders, and the specific use cases will significantly influence the approach taken. Collaborative tools, versioning systems, and regular coordination meetings are often critical to ensuring progress and alignment across parallel developments.

J. Step 10: Documentation, publication and maintenance

The final step in the approach involves systematically documenting, publishing, and maintaining all developments. This encompasses every aspect of the process, from domain and stakeholder analysis to use case definitions, requirement derivations, and the specification and development of individual ontology modules. Documentation should be structured to provide a clear, traceable history of decisions, assumptions, and development artifacts. It should also outline next steps and anticipated challenges, referencing the use case backlog or requirements that have already been addressed and iterated upon. The documentation should answer key questions, such as:

- What has been defined? This includes key use cases, requirements, and assumptions.
- Who has participated? Identifying stakeholders, roles, and contributions.
- How has the work been conducted? Detailing collaboration formats, workflows, and processes.
- What are the scope, development horizon, and timeline of the ontology modules?

Publishing ensures that ontology modules and related artifacts are accessible, reusable, and transparent. Metadata must accompany the ontology, detailing its purpose, version, authorship, licensing, and usage instructions. To enhance credibility, validation results from tools, such as SHACL or OWL reasoners should be included to demonstrate correctness. Additionally, access permissions should be clearly defined, whether open, restricted, or tiered, depending on the collaboration model. A clear process for versioning and publishing updates is also essential to maintain consistency and compatibility over time.

Maintaining ontology modules is critical for ensuring their long-term functionality and relevance. This involves regularly reviewing the use case backlog to identify new requirements and evaluating change requests to decide whether they should be implemented. Alignment modules play a key role in managing the cascading effects of changes across modules, ensuring that dependencies are addressed effectively. Version control systems, such as Git, should be used to maintain clear versioning and compatibility tracking. Updates must be validated using semantic tools like Pellet, HermiT, or SHACL to ensure consistency and correctness, and all changes should be thoroughly documented to keep records of updated requirements and refinements. Lifecycle management processes should be established to retire outdated modules, introduce new ones, and manage transitions. Engaging stakeholders through regular feedback ensures continuous

improvement and alignment with the evolving needs of the data space.

IV. CONCLUSION AND FUTURE WORK

This paper identified key elements for collaborative ontology development in data spaces, analyzed semantic interoperability gaps, and derived challenges inadequately addressed by current methodologies. Building on existing approaches and practical experience, we proposed CODES — a structured ten-step methodology guiding stakeholders from domain analysis through collaborative ontology development via a modular, stakeholder-driven process applicable across heterogeneous organizational contexts.

CODES addresses the identified challenges through its modular approach: ontology domain and alignment modules decompose complex semantic graphs into negotiable subgraphs (C1); meta modules enforce consistent modeling guidelines and clarify roles and decision rights (C2); pre-defined, reusable modules lower participation barriers across expertise levels (C3); structured collaboration formats and standardized mapping procedures enable systematic integration while preserving domain autonomy (C4); and metamodules provide a foundation for machine-readable semantics and incremental adoption of shared data exchange mechanisms (C5).

Several aspects require further attention. Versioning, change management, and comprehensive documentation remain indispensable for long-term sustainability, and appropriate tooling for co-creation, alignment validation, and deployment must be developed. Future work will therefore focus on: empirical validation through real-world use cases with measurable metrics; development of a supporting tooling ecosystem; governance frameworks for managing modules across organizational boundaries; incentive mechanisms encouraging broader participation; and practitioner guidance in the form of a comprehensive handbook. These efforts aim to transform CODES into a proven, tool-supported methodology for data space initiatives.

REFERENCES

- [1] G. Solmaz *et al.*, "Enabling data spaces," in *Proceedings of the 1st International Workshop on Data Economy*, Rome, Italy, 2022, pp. 42–48.
- [2] E. Simperl and M. Luczak-Rösch, "Collaborative ontology engineering: a survey," *The Knowledge Engineering Review*, vol. 29, no. 1, pp. 101–131, 2014, doi: 10.1017/S0269888913000192.
- [3] A. Hutterer and B. Krumay, "The adoption of data spaces: Drivers toward federated data sharing," in *Proceedings of the 57th Hawaii International Conference on System Sciences (HICSS)*, 2024.
- [4] E. Curry, S. Scerri, and T. Tuikka, *Data Spaces: Design, Deployment and Future Directions*, 1st ed. Cham: Springer International Publishing, 2022.
- [5] L. Sánchez-González, A. Iglesias-Molina, O. Corcho, and M. Poveda-Villalón, "On the Governance of Semantic Artefacts in Dataspaces," *CEUR Workshop Proceedings*, vol. 3705, 2024.
- [6] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering: With examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, 1st ed. London: Imprint Springer; Springer London, 2004.

- [7] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology for Ontology Engineering," in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds., Berlin, Heidelberg: Springer Berlin, Heidelberg, 2012, pp. 9–34.
- [8] S. Copei and L. Rüllicke, "Best Practices to overcome challenges and barriers during the implementation of a data space for the energy domain: An experience report," *Data in brief*, vol. 61, p. 111838, 2025, doi: 10.1016/j.dib.2025.111838.
- [9] L. Nachabe, F.-Z. Hannou, M. Lefrançois, and M. Jubault, "Toward Agile Interaction Model based ontology development Methodology (AIME) for FAIR European data spaces," *FAIR Principles for Ontologies and Metadata in Knowledge Management (FOAM)*, 2024.
- [10] S. Steinbuss, A. H. Almeida, B. van den Wouter, and B. Sthefan Gabriel, "Semantic Interoperability in Data Spaces," Zenodo, Nov. 2025. doi:10.5281/zenodo.17630664.
- [11] M. S. Fox and M. Grüninger, "Ontologies for Enterprise Modelling," in *Research Report Esprit, Enterprise Engineering and Integration: Building International Consensus, Proceedings of the International Conference on Enterprise Integration and Modeling Technology, ICEIMT 1997, Torino, Italy, October 28-30, 1997*, K. Kosanke and J. G. Nell, Eds., Berlin, Heidelberg: Springer, 1997, pp. 190–200.
- [12] M. Uschold and M. King, "Towards a Methodology for Building Ontologies," *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, 1995.
- [13] M. Fernández López, A. Gómez-Pérez, and N. Juristo Juzgado, "METHONTOLOGY: From Ontological Art Towards Ontological Engineering," in *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*, 1997.
- [14] Y. Sure, S. Staab, and R. Studer, "Methodology for development and employment of ontology based knowledge management applications," *SIGMOD Rec.*, vol. 31, no. 4, pp. 18–23, 2002, doi: 10.1145/637411.637414.
- [15] Y. Sure, S. Staab, and R. Studer, "On-To-Knowledge Methodology (OTKM)," in *International Handbooks on Information Systems, Handbook on Ontologies.*, S. Staab and R. Studer, Eds.: Springer Berlin, Heidelberg, 2004, pp. 117–132.
- [16] H. S. Pinto, C. Tempich, and S. Staab, "Ontology Engineering and Evolution in a Distributed World Using DILIGENT," in *International Handbooks on Information Systems, Handbook on Ontologies*, S. Staab and R. Studer, Eds.: Springer, Berlin, Heidelberg, 2009, pp. 153–176.
- [17] K. Kotis and G. A. Vouros, "Human-centered ontology engineering: The HCOME methodology," *Knowl Inf Syst*, vol. 10, no. 1, pp. 109–131, 2006, doi: 10.1007/s10115-005-0227-4.
- [18] S. Braun, A. Schmidt, A. Walter, and V. Zacharias, "The ontology maturing approach for collaborative and work integrated ontology development: evaluation results and future directions," in *ESOE'07: Proceedings of the First Interational Conference on Emergent Semantics and Ontology Evolution*, L. L. Chen, p. Cudré-Mauroux, P. Haase, A. Hotho, and E. Ong, Eds., Aachen, Germany: CEUR-WS.org, 2007, pp. 5–18.
- [19] S. Auer and H. Herre, "RapidOWL — An Agile Knowledge Engineering Methodology," in *Lecture Notes in Computer Science*, vol. 4378, *Perspectives of Systems Informatics. PSI 2006.*, I. Virbitskaite and A. Voronkov, Eds., Berlin, Heidelberg: Springer, 2007, pp. 424–430.
- [20] E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi, "An Environment for Distributed Ontology Development Based on Dependency Management," in *Lecture Notes in Computer Science*, vol. 2870, *The Semantic Web - ISWC 2003*, D. Fensel, K. Sycara, and J. Mylopoulos, Eds., Berlin, Heidelberg: Springer, 2003, pp. 453–468.
- [21] M. Andresel, V. Siska, R. David, S. Schlarb, and A. Weißenfeld, "Adapting Ontology-based Data Access for Data Spaces," in *CEUR Workshop Proceedings, SDS@ESWC*, J. Theissen-Lipp, P. Colpaert, S. K. Sowe, E. Curry, and S. Decker, Eds.: CEUR-WS.org.
- [22] S. Bader *et al.*, "The International Data Spaces Information Model – An Ontology for Sovereign Exchange of Digital Content," in *Lecture Notes in Computer Science*, vol. 12507, *The Semantic Web – ISWC 2020*, J. Z. Pan *et al.*, Eds., Cham: Springer, 2020, pp. 176–192.
- [23] S. Auer, "Semantic Integration and Interoperability," in *Designing Data Spaces*, B. Otto, M. ten Hompel, and S. Wrobel, Eds., Cham: Springer, 2022, pp. 195–210.
- [24] R. Mies *et al.*, *Open source hardware development: A handbook for collaborative product creation*. Berlin: Berlin Universities Publishing, 2024.
- [25] E. Konietzko, C. Tanrikulu, F. Schwarz, and K. Lindow, "How to Gaia-X?," *Industrie 4.0 Management*, vol. 38, no. 6, pp. 54–58, 2022, doi: 10.30844/IM_22-6_54-58.
- [26] C. Tanrikulu, S. Gogineni, and K. Lindow, "Decentralized Data Spaces with Gaia-X," in *ProduktDaten Journal 1 2024*. Accessed: Apr. 9 2026. [Online]. Available: https://prostep.epaper-pro.org/pdjl-2024_english/#84
- [27] H. Berg, S. Gogineni, C. Tanrikulu, E. Konietzko, and K. Lindow, "Solution Approach for Asset Integration in Federated Ecosystems," in *Lecture Notes in Business Information Processing*, vol. 536, *Information Systems. EMCIS 2024*, M. Themistocleous, N. Bakas, G. Kokosalakis, and M. Papadaki, Eds., Cham: Springer, 2025, pp. 208–217.
- [28] J. Gessler, M. R. Cencic, C. Metzner, H. Wieker, K. Lindow, and W. H. Schulz, "Business models and organizational roles of data spaces: A framework for value creation in data ecosystems," *Data in brief*, vol. 61, p. 111795, 2025, doi: 10.1016/j.dib.2025.111795.
- [29] E. P. Konietzko and S. Gogineni, "Ontology Based Skill Matchmaking Between Contributors and Projects in Open Source Hardware," in *Communications in Computer and Information Science*, vol. 1789, *Metadata and Semantic Research. MTSR 2022*, E. Garoufallou and A. Vlachidis, Eds., Cham: Springer, 2023, pp. 14–25.
- [30] A. Cockburn, *Writing effective use cases*, 24th ed. Boston: Addison-Wesley, 2012.
- [31] K. Lindow, T. Riedelsheimer, P. Lünemann, and R. Stark, "Betrachtung des Entwicklungsumfeldes durch die methodische Datenflussanalyse," in *ProduktDaten Journal 2 2017*. [Online]. Available: <http://prostep.epaper.pro/journal-2017-02/de/#52>
- [32] R. Rik, S. Stol, and T. Mollema, "Ontology Matching trough alignment and extension: a Best Practice," Jun. 2025. Accessed: Apr. 9 2026. [Online]. Available: <https://docs.crow.nl/ontology-alignment/whitepaper/>
- [33] I. Osman, S. F. Pileggi, S. Ben Yahia, and G. Diallo, "An Alignment-Based Implementation of a Holistic Ontology Integration Method," *MethodsX*, vol. 8, p. 101460, 2021, doi: 10.1016/j.mex.2021.101460.