UNIVERSITY OF BAMBERG

Dissertation to obtain the academic degree of Dr. rer. nat. at the Faculty of Information Systems and Applied Computer Sciences

The Basin Network: A Model for Data Sharing and Exchange

Nasr Kasrin

Bamberg, 2023

URN: urn:nbn:de:bvb:473-irb-912696
DOI: https://doi.org/10.20378/irb-91269

This work was submitted as a dissertation to the Faculty of Faculty Information Systems and Applied Computer Sciences at the University of Bamberg.

Supervisor and reviewer: Prof. Dr. Daniela Nicklas
Second Reviewer: Prof.Dr.-Ing Bernhard Mitschang
Doctoral committee members: Prof. Dr. Andreas Henrich, Prof. Dr. Guido Wirtz

Date of Defense: 10 February, 2023

All things flourish without interruption.

They grow by themselves, and no one posses them.

Abstract

How can we facilitate the exchange of datasets and other data assets across heterogeneous technical and social environments? This is a classical problem that is more relevant today than ever before, relevant both on a global scale as well as the scale of organizations. We need to distinguish between two notions of data sharing and exchange: one based on the integration of data at the schema level, and the other, which is our focus here, is based on the description, publication, discovery, compilation of data *as an asset* (similar to how a book is managed in a library).

The state-of-the art of the data sharing and exchange problem is polarized. On one end we have emerging *'platforms'* which can be susceptible to become walled-gardens that do not interoperate with each other. On the other, we have *recommendations* such as FAIR guidelines, or architectural abstractions such as *data lake* and *data catalog* that help orchestrate vendors in developing interoperable technologies; taking us in the right direction but not too far, due to being too high-level and more normative/prescriptive than constructive. This work fills a gap by making a (1) 'medium-level' proposal, (2) in the form of a constructive recommendation (an architectural pattern), (3) which incorporates several major recommendations.

We propose the *Basin Network*, a distributed architectural pattern which revolves around two novel abstractions: the *Offering* (message exchange format) and the *Basin* (node). We adopt an *indirection* approach: a data asset is managed and exchanged via a *surrogate* (intermediate representation) which identifies, represents, describes, and effectively 'stands in' for it. The Offering is a model for such surrogates; built on the *URI-Resource* framework for hypermedia authoring but avoiding some of its pitfalls. The Basin is structure to author, catalog, and manage Offerings and exchange them with other Basins (publish/subscribe), resulting in a Basin *Network* (akin to a network of data lakes).

We demonstrate the applicability of the proposal by applying it to use-cases in three domains: a computer-aided manufacturing project, an IoT project in smart agriculture, and one in crowd data management.

Zusammenfassung

Wie können wir den Austausch von Datensätzen und anderen Datenbestände über heterogene technische und soziale Umgebungen erleichtern? Dies ist ein klassisches Problem, das heute aktueller denn je ist, die sowohl auf globaler Ebene relevant sind sowie auf einer Organisationsebene. Wir müssen jedoch zwischen zwei komplementären Interpretationen der gemeinsamen Nutzung und des Austauschs von Daten unterscheiden: Der eine basiert auf der Integration von Daten auf Schemaebene und der andere, auf den wir uns hier konzentrieren, basiert auf der Dokumentation, Veröffentlichung und Entdeckung *Daten als Aktivposten* (ähnlich wie ein Buch in einer Bibliothek verwaltet wird).

Der Stand der Technik beim Problem der gemeinsame Datennutzung und Datenaustausch ist polarisiert. Auf der einen Seite gibt es aufstrebende Plattformen dazu neigen, sich in "Walled Gardens" zu verwandeln, die nicht miteinander interagieren. Auf der anderen Seite gibt es *Empfehlungen* wie FAIR-Richtlinien oder Architekturabstraktionen wie *Data Lake* und *Data Catalog*, die dabei helfen, Anbieter bei der Entwicklung interoperabler Technologien zu koordinieren, was in die richtige Richtung geht, aber nicht zu weit. Dies liegt daran, dass sie zu allgemein gehalten und eher normativ/vorschreibend als konstruktiv sind. Diese Arbeit füllt eine Lücke, indem sie (1) einen Vorschlag „mittleren Niveaus" vorlegt, (2) in Form einer konstruktiven Empfehlung (einem Architekturmuster) und (3) mehrere wichtige Empfehlungen enthält.

Die vorliegende Arbeit führt das *Basin Network* ein. Es ist ein ein verteiltes Architekturmuster, das sich auf zwei grundlegende Abstraktionen stützt: das *Offering* (Nachrichtenaustauschformat) und das *Basin* (der Knoten). Der hier verfolgte Ansatz ist der der *Indirektion*: Ein Datenbestand wird als *Surrogat* verwaltet und ausgetauscht, der es identifiziert, repräsentiert, beschreibt und effektiv 'vertritt'. Das Offering ist ein Modell für solche Surrogate; baut auf dem *URI-Resource*-Framework für Hypermedia-Authoring, vermeidet jedoch einige seiner Fallstricke. Das Basin ist eine Struktur zum Erstellen, Katalogisieren und Verwalten von Offerings und zum Austauschen dieser mit anderen Basins (Publish/Subscribe), was zu einem Basin-*Netzwerk* führt.

Wir demonstrieren die Anwendbarkeit des Vorschlags, indem wir ihn auf Anwendungsfälle in drei Bereichen anwenden: ein Industrie 4.0-Projekt, ein IoT-Projekt in der intelligenten Landwirtschaft und eines im Crowd-Data-Management.

Acknowledgements

I want to thank my mother, Sanaa Salloum, and my sister, Zain Kasrin, for their boundless support, especially in the last three years of the Ph.D. which were the hardest and most critical. This work would be a shadow of what it is if it was not for your support.

I want to thank my supervisor Prof. Dr. Daniela Nicklas for all the support, guidance, and enthusiasm. I worked at the Chair of Mobile Systems during the first two-thirds of my doctoral studies. There I was involved in the SIMUTOOL project where I got my first taste of the data sharing and exchange problem. Although I was the only constant member in the project, I was not alone. I want to thank the revolving cast of members which have made valuable contributions to it. Thanks to the dean's office for doctoral affairs and the doctoral committee of my Ph.D.–Prof. Dr. Andreas Henrich, Prof. Dr. Guido Wirtz, and Prof. Dr.-Ing Bernhard Mitschang–for all their support and feedback.

To my colleagues, Aboubakr, who was welcoming and supportive to me when I first arrived at the chair, a time when I needed it the most, and Claudia, Michael, Simon, Stefan, and Golnaz, each one of you added a different color to my working day. To the '5th-floor WIAI WiMi gang:' Michael Siebers, Marcel Großmann, Martin Sticht, and a revolving cast of members, it was nice to have a group to 'talk geeky to' over coffee. Thanks to the Welcome Center at the University of Bamberg, my experience as an international post-grad was enriched by you, and thanks for selecting me for the DAAD STIBET scholarship. Thanks to the Trimberg Research Academy (TRAc) for choosing me for the DAAD IPID4all scholarship. Thanks to the city of Bamberg, the best part of the whole deal. And to the German University in Cairo (GUC) and the city of Cairo, both of which have changed my life.

To my friends, Rayan Al-Aghbari, thank you for your endless support and friendship. Khaled Tolba, my brother from another mother, Mohamed Badawy, thanks for all the laughs and the memories, Christine Rizkallah, it's always great to see you, thanks for all the feedback on my Ph.D. work. Martin, Isabel, and Jan, thank you for your friendship and support.

Nasr, 08 September, 2023

Bamberg, Germany

Contents

1 Introduction

The sharp empirical bend in science and technology has led us to an explosion in data volume, velocity, variety, veracity, and value. That with the continued exponential growth of digital interconnectedness naturally leads us to ask questions about *data sharing and exchange*. However, this problem is not new. Standardization bodies (ex., ISO[1], IETF[2]) have been–and still are–the pioneers in asking these questions since the start of the twentieth century, producing enough foundations for us to be able to talk about "data sharing and exchange" in such a casual tone and high level of abstraction and still be productive. However, even though it is not new, the scale and scope of the problem have exploded. The COVID-19 pandemic hit and only emphasized already pressing requirements regarding the timely and interoperable global data exchange. In addition, data sharing and exchange is not only a problem that is relevant on a global scale; corporations have always been in search of better ways to share and exchange data among each other. Hence, it is a classical problem with renewed interest that is relevant on both a global and a local scale.

Let us start with stating our research question. *How can we facilitate the managing and exchange of data assets across heterogeneous technical, social, and legal environments?*. Despite the importance of this question, there are few open, generalized, formal, technical, overarching *models* we can build on, which can provide us with an intellectual platform for technical, scientific, and design discussions or (dis)agreements. This study documents the efforts toward one such model.

Most similar work tends to fall near two ends of a spectrum. On one end, we have *highly general* approaches–which give us little in the way of such models–such as (1) simple concepts or analogies driven by industry buzzwords (ex., data lakes [HQJ21; SD20]), (2) high-level visions without many technical details or formal foundations (Data Spaces [FHM05]), or (3) guides/references/recommendations/standards (ex. FAIR principles [Wil+16], Data on the Web Best Practices[3]) that *prescribe* the desired properties of such a solution rather than construct one. On the other end, we have *highly specialized or application-dependent* approaches–which provide us with few reusable, general models–such as

[1] https://www.iso.org/home.html (Accessed 08.2022)
[2] https://www.ietf.org/ (Accessed 08.2022)
[3] https://www.w3.org/TR/dwbp/ (Accessed 08.2022)

research data repositories, research objects, data cataloging solutions, and many application-dependent software development solutions.

We define a data asset (*asset*, for short) as a set of one or more closely related datasets or dataset series, published or curated by a single agent, that together form a specific identity or fulfill a particular function[4]. Examples of data assets include binary objects (as in object stores), JSON documents, time series data produced by some sensor composed of periodically released subsets, relational data accessible via a query to some endpoint, and PDF documents available directly for download.

The term 'managing' in the research question above should be understood in the sense of 'data management,' which includes activities such as [data asset] governance, architecture, integration, interoperability, security, and quality [Int17]. Furthermore, we view the activities of management and exchange as *inseparable*: every act of management has exchange in mind, and every act of exchange has management in mind. Another duality exists between the *internal and external* scopes of management: the processes of managing and exchanging data assets within an organization and across organizations are functionally identical–*modulo visibility*.

We adopt an *indirection* [Nil10] approach to management, where an asset is managed and exchanged by managing and exchanging a '*surrogate*' which identifies, represents, describes, and effectively 'stands in' for it. We use the term *metadata* to refer to the 'stuff of surrogates.' We define metadata as *the structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, manage, and exchange a data asset* [NIS04].

Please note that whether we adopt an indirection approach or not, many of the problems studied here, such as metadata representation and interoperability–as well as questions such as the conception and 'artifactualization' of data resources–are common problems that need to be addressed nevertheless.

Borrowing Roy Fielding's conception of a URI-Resource [FT02, pp. 135], we define a surrogate as the "semantics of what the author [of the surrogate] intends to identify, rather than the value corresponding to those semantics at the time the [surrogate] is created." In this conception the identity of a surrogate is held together by a sort of *intentional definition* [Coo09, pp. 155] by the author of the surrogate. An intentional definition is a dictionary-type definition where the meaning of some concept is identified by specifying necessary and sufficient conditions for its identity. It is usually understood by contrast to an extentional definition, which provides a list of all instances in which the defined concept is applicable. For example, whereas an intentional definition of a bachelor is 'an unmarried man,' the extentional definition would be *a list* of all such men [Coo09]. Hence, one way to look at a surrogate

[4]This definition is borrowed from the definition of DatasetSeries in the up-and-coming Data Catalog Vocabulary (DCAT) Version 3 (W3C Working Draft 10 May 2022) (https://www.w3.org/TR/2022/WD-vocab-dcat-3-20220510/ (Accessed 08.2022)

is as an intentional definition of the scope of some data of interest, from the point of view of the surrogate's author.

We need to distinguish between two complementary senses of the phrase 'data sharing and exchange': one, the classical database notion, which is based on the integration of data on the schema level [DA12] and the other, influenced by library science [Gil08], is focused on facilitating the documentation, publication, discovery, compilation, access, and re-use of heterogeneous data. Here we are interested in the latter sense.

This study proposes the *Basin Network (BNet) model*. The BNet model is centered around two fundamental abstractions: the *Offering* and the *Basin*. The Offering is the surrogate of a data asset; influenced from one side by recent insights from the philosophy of the web literature regarding the two-decade-long confusion in the W3C community over the *ontological* status of the URI/Resource and from the other by the research data management literature. A basin is a structure to publish and manage offerings and exchange them with other basins, resulting in a basin '*network*.' The Basin abstraction is a generalization of existing abstractions such as data lakes, data spaces, and data catalogs, making the BNet model a kind of 'network of interacting data catalogs/lakes/spaces.'

From an overarching/infrastructural perspective, the BNet model can be seen as a super-process model for data sharing and exchange in the format of a distributed system, which includes a corresponding information interoperability architecture. It meets two prominent data sharing and exchange recommendations: (1) the FAIR (Findable, Accessible, Interoperable, Reusable) Guiding Principles for scientific data management and stewardship, and (2) the W3C's Data on the Web Best Practices.

An illustration of a simple basin network instance is provided in Figure 1.1. Basins are represented by circles, a party that can be an organization, a team, or a group is depicted as a triangle. Control flow from a party to a basin is represented by solid lines, and subscription contracts between two different basins, in which offerings can flow, are represented as directed, dashed arrows. Offerings are represented by greek symbols, with some shown as part of a set within a basin and others shown on the publication contract arrows, meaning that they are being exchanged between basins. In this example Party A is involved in two basins, one that it controls alone (B_1), for example to carry out internal workflows, and one it shares with another party (B_{12}), which is used for close data-driven collaborations between the involved parties. We can that this shared basin has subscriptions contracts with the two basins controlled by the parties that share this basin, this can be used for example when a clear separation between internal, private, or non-production ready data offerings on one hand, and offerings that need to be shared and form the basis for inter-group collaboration. There is a fourth basin (B_3) which includes two way publication contracts with the B_{12}, which entails a two way data-driven exchange, perhaps the offerings published to B_3 are for carrying out some data-driven activity, and feedback from executing this activity are published back into B_{12}, and so on.

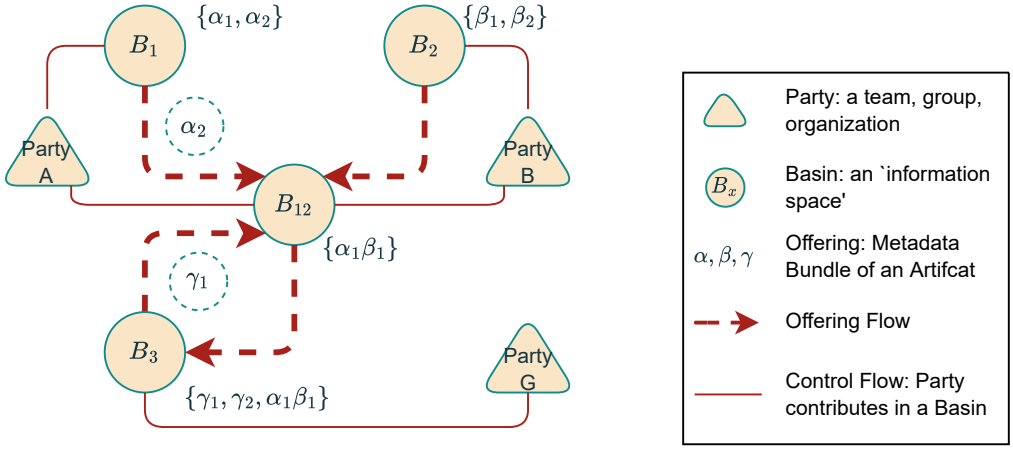


Figure 1.1: Elementary illustration of a basin network instance. A snapshot with four basins and three parties, with two offerings (α_2 & γ_1) published from one basin to another, and other offerings.

With regards to the scientific background of this work, we can divide our pursuit into two parts: *an investigation into metadata architecture* and *a search for an architecture to manage surrogates*.

Metadata architecture (*Metadata*, for short) 'Architecture' here should not be understood in the traditional sense of the word but in the sense of 'data architecture,' which includes "specifications used to describe existing state, define [meta]data requirements, guide [meta]data integration, and control [meta]data assets [...] at different levels of abstraction [...]" [Int17, pp. 98].

Architecture to manage surrogates (*Architecture*, for short) Architecture here is used in the traditional sense of the word as a "fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" [Sta11].

The main contributions of this work are:

- An original model, *the Basin Network model*[5], formalized in set-theoretic notation,
- A investigation of three architectures related to managing digital surrogates (linked data, data lakes/spaces, URI-web architecture),
- A study of key concepts of metadata representation and interoperability,
- Three use case studies in data sharing and exchange.

[5]The term '*basin*' in geology means a landform that dips inwards towards a central point [Jam14, pp. 241], and a '*basin network*' is a system where streams carry water through networks of basins [WM]

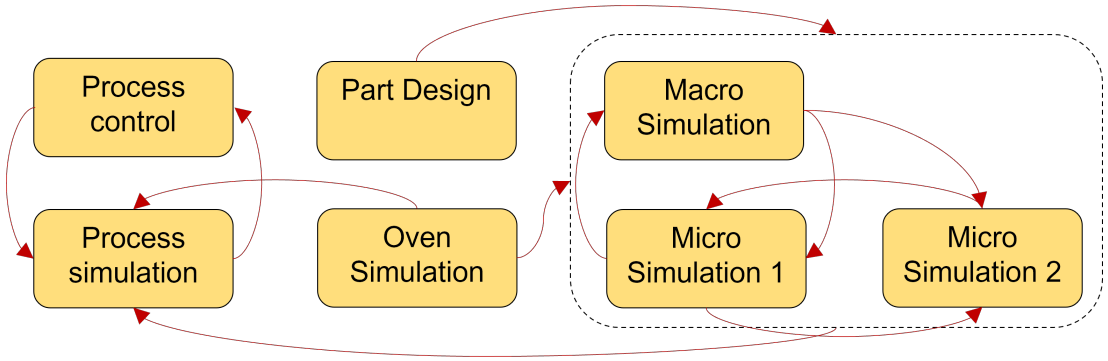


Figure 1.2: Partial example of asynchronous data exchanges between activities and groups.

The work presented here was developed over seven years in three phases. The first included a computer-aided manufacturing project (SIMUTOOL[6], HORIZON2020 project, grant agreement number 286717) consisting of 8 organizations spread across five countries, sharing and exchanging simulation and production data [Kas+18]. The second phase involved exploring the model in an IoT project consisting of 5 organizations sharing and exchanging smart farming data [Kas+21]. In the third phase, the insights and perspectives gained from the previous stages were used to guide the development of the proposal presented here.

1.1 Motivating Scenario

The scenario presented here comes from the domain of computer-aided manufacturing [Wu+15; KMT17; Kas+18; Ram+19; RPC19; Zie+21] in the aerospace and automotive industries in the project of SIMUTOOL[7], a HORIZON2020 project that ran from 2015 to 2019 (grant agreement number 286717). The subject of the project was the *integrated design, novel tooling, and process optimization of microwave processing of composites*. The goal was to further the state-of-the-art in the microwave processing of composites.

Figure 1.2 depicts a partial view of some asynchronous data exchanges between activities and groups in this domain. The approach included the use of several kinds of simulations at different levels of granularity and using various methods such as electromagnetic field simulations, heat transfer simulations, oven simulations (etc.), along with other activities traditionally associated with production technology research lifecycle such as measurements, process control, material development and tooling, prototyping, etc. The project involved partners from eight geographically distributed locations with different areas of specialization and scopes of confidentiality, privacy, and legality, which did not know each other beforehand. It required the organization of various couplings of data-driven cooperations

[6] https://cordis.europa.eu/project/id/680569 (Accessed 08.2022)
[7] https://cordis.europa.eu/project/id/680569

with dense networks of inter-dependencies to increase the turnover time of R&D activities and accumulate reusable data resources.

What is challenging in this domain is not the variety of data assets, but the interrelationships and data-driven dependencies across activities, particularly the simulation activities which are more data-drive. Revisiting Figure 1.2, we have a partial view of some asynchronous data exchanges between activities and groups. The *process control* activity produces production recipes (sometimes called 'curing cycles') on how to manufacture a part of family of similar parts. The process control is dependent on the *process simulation* of the part to be manufactured in the oven, and vice versa. The process control is also influenced by the *oven simulation* activity, which simulates the behavior of the microwave oven based on its power, size, and controls. The *part design* activity produces CAD/CAM models and other artifacts that specify the properties of the part to be produced. In as sense, most of the activities are driven by the data produced the part design, but the closest coupling is with the *micro- and macro-simulation* activities which focus on the modeling the behavior of the part in the whole production process, going from the micro-level (electro-magnetic / Maxwell equations), to macro-level simulation (heat coupling). Several of the previous activities are part of embedded cycles in which several activities closely cooperate on a problem until a stable result is reached. One such cycle is depicted as a dashed square around the micro- and macro-simulation activities. This domain will be studied in depth in Chapter 6. It will require a more complex basin network instance than the one presented in Figure 1.1. For a forward preview of how such a network might look like, the reader is invited to inspect Figure 6.4 in Section 6.3.3.

1.2 Thesis Overview

This thesis is structured as follows: in Chapter 2, we define our problem and its requirements. In Chapter 3, we analyze key concepts and trends related to metadata representation and interoperability, and in Chapter 4, we investigate the state-of-the-art of three relevant architectural aspects of the problem. In Chapter 5, we present the proposed Basin Network model. In Chapter 6, we demonstrate the applicability of the proposal by an in-depth study of the work within a multi-year inter-organizational data sharing and exchange in the computer-aided manufacturing project we where involved in (two additional, smaller, use-cases are provided in Appendices B and C). In Chapter 7, we present a qualitative evaluation of the proposal relative to related work. Finally, the thesis is concluded in Chapter 8.

2 Problem Definition

This chapter defines the problem of interest, research question, and requirements. It also presents the research methodology. It must be noted that historically, the problem definition presented here is the outcome of two inter-related activities: (1) A multi-year cyclical process over several projects in different domains in which pain points and requirements were extracted[1], and (2) literature review into common challenges and requirements for the problem area.

We start with a specification of the main problem of interest, the *Interorganizational Data Management and Exchange (IoDME)*, including the situation of interest, requirements and research question. Then, given the adopted approach (*indirection*: data managed via metadata), we derive an *[approach-dependent] sub-problem*, with its own requirements and research question. This two-tier approach helps us address the problem at two levels of abstraction: (1) the problem regardless of approach adopted, and (2) the problem with respect to a specific approach.

2.1 Problem: Interorganizational Data Management and Exchange

In this section we define the situation of interest, research question, and derive a set of requirements.

2.1.1 Situation of Interest and Research Question

We start with dictionary definitions and build up the situation.

Organization (Definition) The Oxford English Dictionary provides several senses of the term[2]:

- "An organized body of people with a particular purpose, as a business [...], etc."
- "The condition of being organized; systematic ordering or arrangement; spec. the way in which particular activities or institutions are organized. Frequently in social organization,"

[1] These use-cases are discussed in detail in Chapter 6 and Appendices B & C and briefly introduced in this chapter.
[2] "organization, n." OED Online. Oxford University Press, June 2022. Web. 7 July 2022.

- "Chiefly Biology. The development or coordination of parts (of the body, a body system, cell, etc.) in order to carry out vital functions; the condition of being or process of becoming organized (organized adj. 1). Also: the way in which a living thing is organized; the structure of (any part of) an organism"

We define an organization as *a 'gated,' systematically arranged organism with a particular purpose. It includes the following components: (1) a group of actors (humans and systems), (2) a set of activities (that produce and consume data), (3) declarative (know-what) knowledge [MM98], and (4) technical infrastructure.*

Gated (Definition) The adjective *gated* is borrowed as an analogy from the notion of 'Gated Communities' in urban design [GWF06] which refers to residential communities delineated by walls or fences with strictly controlled entrances. We define the term here as a set of conditions that must be satisfied for some organization, which includes legal, security, quality, and privacy conditions. A similar term is *walled garden*.

Take the computer aided manufacturing use case introduced in the previous chapter. Each activity in the environment (product design, oven simulations, etc.) is carried out by a gated organization, be it a large multi-national company, or a division or single team in a company. In fact one of the micro-simulation activities and the macro-simulation activity where carried out by two teams that work in the same multi-national company who never met before the project, due to working in different specializations and being stationed in different countries. So, one point to note about the definition of an organization given above, is that there is a recursive, 'fractal-like' emergent structure of what an organization is, although a legal entity with a single registered named is the typical image one has of what an organization is, it is usually decomposed hierarchically in smaller interacting organizations, which themselves might be decomposed further, and so on. The same pattern also exist in the other direction: two organizations collaborating on some end goal producing and consuming data as an single entity are also and organization, and so on.

Looking at the supporting technologies used across organizations, we observe that heterogeneity not only in data (variety), but in technologies and technical infrastructures, is the norm rather than the exception. An addition, even assuming homogeneous technologies are used, what differentiates one organization from another is its knowledge and information about its data–be it personal knowledge or organizational–as well as conventions. In addition, another key factor that differentiates two otherwise functionally identical organizations are the legal and privacy scopes in terms of data usage. Finally, the dispersion/movement of data assets across organization is anything but simple one-to-all broadcasting: there is a natural need for arbitrary and fine-grained control over the movement of data asset classes across organizations and in the public sphere. Given this introduction we can now define our situation of interest.

Situation of Interest The *Interorganizational Data Management and Exchange (IoDME)* is when we have[3]:

1. a variable set of organizations, **with**
2. disjoint legal and privacy scopes,
3. heterogeneous technologies, infrastructures, and data formats,
4. varying declarative knowledge (know-what) and technical conventions, and
5. arbitrary sharing and exchange configurations of data assets.

The scope of the situation definition above is still large, however. To specialize it further, we focus on specific goals: discoverability ('if you can not find it, it is not there'), persistability [of the reference & metadata], and data-driven collaborability. Furthermore, there is one value we prioritize in approaching this problem: minimum disruption to internal processes and technical infrastructures, because migrating internal systems is costly. Given that we can state our research question:

Research Question *How can we facilitate data sharing and exchange in an Interorganizational Data Management and Exchange situation, in terms of data-driven collaborability and with least disruption to existing internal processes and technical infrastructures?*

2.1.2 Requirements

To recall the computer-aided manufacturing scenario introduced in Chapter 1, the macro-simulation group used a commercial in-house software product. In contrast, the micro-simulation group built their simulations using popular tools such as MATLAB. Additionally, the two groups sometimes used different conventions and terminology when referring to the same concepts. Finally, each group came from different specializations; whereas one worked on electromagnetic simulations, the other used heat transfer functions. Hence we have the requirements for application domain neutrality and transparency in accommodating variable technological conventions. Moore has introduced the concept of infrastructure independence to encapsulate such concerns [Moo08]. We can state our first requirement.

Requirement 1 (Infrastructure Independent) *The proposal must be infrastructure neutral and accommodate heterogeneous infrastructures and technologies.*

There is a set of related issues of life-cycle management, annotation, and curation [FC15; Che+21] of data beyond the border of a single organization. Usually, every organization has local solutions to

[3] There are similar concepts in the literature such as *Information-Powered Collaborations (IPC)* [Tra+18], which focuses on research data management and is defined as "*complex, dynamic and heterogeneous environments that enable information sharing [...] from independently managed organizations [...] supporting knowledge and expertise exchange [...].*" [pp. 3]

these issues; however, what is needed is to address these issues in an interorganizational sphere, which forms our second requirement.

Requirement 2 (Supports Interorganizational Curation) *The proposal must support the curation and annotation of data in an Interorganizational sphere. This involves smaller problems such as provenance, quality, security, trust, compilation, and data discovery.*

Introduced in 2016, the FAIR (Findable, Accessible, Interoperable, Reusable) guiding principles for scientific data management and stewardship [Wil+16; Jac+20] are a popular set of principles aimed at guiding the design of systems for managing 'data assets,' with a focus on machine-actionability. Each of the four principles is usually divided into 3-5 sub-principles; a clear and straightforward presentation can be found on their website[4].

Requirement 3 (FAIR) *It must meet the FAIR principles.*

The *Data on the Web Best Practices*[5] (DWBP) is a W3C Recommendation published in January 2017, which can be considered a high-level data sharing protocol in the form of a list of best practices. The recommendation proposes 35 best practices that data sharing and exchange solutions should meet.

Requirement 4 (Meets W3C's DWBP) *It must meet the W3C's Data on the Web Best Practices*[6] *(DWBP).*

Being constructive around a core allows for more engineering-type structures to emerge across interoperable systems, clarity makes it easy to communicate and understand, transparency makes it easy to assess and analyze, and extendability allows the solution to develop gracefully[7]. As discussed in Chapter 1, there is a gap in the space of similar work a few open, generalized, formal, technical, *overarching models* we can build on to provide us with an intellectual platform for technical, scientific, and design discussions or (dis)agreements. Take several examples: data lake [HQJ21; SD20] is an intuitive and easily understood concept which does not provide enough agreement and technical detail to enable multiple parties to build interoperable systems. Data spaces [FHM05], provides some more information than data lakes (peer-reviewed paper vs. blog post), but not enough to create a technical or conceptual base for building interoperable systems. FAIR principles [Wil+16; Jac+20] provide an overarching view, but it is in the form of a small set of high-level requirements, which are high-level, normative instead of constructive or prescriptive, the same goes for Data on the Web Best practices[8], which is more details but inherits the same problems.

[4] https://www.go-fair.org/fair-principles/ (Accessed 11.2021)
[5] https://www.w3.org/TR/dwbp/ (Accessed 08.2022)
[6] https://www.w3.org/TR/dwbp/ (Accessed 08.2022)
[7] Rosemann [Ros21; OR21] who highlights the growing gap of incomprehensibility between intelligent systems and the ability of users to understand them. A similar dynamic is perhaps present between 'non-foundational' theories in the literature and their audience. This can hinder critical evaluation and lead to diverging interpretations or misunderstandings.
[8] https://www.w3.org/TR/dwbp/ (Accessed 08.2022)

Requirement 5 (Foundational) *The proposal must have a (i) constructive format with a first-principles-like core, that is easy to (ii) communicate, (iii) transparent, and (iv) extendable*[9].

Overall, the abovementioned requirements are not out of the ordinary, and programs with a similar goal usually share several of these requirements. Take the Report of the Interagency Working Group of the National Science and Technology Council [Dig09] for example, which identifies the following sub-problems: data innovation research (scalability, systems integration, design robustness), data discovery and dissemination (finding, understanding data), data protection (data security, privacy, confidentiality, and intellectual property rights), Data quality and disposition (data quality assessment and control, validation, authentication, provenance, and attribution), and integration and interoperability.

2.2 Sub-Problem: Indirection and Data Domain Topology Infrastructures

We will start with specifying the approach and defining some novel conceptualizations to structure it, which leads to asking an *approach-dependent research question (sub-question)* and a set of requirements that need to be met to answer it (*sub-requirements*).

2.2.1 Approach and Sub-Research Question

There are three key concepts that we will introduce, which helps us construct the sub-research question, we start with their definitions.

Indirection and Surrogates (Definition) We adopt an *indirection* approach to data management, where data is managed and exchanged by managing and exchanging a *surrogate* that identifies, represents, describes, and effectively 'stands-in' for it. With *meta-data* being the main content of a surrogate, defined as structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, manage, and exchange a data asset [NIS04]. Given the surrogate as a *basic building block*, we can introduce our next conceptualization.

[9] The term foundational is borrowed from the concept of *Foundationalism* in philosophy [Bla08], the reason for selecting the term is not due to some philosophical approach but because technical proposals which have a core/base are easy to communicate, evaluate, and extend.

Surrogate Management Space (Definition) A *surrogate management space ('surrogate space', for short)* is a gated or walled-garden (in terms of access and privacy) to manage surrogates [of data assets] that share a thematic goal or purpose. As we will see in the discussion of data lakes and data catalogs in Section 4.2, simple data lakes (with a single zone) and data catalogs can be understood as a type of surrogate spaces. A data catalog is a set of entries in which each is a surrogate of some data asset. A simple data lake with a single zone or pond is a structure to manage a set of heterogeneous data assets in a single place. However, so far, there is no mention of *exchanging surrogates across surrogate spaces* for which we will need another conceptualization.

Data Domain Topology (DDT) Infrastructure (Definition) A *data domain topology (DDT) infrastructure* is a set of interrelated technologies, standards, and design methodologies which enable the (1) managing of a set of two or more surrogate management spaces and (2) the exchange of surrogates between them. Just as a surrogate is the basic building block of a surrogate space, a surrogate space is the basic building block of a DDT infrastructure. Hence data lake approaches with multiple zones or ponds that include some notion of movement of surrogates across them are effectively data domain topology infrastructures. A data spaces approache is also a data domain topology infrastructure, and approaches that cover the exchange of surrogates across multiple data catalogs (ex., W3C's Data Catalog Vocabulary[10] recommendation) are a DDT infrastructure as well. Given the approach and above constructs we can now derive an *approach-Dependent* research question, or *sub-research question*.

Sub-Research Question *What is a data domain topology (DDT) infrastructure that meets the requirements of the main problem?*

2.2.2 Sub-Requirements

Given the sub-question, we can now map our requirements to sub-requirements.

Sub-Requirement 1 (Arbitrary Space Membership Condition) *There should be no constraints on the conditions that surrogates must meet to be encapsulated in some Space.*

Sub-Requirement 2 (Arbitrary Inter-Space Exchange) *It should be possible, in principle, to exchange surrogates between any two spaces.*

Sub-Requirement 3 (Arbitrary No. of Spaces) *There should be no upper limit on the number of Spaces.*

Sub-Requirement 4 (Privacy) *There should be a mechanism to forbid the propagation of surrogates to spaces that are not designated audiences.*

[10] https://www.w3.org/TR/vocab-dcat/ (Accessed on 27 August 2021)

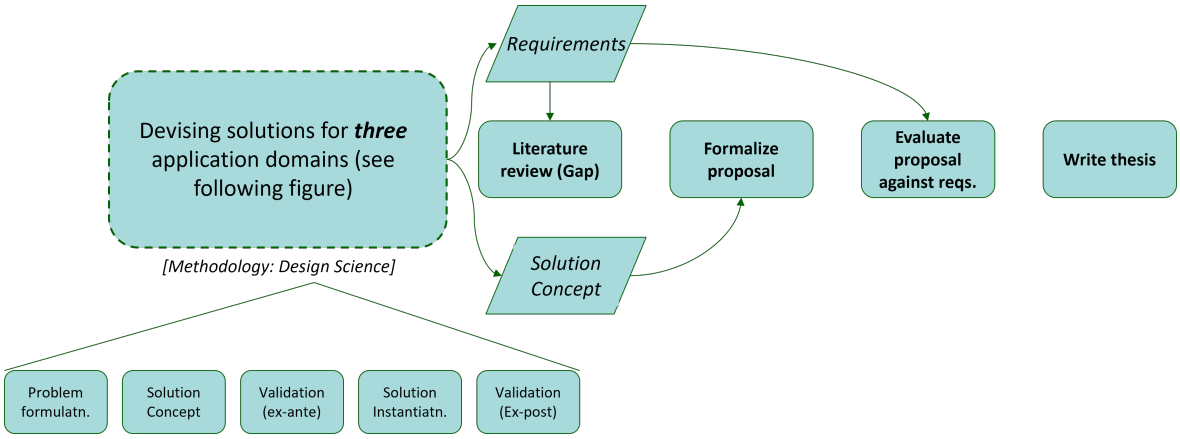


Figure 2.1: High-level view of the process of the research project that led to this work.

Sub-Requirement 5 (Inter-Space Curation) *It should support the encoding and exchange of information about surrogates across multiple spaces.*

Sub-Requirement 6 (Semantic Heterogeneity) *It should allow for some degree of semantic heterogeneity regarding the descriptive/contextual information in surrogates.*

2.3 Methodology and Process

The overacrhing methodology and process started with a 6 year phase which went through 3 cycles where each devised solution on some project in some domain, this phase followed the design science research. This phase was resulted in a set the requirements, as well as a solution concept. It was followed by a 2 year phase of the traditional process of:literature review and gap analysis, followed by developing the proposal and then evaluating against the identified requirements, and writing up the dissertation.

A high-level overview of this process is depicted in Figure 2.1. The first phase, covering two thirds of the total time-frame, included devising solutions for three successive applications domains, computer-aided manufacturing (see Chapter 6), the Internet of Things (see Appendix B), and mobility data management (see Appendix C). The methodology carried out in each application use-case was design science (see below for more details). At the conclusion of this phase, a stable set of requirements were derived as well as a solution concept (see the this sections above of this chapter). Using the requirements a literature review was carried out which identified a gap (Chapters 3 and 4). This led to take the solution concept and generalize it in order to formalize a proposed model (Chapter 5), followed by evaluating the proposal (Chapter 7), and then writing up the dissertation.

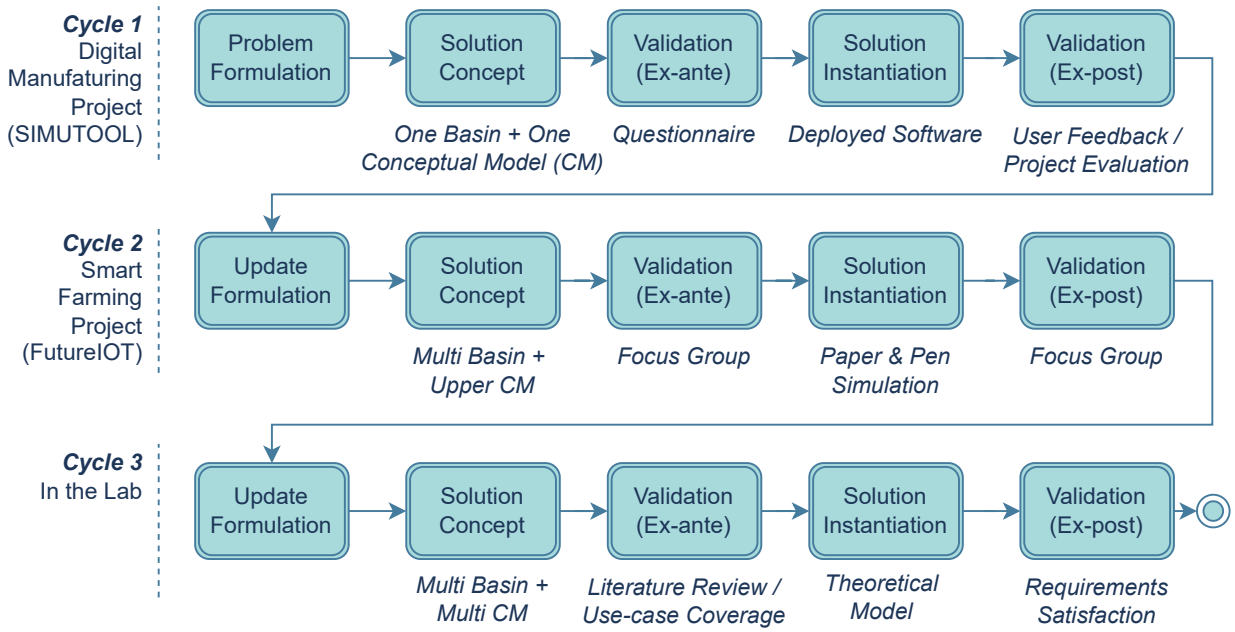


Figure 2.2: Design science-related stages of the research (Adapted from [TVY90; SO11])

With regards to the first phase, the methodology followed in this work was *design science research* [Hev+04], along with some *interpretive research*. Whereas interpretive research involves observations that seek to describe, explain, and predict phenomena, design science research involves actions to intervene and alter these phenomena [LH09], it is primarily about the *creation of possible futures*. One way to understand the design science methodology is as a *search process that involves an iterative design-evaluate loop*.

In the early phases of the work, more interpretive research was carried out to understand the domain and build insights. These two methodologies were interleaved as the project progressed, where interpretive research informs design research. The whole process can be seen as several iterations over different conceptions of the work along with different environments.

A high-level process, adapted from [TVY90; SO11], of the major stages taken in this research is depicted in Figure 2.2. The project consisted in three major cycles, with a similar structure but different subjects and methodologies in the respective sub-activities. We discuss each of these below.

The process started with a problem formulation, this was based on experience observing and taking part of the project or from earlier cycles. This was followed by a formulation of a solution concept (for example, where as in the first cycle the concept was a single repository of data shared by all organizations, the solution concept in the second cycle was expanded to have a repository per group with exchanges between repositories, and the third solution conception, expanded further allowing different groups to use heterogeneous conceptual models for metadata. In the This stage was followed by the ex-ante validation activity. In the first cycle it was a questionnaire, whereas in the second it was

a focus group discussion, and in the third was a literature review and use-case coverage. The fourth activity comprised the instantiation of the solution concept. Whereas in the firs cycle it was a software suite, in the second it was a paper-and-pen simulation, and the third it was a theoretic formal model. The last activity was the ex-post validation, which was user-feedback and project evaluation feedback in the first cycle, and focus groups and peer reviews of the published work in the second, the third was about the requirements satisfaction by the model.

3 Metadata Interoperability

The goal on this chapter is to lay some foundations with regards to the inter-organizational curation requirement (Req. 2 and Sub-Req. 5) and the semantic heterogeneity sub-requirement (Sub-Req. 6). Basically, due to the metadata-driven indirection approach, one of the key elements to facilitate curation and its interoperability, is to facilitate the movement and translation of metadata across heterogeneous environments in terms of semantics, syntax and systems. Hence, in this chapter, we focus on the topics of metadata *representation* and *interoperability*, with a '*(meta)data over the web*' focus. We define the term *metadata* as *the structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, manage, and exchange a data asset* [NIS04]. We select 'semantic data management' as a representative metadata technology to investigate. This is not an unrealistic choice since many of the practices associated with 'semantic data management' in the past three decades–such as ontologies and vocabularies–have become *defacto* standards for metadata representation and interoperability over the web, regardless of technology or approach.

This chapter is structured as follows. In Section 3.1, we start with an introduction, and in Section 3.2, we present a metadata typography for the domain of data sharing and exchange. In Section 3.3 we discuss the intertwined histories and issues related to data and domain modeling [of metadata]. In Section 3.4 we present an approach for data model *modeling*, and in Section 3.5 we present an approach for domain model *modeling*. In Section 3.6 we discuss two additional tools relevant to metadata management, and in Section 3.7 we present approaches to metadata interoperability. Finally, we conclude the chapter in Section 3.8.

3.1 Introduction

Metadata is informally defined as 'data about data.' However, it helps to look at how metadata has been understood in use by different communities. We start with presenting different conceptions and then discuss different aspects of the subject.

Gilliland gives a historical perspective of metadata across different communities [Gil08]:

- *"For [data management and systems design and maintenance] communities, metadata refer[s] to a suite of industry or disciplinary standards as well as additional internal and external documentation and other data necessary for the identification, representation, interoperability, technical management, performance, and use of data contained in an information system."* [pp. 1]
- *"[Cultural heritage communities] often apply the term metadata to the value-added information that they create to arrange, describe, track, and otherwise enhance access to information objects and the physical collections related to those objects."* [pp. 2]
- *"Library metadata includes indexes, abstracts, and bibliographic records created according to cataloging rules (data content standards)[...]"* [pp. 2]

Metadata, as understood in this study, builds on all three concepts of metadata presented above.

From a structural standpoint, there are two types of metadata: that *embedded within* an asset and that *separate* from it. Examples of the former include properties of a PDF document (creator, date of creation), MPEG-7[1] document, and data used to manage the contents of a relational database (indexes, definitions of columns, and tables). Examples of the latter include data catalogs, reference and master data in enterprise systems, and metadata used to manage heterogeneous data resources, as in Data Lakes, Data Spaces, and Data Grids. Here we are interested in the latter sense, where metadata is managed separately from the 'data.' Finally, we can distinguish between two degrees of 'separateness:' (1) metadata in representation/formats *distinct* form the 'data' but *colocated* with it (ex., MPEG-7) and (2) metadata in *distinct* representation/format and *separated* from the data (ex., data catalogs).

The principal value proposition of managing metadata of assets (as opposed to the assets themselves) is that it enables systems, applications, and users to manage assets without direct interaction with them: a phenomenon known as *indirection* [LS07; Nil10]. This indirection is the cause of this approach's key strengths and weaknesses. When done correctly, it allows for a separation of (technical) concerns as well as the ability to scale (out and up) without material barriers (logistical, operational, etc.). However, when unsuccessful, metadata can become detached from 'reality' and create new problems without solving the original ones.

The Metadata Standards Directory WG of the Research Data Alliance (RDA) adopts the notion that the only difference between metadata and data is in the mode of use [Dat14]. Here are some examples to demonstrate this perspective.

A library catalog card, when used by a researcher to find a scholarly paper, is metadata. However, when used by a librarian to count articles on river pollution, it is data [Qui+20]. Take this example from sensor data management. A sensor's readings to an entity recognition system are data; the readings

[1] https://mpeg.chiariglione.org/standards/mpeg-7 (Accessed 07.2022)

of a sensor used by an automated system to correct the readings of another sensor are metadata. And an example from the Semantic Web sphere. A dataset represented in the Resource Description Framework[2] (RDF) is metadata when it describes some web resource, and within the context of a data catalog, it is data.

Once we consider the management and exchange of metadata in a heterogeneous distributed environment, the question of information interoperability comes to the forefront. There are four levels of heterogeneity hindering information system interoperability [OS99]:

- *System*: hardware / software heterogeneity.
- *Syntax*: heterogeneity in encodings and representation.
- *Structural*: variations in data models, data structures, and schema.
- *Semantic*: inconsistencies in terminologies and meanings.

In this chapter we focus on syntax, structural, and semantic interoperability levels.

3.2 A Metadata Typography

There are a couple of typographies of metadata in the literature [Tay99; Ora15; Ril17; RZ19; Saw+19; SD20; Qui+20] [ZQ16, pp. 45] [Int17, pp. 422-424], below we synthesize several of them in the context of the management and exchange of data to give us a rough guide for the types of metadata we should consider.

- *Descriptive* metadata. About the content of a resource. Includes title, description, keywords, and contact points (creator, author, editor). Used for discovery and curation purposes.
  - *Identifying* metadata: titles, dates of publication or distribution, languages, identifiers
  - *Content* metadata: subjects, coverage, tables of contents, classification notations, categories.
- *Administrative* metadata. Information for supporting the management and organization of the resources.
  - *Technical* metadata: version control, formats, encoding
  - *Rights & access* metadata: ownership, permissions, usage restrictions, rights, access categories, copyright, license, rights holders, terms and conditions, periods of availability, and payment options. The Open Digital Rights Language[3] (ODRL) is an example policy expression language for rights & access metadata.

[2] https://www.w3.org/RDF/ (Accessed 07.2022)
[3] https://www.w3.org/TR/odrl-model/ (Accessed 07.2022)

- *Provenance* metadata: lineage, why, by whom, and in which context
- *Operational* metadata: API access specification, resource location, reports, and query access patterns, service-level agreements, requirements and provisions, data sharing rules and agreements
- *Authentication, integrity, & security* metadata: encryption, integrity information.
- *Global* metadata: overarching data, such as indexes, logs, and semantic resources, such as vocabularies.

- *Structural* metadata
  - *Intra-structural* metadata. Structural information of a resource: document types and their structures, object behaviors or functionality, aggregation of items, lists, and parts.
  - *Inter-structural* metadata. Relations between resources: groupings, similarity, containment, partial overlap, logical clusters of related resources.

This listing gives us an impression of the various subjects of metadata elements in our problem domain. We will return to this issue when presenting our proposal in Chapter 5.

3.3 On Data and Domain Modeling

In this section, we provide a historical background into the activities of data modeling and domain modeling and their interrelations.

3.3.1 The 'Semantic' Data Model

The topic of 'semantic' data management is intimately and intricately related to the issues of metadata management, metadata standards, and interoperability which we are interested in here; hence we start our discussion with it.

'Semantic data modeling' is usually understood in contrast to the classical relational, hierarchical, binary data modeling approaches. The motivation is that the low-level, machine-oriented representations of the models mentioned above do not reflect the structure of how humans intuitively perceive the world. This creates a *discrepancy* (and discontinuity) between human perception and conceptualization of the world, and low-level machine data structures that represent it, which results in a 'schism' between conceptual models/semantics on the one hand, and schema/syntax on the other [HM78]. This results for example in causing the data models and queries to loose their conceptual naturalness [W81; Jag+88] [KS95, chap. 5].

The degree of how 'rich' a semantic data model is can vary and has been stretched in the last 2-3 decades to include simpler models such as association lists. An example from the middle of the spectrum is the Resource Description Framework[4] (RDF) which models data as a set of subject-relation-object triples.The Web Ontology Language[5] (OWL) adds more expressiveness on top of RDF as well as some reasoning and model verification capabilities. On the far end of the spectrum, we have fully-fledged logic-based knowledge representation and reasoning systems that allow for deductions and theorem proving. Semantic models within the metadata on the web spheres usually refer to the 'lighter side' of the spectrum.

Although the details differ, many 'light' semantic data models have common features. A set of *classes* are defined, usually each with a distinct set of properties. Instances are (extensionally or intentionally) organized into classes. Inheritance relations between classes (*type-of*) are usually supported, too. Furthermore, to make factual statements about the world, semantic data models allow for a notion of stating properties of instances, usually in the form of triples[6] <instance-x, property-name, property-value>. Properties usually cover simple attributes (ex., date of birth) or object relationships (father of). In the case of simple attributes, the property value is a string of characters or numbers. In object relationships, the value refers to another instance [HM78; STW84].

It is helpful to note that, from a database model perspective, the semantic data model is best understood as a higher-level data model (abstraction), meaning that it can be implemented on top of simpler data models, some of which can be the classical data models themselves [Bor78; HM78].

So far, we have only discussed the data modeling side. However, one of the most exciting developments in the past decade in the practice of semantic data management on the web has not been on the data modeling side, *but on the 'domain modeling' side*. *Knowledge Organization System (KOS)* [LEI08; Soe09] is an umbrella term for systems and artifacts that play a similar role to *domain models* but are usually represented in standardized formats or technologies (such as RDF), hence facilitating exchange and adoption (we discuss KOSs in Section 3.5). In specific *the community and social practices* of creating, sharing, and re-using KOSs have been steadily growing in the last decade.

However, the story is not over yet. The typical usage pattern of KOSs across metadata applications has not revolved around a single KOS that all applications adhere to (a so-called canonical model or global ontology [UG04]), but the selective reuse and combining of elements from different KOSs to fulfill the pragmatic information representation needs of applications, which we can sum up in the following catch-all phrase: *'laissez-faire element-mix-and-match free-form metadata semantics'*. Take data.world [7] for example, which is a data cataloging and hosting service. A simple inspection of the metadata of

[4] https://www.w3.org/RDF/ (Accessed 07.2022)
[5] https://www.w3.org/TR/2012/REC-owl2-primer-20121211/ (Accessed 07.2022)
[6] Probably influenced by the Entity–attribute–value database model.
[7] https://data.world/ (Accessed on 30 September 2021)

some published datasets displays properties imported from around 4-6 KOSs, such as CSVW, Dublin Core (DCMI), DCAT, and others.

This was complicated further by the emergence of multiple standards and languages with *incompatible abstract models and syntaxes* for representing metadata. We have two majors 'camps:' those built on the XML tree structure such as the *Metadata Object Description Schema*[8] (MODS), and those built around the Entity-relationship structure such as DCMI and RDF [9] [Nil10]. This naturally complicates the issues of metadata interoperability.

More recently, since the rise of Neo4j[10] and its corresponding property graph model, we have witnessed another data model 'war' between the RDF model and the property graph model. Although the abstract model incompatibility gulf is arguably smaller[11], there are still challenges that need to be resolved, with some efforts trying to define a 'compatibility subset' over both abstract models. However, the problem remains open [Las+21].

Another key trend in semantic data management in the past three decades has been separating data modeling and domain modeling activities. We turn our attention to this issue below.

3.3.2 On Separating Data & Domain Modeling

In the (pre-web) past, the developments of the different sub-components of a knowledge representation and reasoning system were intimately co-dependent. These included (1) representation languages, (2) factual assertions (knowledge bases), and (3) conceptual models (axioms and theories). Take the influential Cyc project [Len95] as a demonstration. The group behind Cyc developed their own representation language (CycL) [LG91], ontology, knowledge base, and reasoning engine [LG89].

However, within the data-on-the-web sphere, this co-dependency has been minimized in the past three decades. This is due, in part, to the rise of specialized and reliable *data modeling frameworks* such as the Extensible Markup Language[12] (XML), the Resource Description Framework (RDF), JSON[13], and JSON-LD[14]. These frameworks included rich toolsets to handle many data modeling requirements such as schemas (XML schema, RDF Schema, JSON Schema), as well as constraint definitions (as simple

[8] http://www.loc.gov/standards/mods/ (Accessed 07.2022)

[9] Although we can technically distinguish between DCMI which is a metadata element set for the domain of library science and RDF which is merely a data model for knowledge representation and reasoning on the web, both have underlying abstract data models nevertheless and in that sense are similar. We will come back to this issue in the following section.

[10] https://neo4j.com/ (Accessed 08.2022)

[11] Qualified relations in the property graph model have no direct analog in RDF. Named graphs come close but are not meant to be used on a per-triple basis.

[12] https://www.w3.org/XML/ (Accessed 07.2022)

[13] https://www.json.org/json-en.html (Accessed 07.2022)

[14] https://www.w3.org/TR/json-ld/ (Accessed 07.2022)

as XSD Types or as sophisticated as OWL ontologies[15] for model validation), and serialization formats (Turtle[16] & RDF/XML[17] for RDF, UBJSON[18] & JSON→URL[19] for JSON), and so on.

With the availability of data modeling frameworks, people developing conceptual frameworks that model some aspects of the world were freed from the burden of developing a corresponding data model and could focus on building their artifacts, usually with some data modeling framework in mind. However, this separation produced some side effects.

The first challenge is what we will call the *data/domain model modality proliferation phenomenon*. With the separation of modeling concerns, many implicit assumptions, ontological commitments, and logical constraints need to be communicated and documented across people that never met; however, due to a lack of clear understanding regarding *where and how* these concerns are communicated, groups usually develop their own (sometimes esoteric) conceptual structure of recommendations, what is one man's data model can include elements of another man's abstract or formal model. For example, the deliverables of W3C's RDFCore Working Group[20] or W3C's Provenance Working Group[21], include various sub-components covering different perspectives to the core proposal, such as data model, notation, serialization(s), mathematical models, constraints, bindings, mappings, etc. From the analogy of multi-modality in human perception (vision, auditory, taste, touch, etc.), we can view these sub-components as different 'modalities of models' that try to encode various aspects of the core proposal.

The second challenge, already hinted at in the previous section, is the *abstract data model incompatibility challenge* (chiefly the incompatibility between hierarchical-based data models and triple-based ones, see below for an example). Studied by Nilsson [NPB03; NN10; Nil10], this challenge is about the problems associated with harmonizing metadata standards built on incompatible abstract data models. This problem can be demonstrated by a simple example [Nil10]. The Learning Objects Metadata (LOM) developed by the IEEE Learning Technology Standards Committee[22] (TSC) has an abstract model influenced by the hierarchical structure of *elements-within-elements abstract model* influenced by XML. For example, the element '*language*' means different things based on whether it is located in the *General* category or the *Educational* category. It is not clear how such an ambiguity can be resolved if the data is mapped into the *subject-relation-object abstract model* of RDF or the similar, yet distinct *abstract model of Dublin Core* (DCMI) [Nil10, pp. 41-42]. Another example is XML attributes (as opposed to elements) in XML-based models; there is no universal mapping from those to the subject-relation-object model.

[15] https://www.w3.org/OWL/ (Accessed 07.2022)
[16] https://www.w3.org/TeamSubmission/turtle/ (Accessed 07.2022)
[17] https://www.w3.org/TR/rdf-syntax-grammar/ (Accessed 07.2022)
[18] https://ubjson.org/ (Accessed 07.2022)
[19] https://jsonurl.org/ (Accessed 07.2022)
[20] https://www.w3.org/2001/sw/RDFCore/ (Accessed 07.2022)
[21] https://www.w3.org/2011/prov/wiki/Main_Page (Accessed 07.2022)
[22] https://ieee-sa.imeetcentral.com/ltsc/ (Accessed 07.2022)

A metadata interoperability system should be able to represent such subtleties as data/domain model modalities and compatibility issues between data models based on incompatible abstract data models. In what follows, we separate data from domain model modeling and approaches to model each activity.

3.4 Modeling Data Models

Here we are interested in a structure to encode properties of data models, in other words, to *model* data models for interoperability. From a database and information systems interoperability perspective, the classical view of the 1975 ANSI/X3/SPARC report identifies three kinds of models [Ste75]: *external data models* specify how to encode data in a specific data representation format or technology targeted to some application or system. Examples include a row in a relational in RDBMS and a logical record in COBOL. *Conceptual data models* represent the entities in the domain of interest and their properties in a system in a single source-of-truth representation layer in a DBMS, and *internal [physical] data models* specify how both the above types of data models are encoded and stored internally on disk.

Although this model is very different than what we are looking for here, we can get some inspiration from it. With regards to modeling data models, we have three layers:

1. *Abstract Data Model Model (Abstract DMM)* describes elements and structures of a data model and their 'semantics.' It can also encode axiomatic/ontological commitments. Examples include the DCMI abstract model and the XML abstract model.
2. *Logical Data Model Model (Logical DMM)* specifies a language (syntax and semantics) to represent data in some technology-specific manner. Examples include the SQL syntax, the RDF model, and the XML model.
3. *Physical Data Model Model (Physical DMM)* specifies the format and structure of objects containing the stored data. They are usually optimized for one or more purposes, such as transmission, storage, etc. We can think of them as *models of serialization formats* (in the web context) or internal table storage formats (in the DBMS context). Examples include JSON, Turtle TTL[23], and BSON[24].

We call this proposed hierarchy *the Data Model Modeling (DMM) Hierarchy*[25]. Table 3.1 demonstrates this hierarchy with several data models (content extracted from discussions in [Nil10]). Please note that

[23] https://www.w3.org/TeamSubmission/turtle/ (Accessed 07.2022)

[24] https://bsonspec.org/ (Accessed 07.2022)

[25] The DMM hierarchy shares similarity with the Meta Object Facility (MOF) by the Object Management Group by having multiple levels with progressive levels of abstractions, however the MOF is concerned with modeling data objects within an object-oriented software engineering environment (i.e., objects and classes), whereas the DMM is concerned with modeling data models for the purposes of mapping across each other and interoperability. From a formal standpoint, the Abstract DMM roughly corresponds to M3 & M2 (assumption about the domain–such as OO and classes–is non-existent in the DMM hierarchy), and the the two lower levels correspond to the two lower DMMs.

Table 3.1: Demonstrating the DMM hierarchy on several data models and metadata standards (Content from [Nil10]).

DM \ *Layer*	*Abstract DMM*	*Logical DMM*	*Physical DMM*
RDF	RDF abstract data model	RDF/XML syntax	TTL, JSON-LD
DCMI	DCMI abstract model	Various / DCMI RDF syntax	TTL, JSON-LD
XML	XML abstract data model	XML syntax	XML
MODS	Adopts XML abstract data model	MODS XML syntax	XML
LOM	LOM abstract data model	LOM XML syntax	XML

the table includes generic data modeling languages (RDF, XML) alongside domain-specific metadata standards. Although they are ontologically distinct, they share the property of having an (implicit or explicit) abstract data model and one or more Logical and Physical DMMs.

We will use the DMM Hierarchy in the metadata interoperability component of the proposed work in Chapter 5. We now turn from data model modeling to domain model modeling.

3.5 Modeling Domain Models

As opposed to data modeling, which is concerned with the syntax and semantics of the models that encode data, we have another class of concerns, usually called domain modeling. We will adopt the notion of *Knowledge Organization Systems* (KOS), which conveniently includes domain modeling and other relevant activities. Below are some definitions from the literature.

"*The term knowledge organization systems is intended to encompass all types of schemes for organizing information and promoting knowledge management [...] Knowledge organization systems are used to organize materials for the purpose of retrieval and to manage a collection.*" [Hod00, pp. 3].

"*These systems model the underlying semantic structure of a domain and provide semantics, navigation, and translation through labels, definitions, typing, relationships, and properties for concepts [...] Embodied as (Web) services [...]*" [LEI08, pp. 1]

"*Prototypically, a KOS provides a framework or schema for storing and organizing data, information, knowledge about the world [...]*" [Soe09, pp. 3]

"*A concept is the basic structural element of the knowledge organization system. A vocabulary, that is, the formal expression of concepts, forms the core of the physical representation of each knowledge organization system. The vocabulary is utilized to express the semantics and the syntax of the organized whole, or, as the case may be, the rules defining how a structure is to be used.*"[BK14, pp. 8-9]

As we can see, the definitions are varied. However, the utility of the term KOS is not in its precision but in acting as an umbrella term for related concepts and principles that would otherwise not be discussed under one roof.

Several typographies and classifications of KOSs exist [Hod00; LEI08; ZM19], some that organizes them by function [CZ06], and some by level-of-abstraction [HP20]. In what follows, we propose a classification for each approach.

3.5.1 Knowledge Organization Systems by Function

There are two functions that KOSs can play: one is defining metadata elements, their properties, and interrelations (Linked Open Data vocabularies (LOV)), and the other is for encoding possible values of these elements, properties, and interrelations (Subject value vocabularies / Reference Data).

Linked Open Data vocabularies (LOV): LOVs are what most Linked Data practitioners usually refer to when they say '*vocabulary*.' These are element sets or property vocabularies published using *some or all* of the Linked Data best practices for publishing RDF Vocabularies[26]. These typically include classes (types), attributes (per class), and relations between classes [ZM19]. Examples include *Dublin Core and DCMI Metadata Terms* and most KOSs used in semantic data management (see Top-Domain & Domain KOS discussion below, for example). For vocabularies not in a Linked Data-ready format, bindings or portings into a LOV are usually produced (using tools such as SKOS[27], see 'meta-KOS tools' below).

Subject value vocabularies / Reference Data: Defines resources to be used as values in metadata records, usually covering subjects, themes, or taxonomies.

- *Specialized Authority lists*: MARC authority control standards, DCMI Type Vocabulary
- *Classification Systems*: Dewey Decimal Classification (DDC)
- *Subject headings/subject authority files*: Library of Congress Subject Headings (LCSH)
- *Thesauri*: Thesaurus for Graphic Materials (TGM)
- *Other*: Language Metadata Table standard[28]

3.5.2 KOSs by Level of Abstraction

Haller et al. [HP20] building on Guarino [Gua97], proposes a hierarchy of ontologies by levels of abstraction. We borrow this system by analogy and adapt it to KOSs. A similar classification can be

[26] https://www.w3.org/TR/swbp-vocab-pub/ (Accessed 07.2022)
[27] https://www.w3.org/TR/2009/NOTE-skos-primer-20090818/ (Accessed 08.2022)
[28] https://www.mesaonline.org/lmt (Accessed 08.2022)

Table 3.2: Examples of LOVs by the level of specialization

Stack Element	*Examples*
Upper Ontology	Upper Mapping and Binding Exchange Layer (UMBEL), Basic Formal Ontology (BFO)
Top-Domain KOS	DCMI Terms, most W3C KOSs: Semantic Sensor Network, PROV; Materials Ontology [Ash10]
Domain KOS	Vehicle Signal & Attribute Ontology (Builds on SSN), MDO [LAL20]
Use-Case KOS	KOS for electromagnetic modeling, KOS for thermal modeling

found in Patel et al. [Pat+05]. We define four levels of abstraction. Table 3.2 includes some examples of each. We focus mainly on LOVs in this classification and not subject value vocabularies.

Upper Ontologies Upper Ontologies are intended as abstract, universal KOS, which any KOS can be specialized from or built on top of. These are not vocabularies proper because they usually contain axioms and rules modeling some elementary aspects of the world (space, time, etc.), so upper ontology is a better term. Although valuable in principle, none of these undertakings have seen widespread success in practice. Two distinct concepts fall on the same level of abstraction of upper ontologies: *reference ontologies* and *bridge ontologies*. These are discussed in Section 3.7.

Top-Domain KOSs Top-Domain KOSs are KOSs that act as a basis for knowledge organization for a single domain. They are typically independent of any other KOS. Mappings between Top-Domain KOSs are desirable but not always possible.

Domain KOSs Domain KOSs specialize or build on one or more Top-Domain KOSs.

Use-Case KOS Use-Case KOSs are highly dependent on a specialized use case. They specialize or build on one or more Domain KOSs.

Due to their importance, we give more examples of Top-Domain KOSs below.

- DCMI Metadata Terms[29]. The DCMI Terms is a popular generic metadata vocabulary originally proposed for the library sciences domain. It has grown in influence and has become almost ubiquitous in the context of Linked Data.
- W3C Provenance Ontology[30] (PROV) used to describe provenance of entities.
- W3C Data Catalog Vocabulary[31] (DCAT), developed to facilitate interoperability between data catalogs published on the web.

[29] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/ (Accessed on 27 August 2021)
[30] https://www.w3.org/TR/prov-o/ (Accessed 27 August 2021)
[31] https://www.w3.org/TR/vocab-dcat/ (Accessed on 27 August 2021)

- CSV on the Web[32] (CSVW), a vocabulary used to describe tabular data exchanged over the internet.
- Open Digital Rights Language[33] (ODRL) a policy expression language for rights & access metadata.
- Schema.org[34], an industry-backed initiative steered by Bing, Google, Yahoo, and Yandex[35], whose main product is a set of schemas describing entities on the web.
- Linked Data Notifications[36], a protocol that describes publish-subscribe over systems dealing with structured information.
- Data Privacy Vocabulary[37] (DPV), a vocabulary to represent information retaining to the processing of personal data based on the EU General Data Protection Regulation (GDPR). As of September 2021, the vocabulary is still in draft status in version 0.2.
- WAIVER[38], a vocabulary for waivers of rights over data and content.
- Friend of a Friend[39] (FOAF) vocabulary is a popular vocabulary used to describe people and things on the web and their interrelations.

3.6 Other Relevant Tools

There are two categories of tools that are important to the organization and representation of metadata but play a different role from KOSs. These are *Meta-KOS tools* and *Authority Control Files*. We discuss them below.

3.6.1 Meta-KOS Tools

Meta-KOS tools provide systems to describe, define, or manage KOSs themselves. Below are some examples of published KOSs. Efforts that did not produce results in the form of KOSs (such as NISO's Issues in Vocabulary Management [WG17] for example) are excluded.

- Simple Knowledge Organization System[40] (SKOS & SKOS-XL)) used for expressing concept schemes such as thesauri, heading lists, taxonomies, folksonomies, etc.

[32] https://www.w3.org/TR/2016/NOTE-tabular-data-primer-20160225/ (Accessed on 29 September 2021)
[33] https://www.w3.org/TR/odrl-model/ (Accessed 07.2022)
[34] https://schema.org/ (Accessed on 29 September 2021)
[35] https://schema.org/docs/about.html (Accessed on 29 September 2021)
[36] https://www.w3.org/TR/2017/REC-ldn-20170502/ (Accessed on 29 September 2021)
[37] https://w3c.github.io/dpv/dpv/ (Accessed September 29, 2021)
[38] https://vocab.org/waiver/ (Accessed 29 September 2021)
[39] http://xmlns.com/foaf/spec/ (Accessed 27 August 2021)
[40] https://www.w3.org/TR/skos-primer/ (Accessed on 27 August 2021)

- Asset Description Metadata Schema[41] (ADMS), an application of DCAT, it is used to describe KOSs and related assets such as XML schemata, generic data models, and reference data.
- Networked Knowledge Organization Systems Dublin Core Application Profile[42] (NKOS AP): A metadata schema for describing KOSs to facilitate the discovery, sharing and reusing of KOSs.
- VANN[43], a vocabulary for annotating vocabulary descriptions with examples and usage notes.
- INSPIRE metadata code list register[44], a subject value vocabulary for metadata *element types* (as opposed to metadata element values).
- Lexicon Model for Ontologies[45] (lemon), aimed to provide a rich linguistic grounding (morphological and syntactic) of lexical entries in KOSs (i.e., linguistic description of the textual *labels* usually found for each entity in a KOS.)
- VoID [46], a vocabulary for expressing metadata about RDF datasets, also useful in describing relations between different RDF datasets.
- JavaScript Object Notation for Simple Knowledge Organization Systems[47] (JSKOS), a data format to represent KOSs, built on top of SKOS but can be used independently.

3.6.2 Authority Control Files / Master Data

This category of systems is logically different from KOSs. Authority control systems are concerned with naming/identifying individuals or instances of things (locations, persons, thematic subjects, etc.). They play a key role in metadata management but are better considered separated from KOSs. We can identify two categories: *authority control files* and *knowledge bases*.

Authority control files (ACF) / master data. Authority control files are usually systems specialized in some topic or theme (ex., persons, geographical locations). They name and identify instances of real-world things and are used for bibliographic citation purposes. In our domain here, they can be used as shared reference points for real-world entities. Take, for example, the problem of referring to the city of Paris (capital city of France) across multiple metadata records for linking and discovery purposes. We can use the ISNI (International Standard Name Identifier) record `000000012114268X` (or in URI form https://isni.org/isni/000000012114268X) to refer to the city of Paris. In this sense, authority control files share similarities with master data sets. Below are several examples grouped by subject.

[41] https://www.w3.org/TR/vocab-adms/ (Accessed on 29 September 2021)
[42] https://nkos.dublincore.org/nkos-ap.html (Accessed 08.2022)
[43] https://vocab.org/vann/ (Accessed on 29 September 2021)
[44] https://inspire.ec.europa.eu/metadata-codelist (Accessed 08.2022)
[45] https://www.w3.org/2016/05/ontolex (Accessed on 29 September 2021)
[46] https://www.w3.org/TR/void/, (Accessed on 29 September 2021)
[47] http://gbv.github.io/jskos/jskos.html (Accessed 08.2022)

- *Compilations*: Virtual International Authority File (VIAF), various authority control files, such as Faceted Application of Subject Terminology[48] (FAST) and Wikidata (see below) include references to equivalent or related authority records in other authority control systems.
- *Legal personality ACF*: ISNI – International Standard Name Identifier, GND – Integrated Authority File, WorldCat/identities, ORCID
- *Bibliographic ACF*: ISBN, ISSN, DOI

Knowledge Bases: In the past decade, we have seen the rise of community-driven, large-scale knowledge base development projects that have grown enough and become a source of agreement that they can be used as authority control files/master data. The first high-profile usage includes Google's use of Freebase and then Google Knowledge Graph. Below are several popular projects.

- Wikidata[49], is a community-driven knowledge base, following the same content development model as Wikipedia[50]. Considered one of the more promising knowledge bases of its kind due to its community-driven model [HP20] as of 2020, it includes both its KOS alongside statements about individuals in a unified system. Wikidata items also contain a compilation of external authority records for an item (ex. WorldCat, VIAF, ISNI, etc.), making it an aggregate authority control file as well.
- DBPedia[51], one of the early success stories of linked data[52], not community driven *per se* but harvesting the content of Wikipedia, which is community-driven, into the Linked Data framework.
- Freebase [Bol+08], launched in 2007, the same year as DBPedia, Freebase adopted a different approach than DB (i.e., data modeling framework). It involved community-driven development with user-submitted content and harvesting structured data from various sources. It was acquired by Google in 2010[53] and used as a precursor to the Google Knowledge Graph and was officially shut down in 2016 [Pel+16].
- GeoNames[54], an editable geographical database, which includes a semantic web integration[55].

After covering syntax and structural heterogeneity in metadata, we turn to semantic heterogeneity for the remainder of this chapter.

[48] https://www.oclc.org/research/areas/data-science/fast.html (Accessed 08.2022)
[49] https://www.wikidata.org (Accessed on 30 September 2021)
[50] https://www.wikipedia.org/ (Accessed on 30 September 2021)
[51] https://www.dbpedia.org/ (Accessed on 30 September 2021)
[52] https://talis-podcasts.s3.amazonaws.com/twt20080207_TimBL.html (Accessed on 30 September 2021)
[53] https://googleblog.blogspot.com/2010/07/deeper-understanding-with-metaweb.html (Accessed 08.2022)
[54] https://www.geonames.org/ (Accessed on 30 September 2021)
[55] http://www.geonames.org/ontology/documentation.html (Accessed on 30 September 2021)

3.7 Interoperability

In the latter part of the 19th century, the popular worldview was that it was possible to determine a single, absolute meaning if a sufficient effort was made. This influenced standardization organizations such as the International Union of Associations (UIA) and the International Standards Organization (ISO), which were driven by a view that the international was the key to the universally valid and ontologically true. Hence the attempt was to develop these standards in a *top-down, deductive* fashion. Fast forward to the second half of the century, even a project as seemingly simple as devising fields of a library catalog (Machine Readable Cataloging[56], MARC) eludes a single, international, universal, solution: resulting in US MARC, AUS MARC, UK MARC, and CAN MARC [Velo1].

With the advent of the World Wide Web and, later, around the turn of the millennium, and efforts such as Semantic Web, Dublin Core, and related programs: the development of KOSs started to become more bottom-up, democratized, decentralized, ad-hoc, and usually driven by pragmatic goals. For example, as of July 2022, the Basic Register of Thesauri, Ontologies & Classifications (BARTOC), which hosts KOSs, lists 3400 KOSs[57].

Arguably, the most crucial observation today is the *usage pattern* of KOSs by different applications. This pattern is the selective reuse and combination of KOSs by importing properties/classes from different KOSs to fulfill the information representation needs of some domain/application. Take data.world [58] for example, which is a data cataloging service, among other things. A simple inspection of the description of some published datasets displays properties imported from around 4-6 KOSs, such as CSVW, Dublin Core, DCAT, and others.

Semantic interoperability is defined as [ZC15] *"the ability of two or more systems or components to exchange information and use the exchanged information without special effort on either system"* [Metoo].

Focusing on the use case of metadata, we can identify two distinct interoperability approaches: constructing KOSs in the first place with the goal of interoperability and building systems to manage mappings across KOSs. We discuss both these approaches below.

[56] https://www.osti.gov/marc-records (Accessed 07.2022)
[57] http://bartoc.org/stats (Accessed 07.2022)
[58] https://data.world/ (Accessed on 30 September 2021)

3.7.1 Interoperability by Constructing KOSs

Zeng identifies the following typography of approaches to interoperability by KOS construction [Zen19]:

- *Derivation*: in which one KOS is derived from another. This includes adaptation, modification, extension, partial adaptation, and translation.
- *Expansion*: there are several approaches to expansion.
  - *Leaf node*: when a leaf term in one KOS is expanded into a more detailed set of terms in another KOS.
  - *Satellite vocabularies*: a set of KOS developed in a closely coordinated, top-down super-structure in which different KOSs specialize in different aspects of the domain of interest.
  - *Open umbrella structure*: the classical upper ontology with a set of top-domain KOSs inheriting from it (and others inheriting from them, and so on).
  - *Integration/combination*: the combination of multiple vocabularies to form a meta-vocabulary whose scope is the combined scope of all the combined vocabularies. Zeng gives an example of the Unified Medical Language System (UMLS) metathesaurus.
- *Shared/bridge schemes & Reference ontologies*: bridge schemes mediate between particular concepts across multiple ontologies by providing common terms that can be mapped from terms in the source ontology [Fri+17].

The approaches above tackle the interoperability problem by constructing KOSs which facilitate interoperability. However, if the KOSs have already been published and are in use and we would like to achieve interoperability between them, then we have to construct mappings between them.

3.7.2 Semantic Interoperability Architectures

KOS Mapping is the process of establishing relations between terms in two different KOSs and using those relations in automated systems to transform data adhering to one KOS into data adhering to the other. Cross KOS mapping is a multifaceted problem that is an approximation at best since KOSs differ concerning structure, domain, language, and granularity.

Uschold and Gruninger identify five levels of *semantic integration architectures*[59] [UG02; UG04]:

1. *Global/Upper KOS*: this architecture is the easiest technically if such a global KOS can be developed. In this case, no mappings need to be generated since all use the same KOS, but all parties have to agree on the meaning and definitions of everything. A variation of the global KOS is the Upper KOS

[59] We will replace the term 'ontology' with the term 'KOS.'

model, in which KOSs specialize from an Upper KOS. As discussed in this chapter, this architecture is unrealistic unless we consider it in a small, specialized community.

2. *Manual Mapping*: in this architecture, humans have to design a mapping between the KOSs before every two parties need to exchange data.
3. *Interlingua KOS*: here, a KOS is used as a mediating KOS, and parties define mappings to and from this KOS. Here the agreement must be on the interlingua KOS. Parties are free to do what they please with their KOSs as long as they provide a mapping to it.
4. *Community KOSs*: in this architecture, we have a shared common library of KOSs that acts as a set of interlingua KOSs, and parties define mappings to and from these KOSs. This architecture is similar to how KOSs are used on the web within the Linked Data community.
5. *KOS Negotiation*: in this architecture, systems auto-generate mappings at interaction time. This is considered the 'Holy Grail' of semantic integration.

These architectures are not mutually exclusive and can be mixed and matched based on the application domain. For example, combining the Upper KOS architecture with an Interlingua or Community KOS model results in a hub-and-spoke architecture.

3.8 Conclusion

In this chapter we explored some key foundational topics for our problem. These are derived from the approach adopted (indirection). These topics lay some groundwork for the the inter-organizational curation requirement (Req. 2 and Sub-Req. 5) as well as the semantic heterogeneity sub-requirement (Sub-Req. 6). If we are to facilitate curation and its interoperability in an indirection environment, we need to answer questions related to the *movement and translation of metadata* across heterogeneous environments in terms of semantics, syntax and systems. Hence in this chapter we investigate topics of metadata *representation* and *interoperability*.

Arguably, this chapter set out to chart a space that is beyond the scope of this dissertation. However, there is enough insights to to extract some concrete structures and models which aid us in the proposed model of the this work (particularly in the *Metadata Interoperability Structure* within the Basin Network model presented in Chapter 5). As this chapter showns, the problem of metadata representation and interoperability for inter-organizational curation is a big problem with many branches. However, as we will see, what this effort achieves is in setting a framework in which further and more specialized questions can be pursued in the future: in particular, who/how/what does the mapping across KOSs, and how can an eco-system for such mapping emerge?

4 Architectural Aspects

The term *architecture* in the title if above is used loosely here, not in the sense of a typical software/hardware architecture[1], but in the sense of *overarching and cross-cutting concerns* in a decentralized or distributed system environment. There are various such problems that needs to be addressed to be able to build a running solution of our problem, however, due to scope limitations, we focus on three major sub-problems: (1) information management in Section 4.1, (2) data domain topology (surrogate management spaces and the exchange of surrogates across them) in Section 4.2, and (3) abstraction for digital surrogates in Section 4.3. For each of these sub-problems, we investigate one or more influential approaches and show how none of them is satisfactory, which lead us to our proposed model in the following chapter.

4.1 Information Management: The Linked Data Framework & Its Problems

The *Semantic Web* [Ber98] started as a vision in which content is published in machine-readable and -understandable on the web with the goal of enabling automation of information-driven tasks [BHL01]. The term also refers to a set of standards and techniques developed around this vision [Hog20, pp. 654]. It came out partly as a successor of the *Platform for Internet Content Selection* (PICS), a specification for attaching simple metadata to internet content for enabling parental control over web content[2]. Influenced by the field of knowledge representation and reasoning (KRR), one of its goals was to do KRR over metadata embedded in web content, distributed over the web, as stated by the first high-profile introduction in an article in Scientific American magazine [BHL01]. In the now famous article, the authors compare the relation of "KRR to the Semantic Web" as that of "hypertext to the web:" a mature technology waiting for a large-scale application.

Whereas in databases, a dataset usually refers to a collection of observations or readings, in the Semantic Web, a dataset refers to a collection of statements (subject-relation-object triples) where each asserts

[1] A standard definition of architecture is: *"[a] fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution"* [Sta11].

[2] https://www.w3.org/PICS/ (Accessed 07.2022)

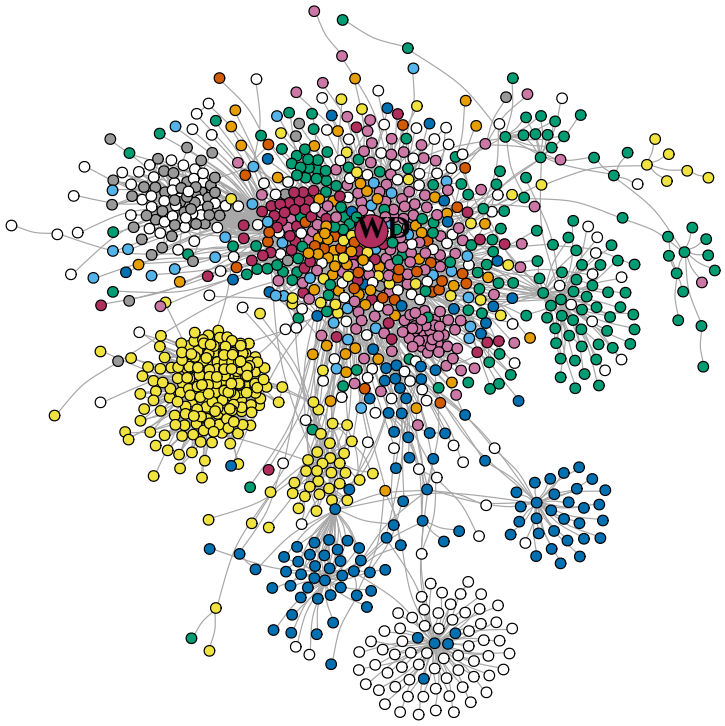


Figure 4.1: An example of the linked data cloud generated using data from lod-cloud.net. (Copyright CC BY 4.0, Thomas Shafee)

that some relation or property holds on some individuals. These statements are usually represented in the Resource Description Framework (RDF) [Ber98; SR14].

Linked Data [Ber06] refers to a set of rough architectural principles for *publishing* and *cross-referencing* entities across semantic web datasets, the term is also used to denote the semantic web datasets published based on these principles [Hog20, pp. 649], [DFH11, pp. 67]. It is often visualized as a set of vertices (representing RDF datasets) with edges between them denoting 'references' (see Figure 4.1). 'Reference' here is sometimes understood when a subject or object of some assertion is a Uniform Resource Identifier (URI) with a namespace from another semantic data set. However, there is still no common agreement on the subject in practice [Hal+20]. In what follows, we discuss some challenges with this approach.

Semantic Web: More Representation than Reasoning More than 20 years later, the most successful work influenced by the Semantic Web has been more knowledge representation than it is reasoning [Pol+20]. This can be observed, for example, by bringing to mind two success stories: DBPedia[3] one of the earliest success stories as stated by Tim Berners-Lee in a recent interview[4], is a knowledge

[3] https://www.dbpedia.org/ (Accessed on 30 September 2021)
[4] https://talis-podcasts.s3.amazonaws.com/twt20080207_TimBL.html (Accessed on 30 September 2021)

base. Wikidata[5], also a knowledge base, part of the Wikimedia Foundation[6], considered as one of the most promising projects of its kind [HP20] and a worthy successor of DBPedia. Both are knowledge representation efforts. This issue begs the question: if the reasoning aspect has not been widely adopted in practice, then many of the complexities introduced just for supporting it are complexities we have to bear for no practical reason.

Linked Data Architecture: Lack of Real-Time Support Although not explicitly stated in Tim Berners-Lee's early notes, the semantic web and linked data architectures view RDF datasets as discrete products in time packaged as versions, releases, or snapshots. This can be observed by looking at the recent advances of one of major linked data projects, DBpedia[7], with 850 million facts as of a June 2021 official announcement[8]. In the same announcement, a release frequency of one snapshot per four months is given. However, although they do mention new initiatives such as DBpedia Live [9], it remains to be seen how they will develop. An approach using SPARQL endpoints attempts to minimize these issues, however, there are no agreed-on standards regarding the underlying back-end and its real-time properties.

Lack of Privacy Considerations Privacy is one of the challenges of Linked Data [BHB09]. Neither the original[10] nor the updated[11] Semantic Web stack contains a privacy component. The specifications of RDF [SR14] and OWL [Hit+12] as well do not mention the topic. Vocabularies such as Web Access Control (WAC)[12] and the Privacy Preference Ontology (PPO) [SP11] are tools for representing information about controlled access of external resources (ex. web resources, files , ...etc), but not the contents of RDF datasets themselves.

Breadth over Depth, and the Crisis of Trivialities Due to trying to achieve an impressive breadth of scope and vision, it can be argued that some depth has been sacrificed in the process. There are significant gaps when it gets to the technical details, which leads to diverging independent interpretations when implementing systems, breaking the original distribution and interoperability goal. One could say there are several linked data and semantic web interpretations, and two decades on, some researchers still find a need to define notions such as URI, dataset, etc., due to a lack of consensus and an interpretation gap (see for example [Hal+20]). A similar point can be made about the separation between theory and practice in the Semantic Web sphere [VV20]. Take, for example, the RDF dataset

[5] https://www.wikidata.org (Accessed on 30 September 2021)
[6] https://www.wikimedia.org/ (Accessed 07.2022)
[7] https://www.dbpedia.org (*Accessed on 05 August 2021*)
[8] https://www.dbpedia.org/blog/snapshot-2021-06-release/ (*Accessed on 05 August 2021*)
[9] https://www.dbpedia.org/resources/live/ (*Accessed on 05 August 2021*)
[10] https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html (*Accessed on 05 August 2021*)
[11] https://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24) (*Accessed on 05 August 2021*)
[12] https://github.com/solid/web-access-control-spec (*Accessed on 05 August 2021*)

definition, the varying understandings of what constitutes an RDF dataset leave room for each application to make diverging interpretations, leading to interoperability challenges [Hal+20]. Pollores et al. state in 2020: "ten years into Linked Data there are still (too?) many unresolved challenges" [Pol+20].

Open Problems Regarding 'URI' & 'Resource' The semantic web / linked data standardization efforts at the W3C have stumbled into some open philosophical challenges. These include what has been called the Identity Crisis[13] which revolved around a confusion over the ontological status of the 'URI-resource,' which, besides leading to philosophical problems, leads to technical problems the follow from them such as the *httpRange-14* conundrum[14], where the range of the HTTP Get method becomes undefined. A related problem is that of the semantics of URIs [Hal11]. We discuss these issues in Section 4.3. One can ever name other problems such as the non adoption of the unique names assumption and the challenge of the various interpretations of `Owl:SameAs` [HH10; VV20]. We discuss these problem in more details as well as recent proposals from the literature to tackle them.

The Semantic Web started as a vision to make the web machine-understandable and machine-actionable for automated software agents that (inter-)act over structured information on the web. Two decades on, its key contributions are a suite of standards, recommendations, tools, and technologies for knowledge representation and reasoning on the web.

4.2 Data Domain Topology & Surrogate Spaces: Lakes, Spaces, and Their Kins

In this section two closely related architectural aspects of the solution will be discussed: surrogate spaces and data domain topology infrastructures. We will look at the major approaches and show how none can meet all our requirements.

4.2.1 Lakes, Spaces, and Their Kins

Data lake is an industry buzzword for a set of heterogeneous architectural practices usually centered around storing heterogeneous data in its original format, schema-on-read, and managing the lake's contents with the support of metadata [HQJ21; SD20]. However, there is little consensus on the specifics.

[13] https://www.xml.com/pub/a/2002/09/11/deviant.html (Accessed 08.2022)
[14] https://www.w3.org/2001/tag/group/track/issues/14 (Accessed 08.2022)

Table 4.1: Data lakes versus data warehouses, from Laurent et al. [LLM20, pp. 12].

	Data Lake	Data Warehouse
Data storage	Hadoop Distributed FS, NoSQL, relational	Relational
Data preparation	On the fly	Before integration
Schema	On read	On write
Information architecture	Horizontal	Vertical
Metadata	Context/descriptive	operational
Data analysis method	Unique	Repetitive
(End-)Users	Computer/data scientists	Decision-makers
Conception	Information Driven	Data driven

Chihoub et al. [LLM20] give an abstracted definition of data lakes influenced by one of the popular proposals, that of IBM [Che+14]:

> *"A data lake is a set of centralized repositories containing vast amounts of raw data (either structured or unstructured), described by metadata, organized into [logically or physically] identifiable datasets, and available on demand."* – Chihoub et al. [LLM20, pp. 23] *(Text in square brackets borrowed from other definitions [ML16; SD20]).*

Data governance and quality are two themes usually discussed in conjunction to data lakes [Che+14; ML16; Mat17; SD20], as well as data processing [Mat17; HQJ21]. Data lakes are sometimes understood in contrast to data warehouses. Table 4.1 presents one such comparison. The first column includes functional aspects of the storage architecture, and the following two columns specify the approach used to implement each aspect of the storage architecture. Based on this table, one way to describe a data lake would be as *a horizontal system supported by metadata that stores heterogeneous data and is used by computer/data scientists for carrying out unique one-off data-driven tasks*.

Sawadogo et al. categorizes existing data lake architectures into *functional*, *data maturity-based*, and *hybrid* architectures [SD20]. Functional architectures [JM17; QHV16; Meh+19] are built around fulfilling the basic data lake functionalities [LS16; SD20]: data ingestion, storage, processing, data access / querying. Data maturity-based architectures [ZdB14; LS16; Inm16] on the other hand are built around data refinement level [SD20]. There are two varieties: pond [Inm16] and zone [Gie+19; LS16] architectures.

A data pond is a subdivision of the data lake focused on a specific type/nature. Each pond is associated with its storage and processing systems. Inmon [Inm16] identifies five data ponds within a lake: raw data pond, analog data pond, application data pond, textual data pond, and archival data pond. The raw data pond deals with newly ingested data. The analog pond deals with (semi-structured) high-velocity data from sources such as IoT systems. The application pond receives structured relational data from software applications. Inmon identifies the data warehouse as a type of application pond, *effectively*

positioning the pond architecture as a generalization of the warehouse architecture. The textual data pond manages unstructured textual data with its disambiguation and text analysis systems. Finally, the archival data pond stores data not actively used in other ponds [SD20].

Zone architectures assign data to zones based on their level of refinement. In one architectural variant [LS16], there are the following zones: transient loading zone, raw data zone, refined data zone, trusted data zone, discovery sandbox, and consumption zone.

Finally, from the name, hybrid architectures are consider functional and data maturity-based considerations [Inm16; RZ19].

The notion of *Data Spaces* was introduced by Franklin et al. [FHM05] in 2005 as an 'agenda for data management.' The motivation was the expanding diversity of heterogeneous data sources belonging to an organization (ex., an enterprise, government agency, library, and smart home) that need to be managed in an integrated fashion. A data space is defined as the total of all data sources/assets of an organization. Although not defined formally, the authors gave various specifications to clarify the concept. Data spaces aim to provide base functionality over all data sources and develop further integration measures incrementally in a 'pay-as-you-go' fashion. This thinking has influenced the NOSQL/big data management/data lake 'ethos.' We discuss a recent project of International Data Spaces [JQ17; OJ19; Bad+20] in the scope of similar work in Chapter 7.

Data catalogs are defined as systems that exist to "collect, create and maintain metadata" [Qui+20]. The following keywords are usually mentioned in conjunction of data catalogs [Ehr+21]: *metadata management*, *business context*, *data responsibility roles*, and *FAIR Principles*. Hence, Data catalogs also fall under our definition of a surrogate management space.

4.2.2 Surrogate Spaces and Data Domain Topology Infrastructures

We defined a *surrogate management space* as a gated/walled-garden (in terms of data access and privacy) to manage surrogates of data assets that share a thematic goal or purpose. Looking at the discussions of data lakes, spaces, catalogs above, it becomes clear that the basic unit of these approaches is a surrogate space. A data catalog is a set of entries in which each is a surrogate of some data asset. A simple data lake with a single zone or pond, is similarly a surrogate space, and so on. However, so far in this conceptualization, there is no mention of the process of *exchanging surrogates across surrogate spaces* of which we need another conceptualization.

Table 4.2: Properties of several data domain topology infrastructures.

Architecture	*Space Membership Condition*	*Inter-Space Exchange*	*No. of Spaces*	*Privacy*	*Inter-space Curation*	*Semantic Heterogeneity*
Lake (functional)	Stage of data lifecycle	preset	<10	no	no	no
Zone	Refinement	preset	<10	no	preset	no
Pond	Type/nature	no	<10	no	no	no
Lake (hybrid)	Refinement or type	preset	<20	no	preset	no
Data Catalog	Arbitrary Scope	no	1+ per org.	no	yes	no
Data Space	Organizational Scope	yes	1 per org.	yes	yes	no

A *data domain topology (DDT) infrastructure* is a set of interrelated technologies and standards which enable the (1) managing of a set of two or more surrogate management spaces and (2) the exchange of surrogates between them. Just as a surrogate is the *basic building block* of a surrogate space, a surrogate space is the basic building block of a DDT infrastructure. Hence data lake approaches with multiple zones or ponds are effectively data domain topology infrastructures. A data spaces approaches is also a data domain topology infrastructure, and approaches that cover the exchange of surrogates across multiple data catalogs (ex., W3C's DCAT[15] recommendation) are a DDT infrastructure. Table 4.2 compares key properties of different data domain topology infrastructures. Below is a discussion of each property.

- *Space membership condition*. This property refers to restrictions, or lack thereof, that a model places on the contents of a single space within it. Data lake variants typically have a 'hard-coded' condition on the contents of each space related to some data lifecycle property maturity, refinement, data lifecycle stage, etc.). The data spaces model defines the scope of a space membership to cover that of all the data asset surrogates of an organization (one space per organization or per person, in the case of personal data management). Data catalog models leave the spce membership condition arbitrary, making them the most permissive in this regard.
- *Inter-space exchange*. This property refers to the consideration of exchanging surrogates across spaces. The exchange mechanisms between spaces in the data lake variants are preset based on the typical exchanges between the different spaces in the data life-cycle process (ex. raw data, to cleaned data space; or in the case of the pond variant, surrogates move across spaces only when they are mapped to a different data type/nature. The data spaces model assumes data exchange across spaces as a key use case, however, no details are given as how to achieve it. Data catalogs typically do not consider exchanges of records across catalogs, however, special purpose models such as the Data Catalog Vocabulary (DCAT) W3C recommendation, consider that use case.
- *No. of spaces*. This property refers to how many spaces does a model allow. Data lake variants usually allow from 1-20 spaces, where typically it is less that 5. Data spaces allow for one per organization, and data catalogs, although not usually specified, usually assume the context of one catalog per organization or sub-domain within an organization.
- *Privacy*. This column refers to privacy considerations for assets moving across spaces, in specific we are considering visibility, access, and some simple forms of annonymization. This feature is usually not considered in the data lake model variants because the context of a data lakes is understood as a shared environment between data workers in an organization. For data spaces, visibility and audience are mentioned in the original paper [FHM05], however detail is given about the how. Finally, data catalogs usually do not have these considerations.

[15] https://www.w3.org/TR/vocab-dcat/ (Accessed on 27 August 2021)

- *Inter-space Curation*. In this requirement we are concerned with whether the approach includes an explicit consideration of the issue of how surrogates move across spaces. The data lake approaches at best have some preset system, and usually none. The data catalog approaches usually do consider this issue, in specific the popular W3C Data Catalog Vocabulary (DCAT) is built with the use case of exchanging entries across data catalogs. The data space approach as presented in the original papers, considers this issue, but only implicitly, so each approach implementing the model would have to decide how to realize it.
- *Semantic Heterogeneity*. Considering semantic heterogeneity entails that the approach allows for different parties to use metadata with different semantics, regardless of an upper or reference ontology. The data lake approach do not explicitly consider this issue, and neither does the data catalog approaches. The data space approach, as it is presented in the original papers, does not explicitly touch on this topic. Some implementations, such as the International Data Spaces project (see Chapter 7) have a vocabulary management component, whereas others, such as the model developed by the Gaia-X Association[16], do not.

As we can see each approach provides different features, data spaces appear the most general with regards to the considered properties, however it remains a "vision for data management" in the words of its authors, hence the details regarding things such as inter-space exchange or privacy are left for the adopters of the model. In a sense we are looking for a model that can inherits all the strengths of the above approaches while overcoming their limitations. And for this we will propose the Basin model in Chapter 5.

4.3 Surrogate (Artifactualization of): The URI-Resource

Artifactualization is a term introduced by Monnin [Mon09] which is defined as a process in which a concept gives birth to a digital abstraction; also, the product thereof. Files/folders in the Windows operating system are artifactualizations of paper-based files/folders. The study of artifactualizations has to do with the conceptual, semantic, and philosophical foundations of such newly devised digital abstractions. What we need here is a *shared artifactualization of surrogates* to be used in our data domain topology infrastructure. Such a shared artifactualization achieves two goals simultaneously:

- Accounts for the meta-data of data assets along with their data in a unified conception,
- Acts as an umbrella for data assets of heterogeneous types, formats, and technologies.

[16] https://gaia-x.eu/ (Accessed May.2023)

The URI-resource[17] is a prominent artifactualization of surrogates. However, it has two major problems which have been collectively called the *URI Identity Crisis*: one ontological[18] and one related to semantic interpretation [Hal11; Hal19]. In general, these problems emanate from the conflict between using URIs used as locations on the internet (i.e., URLs) and URIs used as names (i.e., URNs) for entities that do not fit the mold of web resources as they are commonly understood (e.g., telephone numbers, persons, etc.). We will discuss these problems and recent proposals in the literature to overcome them.

4.3.1 The Ontological Identity Crisis

There is a long-running theoretical open problem in the Semantic Web community called the httpRange-14[19] by the W3C Technical Architecture Group (TAG). The problem is related to the *duality* held in the minds of semantic web theorists regarding the *kinds of things* that are the primary content of the 'web of data.' The first kind, called '[web] documents' by practitioners, refers to web pages and other entities that can be represented as data formats that can be hosted and retrieved using HTTP content negotiation and streamed across the network (ex., images, CSS files, streaming video). The second kind includes things that do not fit this categorization, including things that are supposed to 'identify/stand-for' people, locations, and abstract concepts (ex., tree), etc., which we can call 'non-documents' for the sake of argument.

From the name of the problem, the issue is about the range of the HTTP GET function when retrieving a 'document' or a 'non-document' identified by some name/URI that is also a web location (URL). Consider the web page identified by the following URI https://www.w3.org/2001/tag/group/track/issues/14. When we execute an HTTP GET request with the URI above as a parameter, we get an HTTP document located at that location, the one-and-the-same entity that the URI identifies. This is pretty well understood. However, consider a 'non-document' like the planet *Venus* associated with the name/URI www.example.com/planets/venus. What should we get if we execute an HTTP GET request at that location? Nothing? Some web resource with *information about* the planet? A web resource that should be understood as representing the planet and not just a description?

The term used to denote the basic building block of the URI/web-of-data architecture came to be known as *resource*. The TAG could not reach an agreement at a conceptual and technical level on this problem, with different workarounds arising and proposing the problematic notions of *information resource* and *[non-information] resource*[20].

[17] https://www.w3.org/TR/webarch/ (Accessed 08.2022)
[18] https://www.xml.com/pub/a/2002/09/11/deviant.html (Accessed 08.2022)
[19] https://www.w3.org/2006/04/irw65/urisym (Accessed 08.2022)
[20] https://www.w3.org/TR/webarch/ (Accessed 08.2022)

One of the key reasons for the httpRange-14 conundrum is the *confusion over the ontological status* of the URI-resource[21]. Roy Fielding, associated with the REST architectural style, defines a resource as "*the semantics of what the author [of the URI] intends to identify, rather than the value corresponding to those semantics at the time the reference is created*" [FT02, pp. 135]. This definition of what a resource might mean in the context of the web is one of the better definitions around, although it is more than two decades old. It clarifies several key properties of URI/resources.

First, a resource is the idea of something an author deems worthy to *single-out/name/identify*, this mirrors the common understanding that resources can be anything, and an abstract artifact [Mon12]. Second, the resource's value (i.e., 'semantics') is independent of the value *at the time of the creation* of the technical object. This separates the lifecycle of the identified resource from the temporal changes in value it can undergo, which makes a resource akin to a *virtual trajectory* [Mon12]. The following example will help elucidate the second property.

The TAG's web architecture recommendation document[22] provides us with the following example. The resource identified with the URI `http://weather.example.com/oaxaca` is *weather information for Oaxaca, Mexico*. Some user (Dirk) adds the URL on their webpage and labels it 'report on the weather in Oaxaca on 1 August 2004.' Another user (Naida) alerts him that "The Oaxaca weather site policy is that the URI in question identifies a report on the current weather in Oaxaca–on any given day–and not the weather on 1 August."

Monnin proposes a new perspective to look at the resource Identity Crisis [Mon12, pp. 17]:

> Nowhere do we need to ask ourselves whether or not URIs refer to something permanently and how. URIs do *not* refer for the aforementioned reasons. We rather publish Web resource identified by the latter and, if given enough resources (in the traditional meaning of the word), then we maintain a positive feedback loop between Web resources and accessible representations; all in all, a very different story.
> [...]
>
> While regular (philosophical) proper names may possess the distinctive feature to refer if used accordingly, neither identification nor access on the Web have to do with reference. In other words, the artifactualization of proper names is tantamount to replacing reference with other (technical) processes, coupled to one another. Thus the explanation of the binding between URI and resources rests upon entirely new principles. The issue at stake is no longer to *point* at or *designate*, but rather to *maintain* a coupling between two kinds of processes through socio-technical means. *[Emphasis in the original]*

[21] https://www.xml.com/pub/a/2002/09/11/deviant.html (Accessed 08.2022)
[22] https://www.w3.org/TR/webarch/ (Accessed 08.2022)

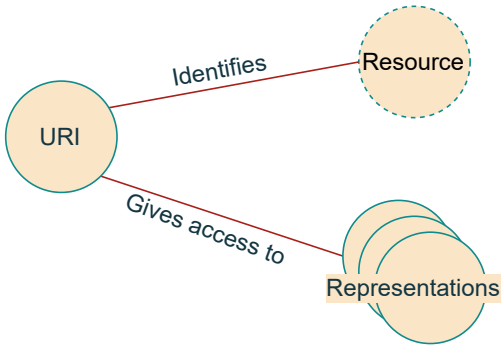


Figure 4.2: A depiction of the technical triangle of URI, resource, representations.

In other words, Monnin–influenced by Roy Fielding's definition of the resource–circumvents the ontological identity crisis in two steps:

1. **URIs do not refer**, they merely have to do with identification and access; a set of (technical) processes in which different representations/semantics are coupled/associated an ID over time. We can distinguish between two technical roles: (1) the URI *minter*, the party that executes a request for the creation of a new URI (to identify some resource) based on the possession of a domain name, and (2) the *service provider*, the party (usually company) that maintains the access to a given set of representations. See Figure 4.2 for a depiction of this technical triangle.
2. **All resources are abstract**. Resources are abstract entities that have a URI associated with them; this follows directly from Fielding's definition of resource.

Given this view, we can understand the situation by looking at it as a technical triangle of a URI (the identifier), a resource (the thing the URI is intended to identify), and a set of representation over time in which some service provider associates with the URI. This is visualized in Figure 4.2) [Mon12]. Take the homepage of the Guardian newspaper website (URI: theguardian.com), depending on the day the URI is referenced, the service provider which hosts the website will associate different contents (representation) with the URI of the homepage.

Now that we have the ontological problem out of the way (there is not two types of resources but only abstract ones), we can turn to the Semantic Interpretation URI-resource Identity Crisis.

4.3.2 The Semantic Identity Crisis

For a majority of users and institutions that use and author URI/resources take the following two questions to be closely related or even *identical*:

- *What should a URI evaluate to when dereferenced?*, and
- *What is the 'content/value' of a URI/resource?*

These are both questions about regarding the semantic interpretation of URI-resources. There are two prominent competing positions regarding the question of what the 'semantics' (value/evaluation) of a URI/resource come from the discipline of mathematical logic. This is partly due to the founders of the semantic web *viewing it* as a problem of knowledge representation and reasoning over the web. The positions are the *direct reference* position, where the meaning of a URI is what was defined by its owner, and the *logical* position, which adopts a Tarksian model-theoretic semantics where the meaning of a URI is given by the set of all things that can stand in its place and still satisfy some provided formal model [Hal11].

When implementing URIs, many practitioners follow the direct reference (or declarative) position. The logical position would require dataset providers to construct rich models associated with their data. As we have seen in our discussion of semantic data practices on the web, semantic data is rarely specified with rich models. Hence, although formally the desirable position, the model-theoretic approach is not realistic in practice, leaving the first position the way most practitioners understand URI 'semantics.'

Halpin [Hal11] proposes a third position, that which he calls *social semantics* [of URIs], taking inspiration from Wittgenstein's notion of *public language* and with the example success story of web search engines, where the meaning of webpages have come to be associated with keywords, statistics, cross-linking from other pages, and past user behavior; many things outside of the original owner or model specifier of the resource [Hal11].

An approach to managing URI/Resources must adopt one or more of the three positions above. Whereas model-theoretic semantics is not consistent with the other two, the direct reference and social semantics seem more consistent.

4.4 Conclusion

In this chapter, we discussed three major architectural aspects of the solution: the information management aspect, the surrogate space + data domain topology infrastructure, and the surrogate artifactualization aspect. We discussed the prominent approaches for each aspect and the challenges and limitations in them. This chapter concludes the study of the background and related work in all the different aspects of the solutions space. In the next chapter we present our proposed model, the Basin Network.

5 The Basin Network Model

In this chapter, we present our proposal: the Basin Network (BNet) model. The model can be divided into three major components: the *Offering* abstraction, the *Basin* abstraction, and the *Metadata Interoperability Structure*. Whereas the third builds on ideas from Chapter 3, the first two are original abstractions that benefit from some background and intuitive introductions. We start with discussing them and then move on to present a formalization of the BNet model. Although some depictions and examples of different elements of the model are mentioned in this chapter, a full example of the proposed model is given in Section 6.3 in the following chapter.

5.1 The Offering Abstraction

An offering is a type of resource, where resource here must be understood as the term discussed in Section 4.3. Three examples of a Resource are: (R1) 'weather information for Oaxaca, Mexico,' (R2) 'weather in Oaxaca on 1 August 2004', and (R3) 'simulation data for part x with material z using profile y.' For example, whether the values associated with R2 are correct is irrelevant because it is about the author's intention and worldview (be it a human or machine).

As Monnin notes [Mon12], the relation between a Resource and the values corresponding to it over time is not a mysterious abstract/theoretical/philosophical 'bond,' but merely a technical task of maintaining a coupling between the two; a technical task of maintaining a close coupling between an *identifier* and one or more *representations* over time.

An offering is *a type of resource which (1) identifies or individuates a data asset, and (2) its coupled representations adhere to the* offering token *specification (Definition 5 below).*

For the first clause, there is no offering without a data asset. The core purpose of authoring an offering is an act of communicating, of advertising, *of offering*–so to speak–some data-driven value for others to discover, use, etc. For the second clause, the *offering token* specification is constructed with the data domain topology (DDT) infrastructure problem identified in Chapter 2. The offering token

specification is intended to facilitate the fulfillment of technical bookkeeping duties (ex., integrity, traceability, license, security, authorship, metadata interoperability, etc.) of this domain.

The discussion so far has resorted to intentional definitions and motivations. Another way to communicate the intuition behind this abstraction is to give a process-oriented explanation. The process of authoring an offering involves the following stages:

- Identifying an intention/goal.
- Selecting a data asset.
- Describing the data asset *[Following the Offering Token specification]*.
- Declaring the properties of the Offering (ex., integrity, audience) *[Offering Token specification]*.
- Publishing the offering in a basin.

Next, we move on to the second original abstraction, the Basin.

5.2 The Basin Abstraction

Turning over to the second fundamental abstraction, the *Basin*, we can also introduce it from several angles. The abstraction is built around an analogy *(and term)* borrowed from geology. A *Basin* is defined as a landform that dips inwards towards a central (lowest) point [Jam14, pp. 241]. A related concept, *drainage basin* refers to open systems where rivers and streams carry water (usually from rain) through *networks* of permanent freshwater lakes (i.e., basins), eventually draining into the ocean[1].

From an operational perspective, a basin as an information system playing multiple roles:

- A gated scope or walled garden, unified by a goal, use-case, or intention, used for the management and co-management of offerings.
- A publisher of offerings.
- An agent to manage the exchange of offerings with other basins *[Publication Contracts, Definition 8]*.
- A checkpoint where privacy management and processing (ex., anonymization) takes place.
- A checkpoint where model mapping and content negotiation of offering tokens takes place.

Below we give a formal definition of the different elements of the model.

[1] https://www.britannica.com/science/inland-water-ecosystem (Accessed 05.2021)

5.3 Formalization

To ease comprehensibility, we divide the formal definition of the model into two parts: the core and the Metadata Interoperability Structure. This section will be structured as follows, one or more related definitions will be given, and then an example depiction or textual example is given.

5.3.1 The Core

Definition 1 (Dataset) *A* Dataset *is a collection of data, published or curated by a single agent, which is available directly for access/download or in the form of a group of operations that provide access to them* [2].

Examples include:

- relational data accessible via a query to some API,
- hierarchical data,
- one or more JSON documents,
- data formatted in some unified format (ex., office documents, CAD/CAM), and
- binary blobs.

Definition 2 (Dataset Series) *A* Dataset Series *is a collection of Datasets that are published separately but share some common characteristics that group them*[3].

Examples include:

- time series composed of periodically released subsets, and
- map-series composed of items of the same type or theme but with differing spatial footprints.

Definition 3 (Data Asset) *A* data asset (asset, *for short) is a set of one or more closely related Datasets or Dfataset Series, published or curated by a single agent, that together form a specific identity or fulfill a particular function.*

Definition 4 (Offering) *An* offering *is a type of URI/resource. It is defined as the semantics of what the author intends to identify related to some data asset rather than the value corresponding to those semantics when the reference is created.*

[2] This definition is influenced by the definitions of Dataset and DataService in the W3C Data Catalog Vocabulary (DCAT) V2 (https://www.w3.org/TR/vocab-dcat-2/).

[3] This definition and its examples are borrowed from the definition of DatasetSeries in the up-and-coming Data Catalog Vocabulary (DCAT) Version 3 (W3C Working Draft 10 May 2022) (https://www.w3.org/TR/2022/WD-vocab-dcat-3-20220510/ (Accessed 08.2022)

To associate an offering with the representations corresponding to its semantics at different states or points in time, we will introduce the *offering token* (*token*, for short) construct, where 'token' here is borrowed by analogy from the *type-token distinction* in logic [Wet18]. The offering token construct serves as the basic building block and message exchange format of the BNet model. It is immutable; once created, it cannot be changed. An offering is associated with a set of tokens via a coupling maintained by an *offering registry* (Definition 6).

Definition 5 (Offering Token) *An* offering token (token*, for short) is a declarative specification of the state of an offering in some state or point in time. It is an immutable representation and should not be changed once created. The contents of a token are populated by the author of the offering or token. Formally, a token is a tuple* $T = (a, d, i, u, l)$*, where*

- *a is called the* access and structural *element (includes access instructions and structural information for a data asset),*
- *d is called the* token description *element (includes contextual, operational, descriptive information), is a set of Semantic Data Sets (Definition 14),*
- *i is called the* integrity and authorship *element (includes authorship information, signatures, dates),*
- *u is called the* usage *element (includes license, audience, access rights), and*
- *l is called the* links *element (for links to offerings/tokens such as lineage but also other uses).*

Example Datasets from the computer-aided manufacturing scenario introduced in the introduction chapter include an instance of simulation results for some part (say a part of an airplane wing) using some specific parameters (using some carbon fiber variant material) and inputs (material properties of the said material). Another would be CAD/CAM models of desired parts. However, without the associated metadata this data will only be useful, accessible, findable by the persons that generated it. Whereas a dataset would be a specific instance of simulation data, an offering can represent a sequence of simulation datasets associated with some parts. This is typical, because the first simulation result is rarely the last, and the need to collect several relation results under a single structure affords many desired properties. Hence instead of being a single instance of a single simulation run, an offering organizes a set of related simulation runs (for example, sequential).

As defined above, the offering is abstract, meaning there is no single data structure associated with it. That is realized using the offering token. Whereas the simulation data offering above would be about above a sequence of simualtion runs for some material using some system (etc), possibly including past, present, or future runs, an offering token identifies, and describes, a specific run, say, a simulation run, on a specific date, using a slight variation on one of the inputs to the simulation system, and resulting in a different end state. The contents of this offering token would be includes annotation about the simulated part, the operator, date (in the *d* element), instructions to access the data (in the *a* element),

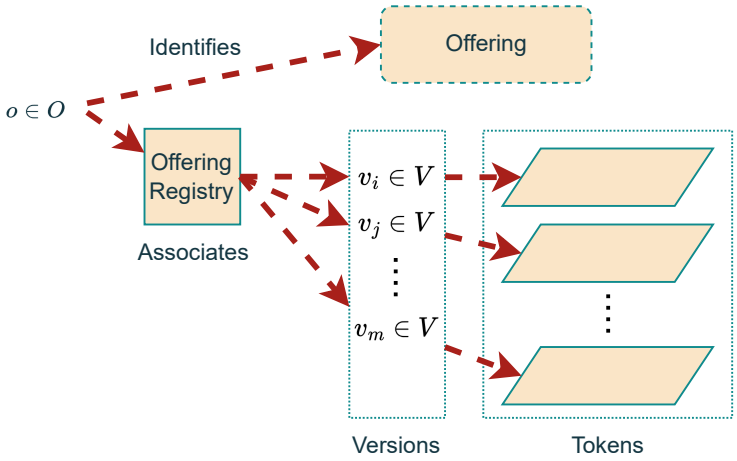


Figure 5.1: Depiction of the relation between the offering, token, versions, and the offering registry.

metadata about the offering token itself (in the *i* element), usage rights and audience of this offering token, such as users in a specific organization (in the *u* element), and other offering tokens that where used in generating the current token, such as the result of a measurements activity of some material of interest (in the *l* element).

Next, to formalize the relation between the offering and the offering token, we define the *offering registry*. A depiction of the conceptual structure of the offering registry is given in Figure 5.1). Expanding on the URI technical triangle introduced at the end of Chapter 4, the offering registry introduces (formalizes) the notion of versions that identify different representations (offering tokens) that are associated with an offering.

Definition 6 (Offering Registry) *An* offering registry *is a tuple* $R = (P, \mathbf{O}, \mathbf{T}, V, \upsilon, \delta)$*, where:*

- P *is a set of participants,*
- $\mathbf{O} = \{O_1, O_2, \ldots\}$ *is a set of offerings,*
- $\mathbf{T} = \{T_1, T_2, \ldots\}$ *is a set of offering tokens,*
- $V = \{v_1, v_2, \ldots\}$ *is called a set of* versions*,*
- $\upsilon : \mathbf{O} \longrightarrow \mathcal{P}(V)$ *is called the* version mapping *function, where* $\mathcal{P}(.)$ *denotes the power set function, maps an offering to a set of versions, and*
- $\delta : \mathbf{O} \times V \times P \longrightarrow \mathbf{T}$ *is called an* token dereferencing *function. It is not defined for participants who are not the target audience of an offering/token.*

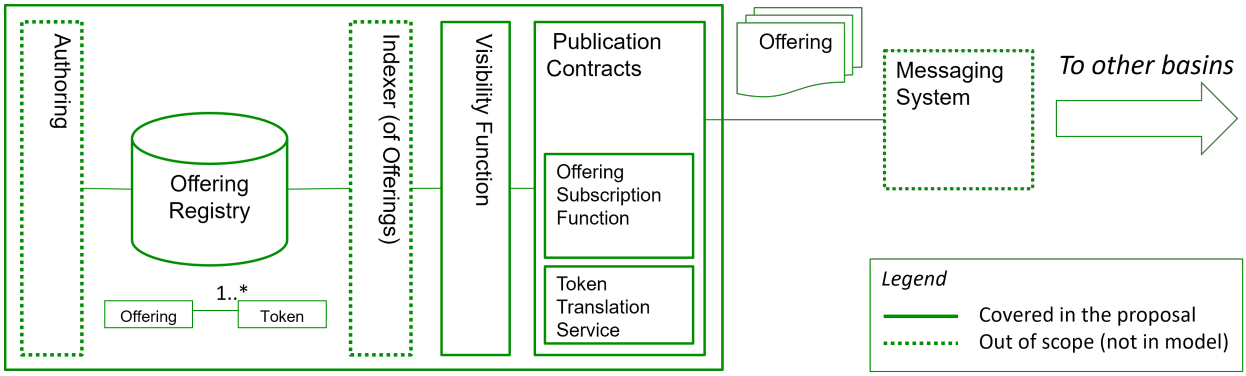


Figure 5.2: Visual aid depiction of the basin as a software architecture (suggestion only, not normative).

Definition 7 (Basin) *A* basin *is a tuple* $B = (R, d, P, o, \nu, \mathbf{C})$*, where:*

- R *is an* offering registry*,*
- d *is called the* basin description*,*
- P *is called the* set of participants*,*
- $o \in P$ *is called the* basin owner*,*
- ν *is called the* offering visibility *function, which maps participants to tokens, and*
- $\mathbf{C}$ *is a set of* publication contracts *(see below).*

We can think of the basin as a node in a distributed information system, that publishes and exchanges offerings with other basins. Figure 5.2 gives one possible software architecture of a basin[4]. Looking at the components of the basin structure, all except the last are self explanatory, or follow from previous definitions. The new novel construct here is the last element, which we define below.

Definition 8 (Publication Contract) *A* publication contract $C_i \in \mathbf{C}$ *is a tuple* $C_i = (p, \sigma, D, K, \tau)$*, where:*

- $p \in P$ *is the subscriber basin,*
- S *is the set of offerings subscribed to* $(S \subseteq \nu(p))$*,*
- $D \subset \mathbb{D}$ *is a set of* Data Model Stacks (see Definition 10) *supported by the subscriber,*
- $K \subset \mathbb{K}$ *is a set of* KOSs (see Definition 11) *supported by the subscriber, and*
- τ *is called a* token translation service*. It maps or transforms tokens for privacy, mapping, or other purposes.*

[4]The architecture as presented in the figure is for demonstration purposes only.

The token translation service can play several roles. One is an anonymization function for a token, another a *model mapping* function which takes a token adhering to a set of Data Model Stacks $D' \subset \mathbb{D}$ and KOSs $K' \subset \mathbb{K}$ and generates new token adhering to D and K. In generating a new token, a token translation service will add lineage information in the sinks element l of the new token, which denotes the actions performed and a reference to the original token. As a visual aid, Figure 5.2 depicts a basin as a node in a distributed system and gives one possible high-level software architecture for it, however, please note that the figure is for demonstration purposes only, and not an implementation recommendation!

Given the basin structure, now we can expand our running example. Say, the previously presented example of offering, that of simulation results, is part of the data driven activities carried out by one organization or group which are concerned with simulating the production process on a micro-sale (ex., electro-magnetic simulation), on a specific location within the produced part. Another group uses some of the results of the micro-scale simulations to run macro scale simulations (ex., heat transfer simulations). What the basin structure introduces is an organization of this domain into two different basins, with subscription contracts between them. Say for example, that an agent in the macro-simulation group is interested in all the offerings produced by the micro-simulation group that simulate an airplane wing part, using a specific material, and whose final state is a successful part productions. Such a desired result can be achieved using publication contracts.

Given the definitions of basins, now we can define a Basin Network.

Definition 9 (Basin Network) *A* basin network *is a tuple* $N = (P, \mathbf{B}, I)$*, where:*

- P *is a set of participants,*
- $\mathbf{B}$ *is a set of basins, and*
- I *is a Metadata Interoperability Structure* (Definition 12 below)*.*

Figure 5.3 depicts a basin network instance with four basins, with subscription contracts between them. In this example, basin B_{12} subscribes to B_1 and B_2 for tokens related to successful micro-macro simulation alignment activities that have been generated in the last 24 hours (this information can be found in $\mathbf{d}$). Let us assume all statements in B_1 adhere to KOS $k1$ and those in $B2$ adhere to KOS $k2$ and that there is a KOS $k3$ to which all statements in B_{12} adhere. We need a KOS mapping system that can map $k1$ to $k3$ and $k2$ to $k3$. Another example is that where $B3$ subscribes for successfully aligned simulation results from $B12$ (see Fig. 5.3), and so on.

With the basin network strucutre defined, we can consider the high-level definition of the model complete. What remains is the Metadata Interoperability Structure, which we define below.

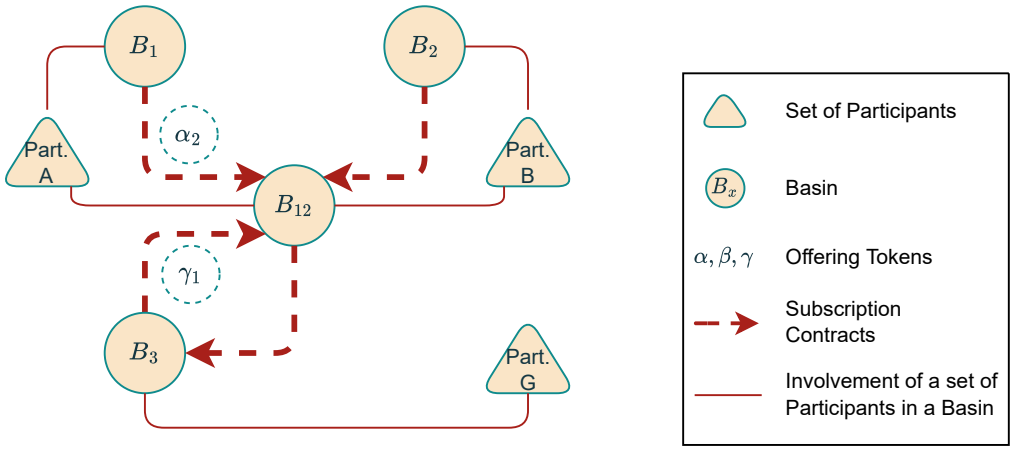


Figure 5.3: A basin network instance with four basins.

5.3.2 Metadata Interoperability Structure

The following constructs build on insights of Chapter 3. Examples are provided in Section 6.3.

Definition 10 (Data Interoperability Structure) *A* Data Interoperability Structure *is a tuple* $I_D = (D_P, D_L, D_A, B, S, T, \mathbb{D})$*, where:*

- $D_P = \{d_{p_1}, d_{p_2}, \ldots\}$ *is a set of physical data model models,*
- $D_L = \{d_{l_1}, d_{l_2}, \ldots\}$ *is a set of logical data model models,*
- $D_A = \{d_{a_1}, d_{a_2}, \ldots\}$ *is a set of abstract data model models,*
- $B = \{b_1, b_2, \ldots\}$*, where* $b_i \in D_A \times D_L$ *is called a binding,*
- $S = \{s_1, s_2, \ldots\}$*, where* $s_i \in D_L \times D_P$ *is called a serialization,*
- $T = \{t_1, t_2, \ldots\}$*, where* $t_i \in D_A \times D_A$ *is a translation, and*
- $\mathbb{D} \subset D_A \times D_L \times D_P$ *is called the set of all legal data model stacks.*

Definition 11 (KOS Interoperability Structure) *A* KOS Interoperability Structure *is a tuple* $I_K = (\mathbb{K}, M, D, \alpha, \prec, \sqsubseteq)$*, where:*

- $\mathbb{K} = \{k_1, k_2, \ldots\}$*, is a set of KOSs,*
- $M = \{m_1, m_2, \ldots\}$*, is a set of KOS modality types,*
- $D = \{d_1, d_2, \ldots\}$*, is a set of KOS definitions,*
- $\alpha : \mathbb{K} \times M \longrightarrow D$ *is a function that maps from KOSs and modality types to definitions,*
- $\preceq$ *is called the* Specializes *order. It is a partial order relation on* $\mathbb{K}$*, and*
- $\sqsubseteq$ *is called the* Imports *order. It is a partial order relation on* $\mathbb{K}$*.*

Definition 12 (Metadata Interoperability Structure) *A* Metadata Interoperability Structure *is a tuple* $I = (I_D, I_K)$*, where:*

- I_D *is a Data Interoperability Structure, and*
- I_K *is a KOS Interoperability Structure.*

Definition 13 (The Set of URIs) *A set of URIs* $\mathbb{I}$ *is a countably infinite set of URIs.*

Definition 14 (Semantic Data Set (SDS)) *Given a Metadata Interoperability Structure* $I = (I_D, I_K)$*, a* Semantic Data Set *(SDS) is a tuple* (u, S, ι, ς)*, where:*

- $u \in \mathbb{I}$ *is the URI of this SDS,*
- $S = s_1, s_2, \ldots$ *is a set of statements,*
- $\iota : S \longrightarrow \mathbb{I} \times \mathbb{I} \times \mathbb{I}$ *is a mapping from statements to a triple of URIs for the subject, relation, object, respectively,*
- $\varsigma : S \longrightarrow \mathbb{K} \times \mathbb{K} \times \mathbb{K}$ *is a mapping from statements to a triple of KOSs that model the subject, relation, object, respectively, and*
- $d \in \mathbb{D}$ *is a Data Model Stack to which the Semantic Data Set adheres to.*

5.4 Conclusion

This chapter presented a high-level formalization of the Basin Network model. The central elements of the model are the novel abstractions of the Offering and Basin and the Metadata Interoperability Structure.

The definitions presented in this chapter serve as a 'blueprint' of the model. However, there are still several areas that require further investigation.

Concerning the Offering/Token abstractions, there are several questions. What is the relation between an offering and a token, and between different tokens of a single offering? What are the details of each element of the Token?o Can they be standardized into some decomposition? For example, the Links element contains provenance between Tokens and provenance across Offerings, as well as back-links and references of the offering/token. It might be beneficial to divide the Links element into three sub-elements. In addition, what kind of relations exist between tokens of the same offering? Regarding the elements of the Token, it might also benefit from differentiating between elements about the data asset (the data) versus elements about the offering/token (the surrogate). For example, the Access and Structural element is exclusively about the data asset, whereas the Links element is exclusively about

the offering/token. The 'Integrity and Authorship' and 'Usage' elements can cover information about both: should they be mixed or separated?

About the Basin, further details are required about this system's structure and internal components. Also, the Offering Registry, which can be seen as a generalization/interpretation of a URI dereferencing system, still requires further investigation. For example, what is the relation between the service provider and the offering/token author? What about the authoring of new tokens of an offering? Should the access of offerings be limited to a specific audience, or should all offering/tokens be accessible to all, as URIs usually are?

6 Application: Computer-Aided Manufacturing

In this chapter we present the main application scenario of this work. There are two additional, smaller, applications presented in Appendix B and Appendix C as a demonstration that the model can be applied across different domain, the reader is adviced to consult these after completing this chapter.

The use case presented here comes from the domain of computer-aided manufacturing [Wu+15; KMT17; Kas+18; Ram+19; RPC19; Zie+21] in the aerospace and automotive industries in the project of SIMUTOOL[1], a HORIZON2020 project that ran from 2015 to 2019 (grant agreement number 286717). The subject of the project was the *integrated design, novel tooling, and process optimization of microwave processing of composites*. The goal was to further the state-of-the-art in the microwave processing of composites.

The approach included the use of several different kinds of simulations at different levels of granularity and using various methods such as electromagnetic field simulations, heat transfer simulations, oven simulations (etc.), along with other activities traditionally associated with production technology research lifecycle such as measurements, process control, material development and tooling, prototyping, etc. The project involved partners from eight geographically distributed locations with different areas of specialization and scopes of confidentiality, privacy, and legality, which did not know each other beforehand. It required the organization of several various couplings of data-driven cooperations with dense networks of inter-dependencies to increase the turnover time of R&D activities and accumulate reusable data resources. Take the following close-up scenario from the project to demonstrate the non-triviality of this domain.

> The task of simulating the behavior of some composite material during production is dependent on various processes. The simulation of the microwave oven itself needs to be developed to simulate the material's behavior during production. Additionally, the simulation scientists require multiple parameters and properties of the material under question, some of which are unknown to them, especially with composite materials

[1] https://cordis.europa.eu/project/id/680569

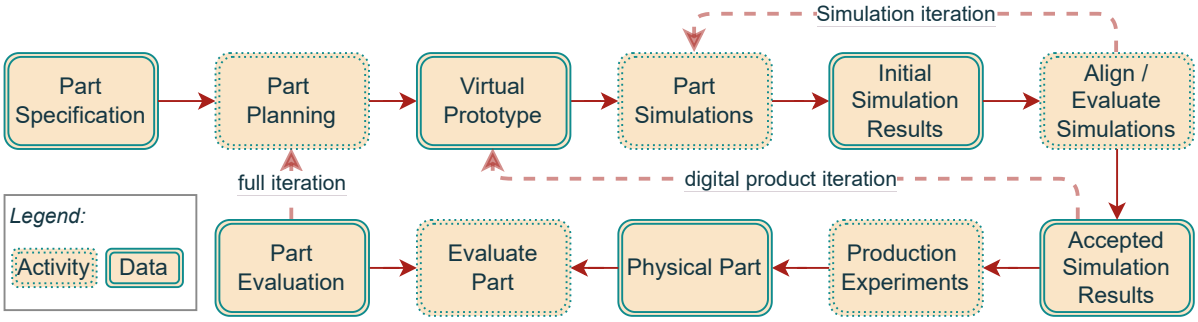


Figure 6.1: Overview of the processes in the computer-aided manufacturing domain (Adapted from [Zie+21]).

(i.e., synthetic), which can vary based on the producer and particular instance of the material. Solving this challenge involves several inputs. The end-user or product designer would specify specific requirements and specifications of the material, and the tooling or material manufacturer would provide further properties. The physical part might already be available, but there might still be properties required by the simulation that are not yet known. Some of these properties (ex., dielectric properties) can be discovered by physical experiments and measurements on the part, ...etc.

All of these complications and we are only considering a single aspect, of a simple one-material part, in a single one-time-only run, where all the participants are in the same 'room' with the same data formats and standards, and where the results and data produced are not reused and combined in other use cases, ...etc.

Fig. 6.1 depicts a simplified project process. Below is a description of some of the stages in the process with examples of data assets. The part specification produces the major requirements and desired properties (part specification, material requirements, etc) of the manufactured part. This leads to another activity which can be joined or separate to plan a '360 degrees' and the requirements needed to achieve it (ex.: Do we know the properties of the all the materials used in the part or do we need to run experiments to measure some?). A follow-up activity, *virtual prototyping*, would be to build CAD/CAM products and other digital representations of the part. This is followed by a complex, closely coupled *simulation cycle* of activities (oven simulation, process simulations, etc.), which is itself a part of a the *digital product cycle*. When a stable digital product is reached, production experiments on physical parts in an oven are carried out to verify the results so far and produce feedback to the part planning activity.

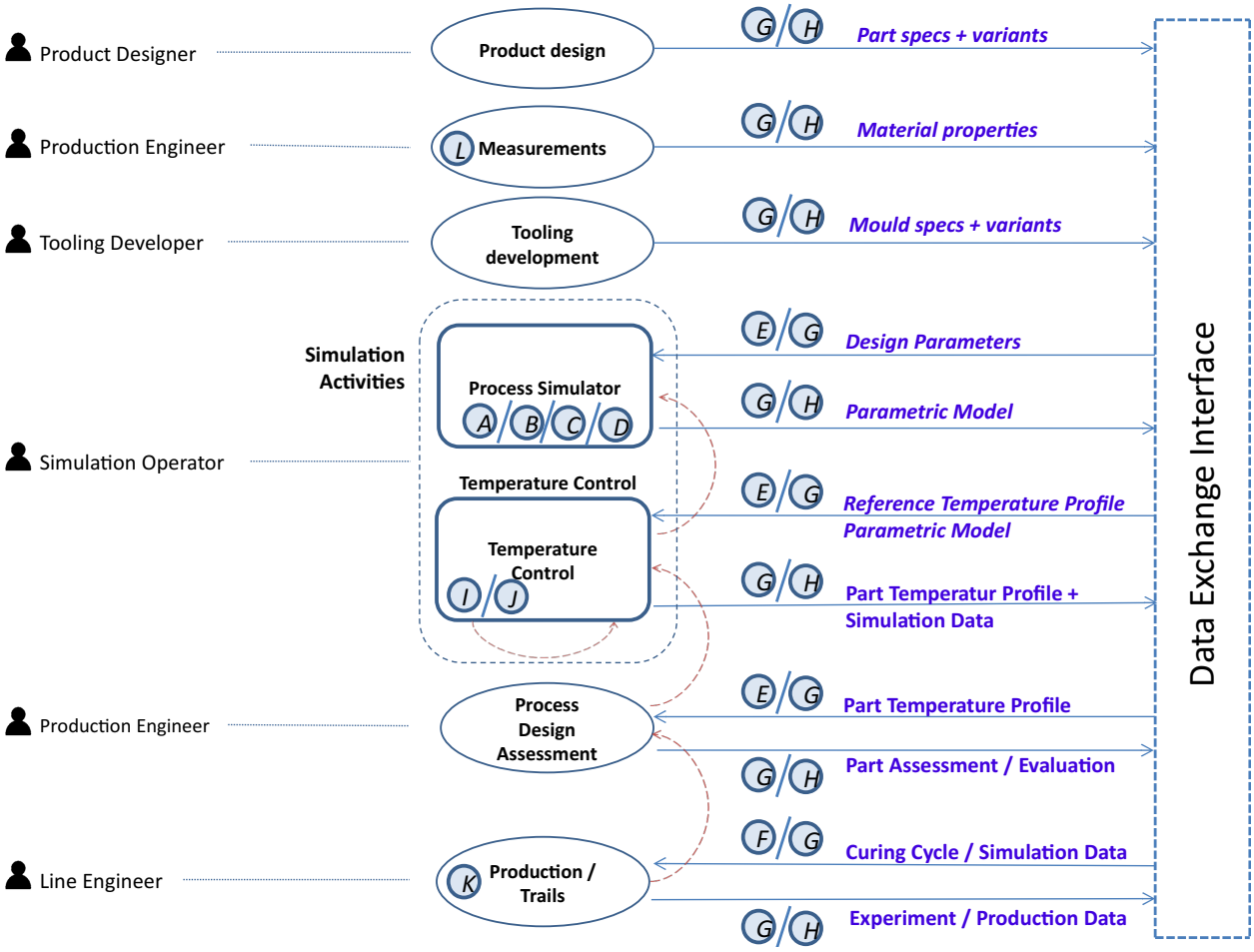


Figure 6.2: Integrated view of the SIMUTOOL EMTD process. Letters denote data producing/consuming systems in Table 6.1.

6.1 Experimental Manufacturing Technology Development (EMTD)

The efficient development of new and experimental technologies is a crucial driver of innovation and growth in production; we will call this domain *Experimental Manufacturing Technology Development (EMTD)* [Kas+18]. The EMTD domain is a sub-type of the computer-aided manufacturing domain. Experimental manufacturing technologies play a significant role in improving production processes. For example, the microwave-assisted manufacturing of composites can save energy and reduce turn-around times compared to the traditional heating in ovens. However, since this technology is not yet well-understood, it requires more research and development activities (e.g., simulation or production experiments) to enable stable and efficient production with controlled product quality. These activities span multiple divisions over (possibly) various organizations and require close cooperation and communication. In addition, this process proceeds iteratively and produces a lot of data and documents.

Most major activities in this domain are data-driven: a production experiment produces various types of sensor data, a simulation experiment has several data sets associated with it, the production part has CAD/CAM design data defining it, and others. Additionally, the amount of information about each entity grows with complexity, and, naturally, the ability to find it in the volume of all data becomes harder.

In addition, a set of challenges of this domain emerges, some of these are as follows. Groups or divisions can have their own internal heterogeneous (data-management) systems. Some of these systems might not be accessible to the outside, and others have to provide access to specific data explicitly. Additionally, there is a need to have a shared view describing all the relevant entities and their data. Data will be naturally located on various systems, and sometimes access will be provided. In others, a read-only copy can be shared; in yet another case, the raw data behind the entities might not be accessible. But in all these cases, the standard requirement is to have a description of all these items and relate them to all the relevant entities in the project. We start with the process modeling. In what follows, we describe the various roles and stages in the process.

Product Design Product design includes part specification and part evaluation. Sometimes understood as end users because they will also use the final product.

- *Role*: The product designer (a.k.a. end-user) produces product specifications and is the final part's consumer, for example in the form of CAD/CAM and specification documents describing different aspects of the desired part to be produced. They are also responsible for evaluating the final product.
- *Data Exchange*: Data exchanged in the product design includes: adding part specifications, CAD/-CAM models, and material properties.

Measurements Not all the material properties and their behavior are known beforehand. Because there properties are needed inputs for the simulation activities, they must be measured in a real-life production environment.

- *Role*: A production engineer carries out material property measurements to identify dialectical properties of the material.
- *Data Exchange*: Data exchanged are data sets and other artifacts (excel sheets, graphs, etc) which specify the several properties for the material.

Tool (Mold) Development Some of the components needed in the production process themselves require production. In collaborative projects, these will not be the same as the product designer / final part manufacturer. And the roles and the data consumed and produced will also differ.

- *Role*: Uses the specification of the part provided by the product designer to specify and manufacture a corresponding mold.
- *Data Exchange*: The activity produces little to no data, its main outcome is a physical mold to be used in experiments by later stages.

Simulation The simulation activity uses the specification of the part to be produced, which was generated from the product design activity, as well as the material properties and other parameters. Simulations and their results are added to the system with corresponding metadata.

- *Role*: Simulation scientists create and run simulations. They have access to geometrical models developed by product designers and tooling developers. The interface between simulation systems and the System can be automated or manual. A simulation operator selects the successful runs to add to the system.
- *Data Exchange*: Exchanges in the simulation activity include: consuming part, material, and oven specification data and generating data of simulation runs and curing cycles with different variants. This information can be used for future simulations or during an experiment activity. An Agent can query the part specifications, the curing cycle, and the oven specification from the previous simulation runs.

Process Assessment The experiments activity is carried out to understand the production process of the part further and to derive essential parameters for production. The experiments make use of the simulation data in the monitoring of the process. The curing cycle is one of the primary inputs for the experiments. When experiments are concluded, a new experiment activity with a curing cycle is generated. It is also possible to create multiple versions and select the definitive versions.

- *Role*: Production engineers assess curing cycles and control solutions before they are propagated to the line production stage.
- *Data Exchange*: In the scope of the experiments activity, production engineers will consume simulation runs and curing cycle information with variations of parameters. With regards to data produced, this includes experiment sensor data, control models, curing cycles, and part evaluations.

Line Production The goal is to control, monitor, and report on the production process. The produced part evaluation and sensor data of the process are vital data resources that must be communicated and integrated with the data of other groups.

- *Role*: The role of line engineers includes operating the oven and using the different data assets discovered in the system as a guide to increasing the probability of a successful production run. The Line Engineer has the authorization to add final part evaluations to the system.
- *Data Exchange*: During the production activity, the curing cycles, control models, and sensor data are retrieved from the system, and production sensor data and final part evaluation are added to the system for future reference.

Next, we describe the specific systems data-producing and consuming systems that where involved in the SIMUTOOL project.

6.2 The SIMUTOOL Project

The SIMUTOOL project involved eight participating organizations:

- TWI Limited: Production engineering research and development organization based in the UK.
- LOIRETECH SAS: Tooling[2] design and manufacturing organization based in France.
- Ecole Centrale De Nantes (**ECN**): A engineering school and research university in Nantes, France, which developed (micro-scale) simulations and solutions for the project.
- Engineering Technology Solutions (**ETS**): a company based in Greece focused on industrial measurements/automation and process monitoring.
- ESI Group (**ESI**): A company that develops virtual prototyping software for simulating products during different phases of production. Headquartered in Rungis, France, with offices worldwide.

[2] Tooling refers to various elements needed to support production manufacturing, most prominently molds which are used to shape the produced part to its desired form.

The team involved in the project was focused on the (macro-scale) curing simulation and MW oven simulation.

- University of Bamberg (**UBA**): University in Bamberg, German. The team involved in the project was focused on data and information management.
- Faurecia Automotive Composites (**FAC**): An automotive supplier of internal and decorative aspects of cars, headquartered in Nanterre, France. FAC was considered an 'end-user' in the project.
- Airbus Group SAS: European multinational aerospace corporation that manufactures and sells civil and military aerospace products (considered an 'end-user' in the project).

The SIMUTOOL project contained 12 major systems which produced and consumed shared data in the project. They are listed in Table 6.1. The sub-systems of the (now outdated) Basin Network model implementation are systems **E**-**H**. The solution concepts at the time was akin to a single basin with several systems interacting with it, mostly to add, manage, and discover 'proto-offerings.' The interaction of these systems is outlines in Figure 6.3. The Domain Model Creator is a tool to generate machine readable representations of domains models/vocabularies, from excel documents which where populated by domain modelers working with end-users. The DM Reader is tool to load the product of the previous tool into the knowledge graph service.

Table 6.1: List of data producing and consuming systems in the SIMUTOOL project

ID	*Systems*	*Owner*	*License*	*Platform*	*Description*
A	PGD EM Solver	ECN	n/a	MATLAB	3D Maxwell solver for microscopic analysis of stratified media.
B	PGD Parameterization Tool	ECN	n/a	MATLAB	For constructing a parametric model based on simulation data.
C	ESI CEM One	ESI	Proprietary	Windows/Linux	A computational electromagnetic solution for virtual testing of large-scale industrial applications .
D	ESI PAM-COMPOSITES	ESI	Proprietary	Windows/Linux	Simulator suite for modeling the manufacturing process of composite structural components.
E	Knowledge Graph Service	UBA	Apache 2.0	Python	Back-end knowledge graph store on manage and index heterogeneous data assets via metadata.
F	Online Monitoring Tool	UBA	Apache 2.0	Spring/Node.js	Visualizes sensor data, and assists user in metadata entry.
G	KMS Web interface	UBA	Apache 2.0	Python/Web2py	Web interface to manage and explore the contents of the Knowledge Graph Service.
H	Automatic Knowledge Uploader	UBA	Apache 2.0	Java	A user-assisted meta-data extraction and entry application.
I	Process Control Simulation GUI	ETS	Proprietary	Windows	Software supporting automatic and user-guided temperature and power control simulation.
J	Temperature Variation Controler	ETS	Proprietary	LabVIEW	Module providing the optimum allocation (distribution) of the MW oven power during manufacturing process.
K	Temperature Virtual Sensor GUI	ETS	Proprietary	Windows	GUI for controlling the Virtual Sensor hardware.
L	Microwave dielectric measurement	ETS	Proprietary	Windows	For controlling the microwave dielectric measurement system.

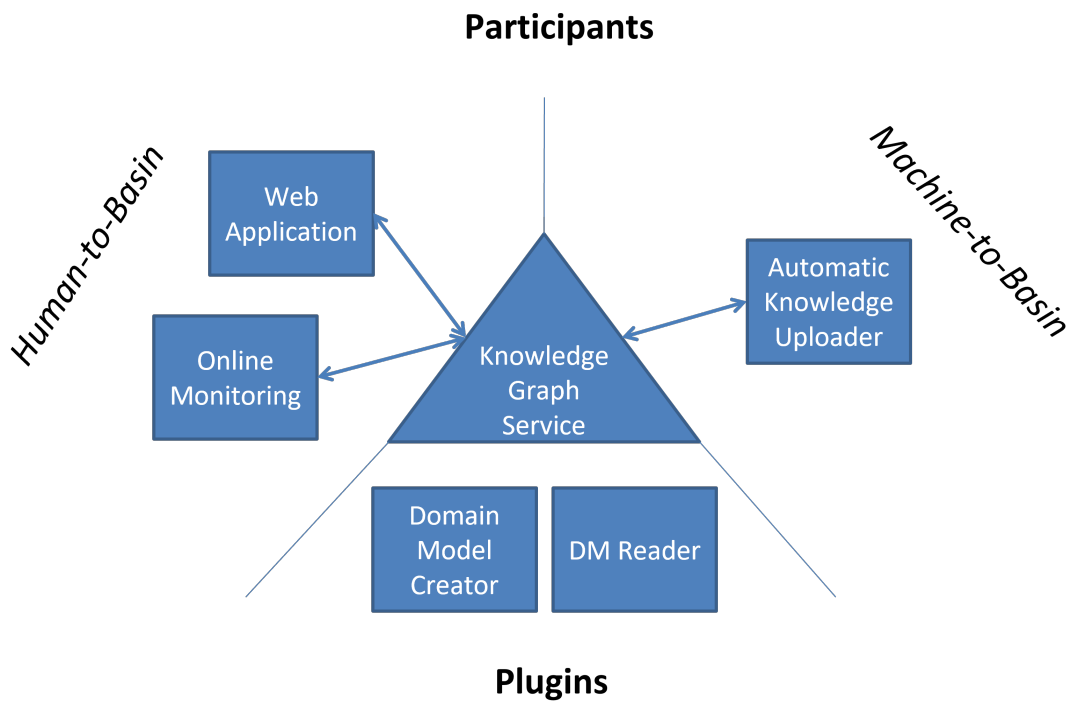


Figure 6.3: Systems around a basin

Next we show how the Basin Network model, as presented in Chapter 5, can be applied to solve this domain.

6.3 Application Scenario

In this section we demonstrate the Basin Network model by example by applying it to the EMTD domain. The application is divided into two parts: (1) introducing a KOS we developed for this domain, and (2) showing one example of each construct in the Basin Network model, starting with an example token and concluding with an overarching basin network view which shows the set of basins and subscription contracts between them. Although we will use the KOSs developed during the work (the D^4 KOS in the Appendix A and its specializations as presented below), the proposal is not tied to any such KOSs (several of the KOSs mentioned in Chapter 3 can be used).

In what follows we will use the following shorthand conventions in the example to clarity. Each offering will have an ID of the form of `off://b.net/b[0-9]+/o[0-9]+`. As a shorthand we will write this as `o[0-9]+` (ex., for `off://b.net/b12/o34` we will write `o34`. As defined in the model, an offering will have one or more offering tokens. These we will represent in the shorthand `o[0-9]+/t[0-9]+` (ex., `o61/t42`). In the graphical representation of offering tokens, the shorthand ID of each offering token will be added on top of it (such as that at the top of Table 6.3). Second to identify entities that are not offerings or tokens

(ex., people, systems, locations, etc), we will use a different ID scheme in this example with the following format: `id://[domain-name]//[path]`. It is possible that in the future both can use the same scheme, however, here, no such assumptions are made. As a human-readable shorthand for non-offerings / non-token IDs, we will display them in **`bold typewriter`** text (ex., `id://b.net/entities/ds-124` will be represented as **`ds-124`**. Furthermore, with regards declaring the KOSs and data types of statements we will use the following conventions in the example. For declaring the data model of statements in an offering token, we will assume that all statements in an offering adhere to a single *data model stack*, which will be identified in brackets following the ID of the offering token at the top of the graphical representation of token.

Furthermore, we will assume a simplified pseudo-TTL format (Turtle: Terse RDF Triple Language), in which punctuation will be skipped to readability of the examples, each statement will be a single subject, a single object, and a single relation, and on each new line a new statement is begun. Finally, in Definition 14 in the model each of the three components of a statement must state some KOS of which they adhere to, to represent this we will use the typical convention of stating namespace, in which a shorthand acronym of a KOS is written before a double column preceding each component of a statement (ex., `dct:creator` to indicate the popular DCTERMS KOS discussed in Chapter 3). At the end of the example we will list several such KOSs when giving example contents of the Metadata Interoperability Structure. Furthermore, we will also use the same convention state the type of some entity (ex., **`d4:DataSet:ds-12`** to indicate that the entity identified with ID **`ds-12`** is of the type `DataSet` which is defined in the `d4` KOS proposed in this dissertation in Appendix A). Finally, although the integrity and links elements of the offering token will typically use a simple key-value, non-semantic format, for simplicity we will assume that they we use the same data format as the first two elements, to achieve this we will use a shorthand `this` to play a similar role to the `context` element in JSON-LD.

6.3.1 EMTD Domain Model

In this section, we give a domain model for the EMTD problem. It is a specialization of the D^4 domain model shown in Appendix A. Hence, it assumes that the reader is acquainted with that KOS. Starting with activities, we can identify six significant activities in the domain: five technical and one general. Table 6.2 defines all the specialized entities modeled in the domain.

6.3.2 Token, Basin, & Metadata Interorperability Structure

Starting from the offering token presented in Table 6.3, it revolves around a data asset which is composed of two datasets. The creator of this dataset is asserted as entity **`d4:Simulator:sys-73`**, which is of type `d4:Simulator`, which itself is a type of `d4:System` as defined in the `d4` KOS given in the appendix. Furthermore, this entity has a query associated with it, that gives guidance on how to access the dataset.

Table 6.2: EMTD Domain model, specialized from the D^4 domain model given in Appendix A.

Title	Subclass_of	Description
ProductDesign	Activity	Activity to design a production part.
ToolDevelopment	Activity	Activity to develop moulds / casts.
Simulation	Activity	Activity to simulate a production process.
EMSimulation	Simulation	A example sub-type of Simulation.
Production	Activity	A streamlined production process.
Experiments	Production	A one-off production experiment.
Measurements	Production	An experiment to measure material properties.
EvaluationSucess	Activity	A successful curing activity (real or simulated).
EvaluationFailure	Activity	A failed curing activity (real or simulated).
Project	Activity	An undertaking requiring concerted effort by Agents.
Part	Entity	Represents a part to produce.
Tool	Entity	Represents a mould.
Oven	Entity	Represents an oven.
Material	Entity	Represents a material.
CarbonFiber	Material	A example sub-type of material.
Simulator	System	A simulation system.
EMSolver	Simulator	Electromagnetic simulator / solver, micro-scale.
CuringSimulator	Simulator	Manufacturing process simulated, macro-scale.
TempVarContr	System	Temperature Variation Controller of the microwave oven.
Sensor	System	A system to transform a physical phenomenon into a reading.
VirtualSensor	System	Transforms a sensor-based value to another domain.
SimulatedSensor	System	An algorithm to simulate a sensor.
PartData	Data	Defines a Part to be produced.
MaterialDB	DataSet	A Dataset defining a set of material properties.
MonitoringData	DataSet	A Dataset produced during a Experiments activity.
FreqMeasurements	Document	Microwave frequency readings from a Production.
EMSimData	DataSet	Micro-scale simulation data.
CuringSimData	DataSet	Macro-scale simulation data.

Table 6.3: An offering token.

o36/t921 (*Pseudo-TTL*)		
	Access / Structural	
d4:CuringSimData:asset-157	dct:hasPart	**d4:DataSet:ds-12**
d4:CuringSimData:asset-157	dct:hasPart	**d4:DataSet:ds-13**
d4:DataSet:ds-12	dct:creator	**d4:Simulator:sys-73**
d4:DataSet:ds-12	dct:requires	**d4:Query:q-23**
d4:Query:q-23	d4:queryString	``jdbc:oracle:thin:@hostname:1521:db''
d4:DataSet:ds-13	dct:creator	**d4:Simulator:sys-73**
	Description	
d4:CuringSimData:asset-157	dct:creator	**d4:User:yann-d-73**
d4:CuringSimData:asset-157	dct:subject	**d4:Part:p-19**
d4:CuringSimData:asset-157	d4:used	**d4:MaterialDB:mbd-12**
	Integrity / Authorship	
this	dct:creator	**d4:System:knowledge-uploader**
this	dct:audience	**d4:Agent:ecn**
this	dct:rightsHolder	**d4:Agent:esi**
	Links	
this	prov:wasDerivedFrom	o30/t41
this	dct:replaces	o36/t921
this	dct:requires	o21/t2

Moving on to the second element, the *Description*, it contains context information about the data asset. Here some examples are given such as creator, and the subject being a specific manufacturing part, and the *material database* used in the simulation (machine readable document that states properties of key materials). The third element, *Integrity / Authorship*, is concerned with the authoring of the token (i.e., meta-metadata). The token above has been authored by the a system with the ID **knowledge-uploader**. It states a rights holder and an audience (an agent, which in this example, is a single organization). Finally, the *Links* element includes relations to other tokens, one in which the current was derived from, another is is replaced by the current token, and one that is required by the current token.

An example of an offering registry that includes the token above as well as other tokens and offerings can look as follows. $R_{12} = (P_{12}, \mathbf{O}_{12}, \mathbf{T}_{12}, V_{12}, \upsilon_{12}, \delta_{12})$, where:

- $P_{12} = \{$**d4:User:yann-d-73**, **d4:User:eric-21**, …$\}$ is a set of participants,
- $\mathbf{O}_{12} = \{$o36, o37, … $\}$ is a set of offerings,
- $\mathbf{T}_{12} = \{$o36/t921, o36/t920, o37/t33, … $\}$ is a set of tokens,
- $V_{12} = \{$t921, t920, t33, …$\}$ is a set of versions,
- $\upsilon_{12} = \{$(o36, {t921, t920}), (o37, {t33}), … $\}$ is a version mapping function,
- $\delta_{12} = \{$((o36, t921,**d4:User:yann-d-73**), [[o36/t921]]), …$\}$ is an offering derenferencing function.

We have two example participants, one of which is **d4:User:yann-d-73**, which mean that the ID is yann-d-73, and is of type d4:User, so this means there is a KOS d4 defined in the metadata interoperability strucutre, which defines a type User. In the second elements we have a set of offerings. o36 is the offering of which the token given in Table 6.3 is associated with it. Next the set of tokens, o36/t921 we can think of as the ID of the token given in the Table above. Then we have the set of versions. As discussed in the formalization chapter, an offering and a version are associated with a token, and there are usually many versions associated with one offering meaning that there are many tokens for each offering (although not precise, but it might help to think of tokens as frozen, time-stampted, snapshots of different values associated with a variable, the offering, over time). The version mapping function[3], which maps an offering to what is similar to a set of 'version numbers.' So, for example, offering o36 mentioned above, as per this example, has two 'versions.' Finally, for the last element, the concept of the offering dereferencing function is hard to demonstrate on paper, one way to visualize it is to picture something like Table 6.3 in place of [[o36/t921]], so then given an offering, version, and participant, this function 'evaluates' to some data structure with specific values.

Given the offering registry above, we can construct a basin as follows. $B_{1b} = (R_{1b}, d_{1b}, P_{1b}, o_{1b}, \nu_{1b}, \mathbf{C}_{1b})$, where:

- R_{1a} is the offering registry defined above,
- $d_{1a} = \{(\texttt{this}, \texttt{dct} : \texttt{subject}, \texttt{d4} : \texttt{EMSimulation}), \ldots\}$ is a basin description,
- $P_{1a} = \{$**d4:User:yann-d-73**, **d4:User:eric-21**, **d4:Agent:b12:basin-simulations**, $\ldots\}$ is a set of participants,
- $o_{1a} =$ **d4:User:yann-d-73** is the owner of the basin,
- $\nu_{1a} = \{($**d4:Agent:basin-simulations**, $\{\texttt{o37/t33}\}), \ldots\}$ is an offering visibility function,
- $\mathbf{C}_{1a} = \{C_1, C_2, \ldots\}$, is a set of publication contracts

We will look at the specifics of one basin in depth, in particular, basin B_{1b}, which is one of two basins operated by the micro-simulation party. This one is used to publish stable micro-simulation data (i.e., 'production-ready'). We will see below how this basin publishes offerings to another basin **d4:Agent:b12:basin-simulations** which is where stable micro-simulation data is managed along with stable macro-simulation data to create complete simulations of some part (this basin refers to B_{12} in the full basin network scenario explained at the end of this section). The elements of this basin can be demonstrated as follows. The first is the offering registry discussed above. The second is a basin description, we can think of it as a set of triple that state the subject of the basin and other information helpful to discover and understand the purpose of the current basin. This is followed by a

[3] In this example, for brevity, the convention of representing functions and mappings as a set of tuples (ordered sets) is used. For example a function with one set as domain and one as range can be represented as a set of pairs where the first is from the domain, the second from the range.

set of participants in the basin, one of which, **d4:Agent:b12:basin-simulations**, identifies a participant that is another basin, in specific this is the basin B_{12} in Figure 6.4 which we will circle back to in the following parts of this example. Then we have an owner of the basin. After that we have the offering visibility function, which maps a participant to a set of offering they are allowed to read. Finally we have a set of publication contracts where each contract is a tuple $C_1 = (p_1, S_1, D_1, K_1, \tau_1)$, where:

- p_1 = **d4:Agent:b12:basin-simulations** is the subscriber basin,
- $S_1 = \{\texttt{o37/t33}\}$ is the set of offering subscribed to,
- $D_1 = \{\texttt{Pseudo} - \texttt{TTF}\}$ is the set of data model stacks supported by the subscriber,
- $K_1 = \{\texttt{d4}, \texttt{dct}, \texttt{prov}, \ldots\}$ is the set of the KOSs supported by the subscriber,
- τ_1 is a token translation service which we will assume is the identity function for the sake of brevity.

We can demonstrated the content of a publication contract as follows. The first element being the subscriber basin **d4:Agent:b12:basin-simulations**, the same one we highlighted above. The second elements is the set of offerings subscribed to. The third is the set of data model stacks supported by the subscriber. For brevity of this example, we will assume both basin support the same data model stack (namely, `Pseudo-TTF`). Finally, we will also assume that token translation service which we will assume is the identity function for the sake of brevity (i.e., it maps a token instance to itself).

With the basins, offerings, and subscription contracts covered above, the so-called core of the model is covered. What remains is the *metadata interoperability structure*, which is responsible for supporting semantic heterogeneity of the contents of offerings (metadata), and then the basin networks structures which is simply a triple of: a set of participants, a set of basins, and a metadata interoperability structure. We will give an example of the latter first and then put it all together by showing a visualization of network to model the EMTD domain composed of sets of participants, the basins the operate, and publication contracts between those. The metadata interoperability structure of this use case is a tuple $I_0 = (I_{D0}, I_{K0})$, where I_{D0} is a data interoperability structure, and the I_{K0} is a KOS interoperability structure. The data interoperability structure is a tuple $I_{D0} = (D_{P0}, D_{L0}, D_{A0}, B_0, S_0, T_0, \mathbb{D}_0)$, where:

- $D_{P0} = \{\texttt{Pseudo} - \texttt{TTL}, \texttt{TTL}, \texttt{JSON} - \texttt{LD}, \texttt{BSON}, \texttt{BEACON}, \texttt{XML}, \ldots\}$ is a set of physical data models (i.e., serializations),
- $D_{L0} = \{\texttt{RDF} - \texttt{XML}, \texttt{DC} - \texttt{RDF}, \texttt{XML}, \texttt{PROV} - \texttt{DM} \ldots\}$ is a set of logical data models,
- $D_{A0} = \{\texttt{RDF} - \texttt{Abs}, \texttt{XML} - \texttt{Abs}, \texttt{DC} - \texttt{Abs}, \texttt{PROV} - \texttt{ADM} \ldots\}$ is a set of abstract data models,
- $B_0 = \{(\texttt{RDF} - \texttt{Abs}, \texttt{RDF} - \texttt{XML}), (\texttt{XML} - \texttt{Abs}, \texttt{RDF} - \texttt{XML}), (\texttt{PROV} - \texttt{ADM}, \texttt{PROV} - \texttt{DM}), \ldots\}$ is a set of bindings,
- $S_0 = \{(\texttt{RDF} - \texttt{XML}, \texttt{TTL}), (\texttt{XML}, \texttt{XML}), (\texttt{RDF} - \texttt{XML}, \texttt{JSON} - \texttt{LD}), (\texttt{RDF} - \texttt{XML}, \texttt{BEACON}), \ldots\}$ is a set of serializations,

- $T_0 = \{(\texttt{PROV}-\texttt{Abs}, \texttt{RDF}-\texttt{Abs}), (\texttt{DC}-\texttt{Abs}, \texttt{RDF}-\texttt{Abs}), \ldots\}$ is a set of translations,
- $\mathbb{D}_0 = \{(\texttt{RDF}-\texttt{Abs}, \texttt{RDF}-\texttt{XML}, \texttt{TTL}), (\texttt{PROV}-\texttt{Abs}, \texttt{PROV}-\texttt{DM}, \emptyset), \ldots\}$ is a set legal data model stacks.

For the data interoperability structure, the first element is a set of physical data models (called serializations in the web context). Examples include TTL[4], BEACON[5] link dump format, JSON-LD, etc. The second element is a set of logical data models such as RDF-XML, PROV-DM[6], etc. Then we have a set of abstract data models (see Chapter 3 for discussion about those). Then we have bindings, which associate pairs of abstract to logical data models, and serializations which associate logical data models to physical data models (i.e., legally defined serializations of some data model). Then, we have legally defined translations from one data model stack to another (when such exists). Finally, we have a set of legal data model stacks, which can be derived from the earlier elements, but is convenient to have explicitly.

A KOS interoperability structure is a tuple $I_{K0} = (\mathbb{K}_0, M_0, D_0, \alpha_0, \prec_0, \sqsubseteq_0)$, where:

- $\mathbb{K}_0 = \{\texttt{d4}, \texttt{dct}, \texttt{prov}, \texttt{pav}, \ldots\}$,
- $M_0 = \{\texttt{data}-\texttt{model}, \texttt{formal}-\texttt{def}, \texttt{primer}, \texttt{schema}, \texttt{constraints}, \ldots\}$ is a set of KOS modality types,
- $D_0 = \{\texttt{prov}-\texttt{dm}, \texttt{prov}-\texttt{n}, \texttt{d4}-\texttt{xsl}-\texttt{readable}, \texttt{d4}-\texttt{python}-\texttt{pickle}, \texttt{dc}-\texttt{rdfs}, \ldots\}$ is a set of KOS definitions,
- $\alpha_0 = \{(\texttt{prov}, \texttt{data}-\texttt{model}, \texttt{prov}-\texttt{dm}), (\texttt{dc}, \texttt{formal}-\texttt{def}, \texttt{dc}-\texttt{rdf}), \ldots\}$ is a KOS to modality mapping,
- $\prec_0 = \{(\texttt{pav}, \texttt{prov}), \ldots\}$ is the specialization order over $\mathbb{K}_0$,
- $\sqsubseteq_0 = \{(\texttt{d4}, \texttt{dct}), (\texttt{d4}, \texttt{prov}), \ldots\}$ is the imports order over $\mathbb{K}_0$.

For the last structure in this example, the KOS interoperability structure above includes the following elements. The first is a set of KOSs (think of them as namespaces), the second is a set of KOS modality types. As discussed in Chapter 3 KOSes are usually multi-faceted artifacts that require different types of specification and communication formats, we called these KOS modalities. Here we have some types typical types: `data-model` which refers to a formal (usually a human readable document, but can also be machine readable in some cases). Examples of this are the PROV-DM data model referenced above. There are others such as schema (using, for example, RDFS), constraints (SHACL[7] is one tool usually used towards this end), and so on. In the future it will be helpful if an ontology of such modalities is

[4] https://www.w3.org/TeamSubmission/turtle/ (Accessed December, 2022)
[5] https://github.com/gbv/beaconspec (Accessed March 2023)
[6] https://www.w3.org/TR/prov-dm/ (Accessed December, 2022)
[7] https://www.w3.org/TR/shacl/ (Accessed December, 2022)

developed. Next element D_0, is a set of definitions, think of these as the instances of the modality types in M_0. The next element (α_0), maps a KOS and a modality type to the definition. Finally, we have two orders over K_0, one, the specialization order, and the other imports order. The first specifies KOSs as specializations of another, for example, PAV[8] (Provenance, Authoring and Versioning) states on its homepage that it is a specialization of the PROV KOS. The second order, specifies when some KOS imports the whole of another KOS. The D^4 proposed in this thesis (Appendix A), imports the Dublin Core Terms KOS and the PROV KOS.

Given the examples of each of the constructs, we can now define a basin network instance as a triple $N_0 = \{P_0, \{B_{1b}, B_{12}, \ldots\}, I_0\}$, where P_0 is the union of all sets of participants of all basins in the network, the set of all basins, and finally the metadata interoperability structure defined above.

6.3.3 Basin Network: Putting it all Together

Looking at an overarching picture of the basin network instance we look at what basins and subscriptions between them do we need to model the EMTD domain. Figure 6.4 depicts one possible way to model a basin network for this use case. Triangles represent parties (a group of related participants), circles represent basins, solid thin lines denote that a party participates in a basin, and dashed bold arrows indicate subscriptions between basins. Textual titles are added for clarification purposes; more basins can be created based on emerging needs and groupings. We describe the network below.

Micro-Simulation Cluster Starting from the top left corner, we have the party $P1$, which stands for the party responsible for micro-scale simulations (e.g., electromagnetic simulation). They are participants in three basins, two of which they are owners of (B_{1a}, B_{1b}), and one of which they share with another party (B_{12}). B_{1a} is the basin used to manage prototype simulations being developed that are still not stable or complete. We can call basins with no publication contracts to to external organizations *private basins*. Basin B_{1b} is used to manage production-ready micro-simulation data. We can see that B_{1b} subscribes to B_{1a}, and this subscription is concerned with offerings that denote production-ready simulations. This kind of information can be easily represented in the statements of the Offerings. Most simply, this can be a binary property that means that an Offering is production ready.

We have two basins because the processes for prototyping and preparing early-stage simulations differ from managing stable simulations. Also, there might be a need sometimes to compartmentalize different data for organizational purposes.

Organizational purposes can be internal or external. Regarding internal organizational purposes, if the micro-simulation party was a one-person party, for example, it might be better to merge both B_{1a}

[8] http://pav-ontology.github.io/pav/ (Accessed December, 2022)

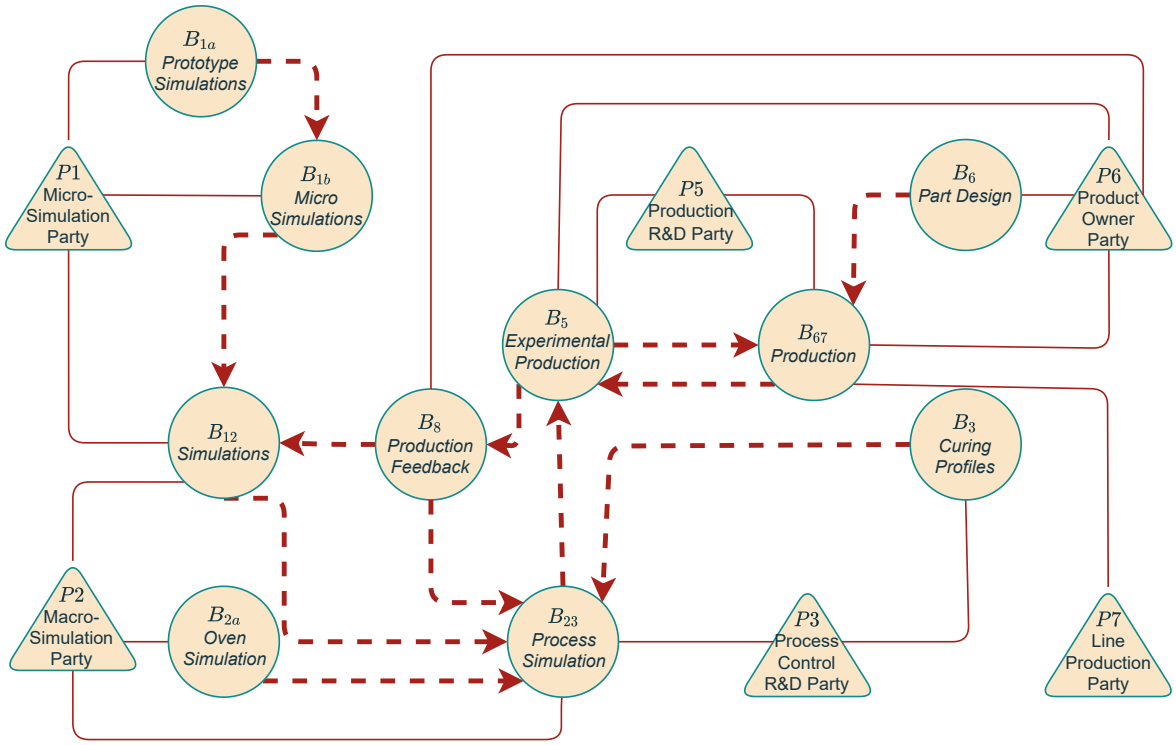


Figure 6.4: A basin network instance for the computer-aided manufacturing domain outlined above.

and B_{1b}. And if it were a division of a global organization that is large and modular, then it would be more desirable to split the two basins further, and so on.

External organizational purposes usually relate to collaboration dynamics between two parties participating in a shared basin. The external party should not have access to all the data resources of the party, so isolating common resources or subscribing to them into a basin and leaving more resources in private basins might be desirable in some cases.

Macro-Simulation Cluster Basin B_{12} is shared by both $P1$ and $P2$ (responsible for developing maro-simulations). It is a single point of truth for all simulation data; it contains the results for a complete *simulation iteration* (see Figure 6.1). It contains the results of successfully aligned simulations which include: micro-scale simulations, macro-scale simulations, oven simulations, as well as process simulations, each revolving around one simulation scenario. The oven simulation data is handled in B_{2a}, which is also operated by the macro-simulation party. Basin B_{23} is a shared basin for managing process simulation data, shared between the party mentioned above and the Process Control R&D party ($P3$). This basin also subscribes to the Oven Simulation basin (B_{2a}) to receive successful oven simulations about relevant entities.

Process Control R&D Cluster Introduced above, the Process Control R&D party ($P3$) is responsible for devising procedures and protocols for operating the oven during production. It is a party whose

role lies at the intersection of simulation and experimental production. The Process Control R&D party operates another basin in which it publishes stable curing profiles, an essential ingredient for the experimental production process.

Production R&D Cluster The Production R&D party ($P5$) is on the interface between experimental and stable line production. This can be seen when observing that this party participates in the experimental production basin (B_5), in which it solely operates, and the production basin (B_{67}), in which it participates. The experimental production basin is a hub for the critical data resources related to the explorative R&D phase. It is where the product owner party adds early part designs and requirements. Fully aligned simulations, curing cycles, and experiment results about a single production scenario can be found.

Product Owner & Line Production Clusters The product owner party ($P6$) defines the required directions and use cases. It participates in three basins, two of which it operates (B_{67}, B_6), and one of which it is a participant (B_5). Basin B_6 is centered around product designs and specification data, and basin B_{67} is a basin to manage production data. It involves the Production R&D party and the Line Production party (P_7). The Line Production party ($P7$) is responsible for evaluating the produced parts.

6.4 Conclusion

The use case presented above (and the two others in the Appendix) have varying structures and requirements, which lead us to observe different sides of the Basin Network model, showing us various features the model affords.

The computer-aided manufacturing use case involves a sophisticated web of interdependencies with varying levels of granularity (a material, a production cycle, a part). It is driven more by humans than machines, acting more like an interorganizational 'knowledge management system' than a data management system. The digital agriculture use case (Appendix B) involves a slightly simpler structure primarily driven by machines. It looks more like an inter-technology data lifecycle integration system, similar to an *enterprise integration bus* use case. In the final use case, that of integrating IoT functionalities in the scope of live festival management (Appendix C), we view the Basin Network as playing a slightly different role as an overarching, interorganizational architecture for data management.

Regarding the structural modeling of a BNet instance, one limitation of static use cases presented on paper is that it is hard to show how the structure has been arrived at and the pros and cons of each configuration. The arrival to a specific structures such as those depicted in Figures 6.4, B.1, and C.1 involved an iterative process, which included three actions: create a basin, merge two basins, split

a basin into two. Some of these actions will involve coordination across several organizations, and others are related to the functioning of a single organization. Therefore, in implementing the Basin Network model, the merging and splitting of basins must be a technically straightforward task from the users' perspective. One way to achieve that would be to separate the functionality from the content of basins and perhaps propose a standard to represent the contents of basins. Another related question is whether a set of design patterns for basin network modeling will emerge in practice, or whether it will be better to let the design of the network evolve organically.

Turning over from (infra)structural observations to *content-related observation*; *delegating the responsibility of [offering] metadata creation/curation as well as the discovery (of Offerings) to the (manual efforts of the) author/user, is an unproductive and perhaps misled approach.* We can think of three complementary strategies which can mitigate this challenge. In order of difficulty, these are: (1) the separation of concerns between human-assisted and non-human-assisted metadata, (2) metadata generation and assistance, and (3) social 'semantics' of web-of-data Resources.

The modularization of the token offering (Definition 5 in Chapter 5) into thematic metadata components was motivated by the first approach. For example, the Description element requires the most human involvement. In contrast, the others (access and structural, integrity and authorship, usage, etc.) can be populated mainly by data-producing systems, embedded metadata, or organizational-internal rules. The internal structure of these elements can be further subdivided when possible to achieve further separation of concerns.

Metadata generation and assistance is a domain-specific problem; we have experimented with simple solutions in the SIMUTOOL project, such as the Automated Knowledge Uploader daemon/wizard and the Online Monitoring system, with varying results. This problem is out of the scope of this work. It is another area for further investigation.

The 'Social Semantics' of Resources is to the web-of-data Resources what search engines/folksonomy/-social web have been to the web-of-documents. This is another fruitful direction of research.

One final observation from the use cases. Adding *yet another system* in the technological pipeline of organizations can face serious adoption challenges. Hence, this might lead one to ask whether a proposal for data exchange can be formulated in a way other than an information system. We come back to these issues in the conclusion chapter.

7 Evaluation

In this chapter, we evaluate the proposal with respect the requirements extracted earlier. We do that on two levels: in comparison to approaches that adopt an *indirection* approach, which we call Data Domain Topology approaches, which we call the *sub-problem*, and to more general approaches that target the problem of data sharing and exchange irrespective of approach, which we call the problem.

7.1 The Sub-Problem: Indirection & Data Domain Topology

In this section, we show how the Basin abstraction meets the surrogate management space requirements introduced in Chapter 2. We also show how the Basin abstraction is a generalization of data catalogs, lakes, ponds, zones, and data spaces. We start with the The Data Catalog Vocabulary[1] (DCAT). Although DCAT is a W3C recommendation intended as a vocabulary to facilitate interoperability between data catalogs, we can see it as a model for data catalogs. Hence we will show how the Basin & Offering abstractions (and related structures) cover the technical scope of the DCAT.

Table 7.1 reproduces the Table discussed in Section 4.2, and adds the BNet model into the table. With regard to the space membership condition in the BNet model, there are no constraints on the contents of a basin in the model, and the number of basins in a network. Inter-space exchange is supported by the publication contracts element. With regard to privacy consideration, the BNet model includes two components related to privacy, although their details are left for future work. The first is the offering visibility function in the basin, where the visibility of offering to participants can be controlled, and the second is the Token Translation Service in each Publication Contract where processes such as anonymization of the content of offering tokens can be carried out (access policy + anonymization). Inter-space curation is supported by the publication contracts as well as the common distributed identity scheme (URI-like). Finally, semantic heterogeneity is supported by the Metadata Interoperability Structure.

[1] https://www.w3.org/TR/vocab-dcat-2/ (Accessed 08.2022)

Table 7.1: The Basin as compared to data catalogs, lakes, zones, ponds, and spaces

Architecture	*Space Membership Condition*	*Inter-Space Exchange*	*No. of Spaces*	*Privacy*	*Inter-space Curation*	*Semantic Heterogeneity*
Lake (functional)	Stage of data lifecycle	preset	<10	no	no	no
Zone	Refinement	preset	<10	no	preset	no
Pond	Type/nature	no	<10	no	no	no
Lake (hybrid)	Refinement or type	preset	<20	no	preset	no
Data Catalog	Arbitrary Scope	no	1 per org.	no	yes	no
Data Space	Organizational Scope	yes	1 per org.	yes	yes	no
Basin Network	**Arbitrary Scope**	**Yes**	**Arbitrary**	**yes**	**yes**	**yes**

7.2 The Problem: Interorganizational Data Management & Exchange

In this section, we take a step up the ladder to the main problem and its requirements. We show how the proposed work meets the requirements as opposed to similar work. We introduce the approaches, present a classification based on the design space, and discuss requirement satisfaction.

7.2.1 Approaches

There are several works influenced by the data management challenges faced in digital manufacturing and related industries [RPC19]; however, none of them is a complete solution yet, and many remain research visions. These will not be included here. These include for example efforts to establish agendas for cloud-based design and manufacturing [LR15; Wu+15], establishing virtual manufacturing enterprises for collaboration [Rei+11; KMT17; Zie+21], semantic data management for manufacturing [Kas+18], viewing the problem as an Internet of Things problem in the production environment [Pen+19a; Pen+19b; BM20; Gle+20; KHM20; LJ20; Kas+21]

We will consider four approaches: the International Data Spaces (IDS) project, the scientific data management solutions, Research Objects (RO), and Data.World as a representative of similar commercial online solutions.

International Data Spaces (IDS) The International Data Spaces (*IDS*) includes a cluster of activities such as publications [JQ17; OJ19; Bad+20], an association[2], and funded projects. Some information about the model can be found in the "IDS Reference Architecture Model" white paper [Ott+18]. It adopts a highly centralized form of governance. It appears the project is in development phase with an emerging Github group[3], so it is not straightforward to identify the specifics. It is centralized, with high governance, and is specialized to some degree.

Scientific Data Management Scientific, Research Data Management, and Research Data Repositories (*RDR*) [Amo+17] are either platforms that (1) support research data workflows or (2) are data repositories. Another trend has been open science [FF14; Mon+17] and the support of data-intensive research disciplines [OJW08; Edw+11]. The class of works that aim at supporting scientific workflows [Ber+05; Gra+05; Atk+17; Dee+18; ZMW20] is an early influence on many of the works discussed here. It has spawned early question about provenance and identification [SPG05], and has produced architecture recommendations [Li+13; Che+00], as well as questions about data citation principles [Fen+19], and the FAIR (Findable, Accessible, Interoperable, Reusable) data principles [Wil+16; Jac+20].

[2] https://internationaldataspaces.org/ (Accessed 11.2021)
[3] https://github.com/International-Data-Spaces-Association (Accessed 11.2021)

Regarding research data repositories, there are general purpose ones such as Zenodo[4] and specialized ones such as Agri-Environmental Research Data Repository[5]. There are also tools to manage and build repositories, such as CKAN[6], which offer website content management tools to create research data repositories.

Research Objects Research Objects[7] [Bec+10; Fen19] (*RO*) where introduced in 2010 with recent attempts at standardization[8]. ROs are a format to package research data. It is a framework for packaging data alongside metadata that describes it. Think of the BagIt IETF file specification format[9] (RFC 8493) and the notion of *manifest files* applied to the case of exchanging and describing dataset files. ROs do not include infrastructure considerations, and their current format is a JSON-LD schema specification.

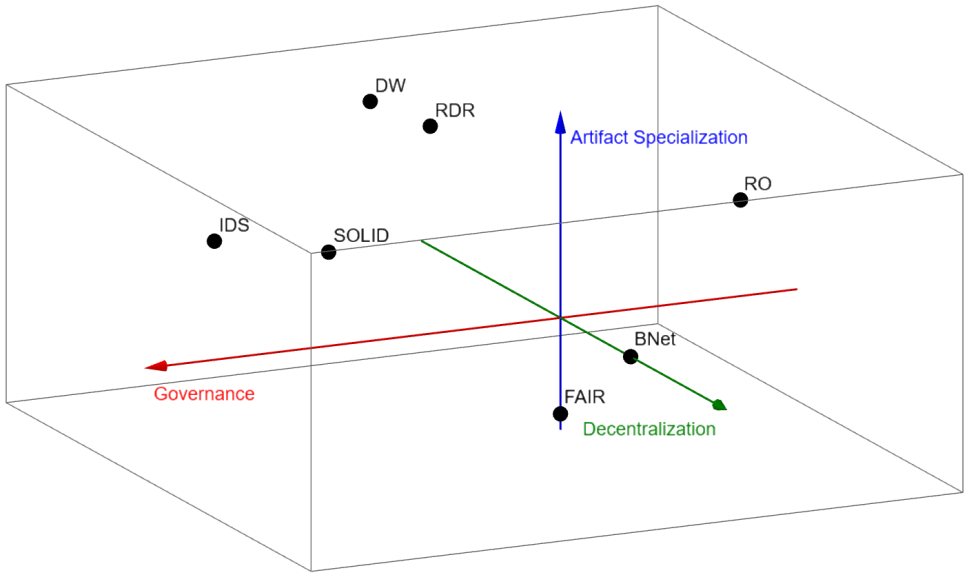
Data.World Data.world[10] (*DW*) is a web-based commercial data cataloging service. It is a data publishing service with analytics integration options and a simple social network structure. Annotation is usually restricted to a set of standard data cataloging and profiling vocabularies (ex. CSVW[11], VoID[12]). The service is designed with a specific use case; using it in general use cases such as custom annotation vocabularies or advanced provenance is not possible.

7.2.2 Design Space

Whereas requirement satisfaction (discussed in the following subsection) is intended in a prescriptive/normative sense (i.e., what should be), the design space presented here is descriptive. Based on where the different approaches fall in the design space, they inherit particular properties with pros and cons. We will consider three dimensions in this space: *Artifact Specialization*, *Decentralization*, and *Governance*.

Artifact specialization Artifact specialization relates to domain and technology specialization, with the 'artifact' being the proposed approach. Proposals that are too specialized are usable in a single domain and dependent on some technology stack. This makes cross-domain data sharing and exchange not viable and ties the future of the artifact to a specific technology. Too little specialization has its challenges too. Take FAIR data sharing principles, for example. They are the least specialized approach here; however, they include little structures or abstractions that guide constructing technical systems.

[4] https://zenodo.org/ (Accessed 11.2021)
[5] https://dataverse.scholarsportal.info/dataverse/ugardr (Accessed 11.2021)
[6] https://ckan.org/ (Accessed 11.2021)
[7] https://www.researchobject.org/ (Accessed 11.2021)
[8] https://www.researchobject.org/ro-crate/ (Accessed 11.2021)
[9] https://www.rfc-editor.org/rfc/rfc8493 (Accessed 08.2022)
[10] https://data.world/ (Accessed 11.2021)
[11] https://www.w3.org/ns/csvw (Accessed 11.2021)
[12] https://www.w3.org/TR/void/ (Accessed 11.2021)



DW	Data.World	RO	Research Object
RDR	Research Data Repositories	BNet	Basin Network
IDS	Internation Data Spaces	FAIR	FAIR Principles

Figure 7.1: Positioning of related work. Approaches on the 0-point of the x- or y-axis denote they are neutral / application specific.

Specialized artifacts are usually packaged services or solutions targeted at non-technical end-users, whereas ones with a low level of specialization are targeted at scientists and technology developers.

Decentralization Decentralization, an old concern, has recently moved back into the forefront, with recent efforts in the direction of decentralized social web and identifiers [Hal19]. This dimension is helpful to include here because it is related to governance (see below) and because different domains favor different flavors of decentralization.

Governance We can define (data) governance as "what decisions must be made to ensure effective management and use of IT [...] and who makes the decisions [...]" [KB10]. Governance and decentralization are not entirely separate problems. Khatri and Brown mention different governance domains which fall on various points of the decentralized/centralized spectrum [KB10]. The SOLID project [Ver20] for personal data management on the web, for example, is decentralized but with a high level of governance on the decentralized 'leaves' of the model where every person chooses how their data is transferred and processed by others. Usually, decentralized proposals allow for centralized and decentralized governance, whereas centralized approaches are limited to centralized governance.

Table 7.2: Requirement satisfaction by similar work (? = unclear/unknown). **RO** = Research Objects, **RDR** = Research Data Repositories, **IDS** = International Data Spaces

Requirement	RO	RDR	Data.World	IDS	Basin Network
R1 Infrastructure Independent	✓				✓
R2 Supports Interorg. Curation	✓		✓	✓	✓
R3 FAIR		✓	(?)	✓	✓
R4 Meets Data on the Web Best Practices				(?)	✓
R5 Foundational	✓			✓	✓

Fig. 7.1 depicts similar work within this space. Research Data Repositories are usually specialized artifacts, centralized, with a small set of governance options. They usually go in the opposite direction of generalist solutions[13]. Research Objects include no governance, are neutral to decentralization, and are very specialized. The International Data Spaces (IDS) is centralized, with high governance, and is specialized to some degree. The *SOLID* project [Pol+20; Ver20] is not similar work, proper, but has been included here for reference since it occupies a unique quadrant in the design space. SOLID is a decentralized web architecture intended to separate data from services that use them and give users control and governance over their data. SOLID occupies the positive XYZ (all positive) quadrant. The FAIR principles are neutral to governance and decentralization and are the most general artifact.

7.2.3 Requirement Satisfaction

Table 7.2 presents an overview of requirement satisfaction by related work. Below are brief remarks on each approach.

- Research Object (RO). The RO model is Foundational. It is infrastructure-independent (JSON-LD schema) and supports interorganizational curation. It is not FAIR compatible because it breaks at least the A2 FAIR principle of "Metadata are accessible, even when the data are no longer available"[14], and by its nature of being an exchange format and not some infrastructure model, it does not meet the Data on the Web Best Practices. It supports interorganizational curation.
- Research Data Repositories (RDR). RDRs are usually one-off solutions for a particular domain; hence not infrastructure independent. Their underlying model is elementary, usually not allowing for much expansion, hence not Foundational. There is support for data lifecycle annotation and curation. Legality and privacy issues are not usually considered.
- Data.World (DW). The underlying model of DW is a data catalog with plugins for data analytics services and a social network. It is more of a web application/service than an extendable model, so not Foundational and not infrastructure independent. It is not FAIR compatible because it breaks at

[13] https://www.nature.com/sdata/policies/repositories
[14] https://www.go-fair.org/fair-principles/a2-metadata-accessible-even-data-no-longer-available/ (Accessed 11.2021)

least the A2 FAIR principle of "Metadata are accessible, even when the data are no longer available"[15], and given that is a web application and not an infrastructure of data exchange, it does not meet the Data on the Web Best Practices. It supports interorganizational curation, as long these organizations are registered users/groups on the platform.

- International Data Spaces (IDS). The IDS reference architecture might be considered a standard model. However, there is no formal definition of such a model, and the original data spaces concept also lacks a formal definition. The project is driven by many documents, organizations, and iterations, and it is not straightforward to understand or expand on the core model. Due to these reasons, it is not Foundational. It is not infrastructure independent due to being primarily a software platform[16]. It is FAIR[17], and it is unclear whether it meets the Data on the Web Best Practices. One of IDSs main use cases is interorganizational curation.
- The Basin Network model. The proposal is Foundational (built around a small set of first-principle-like structures). It is infrastructure independent since it does not assume or presuppose any technology or infrastructure in mind. In the section below we show how it meets the FAIR principles and the Data on the web best practices. Due to the metadata interoperability structure and a standard exchange format, as well as Publication Contracts, the Basin Network model supports interorganizational curation.

7.3 BNet & {FAIR, DOTWBP}

In the following section, we use the check mark symbol (✓) to denote that the BNet model meets a requirement. Due to the BNet model being a high-level model, some of the requirements which are application- or implementation-dependent will not be directly applicable to the model. To this end, we will use the box symbol (□) to denote that the BNet model is *compatible* with a requirement, meaning that the design decisions/ontological commitments of the BNet model do not hinder any of its implementations in satisfying this requirement. In addition, when the model meets a requirement, references are given to definition numbers in Chapter 5 that satisfy this requirement.

The FAIR Guiding Principles for scientific data management and stewardship[18] is a recommendation on data sharing and exchange in the context of scientific data management. The BNet model meets all the FAIR principles but one which it is compatible with. Table 7.3 depicts this situation.

[15] https://www.go-fair.org/fair-principles/a2-metadata-accessible-even-data-no-longer-available/ (Accessed 11.2021)
[16] https://github.com/International-Data-Spaces-Association (Accessed 11.2021)
[17] https://internationaldataspaces.org/ids-and-the-fair-data-principles/ (Accessed February 2023)
[18] https://www.go-fair.org/fair-principles/ (Accessed 08.2022)

Table 7.3: The meeting of the FAIR Principles (✓ = compatible, □= neutral [implementation-dependent])

Principle	BNet Satisfaction
Findable	
F1. (Meta)data are assigned a globally unique and persistent identifier	✓(Def. 6)
F2. Data are described with rich metadata (defined by R1 below)	✓(Def. 5)
F3. Metadata clearly & explicitly include the identifier of the data they describe	✓(Def. 5)
F4. (Meta)data are registered or indexed in a searchable resource	✓(Def. 6)
Accessible	
A1. (Meta)data are retrievable by their identifier using a standardised communications protocol	✓(Def. 6)
> A1.1 The protocol is open, free, and universally implementable	✓(Def. 6)
> A1.2 The protocol allows for an authentication and authorisation procedure, where necessary	✓(Def. 6)
A2. Metadata are accessible, even when the data are no longer available	✓(Def. 1, 3, 4, 5)
Interoperable	
I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation	✓(Def. 12, 14)
I2. (Meta)data use vocabularies that follow FAIR principles	□
I3. (Meta)data include qualified references to other (meta)data	✓(Def. 5, 14)
Reusable	
R1. (Meta)data are richly described with a plurality of accurate & relevant attributes	✓(Def. 5)
> R1.1 (Meta)data are released with a clear and accessible data usage license	✓(Def. 5)
> R1.2 (Meta)data are associated with detailed provenance	✓(Def. 5)
> R1.3 (Meta)data meet domain-relevant community standards	✓(Def. 12)

Looking at another data exchange reference, *the Data on the Web Best Practices*[19] (DWBP), a W3C Recommendation published in January 2017, which can be considered a high-level data sharing protocol in the form of a list of best practices. The recommendation proposes 35 best practices, 17 of which are met by the BNet model and 18 of which it is compatible with. Table 7.4 depicts this situation[20].

7.4 Discussion

To recall our research question: *How can we facilitate data sharing and exchange in an Interorganizational Data Management and Exchange situation, in terms of data-driven collaborability and with least disruption to existing internal processes and technical infrastructures?*.

The main Requirements 2,3,4 (Supports interorganizational curation, FAIR, and meets W3C's Data on the Web Best practices) together are aimed at a operational data sharing and exchange approach which supports data-driven collaborability. With regards to the second phrase of the question (with least disruption to the internal processes and technical infrastructures [of the participating organizations]), requirement 5 (Infrastructure Independent) is relevant.

We can view the space of possible data sharing and exchange approaches in a two-dimensional space of permissibility and order, which expand on the property of *artifact specialization* discussed above in this chapter. Permissibility relates to the degree of freedom to the technical implementation and manifestation that is afforded to data sharing and exchange participants and technology developers, and *order* is a two-step ladder that characterizes the format of the data sharing and exchange approach: first order is a software-first solution where technology developers and participating parties must use a preset software ecosystem and have no choice but to adopt its technology stack, which causes *disruption to existing internal processes and technical infrastructures*. Second order contains many various alternatives each with their unique properties, that mitigate the level of disruption causes to existing internal processes and technical infrastructures of participating organizations. The basin network model has the format of a formal model that is meant to act as an input to a formal specification and/or reference architecture. Figure 7.2 depicts the state-of-the-art using the above characterization.

[19] https://www.w3.org/TR/dwbp/ (Accessed 08.2022)

[20] The descriptions of Best Practice 10, 12, 15, 22, 24, 25, and 33 have been shortened in the table (for space purposes) while retaining their original meaning. Also, the names of the benefits have been abbreviated: Discoverability to Discover, Interoperability to Interop, Linkability to Link, Processability to Process, and Comprehension to Comprehend.

Table 7.4: Meeting the DWBP Best Practices (✓= satisfied, □= compatible [implementation-dependent])

No.	Best Practice	Benefits	BNet Satisfaction
1	Provide metadata	Reuse, Comprehend, Discover, Process	✓(Def. 5)
2	Provide descriptive metadata	Reuse, Comprehend, Discover	✓(Def. 5)
3	Provide structural metadata	Reuse, Comprehend, Process	✓(Def. 5)
4	Provide data license information	Reuse, Trust	✓(Def. 5)
5	Provide data provenance information	Reuse, Comprehend, Trust	✓(Def. 5)
6	Provide data quality information	Reuse, Trust	✓(Def. 5, 6)
7	Provide a version indicator	Reuse, Trust	✓(Def. 4, 5, 6)
8	Provide version history	Reuse, Trust	✓(Def. 5, 6)
9	Use persistent URIs as identifiers of datasets	Reuse, Link, Discover, Interop	✓(Def. 14)
10	Use persistent URIs as identifiers in datasets	Reuse, Link, Discover, Interop	✓(Def. 6)
11	Assign URIs to dataset versions and series	Reuse, Discover, Trust	✓(Def. 4, 5, 6)
12	Use machine-readable standard data formats	Reuse, Process	✓(Def. 10)
13	Use locale-neutral data representations	Reuse, Comprehend	□
14	Provide data in multiple formats	Reuse, Process	✓(Def. 1, 3, 5)
15	Reuse (standardized) vocabularies	Reuse, Process, Comprehend, Trust, Interop	✓(Def. 11, 14)
16	Choose the right formalization level	Reuse, Comprehend, Interop	✓(Def. 11)
17	Provide bulk download	Reuse, Access	□
18	Provide Subsets for Large Datasets	Reuse, Link, Access, Process	□
19	Use content negotiation for serving data available in multiple formats	Reuse, Access	□
20	Provide real-time access	Reuse, Access	□
21	Provide data up to date	Reuse, Access	□
22	Provide explanation for data not available	Reuse, Trust	□
23	Make data available through an API	Reuse, Process, Interop, Access	✓(Def. 1)
24	Use Web Standards as the basis of APIs	Reuse, Link, Interop, Discover, Access, Process	□
25	Provide complete documentation for APIs	Reuse, Trust	□
26	Avoid Breaking Changes to Your API	Trust, Interop	□
27	Preserve identifiers	Reuse, Trust	□
28	Assess dataset coverage	Reuse, Trust	□
29	Gather feedback from data consumers	Reuse, Comprehend, Trust	□
30	Make feedback available	Reuse, Trust	□
31	Enrich data by generating new data	Reuse, Comprehend, Trust, Process	□
32	Provide Complementary Presentations	Reuse, Comprehend, Access, Trust	□
33	Provide Feedback to Original Publisher	Reuse, Interop, Trust	□
34	Follow Licensing Terms	Reuse, Trust	□
35	Cite the Original Publication	Reuse, Discover, Trust	✓(Def. 5)

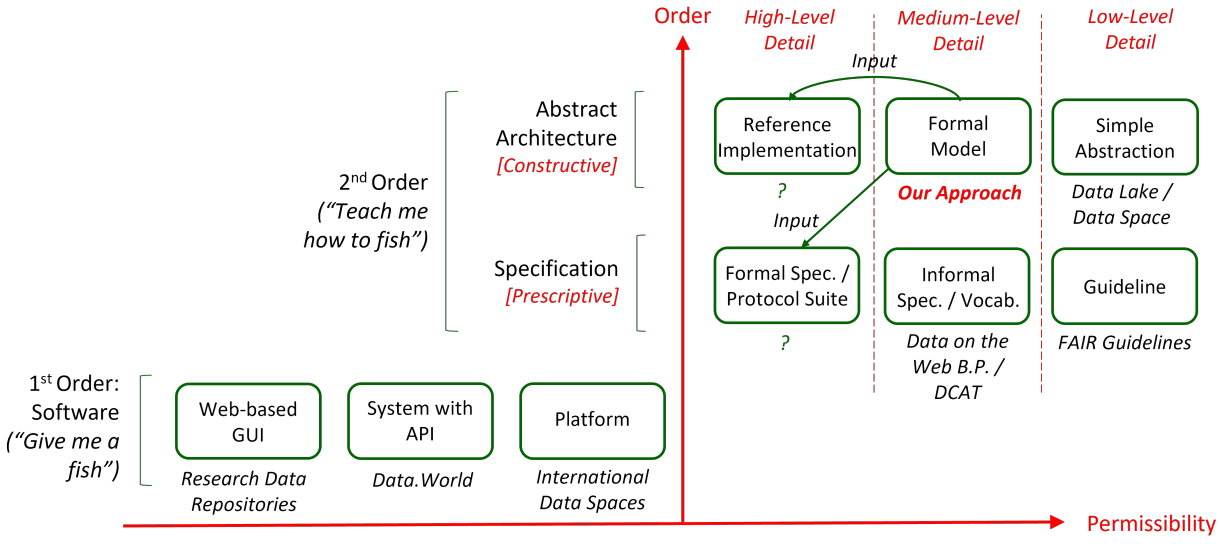


Figure 7.2: Depiction of the space of Permissibility and Order of Data Sharing and Exchange Approaches

8 Conclusion

The growth of digitization and interconnectedness has only emphasized an already pressing need for the streamlined and timely sharing and exchange of data. The problem is not only relevant on a global scale but on a scale of supra-national bodies (EU regulatory initiatives), national boundaries (e-government), cities (citizen science), and corporations (horizontal and vertical integration). All these examples share a common state of affairs: data produced in one environment rarely stays there, and the further it can travel, the more value can be reaped.

We can identify two notions of data sharing and exchange: one concerned with issues such as the integration, federation, virtualization, and transformation of data at the schema level, and the other about the description, publication, discovery, compilation, and re-use of data *as an asset*, similar to how a legal document is managed in a legal information system, for example. We concern ourselves with the second notion here. If we think of data as an asset that needs to be managed and exchanged, then it needs to be cataloged like we would do with other assets, think how a book in a library is managed using a library card catalog, for example. This is typically achieved by using some *intermediate representation* to identify, describe, and effectively '*stand in*' for the asset: what we will call a *digital surrogate*, an approach that has been called *indirection* [Nil10], which we adopt here.

The state-of-the art of the data sharing and exchange problem is polarized. On one end we have the *'platform' approaches*, where a set of closely coupled technologies are built, controlled, and deployed by a single body. On the other we have the *recommendation-based approaches*, which revolve around standards, guidelines, or abstractions intended to orchestrate disparate vendors to develop technologies that interoperate with each other. We discuss both these extremes below.

Upcoming platform initiatives such as the International Data Spaces (IDS) [OJ19] and Gaia-X [Bra+21] provide a technical infrastructure for data sharing and exchange where key functionality such data governance, trust, and identity are handled in a centralized fashion. Due to being led by a single body, the platform approach is more tractable in producing workable solutions that 'go to market' faster in the short run. However, recent history has taught us that platforms and platform companies can become walled-gardens where (1) they do not interoperate with one another [Mas19], (2) are offered on a 'take-it-or-leave-it' basis [Tar22], and (3) frequently result in lock-ins and stifle innovation [ES22].

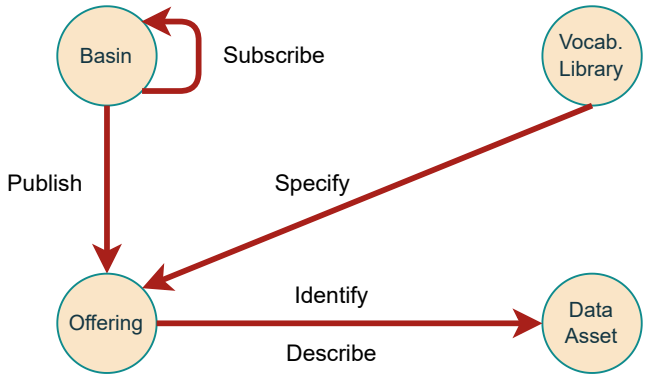


Figure 8.1: Conceptual diagram of the proposed Basing Network Model (*: not discussed here).

Furthermore, it remains to be seen whether these top-down built and administered platforms will be *dynamic* enough to adapt to the ever-changing technology and regulation landscapes, as well as to end-user requirements.

Recommendations-based can be divided into two types: *normative/prescriptive* approaches such as the *FAIR guiding principles for scientific data management* [Wil+16] and W3C's *Data on the Web Best Practices* recommendation [Lós+17], and *constructive* approaches such as the architectural abstractions of *data lakes* [SD20; HQJ21], *data spaces* [FHM05], and *data catalogs* [Ehr+21]. The aim of such recommendations is to provide a common set of agreements and technical structures and leave room for disparate technology developers vendors to build solutions around them, which is a key step in developing interoperable systems. However, there are two key challenges with this group of approaches. First, the current approaches are *low-level detail*, leaving a large interpretation gap to technology developers which leads to diverging implementations and hence hinders interoperability. Second, these approaches have *low-level orchestration*, each focusing on one aspect of data sharing and exchange problem with no constructive vision holding them together.

This work attempts to fill this gap and mitigate some of the problems of the recommendation-based cluster. In this work, we propose the *Basin Network* model. If the challenges of the aforementioned approaches are their low-level orchestration and low-level of detail, then we can characterize the Basin Network model as having *medium-level orchestration* with *medium-level detail*. This is achieved in two steps. First, from the side of *constructive* recommendations, we show how the Basin Network meets the key properties of existing architectural models (data lakes, data spaces, data catalogs), while going further by proposing novel architectural building blocks, hence adding more technical details and structures. Second, from the side of normative/prescriptive recommendations, we show that the design of the Basin Network model meets two major recommendations (the FAIR guidelines and W3C's Data on the Web Best Practices), hence fusing the dominant normative/prescriptive recommendations with the constructive ones in one product-oriented package.

The Basin Network model revolves around two abstractions: the *Offering* and the *Basin* (see conceptual diagram in Figure 8.1). The Offering is novel abstraction for a *digital surrogate of data assets*. Hence, Offerings identify, represent, describe, and effectively *'stand in'* for data assets, re-framing the data sharing and exchange problem into an Offering sharing and exchange problem (discussed below). The Basin is a distributed publish/subscribe-based model for authoring, managing, and publishing Offerings, as well as exchanging them with other Basins, resulting in a Basin *Network* (discussed below). Other aspects of the Basin Network architecture are left for future work, namely: (1) self-sovereign identity/naming, and (2) ledgers (for distributed lineage tracking and governance), and (3) crawlers, indexers, and search engines for Offerings.

8.1 The Offering

The Offering abstraction builds on the *URI-Resource* framework for hypermedia authoring [JWG04]. Nearly pervasive in its adoption, the URI-Resource has come to be considered a 'no-brainier' by today's standards. However, on closer inspection, it turns out that the framework is mired with difficulties [Cla02; Con06; Hal11; Mon12] requiring one to *'tread carefully'* when building on it, we sketch the route we took below.

A URI-Resource is defined as *"anything that has a URI"* [JWG04]. The prominent view of the World Wide Web Consortium's *Technical Architecture Group* is problematic [Cla02; Con06]. To avoid this, we adopt an alternative conception–provided by Roy Fielding–which defines the URI-Resource as *"the semantics of what the author [of the URI-Resource] intends to identify, rather than the value corresponding to those semantics at the time the reference is created"* [FT02, pp. 135]. We can model this conception as an *ID-Resource-Representations technical triangle*. To demonstrate this, consider the homepage of an online news source, say *www.news.example.com*. Whereas the ID is constant (*www.news.example.com*) and the subject of the URI-Resource is stable ("the homepage of Example News"), different contents (representations) will be retrieved each day the URL is accessed [Mon12].

Another challenge is the existence of two competing positions regarding the *semantics* of URI-Resources: the *direct reference* position, which asserts that the semantics are definitions/descriptions provided by the author [of the Resource], whereas the *logical* position adopts the Tarksian model-theoretic interpretation (i.e., set of all Resources that satisfy a formal model) [Hal11]. We adopt the direct reference position.

Given that the semantics of a URI-Resource is its description, we go one step further and propose a specification regarding how these descriptions should be structured: (the *Offering Token Specification*). The Offering Token Specification can be thought as a template which includes four categories of statements about the data asset: (1) data access and structural information (ex., API endpoints, queries,

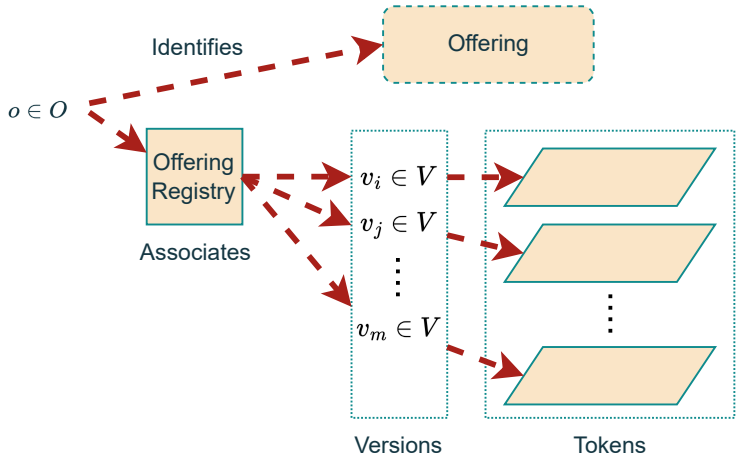


Figure 8.2: The Offering Token: A system modeled on an expanded ID-Resource-Representations triangle.

schema), (2) description/context, (3) integrity, authorship, audience, and usage rights, and (4) links to other Offerings (ex., for lineage purposes). In future work, the static structure of the Offering Token Specification is intended to be replaced with a hierarchical type system of templates influenced by recent work on *data cards* and *model cards* [Don+23].

We can now define an Offering as *a Fieldingian URI-Resource, maintained in a ID-Resource-Representations technical triangle, with direct reference semantics whose associated representations adhere to the Offering Token Specification.* These insights have led to the design of the Offering Registry abstraction, shown in Figure 8.2. This concludes out sketch of the Offering abstraction.

8.2 The Basin Network

With the Offering as a model for digital surrogates of data assets, we search for an architecture to organize and exchange Offerings across groups, a problem we call *data domain topology*. Recent data domain topology architectures include data lakes, data spaces, data meshes, and data catalogs. A key building block in data domain topology frameworks is what we call the *surrogate space* (*space*, for short). A surrogate space is a walled garden (in terms of data access, privacy, and legal scope) to manage surrogates that share a *thematic goal/purpose*. We can distinguish data domain topologies by the constraints they provide regarding constructing spaces and exchanging surrogates between spaces. In addition to these frameworks how a low-level of detail, an issue discussed above, it turns out there is also a gap in the qualitative properties of these architectural abstractions.

With no precise definition of *data lakes* available, each technology vendor tends to define it as close as possible to its own infrastructures, and authors have tried identifying several existing variants. *Functional* architectures [JM17; QHV16; Meh+19] revolve around functions in a data processing pipeline

Table 8.1: The Basin as compared to data catalogs, lakes, zones, ponds, and spaces (*: in theory yes but no details given)

Architecture	*Space Membership Condition*	*Inter-Space Exchange*	*No. of Spaces*	*Privacy*
Data Lake (functional)	Stage of data lifecycle	Preset	<10	No
Data Zone	Refinement	Preset	<10	No
Data Pond	Type/nature	No	<10	No
Data Lake (hybrid)	Refinement or type	Preset	<20	No
Data Catalog	Arbitrary Scope	No	1+ per org.	No
Data Space	Organizational Scope	Yes	1 per org.	Yes*
Basin	**Arbitrary Scope**	**Yes**	**Arbitrary**	**Yes**

[LS16; SD20] (ingestion, storage, processing, etc.). *Data maturity* architectures [ZdB14; LS16; Inm16], of which there are several variants (Zone, Pond) [Inm16; LS16; Gie+19], are built around data refinement level [SD20] (ex., textual, telemetry; or raw, refined, trusted). And *hybrid* architectures [Inm16; RZ19] include some combination of the above. *Data Spaces* [FHM05], although not formally defined, introduced the vision and construct that includes all the heterogeneous data sources belonging to an organization, with a 'pay-as-you-go' fashion in the sense of data integration, querying, and processing functionality. Finally, *data catalogs* are systems that "collect, create and maintain metadata" [Qui+20; Ehr+21].

We focus on four properties of data domain topologies, depicted in Table 8.1. *Space membership condition* denote the rule a framework allows for constructing spaces. *Inter-space exchange* is about any constraints placed on the exchange of surrogates across spaces. *Number of spaces* is about the maximum allowed number of spaces. Finally, *privacy* is about the existence in the framework of some notion of privacy within and/or across spaces. Giving the landscape as depicted in Table 8.1, the data space model is the most general, however, we have observed, in several data sharing and exchange projects we have been involved in, that there is a need to have more than one surrogate space per organization, preferably more than one surrogate space per division/team (ex., staging space for data assets, internal working space). Due to these limitations, we proposed the Basin Network model.

The Basin Network models revolves around a set of Basin exchanging Offerings, and a set of shared administrative components. One of these components are covered in the formal model (Metadata Interoperability Structure: a metadata vocabulary registry, *not discussed here*), whereas others are out of scope (ex., a distributed ledger for lineage tracking and governance of Offerings, Offering crawlers/search engines).

This work has developed through experiences with use-case applications in several domains [Kas+18; Kas+21]. We demonstrate the the applicability of the proposal by applying to three use-cases: a computer-aided inter-organization manufacturing project, an IoT project in smart agriculture, and one in crowd data management. Interested readers can consult Chapter 6 and Appendices B and C.

8.3 Looking Forward

Ultimately, this work has arrived at the conclusion that data sharing and exchange is best facilitated by the emergence of a *constellation of technologies/recommendations*: a set of (typically open and sometimes re-purposed) technologies that play different roles within a loose overarching *architectural model/pattern* to solve some problem domain. These over-aching models/patterns can be top-down (ex., linked data in the 2000s) or bottom-up (ex., the HADOOP ecosystem in the 2010s). The Basin Network model is intended as an step in this direction: a top-down decentralized architectural model of loosely-coupled components for data sharing and exchange.

The strengths of the constellation of technologies/recommendations approach is that, *if it can be achieved at all*, results in dynamic ecosystems that can cope with change, and are less susceptible to consolidation by a single body, although this last property is still little understood [Not23]. However, the current hacker culture *zeitgeist* of 'decentralization-over-everything' of which this work is a part of is still a conjecture at best, that curiously tends to attract two extremes of practitioners with conflicting (political) worldviews which we have seen before in the history of the internet: namely, left-anarchist and right-libertarians [AD22, Ch. 2]. We have seen, for example, how recent decentralization trends have been used sometimes for questionable quick money-making schemes such as some of the cryptocurrency projects. Despite these reservations, there is more to gain, or at least learn, from the emergence of such an ecosystem for data sharing and exchange.

Finally, a classical open problem that still remains an obstacle for sharing and exchanging data assets today is related to the *semantic heterogeneity gap*: a challenge that hinders the *comprehension* and *discovery* of data assets. We have already established the need for digital surrogates within the *indirection* approach to data management. A key component of these surrogates (of which the Offering is an instance of) are statements which assert properties *about* the data asset to be managed: what some refer to as *meta-data*. Different groups and communities use different metadata to describe and catalog their data assets. The challenge–at least today–is not anymore about differences on the technical/system/structural levels [OS99; Arm+02; vdVW08] (hardware, OS, network, syntax, data models, interfaces) but on the *semantic* level. The semantic heterogeneity gap–within the metadata management tradition–ultimately boils down to the use of heterogeneous metadata element-sets or *vocabularies* and mapping between them. We will close this discussion with a look at this problem.

In the latter part of the 19th century, the popular worldview was that it was possible to determine a single, absolute meaning if a sufficient effort was made. This influenced standardization organizations such as the International Union of Associations (UIA) and the International Standards Organization (ISO), which were driven by a view that the international was the key to the universally valid and ontologically true. Hence the attempt was to develop these standards in a *top-down, deductive* fashion. Fast forward to the second half of the century, even a project as seemingly simple as devising fields of a library catalog (Machine Readable Cataloging, MARC) eludes a single, international, universal, solution; resulting in US MARC, AUS MARC, UK MARC, and CAN MARC [Velo1]. Fast forward further to the end of the century, and the search for this universal information model or 'upper/reference ontology' *within information systems* has practically been abandoned, save for specialized niches [Floo8, pp. 158-160].

This paradigm shift was closely associated with the emergence of a large number of metadata vocabularies in a bottom-up, decentralized fashion. For example, as of May 2023, the Basic Register of Thesauri, Ontologies & Classifications (BARTOC) hosts around 3500 such vocabularies. This has resulted in the selective reuse and combination of vocabularies to fulfill the information representation needs of a domain/application. This raises the question: *with all this proliferation of vocabularies, and lack of common upper or reference ontologies, how can we achieve semantic interoperability across metadata adhering to (sets of) heterogeneous vocabularies?* The baseline approach is to build mappings in the order of $O(n^2)$ or peer-to-peer mapping [RS13]. However, the mapping from a single vocabulary to another is already an expensive, manual, and typically imprecise activity.

Unfortunately, in this regard, very little has changed in the last 20-30 years. What has improved, from one side, is the emergence of standards and technologies for *representing and managing vocabularies in machine-readable ways*, and from the other, the explosion in quantity of available vocabularies with permissible usage licenses. However, the *elephant in the room* still eludes us: how can we map across a large set of vocabularies in an affordable, streamlined, and [semi-]automated way? Perhaps recent developments in other technologies (ex., transformer/large language models) might help us out of this predicament.

A The D^4 Domain Model

The D^4 (data driven domain description) model is a proposed *upper ontology* for the use cases presented in this work. The interaction with field experts influenced it during several of the project discussed in the use case chapters to follow. It is build on top of the Dublin Core Terms recommendation (http://purl.org/dc/terms/) and the PROV W3C provenance recommendation (https://www.w3.org/TR/prov-overview/). Dublin Core Terms are prefixed with a *'d:'* and PROV terms with a *'p:'*, and no prefix is given for our terms. Figure A.1 depicts the major types in the model. Figure A.2 depicts the major relations.

A definition of all the terms in the model will be given below, but as an introduction, the main concepts of the model are the following:

- **Agent**. Agents represent information about entities that take actions and produce data and knowledge during the different processes of the domain. So far, we have mostly discussed Agents as Users, but Systems and Organizations are also a subclass of Agents because they take actions and produce results. So the discussion about Agent applies to all of them.
- **Activity**. Activities represent the basic abstraction unit of a loose process in the EMTD domain. Data, Documents, and Datasets are attached to Activities, e.g., simulation data are attached to a Simulation Activity, measurements of material properties are attached to Measurement Activities, and so on.

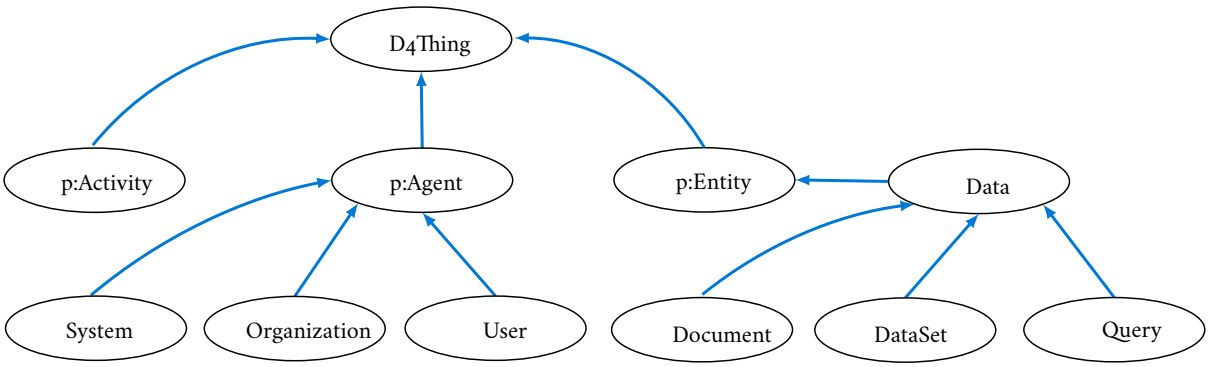


Figure A.1: Overview of the D^4 model, with arrows denoting the *type-of* relation.

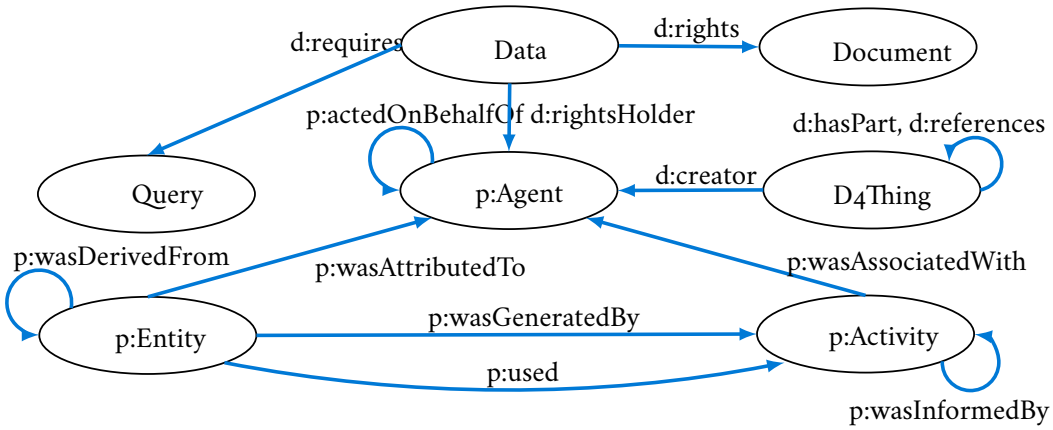


Figure A.2: Overview of the relations in the D4 model.

- **Resource**. Instances of class Resource represents information about material / physical artifacts in the project, things that can run out. Examples include ovens, raw materials, molds (tools), sample parts, and final parts. Resources are linked to Activities, and Data (e.g., specification documents) can be attached to Resources.
- **Data**. Data is a top-level abstraction of both DataSets and Documents, which are structured and unstructured Data, respectively. Data can be attached to the domain's Activities, Resources, and other classes.

In Table A.1, all classes in the D4 model are defined. All the properties are defined in A.2, and the relations in Table A.3.

Table A.1: Classes in the D^4 Model (Definitions for inherited elements adapted from the recommendations)

Namespace	Title	Type	Level	Version	Description
skm	D4Thing	n/a	upper	v3.1	Top level class.
prov	Agent	D4Thing	upper	v6.0	Something that bears some form of responsibility for an activity, entity, or another agent.
prov	Activity	D4Thing	upper	v6.0	Something that occurs over a period of time and acts upon or with entities.
prov	Entity	D4Thing	upper	v6.0	An entity is a physical, digital, or conceptual thing.
skm	Data	Entity	upper	v3.1	Represents data or digital resources stored outside of the system.
skm	Document	Data	upper	v3.1	A data file in a specified format that is human readable.
skm	DataSet	Data	upper	v3.1	A specified set of data that is stored in a data store and can be retrieved using a query.
skm	Query	Data	upper	v4.0	Instructions to locate and access any Data item.
skm	Organization	Agent	upper	v3.1	A social or legal institution such as a company, society.
skm	System	Agent	upper	v3.1	Systems that can create data like sensors or software.
skm	Person	Agent	upper	v3.1	Person agents are people.
skm	User	Person	upper	v3.1	Persons that have anan account on the system.

Table A.2: Properties in the D4 Model (Definitions for inherited elements adapted from the recommendations)

Namespace	Class	Title	Level	Type	Description
dcterms	Data	`type`	upper	XSD:string	The nature or genre of the Data.
skm	Data	`schema`	upper	XSD:anyURI	Information how to locate the relevant aspects of the data.
dcterms	D4Thing	`title`	upper	XSD:string	A name given to the resource.
dcterms	D4Thing	`description`	upper	XSD:string	Abstract, table of contents, or free-text account of the resource.
dcterms	D4Thing	`identifier`	upper	XSD:anyURI	An unambiguous reference to the resource within a given context.
skm	D4Thing	`originType`	upper	XSD:anyURI	A unique, qualified id of the original type if not in the current model.
skm	D4Thing	`type`	upper	XSD:anyURI	A unique, qualified id of the type of the node in the current model.
dcterms	Query	`format`	upper	XSD:anyURI	The data container format.
dcterms	Query	`coverage`	upper	XSD:anyURI	Where to locate the Data provider (server or endpoint address).
foaf	Person	`familyName`	upper	XSD:string	The family name of some person.
foaf	Person	`givenName`	upper	XSD:string	The given name of some person.
foaf	User	`mbox`	upper	XSD:anyURI	Internet mailbox associated with one owner.
prov	Activity	`startedAtTime`	upper	XSD:dateTime	The time at which an activity started. See also prov:endedAtTime.
prov	Activity	`endedAtTime`	upper	XSD:dateTime	The time at which an activity ended. See also prov:startedAtTime.
skm	D4Thing	`secondaryId`	upper	XSD:anyURI	An optional domain specific ID.
skm	Query	`queryString`	upper	XSD:string	A query string or URL in case of Document.

Table A.3: Inherited Relations in the D[4] Model (Definitions adapted from the recommendations)

From Entity	Title	To Entity	Level	Namespace	Definition
D4Thing	references	D4Thing	upper	dcterms	A related thing that is referenced, cited, or otherwise pointed to.
D4Thing	relation	D4Thing	upper	dcterms	A relation between two things (bi-directional version of references)
D4Thing	hasPart	D4Thing	upper	dcterms	A related thing included either physically or logically.
D4Thing	replaces	D4Thing	upper	dcterms	A related resource supplanted, displaced, or superseded by this one.
Data	rights	Document	upper	dcterms	Includes a statement about property rights of the thing.
Data	rightsHolder	Agent	upper	dcterms	A person or organization owning or managing rights over the thing.
D4Thing	creator	Agent	upper	dcterms	A user that created an item in the system.
D4Thing	tableOfContents	Document	upper	dcterms	The related Document describes sub-units of the resource.
Activity	member	Person	upper	foaf	This Person is a member of this Project.
Organization	member	Person	upper	foaf	This Person is a member of this Organization.
Data	requires	Query	upper	dcterms	A related resource that is required to support the function, delivery, or coherence.
Entity	wasGeneratedBy	Activity	upper	prov	Generation is the production of a new entity by an activity.
Entity	wasDerivedFrom	Entity	upper	prov	A derivation is a transformation of an entity into another.
Entity	wasAttributedTo	Agent	upper	prov	Attribution is the ascribing of an entity to an agent.
Activity	used	Entity	upper	prov	Usage is the beginning of utilizing an entity by an activity.
Activity	wasInformedBy	Activity	upper	prov	The exchange of an entity by two activities.
Activity	wasAssociatedWith	Agent	upper	prov	An activity association is an assignment of responsibility to an agent for an activity.
Agent	actedOnBehalfOf	Agent	upper	prov	Assignment of authority and responsibility to an agent to carry out a specific activity.

B Application: Digital Agriculture

The use case presented in this section is influenced by our involvement in a project in the area of digital agriculture (FutureIoT[1]). The primary motivation for intelligent farming solutions is the heavy workload farmers experience in monitoring their dairy cattle's health status. Supported by a sensor-based monitoring system that tracks the activities of the cattle, the farmer is notified about deviations from normal cattle behavior. A brief description of this use case will be given below, followed by a basin network instance modeling this domain.

B.1 Smart Farming

A battery-driven sensor collar is assembled on each dairy cattle consisting of an accelerometer, a gyroscope, and a GPS tracker. This sensor system produces raw data about the cattle's movement, orientation, and location. Besides sensor data acquisition, labeled datasets are created by a domain expert, which depict ground truth for activity recognition. Several video cameras record the cattle on the pasture and in the stable, in addition to the collars. Recorded video files are analyzed offline, and a labeled data set is created using a glossary containing a variety of activity definitions like lying, standing, walking, grazing, and ruminating. Various data sets are made by different sensors covering different animals across time and location, which must be stored and retrieved for multiple uses.

In the second step, sensor data and labeled datasets are used in the training pipeline for developing machine learning models, from pre-processing to post-processing to evaluating and detecting good models for cattle activity recognition. Initially, a learning model should be generated out of an algorithm. Many parameters need to be tuned to find the high accuracy models. In segmentation, you must decide on the window size and stride parameter. In feature extraction, you must choose which and how many features to include. In the training step, supervised machine learning algorithms like Random Forest, Decision Tree (etc.) are used to train models, and so on.

[1] https://www.futureiot.de/ (Accessed 08.2022)

From the description given above, this environment is not a single silo separated from many other activities, systems, and groups. First, there is a sensor data management organization (hereafter referred to as '*Dat*') responsible for setting up sensor networks, collecting, pre-processing data, and so on. Second, there is the organization of domain experts in farming (henceforth '*Dom*'), which define and produce specifications of cattle behavior and extract/label cattle events captured by sensors and video cameras. Third, an organization is responsible for developing and managing machine learning models for cattle activity recognition (henceforth '*Learn*'), which builds on data produced by the previous organizations. The three organizations above are part of a more extensive network of partner organizations with data-sharing agreements. Additionally, the cattle activity group wishes to publish some selective data to the public for educational and research purposes.

Next we show how the Basin Network model, as presented in Chapter 5, can be applied to solve this domain.

B.2 Application

B.2.1 Smart Farming Domain Model

This section presents the domain's main components. It is a specialization of the D^4 domain model given in Appendix A. Let's call this model *CAR* (cattle activity recognition) which is depicted in Table B.1.

B.2.2 Basin Network

Figure B.1 depicts a basin network instance for this use-case. Below we discuss the different clusters of the use case.

Hardware / Sensor Management Cluster Starting from the bottom left corner, we have the Hardware/Sensor Management party ($P1$), which operates the H.W. / Device Installations Basin ($B2$) and participates in the raw sensor data basin ($B3$). The H.W. / Device Installations basin includes data and documentation on installations and device users, with several basins subscribing to it.

Machine Learning Cluster The Machine Learning party (P_1) is responsible for training machine learning models, deploying them, and monitoring them. It operates in two basins, one where it publishes deployed or ready-to-be-deployed models (B_{1b}) and the other where it manages models under development (B_{1a}). The model development basin involves the domain experts party as a

Table B.1: The CAR Domain model, specialized from the D^4 domain model given in Appendix A.

Title	Subclass of	Description
DataProcessing	Activity	An activity for converting raw data into
GroundTruthGenenation	Activity	An activity that happens in the farm for collecting ground truth data
FarmStay	Activity	An activity which represents an expert stays at farm to collect data
Sensing	Activity	An activity that happens in the farm for collecting sensor data
Evaluation	Activity	An activity to evaluate the results of a labeled dataset
Prediction	Activity	An activity to create a labeled data set from a sensor dataset
Training	Activity	An activity to create a labeled data set from a sensor dataset
ModelGeneration	Activity	An activity to generate a LearningModel from a DataProcessor
Cow	Agent	An agent which represent an animal in a Herd
Herd	Agent	An agent which represents more than one animal
LearningProcessor	DataProcessor	A DataProcessor which represents a ML algorithm
GroundTruth	DataSet	a DataSet that contains the ground truth generated from video files
LearningModel	DataSet	A dataset that contains information about a LearningProcessor
LabeledData	DataSet	A dataset that contains sensor data with generated labels
GPSData	DataSet	A dataset that contains GPS updates collected by GPSSensors
Feedbackreport	Document	A document generated by Evaluation activity over labeledData
VideoData	Document	A document generated by Sensing
IMUSensorData	DataSet	A dataset that contains data collected by IMUSensors
BinaryGrayzing	Learning-Model	LearningModel with information for BinaryLearningProcessor
MGNCamera	Sensor	Sn sensor which collects ground truth in the farm
IMUSensor	Sensor	Sn sensor which collects IMU data of an animal
GPSSensor	Sensor	Sn sensor which collects GPS updates of an animal
SensorCollar	Sensor	Sn sensor which represent an animal in s Herd
DataProcessor	System	Sn system that represents an algorithm that can manipulate data
BinaryGrazingProcessor	LearningProcessor	LearningProcessor that generates binary grazing labels

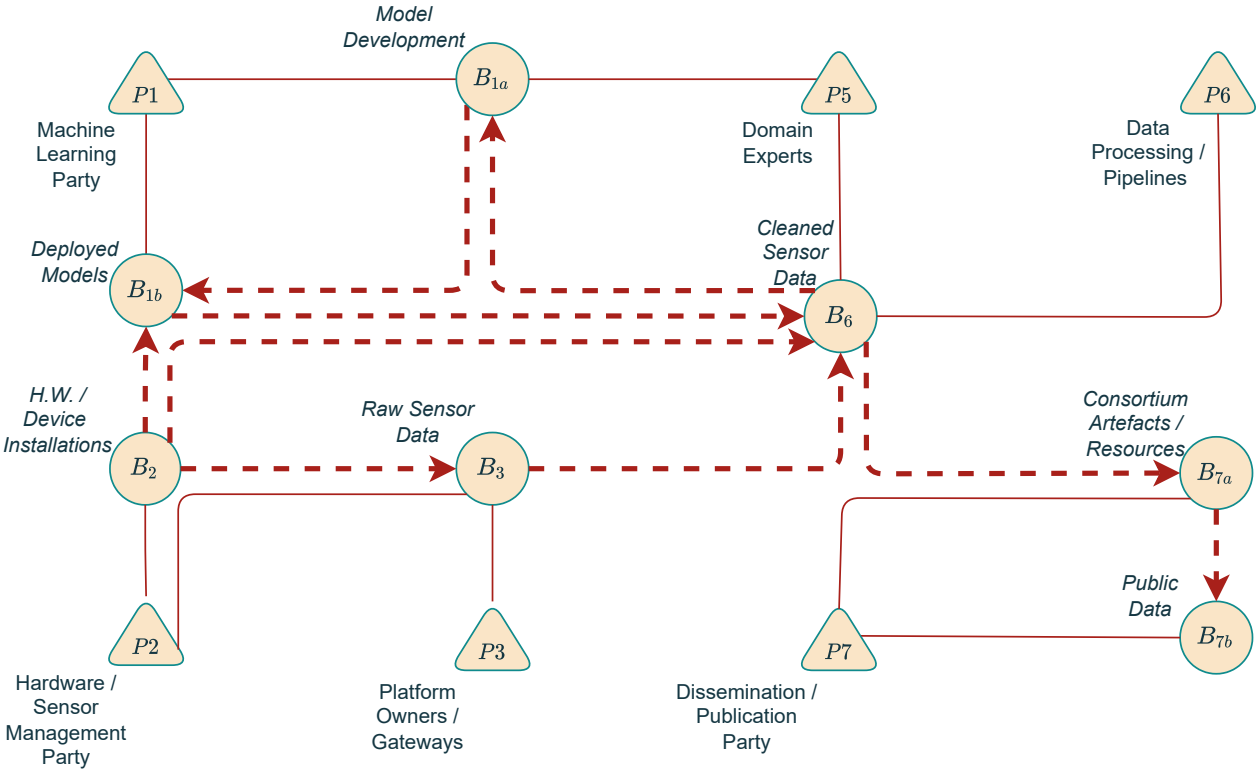


Figure B.1: A basin network instance for the digital agriculture domain.

participant because of the close collaboration of the domain experts in providing feedback on the model development process. This basin also subscribes to the Cleaned Sensor Data basin (B_6).

Platform Owners / Gateways Cluster The Platform Owners/Gateways party ($P3$) represents one or smaller parties responsible for providing the infrastructure for moving and delivering raw sensor data. As opposed to many parties discussed so far in the use cases, many of the participants in this party will be machines. The Cleaned Sensor Data basin (B_6) subscribes to this basin to receive up-to-date offerings about new data sources.

Domain Experts Cluster The domain expert party ($P5$) participates in two basins: B_{1a} where it gives feedback on developed models, and B_6 where it labels data and datasets for providing training data but does not operate its own basin.

Data Processing Cluster The Data Processing/Pipelines party ($P6$) is also primarily machine-driven. It is responsible for producing cleaned sensor data (basin B_6) from raw data. This role is explored in more detail in the following use case study.

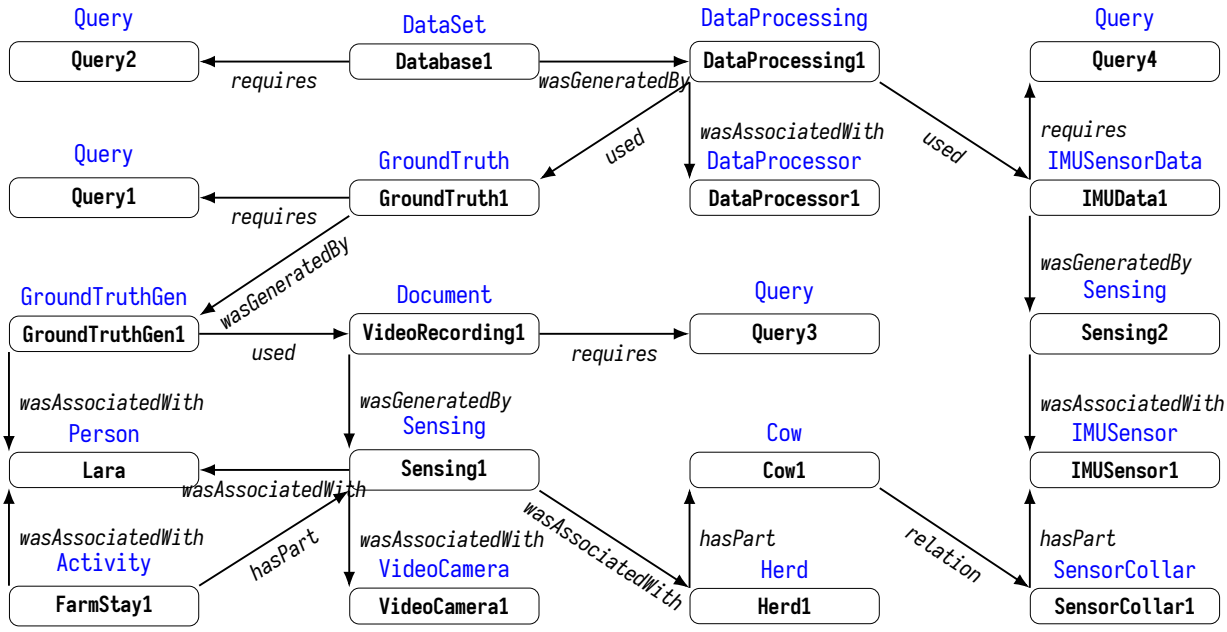


Figure B.2: Example scenario: information contents of the *Dat* market

Dissemination Cluster The Dissemination/Publication party (P_7) is responsible for preparing and publishing data that is shared between all the participants in the consortium, as well as publicly available data; it operates two basins covering these two roles (basins B_{7a} and B_{7b}).

So far, we have looked at the structural side of the Basin Network model. Here we give some examples of the informational content of the Description element d of the offering token. We will use the property graph model notation to represent these examples because it is a recent and popular notation.

Cleaned Sensor Data Basin (B_6) An example scenario of information statements in the description element of a token in the Cleaned Sensor Data basin (B_6) is depicted in Figure B.2. Starting from the bottom left, activity *FarmStay1* is associated with *Lara* which stays in a farm to perform another activity: *Sensing1*. Additionally *Lara* is also associated with the *GroundTruthGen1* activity. A *Sensing* activity is associated with a *Sensor* to generate *Data*, shown as *Sensing1*, *VideoCamera1*, and *VideoRecording1* respectively in the Figure B.2. This same activity is associated with *Herd1* which contains *Cow1*. Additionally, in *GroundTruthGen1*, *Lara* used *VideoRecording1* to generate *GroundTruth1*. As depicted in bottom right of the figure, *Cow1* is equipped with *SensorCollar1*, which is composed of another sensor *IMUSensor1*. Another activity, *DataProcessing1* was associated with an agent *DataProcessor1* and used *GroundTruth1* to generate *Database1*.

Model Development Basin (B_{1a}) An example scenario of information statements in the Description element of a token in the Model Development basin (B_{1a}) is depicted in Figure B.3. Starting from the *Train1* activity on the left hand side: it used *GroundTruth1* and *IMUData1*, was associated with

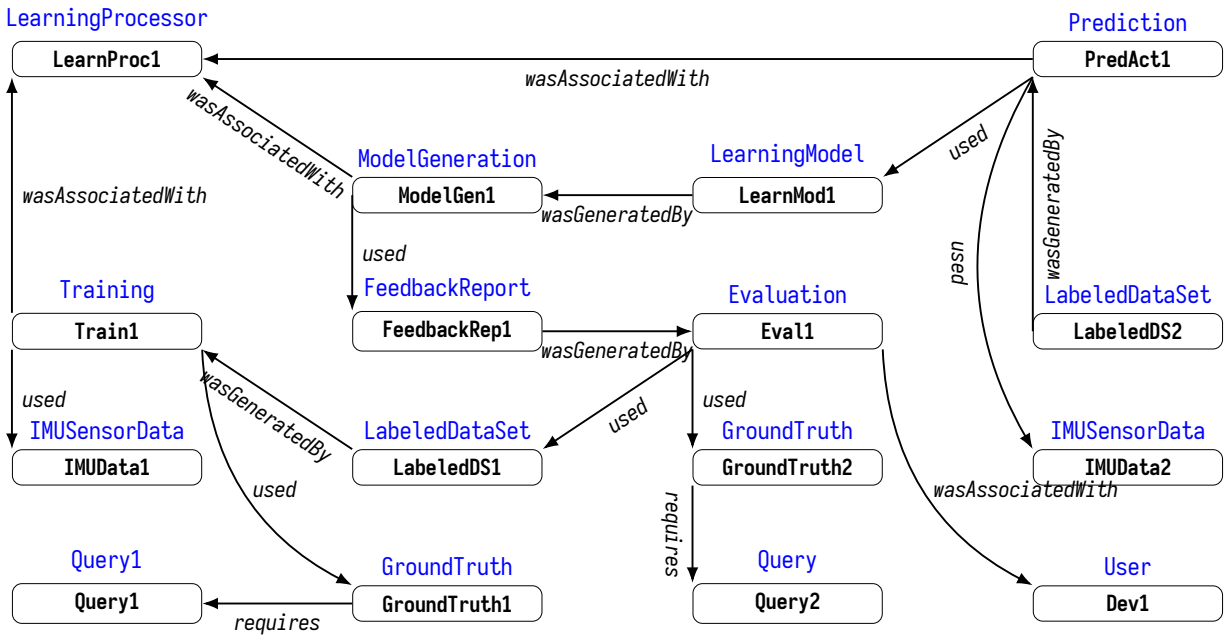


Figure B.3: Example scenario: information contents of the *Learn* market

LearnProc1 which represents a learning algorithm, and generated *LabeledDS1*. *Eval1* used *LabeledDS1* and *GroundTruth2* to generate *FeedbackRep1*. *ModelGen1* activity used *FeedbackRep1* and *LearnProc1* to generate *LearnMod1* which was used by *PredAct1* along with *IMUData2* to generate *LabeledDS2*.

Consortium Artefacts Basin (B_{7a}) Figure B.4 depicts an example scenario of information statements in the Description element of a token in the Consortium Artefacts basin (B_{7a}). The contents include a subset of the Cleaned Sensor Data (B_6) basin contents and a subset of the *Learn* market contents. Note how the types of nodes *Eval1*, *LabeledDS2*, and *PredAct1* have been generalized into the model used in the *MDL* market, becoming of type *Activity*,*DataSet*, and *Activity* respectively. The content of the rest of the basins will follow the same logic.

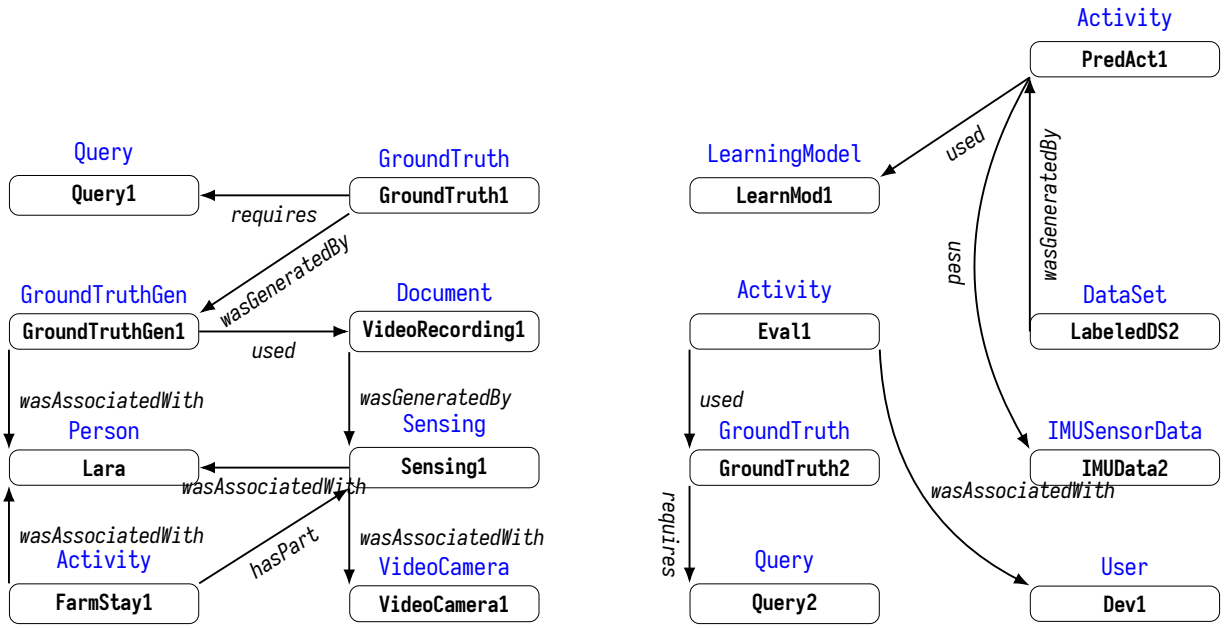


Figure B.4: Example scenario: information contents of the *MDL* market

C Application: Festival Data Management

In this appendix, we demonstrate the Basin Network model in a different usage scenario [Kas+21]. Whereas in the previous two chapters, we explored specific use case scenarios in particular projects and domains with a basin network instance working in a vacuum, here we consider how a basin network can support an existing IoT-oriented data management infrastructure by *integrating disparate data processing functionalities*.

Over the last years, the *Internet of Things* has evolved from a high-level vision of always-connected devices to a real cyber-physical system class that appears in many application domains, from healthcare [Pik+19] over smart cities [Zan+14] to smart farming and precision agriculture [Kam+16]. IoT has introduced radical changes in the way data are processed. The amount of IoT data, the velocity of change, and the variety of sources/formats imply new challenges to process and inter-operate between heterogeneous data sources and formats [Els+19].

In addition, we are faced with another set of challenges on a platform design level. Assuming these emerging data management challenges are solved, how can we extend IoT platforms to integrate new data management functionalities smoothly? This is an especially pressing problem since various available IoT platforms are not easily extendable beyond the original use cases assumed by their designers. We have had firsthand experience with several IoT platforms[1,2] and have faced various issues related to extending platforms with data management functionalities. Currently, data processing-related tasks are typically realized by manually developed code and functions deployed by intelligent devices (AKA edge nodes), in the cloud, or in-between so-called fog nodes. This creates future difficulties in the maintenance and growth of the platforms. Hence it is necessary to explore other approaches to designing IoT platforms that can be extendable to integrate emerging functionalities. Overall these challenges relate to devising methods, techniques, and system platforms that ease the development and maintenance of such systems for the developers and operators.

[1] ThingsBoard, https://thingsboard.io/.
[2] Cumulocity, https://www.softwareag.cloud/site/product/cumulocity-iot.html.

We choose four examples of data management functionalities in IoT that we are already involved in other research activities:

1. **CQuality**. Many IoT applications are based on sensor data which is never perfect; we have to deal with data quality issues which might even change depending on the context of the sensor. Data Quality information and metadata, and the associated sensor data itself, are all valuable and must be exchanged across systems and groups. Hence, the need for methodologies that model sensor data quality for processing and decision-making opens the door to the challenge we call the CQuality challenge.
2. **CPrivacy**. IoT devices are ubiquitous; sensors are present in more devices that produce continuous streams of data about their environment. Representing privacy information and preferences along with the associated data paves the way to automated privacy-preserving data sharing policies and automated decision-making and processing (e.g., pseudonymization on the fly). This opens the door for the challenge of privacy-preserving data processing, which we refer to as the CPrivacy challenge.
3. **CModel**. There is work in utilizing machine learning to train models that eventually replace manually programmed or modeled functions, e.g., for activity recognition. We consider the life-cycle management of these models as part of a (higher-level) data management process, one of which is required to be integrated with many other functionalities. We refer to this challenge as the CModel challenge.
4. **CResource**. Data management is distributed geographically between the sensor/actuators and gateways up to the cloud. This distribution leads to high heterogeneity between the processing nodes regarding computing resources, system security, and connectivity. This also depends on communicating and exchanging information about processed and soon-to-be data, which helps integrate this functionality with other data processing functionalities. We call this the CResource challenge.s

C.1 Live Festival Data Management

To define the challenges and explain and provide appropriate solutions, we describe our use case, a smart city platform, the so-called Living Lab Bamberg. This testbed provides a user-centric environment to test and use different sensing systems. The Living Lab is open to industry partners and citizens who help us receive new sensor systems and find installation locations.

Different kinds of stationary and mobile sensor systems are used. An example of a static sensor is a people-counting camera, and a model for a mobile sensor is a sensor box in a city bus to measure air

quality. These sensing systems produce data that we use in our different application scenarios. We describe some of these application scenarios below.

One application scenario is indoor and outdoor localization management. Every year many people visit a lot of street festivals. The goal was to flow-track the movement of visitors in the area of the festival and learn movement models of the civilians and groups of people. This can help on a short-term basis to predict escape routes on the fly or, in retrospect, to plan for future planning of street festivals.

For the measurement, we used a combination of different sensing technologies like manual counting, camera-based counting, and Wi-Fi tracking of mobile phones. In addition, the people tracking camera system helps us to collect trajectories from people inside buildings (an example of indoor localization would be an iBeacon network). In this environment, we can simulate tourist information panels inside the university or guides for city museums.

Already in such a scenario, we have several data management challenges arising. We also highlight the information-driven aspect of each challenge since this is the crucial property we use in the proposed information integration model, the Basin Network model.

1. **Festival Sensor Data Quality**. Like most other sensors, the above sensors produce readings that are never perfect. However, it should be possible to build quality models for data–using sensor models and environmental context–to enrich them with quality information. This can greatly help in correcting some faultiness or, at least, in describing the nature of the faultiness of the sensor data. For example, such quality information can be invaluable for developing machine learning models (see below). The *CQuality* challenge we present in this paper forms an instance of studying this problem. In the above approach, sensor quality models and quality information attached to data sets (ex., metadata) are structured information that can be represented, exchanged, and processed by others in a basin network.
2. **Visitors' Privacy**. As data is collected about festival attendees, the vulnerability of the visitors' privacy in the Wi-Fi tracking of mobile phones arises. How can we collect data from the festival attendees while protecting their identities? The *CPrivacy* covers this aspect of the problem. Although not as information-driven as the sensor data quality challenge, information representing which parts of the data are privacy sensitive or what algorithms and parameters to run on them, are structured information that drives privacy data processing.
3. **Festival Attendees Movement Models.** Developing movement models helps to predict, manage emergencies and support the planning of future festivals. Sensor data and its quality information can be used to develop these models, which are also structured information that different systems can process. This forms the *CModel* investigation in this paper. Examples of structured information driving this data management functionality include training datasets, learning parameters, and

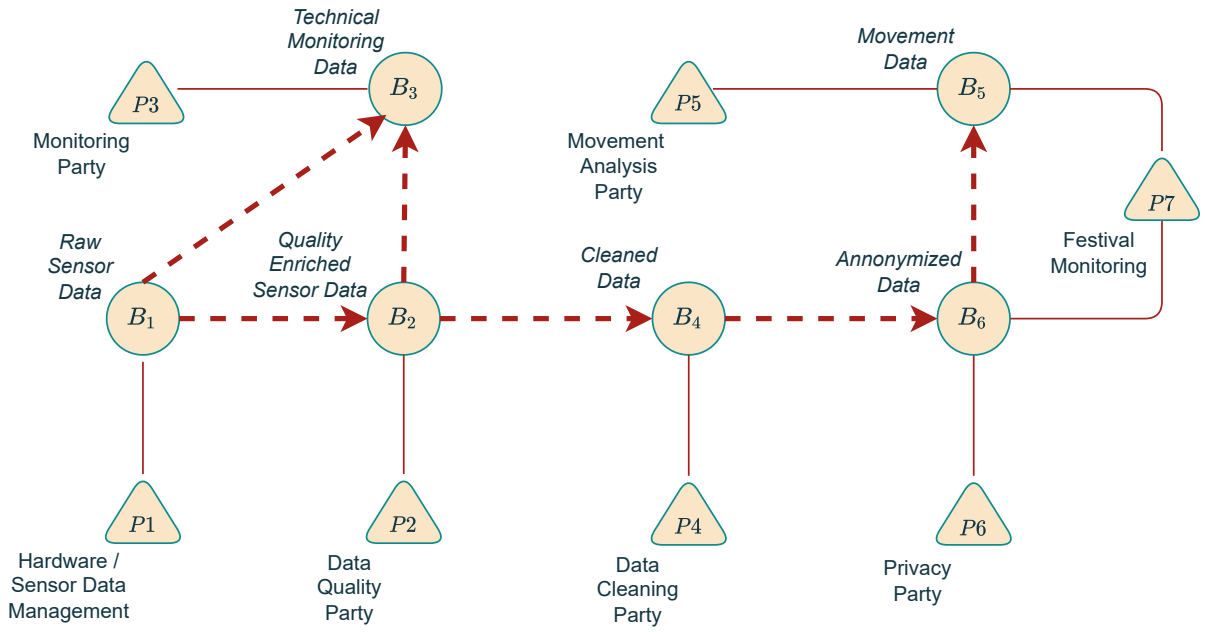


Figure C.1: An basin network instance for the integrating data processing use-case.

the learned models. These are all structured information key to driving machine learning model management.

4. **Resource-aware Computation of Festival Data.** We have assumed that computation is managed locally or by some high-performance machine. But the IoT reality has given us many options to run computations. For privacy, it can significantly benefit if data is, for example, pseudo anatomized on the fog node, or the nearest gateway, if it has such computational capabilities. Also, for developing machine learning models, the resource-aware computation can provide various alternatives to pre-process, clean-up, or run jobs across nodes. This challenge, as opposed to the three above, is a service challenge that supports other data management functionalities.

Various data management functionalities arise in these use cases, and IoT platforms suffer from a lack of fluidity when new features or functions are added to the platform. The Basin Network model can play a different role in serving as an architecture to integrate these different functionalities.

Next we show how the Basin Network model, as presented in Chapter 5, can be applied to solve this domain.

C.2 Application

Figure C.1 depicts one possible way to model the live festival data management use-case.

Starting from the bottom left, the Hardware/Sensor Data Management party ($P1$) operates the Raw

Sensor Data basin (B_1). Two basins subscribe to this basin: the Technical Monitoring Data basin (B_3) and the Quality Enriched Sensor Data basin (B_2). The Technical Monitoring party ($P3$) carries out the typical technical infrastructure monitoring duties. The Data Quality party (P_2) processes and enriches raw sensor data with quality information. The Data Cleaning party (P_4) operates on quality enriched data to clean and various other preprocessing functionalities; it operates its own basin. The Privacy party ($P6$) is responsible for processing data to guarantee the privacy of the participants tracked in it. The Movement Data basin (B_5) and the Movement Analysis party ($P5$) work on anonymized data to detect dangerous or critical movement patterns. The Festival Monitoring party ($P7$) represents the festival organizers, who participate in the Anonymized Data basin (B_6) and the Movement Data basin (B_5) and use the data provided to support them in their duties.

References

[AD22] Janet Abbate and Stephanie Dick. *Abstractions and Embodiments: New Histories of Computing and Society*. JHU Press, 2022. 336 pp. ISBN: 978-1-4214-4438-3 (page 96).

[Amo+17] Ricardo Carvalho Amorim et al. A Comparison of Research Data Management Platforms: Architecture, Flexible Metadata and Interoperability. *Universal Access in the Information Society* 16(4):(2017), 851–862 (page 81).

[Arm+02] William A. Arms et al. A Spectrum of Interoperability, The Site for Science Prototype for the NSDL. *D-Lib magazine;2002 (8) 1*:(2002). ISSN: 1082-9873. https://dspace.library.uu.nl/handle/1874/3088 (visited on 09/06/2022) (page 96).

[Ash10] Toshihiro Ashino. Materials Ontology: An Infrastructure for Exchanging Materials Information and Knowledge. *Data Science Journal - DATASCIENCE* 9:(2010) (page 27).

[Atk+17] Malcolm Atkinson et al. Scientific Workflows: Past, Present and Future. *Future Generation Computer Systems* 75:(2017), 216–227. ISSN: 0167-739X. https://www.sciencedirect.com/science/article/pii/S0167739X17311202 (visited on 11/12/2021) (page 81).

[Bad+20] Sebastian Bader et al. The International Data Spaces Information Model – An Ontology for Sovereign Exchange of Digital Content. *The Semantic Web – ISWC 2020*. Lecture Notes in Computer Science. Cham, 2020, ISBN: 978-3-030-62466-8 (pages 40, 81).

[Bec+10] Sean Bechhofer et al. Why Linked Data Is Not Enough for Scientists. *2010 IEEE Sixth International Conference on E-Science*. 2010 IEEE Sixth International Conference on E-Science. 2010, (page 82).

[Ber+05] Chad Berkley et al. Incorporating Semantics in Scientific Workflow Authoring. *Proceedings of the 17th International Conference on Scientific and Statistical Database Management*. SSDBM'2005. Berkeley, USA, 2005, (page 81).

[Ber06] Tim Berners-Lee. *Linked Data - Design Issues*. 2006. https://www.w3.org/DesignIssues/LinkedData.html (visited on 08/04/2021) (page 36).

[Ber98] Tim Berners-Lee. *Semantic Web Roadmap*. 1998. https://www.w3.org/DesignIssues/Semantic.html (visited on 08/04/2021) (pages 35, 36).

[BHB09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data: The Story so Far. *International Journal on Semantic Web and Information Systems* 5:(2009), 1–22 (page 37).

[BHL01] TIM BERNERS-LEE, JAMES HENDLER, and ORA LASSILA. THE SEMANTIC WEB. *Scientific American* 284(5):(2001), 34–43. ISSN: 0036-8733. JSTOR: 26059207. https://www.jstor.org/stable/26059207 (visited on 07/12/2022) (page 35).

[BK14] Eva Bratková and Helena Kucerová. Knowledge Organization Systems and Their Typology. *Revue of Librarianship* 25(2):(2014), 1–25 (page 25).

[Bla08] Simon Blackburn. *The Oxford Dictionary of Philosophy*. 2nd ed. Oxford Quick Reference. Oxford University Press, 2008 (page 11).

[BM20] Sebastian R. Bader and Maria Maleshkova. SOLIOT—Decentralized Data Control and Interactions for IoT. *Future Internet* 12(6):(6 2020), 105. https://www.mdpi.com/1999-5903/12/6/105 (visited on 09/18/2020) (page 81).

[Bol+08] Kurt Bollacker et al. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD '08. New York, NY, USA, 2008, ISBN: 978-1-60558-102-6. https://doi.org/10.1145/1376616.1376746 (visited on 08/23/2022) (page 30).

[Bor78] Sheldon A. Borkin. *Data Model Equivalence,* MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE, 1978. https://apps.dtic.mil/sti/citations/ADA062753 (visited on 07/22/2022) (page 21).

[Bra+21] Arnaud Braud et al. The Road to European Digital Sovereignty with Gaia-X and IDSA. *IEEE Network* 35(2):(2021), 4–5. ISSN: 1558-156X (page 91).

[Che+00] Ann Chervenak et al. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *J. Netw. Comput. Appl.* 23(3):(2000), 187–200 (page 81).

[Che+14] Mandy Chessell et al. Governing and Managing Big Data for Analytics and Decision Makers. An IBM Redguide Publication. 2014. https://www.redbooks.ibm.com/abstracts/redp5120.html?Open (visited on 03/27/2021) (page 39).

[Che+21] James Cheney et al. Data Provenance, Curation and Quality in Metrology. *CoRR* abs/2102.08228:(2021). arXiv: 2102.08228. https://arxiv.org/abs/2102.08228 (page 9).

[Cla02] Kendall Clark. Identity Crisis. 2002. https://www.xml.com/pub/a/2002/09/11/deviant.html (page 93).

[Con06] Dan Connolly. A Pragmatic Theory of Reference for the Web. *International World Wide Web Conference Committee (IW3C2)*. World Wide Web Conference 2006. Edinburgh, Scotland, 2006. https://www.w3.org/2006/04/irw65/urisym (page 93).

[Coo09] Roy T. Cook. *A Dictionary of Philosophical Logic*. Edinburgh University Press, 2009. ISBN: 978-0-7486-2559-8. JSTOR: 10.3366/j.ctt1g0b2x0. https://www.jstor.org/stable/10.3366/j.ctt1g0b2x0 (visited on 09/05/2022) (page 2).

[CZ06] Lois Mai Chan and Marcia Lei Zeng. Metadata Interoperability and Standardization–a Study of Methodology Part I. *D-Lib magazine* 12(6):(2006), 1082–9873 (page 26).

[DA12] Doan and AnHai (Auth.) *Principles of Data Integration*. Morgan Kaufmann, 2012. ISBN: 978-0-12-416044-6 (page 3).

[Dat14] Research Data-Alliance. Metadata Principles and Their Use. 2014. https://rd-alliance.org/metadata-principles-and-their-use.html (visited on 01/27/2022) (page 18).

[Dee+18] Ewa Deelman et al. The Future of Scientific Workflows. *The International Journal of High Performance Computing Applications* 32(1):(2018), 159–175. ISSN: 1094-3420. https://doi.org/10.1177/1094342017704893 (visited on 11/12/2021) (page 81).

[DFH11] John Domingue, Dieter Fensel, and James A. Hendler, eds. *Handbook of Semantic Web Technologies*. Berlin Heidelberg: Springer-Verlag, 2011. ISBN: 978-3-540-92912-3. https://www.springer.com/gp/book/9783540929123 (visited on 08/04/2021) (page 36).

[Dig09] Interagency Working Group on Digital Data. Harnessing the Power of Digital Data for Science and Society. 2009. https://www.nitrd.gov/pubs/Report_on_Digital_Data_2009.pdf (page 11).

[Don+23] Andy Donald et al. Towards a Semantic Approach for Linked Dataspace, Model and Data Cards. *Companion Proceedings of the ACM Web Conference 2023*. WWW '23 Companion. New York, NY, USA, 2023, ISBN: 978-1-4503-9419-2. https://doi.org/10.1145/3543873.3587659 (visited on 06/05/2023) (page 94).

[Edw+11] Paul N. Edwards et al. Science Friction: Data, Metadata, and Collaboration. *Social Studies of Science* 41(5):(2011), 667–690. ISSN: 0306-3127. https://doi.org/10.1177/0306312711413314 (visited on 09/05/2020) (page 81).

[Ehr+21] Lisa Ehrlinger et al. Data Catalogs: A Systematic Literature Review and Guidelines to Implementation. *Database and Expert Systems Applications - DEXA 2021 Workshops*. Communications in Computer and Information Science. Cham, 2021, ISBN: 978-3-030-87101-7 (pages 40, 92, 95).

[Els+19] T. Elsaleh et al. IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams. *2019 Global IoT Summit (GIoTS)*. 2019 Global IoT Summit (GIoTS). 2019, (page 113).

[ES22] Ariel Ezrachi and Maurice E. Stucke. *How Big-Tech Barons Smash Innovation–And How to Strike Back*. HarperCollins Publishers, 2022. 288 pp. ISBN: 978-0-06-303088-6. Google Books: jXSzzgEACAAJ (page 91).

[FC15] Andre Freitas and Edward Curry. Big Data Curation. *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*. 2015. ISBN: 978-3-319-21568-6 (page 9).

[Fen+19] Martin Fenner et al. A Data Citation Roadmap for Scholarly Data Repositories. *Scientific Data* 6(1):(1 2019), 28. ISSN: 2052-4463. https://www.nature.com/articles/s41597-019-0031-8 (visited on 03/17/2021) (page 81).

[Fen19] Katrina Fenlon. *Interactivity, Distributed Workflows, and Thick Provenance: A Review of Challenges Confronting Digital Humanities Research Objects*. Zenodo, 2019. https://zenodo.org/record/3459770 (visited on 11/08/2021) (page 82).

[FF14] Benedikt Fecher and Sascha Friesike. Open Science: One Term, Five Schools of Thought. *Opening Science: The Evolving Guide on How the Internet Is Changing Research, Collaboration and Scholarly Publishing*. Cham, 2014, ISBN: 978-3-319-00026-8. https://doi.org/10.1007/978-3-319-00026-8_2 (visited on 11/11/2021) (page 81).

[FHM05] Michael Franklin, Alon Halevy, and David Maier. From Databases to Dataspaces: A New Abstraction for Information Management. *ACM SIGMOD Record* 34(4):(2005), 27–33. ISSN: 0163-5808. https://doi.org/10.1145/1107499.1107502 (visited on 08/09/2021) (pages 1, 10, 40, 42, 92, 95).

[Flo08] Luciano Floridi. *The Blackwell Guide to the Philosophy of Computing and Information*. John Wiley & Sons, 2008. 389 pp. ISBN: 978-0-470-75676-8. Google Books: a37OrM9IUagC (page 97).

[Fri+17] Donna Fritzsche et al. Ontology Summit 2016 Communique: Ontologies within Semantic Interoperability Ecosystems. *Applied Ontology* 12(2):(2017), 91–111. ISSN: 1570-5838. https://content.iospress.com/articles/applied-ontology/ao181 (visited on 08/17/2022) (page 32).

[FT02] Roy T. Fielding and Richard N. Taylor. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology* 2(2):(2002), 115–150. ISSN: 1533-5399. https://doi.org/10.1145/514183.514185 (visited on 08/25/2022) (pages 2, 45, 93).

[Gie+19] Corinna Giebler et al. Leveraging the Data Lake: Current State and Challenges. *Big Data Analytics and Knowledge Discovery*. Lecture Notes in Computer Science. Cham, 2019, ISBN: 978-3-030-27520-4 (pages 39, 95).

[Gil08] Anne J Gilliland. Setting the Stage. *Introduction to Metadata*. 2nd ed. 2008, (pages 3, 18).

[Gle+20] Lars Gleim et al. FactDAG: Formalizing Data Interoperability in an Internet of Production. *IEEE Internet of Things Journal* 7(4):(2020), 3243–3253. ISSN: 2327-4662 (page 81).

[Gra+05] Jim Gray et al. Scientific Data Management in the Coming Decade. *SIGMOD Rec.* 34(4):(2005), 34–41 (page 81).

[Gua97] Nicola Guarino. Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*. Lecture Notes in Computer Science. Berlin, Heidelberg, 1997, ISBN: 978-3-540-69548-6 (page 26).

[GWF06] Georg Glasze, Christopher J Webster, and K Franz. Introduction: Global and Local Perspectives on the Rise of Private Neighbourhoods. *Private Cities: Local and Global Perspectives*:(2006) (page 8).

[Hal+20] Armin Haller et al. What Are Links in Linked Open Data? A Characterization and Evaluation of Links between Knowledge Graphs on the Web. *Journal of Data and Information Quality* 12(2):(2020), 9:1–9:34. ISSN: 1936-1955. https://doi.org/10.1145/3369875 (visited on 09/30/2021) (pages 36–38).

[Hal11] Harry Halpin. Sense and Reference on the Web. *Minds and Machines* 21(2):(2011), 153–178. ISSN: 1572-8641. https://doi.org/10.1007/s11023-011-9230-6 (visited on 08/03/2022) (pages 38, 44, 47, 93).

[Hal19] Harry Halpin. Decentralizing the Social Web. *Internet Science*. Lecture Notes in Computer Science. Cham, 2019, ISBN: 978-3-030-17705-8 (pages 44, 83).

[Hev+04] Alan R. Hevner et al. Design Science in Information Systems Research. *MIS Quarterly* 28(1):(2004), 75–105. ISSN: 0276-7783. JSTOR: 25148625 (page 14).

[HH10] Harry Halpin and Patrick J. Hayes. When Owl: sameAs Isn't the Same: An Analysis of Identity Links on the Semantic Web. *Proceedings of the WWW2010 Workshop on Linked Data on*

the Web, LDOW 2010, Raleigh, USA, April 27, 2010. Vol. 628. CEUR Workshop Proceedings. 2010. http://ceur-ws.org/Vol-628/ldow2010_paper09.pdf (page 38).

[Hit+12] Pascal Hitzler et al. *OWL 2 Web Ontology Language Primer (Second Edition)*. 2012. https://www.w3.org/TR/owl2-primer/ (visited on 08/04/2021) (page 37).

[HM78] Michael Hammer and Dennis McLeod. The Semantic Data Model: A Modelling Mechanism for Data Base Applications. *Proceedings of the 1978 ACM SIGMOD International Conference on Management of Data*. SIGMOD '78. New York, NY, USA, 1978, ISBN: 978-1-4503-7342-5. https://doi.org/10.1145/509252.509264 (visited on 07/22/2022) (pages 20, 21).

[Hod00] Gail M Hodge. *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files*. Vol. 91. Digital Library Federation, 2000 (pages 25, 26).

[Hog20] Aidan Hogan. *The Web of Data*. Springer International Publishing, 2020. ISBN: 978-3-030-51579-9. https://www.springer.com/de/book/9783030515799 (visited on 08/04/2021) (pages 35, 36).

[HP20] Armin Haller and Axel Polleres. Are We Better off with Just One Ontology on the Web? *Semantic Web* 11(1):(2020), 87–99. ISSN: 1570-0844. https://content.iospress.com/articles/semantic-web/sw190379 (visited on 09/30/2021) (pages 26, 30, 37).

[HQJ21] Rihan Hai, Christoph Quix, and Matthias Jarke. Data Lake Concept and Systems: A Survey. *CoRR* abs/2106.09592:(2021). arXiv: 2106.09592. https://arxiv.org/abs/2106.09592 (pages 1, 10, 38, 39, 92).

[Inm16] Bill Inmon. *Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump*. Basking Ridge, NJ: Technics Publications, 2016. 166 pp. ISBN: 978-1-63462-117-5 (pages 39, 40, 95).

[Int17] Dama International. *DAMA-DMBOK: Data Management Body of Knowledge (2nd Edition)*. Denville, NJ, USA: Technics Publications, LLC, 2017. 644 pp. ISBN: 978-1-63462-234-9 (pages 2, 4, 19).

[Jac+20] Annika Jacobsen et al. FAIR Principles: Interpretations and Implementation Considerations. *Data Intelligence* 2(1-2):(2020), 10–29. ISSN: 2641-435X. https://doi.org/10.1162/dint_r_00024 (visited on 03/20/2021) (pages 10, 81).

[Jag+88] D. Jagannathan et al. SIM: A Database System Based on the Semantic Data Model. *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*. SIGMOD '88. New York, NY, USA, 1988, ISBN: 978-0-89791-268-6. https://doi.org/10.1145/50202.50207 (visited on 07/22/2022) (page 20).

[Jam14] Reed Wicander James S. Monroe. *The Changing Earth: Exploring Geology and Evolution*. 7th ed. Cengage Learning, 2014 (pages 4, 50).

[JM17] Tomcy John and Pankaj Misra. *Data Lake for Enterprises: Lambda Architecture for Building Enterprise Data Systems*. Packt Publishing, 2017 (pages 39, 94).

[JQ17] Matthias Jarke and Christoph Quix. On Warehouses, Lakes, and Spaces: The Changing Role of Conceptual Modeling for Data Integration. *Conceptual Modeling Perspectives*. Cham, 2017, ISBN: 978-3-319-67271-7. https://doi.org/10.1007/978-3-319-67271-7_16 (visited on 03/27/2021) (pages 40, 81).

[JWG04] Ian Jacobs, Norman Walsh, and W3C Technical Architecture Group. *Architecture of the World Wide Web, Volume One*. Recommendation. 2004. https://www.w3.org/TR/webarch/ (page 93).

[Kam+16] Andreas Kamilaris et al. Agri-IoT: A Semantic Framework for Internet of Things-enabled Smart Farming Applications. *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). 2016, (page 113).

[Kas+18] Nasr Kasrin et al. Semantic Data Management for Experimental Manufacturing Technologies. *Datenbank-Spektrum* 18(1):(2018), 27–37 (pages 5, 59, 62, 81, 96).

[Kas+21] Nasr Kasrin et al. Data-Sharing Markets for Integrating IoT Data Processing Functionalities. *CCF Transactions on Pervasive Computing and Interaction* 3(1):(2021), 76–93. ISSN: 2524-5228. https://doi.org/10.1007/s42486-020-00054-y (visited on 10/21/2021) (pages 5, 81, 96, 113).

[KB10] Vijay Khatri and Carol V. Brown. Designing Data Governance. *Communications of the ACM* 53(1):(2010), 148–152. ISSN: 0001-0782. https://doi.org/10.1145/1629175.1629210 (visited on 03/18/2022) (page 83).

[KHM20] Petri Kannisto, David Hästbacka, and Arto Marttinen. Information Exchange Architecture for Collaborative Industrial Ecosystem. *Information Systems Frontiers* 22(3):(2020), 655–670. ISSN: 1572-9419. https://doi.org/10.1007/s10796-018-9877-0 (visited on 10/13/2021) (page 81).

[KMT17] Benjamin Knoke, Michele Missikoff, and Klaus-Dieter Thoben. Collaborative Open Innovation Management in Virtual Manufacturing Enterprises. *International Journal of Computer Integrated Manufacturing* 30(1):(2017), 158–166. ISSN: 0951-192X. https://doi.org/10.1080/0951192X.2015.1107913 (visited on 09/18/2020) (pages 5, 59, 81).

[KS95] Wolfgang Klas and Michael Schrefl. *Metaclasses and Their Application: Data Model Tailoring and Database Integration*. Vol. 943. Springer Science & Business Media, 1995 (page 20).

[LAL20] Huanyu Li, Rickard Armiento, and Patrick Lambrix. An Ontology for the Materials Design Domain. 2020. arXiv: 2006.07712 [cs]. http://arxiv.org/abs/2006.07712 (visited on 09/30/2021) (page 27).

[Las+21] Ora Lassila et al. Graph? Yes! Which One? Help! *CoRR* abs/2110.13348:(2021). arXiv: 2110.13348. https://arxiv.org/abs/2110.13348 (page 22).

[LEI08] Marcia LEI ZENG. Knowledge Organization Systems (KOS). *Knowledge Organization Systems (KOS)* 35(2-3):(2008), 160–182. ISSN: 0943-7444 (pages 21, 25, 26).

[Len95] Douglas B. Lenat. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11):(1995), 33–38. ISSN: 0001-0782. https://doi.org/10.1145/219717.219745 (visited on 09/30/2021) (page 22).

[LG89] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. 1st. USA: Addison-Wesley Longman Publishing Co., Inc., 1989. 372 pp. ISBN: 978-0-201-51752-1 (page 22).

[LG91] Douglas B. Lenat and R. V. Guha. The Evolution of CycL, the Cyc Representation Language. *ACM SIGART Bulletin* 2(3):(1991), 84–87. ISSN: 0163-5719. https://doi.org/10.1145/122296.122308 (visited on 03/23/2022) (page 22).

[LH09] Allen S. Lee and Geoffrey S. Hubona. A Scientific Basis for Rigor in Information Systems Research. *MIS Quarterly* 33(2):(2009), 237–262. ISSN: 0276-7783. JSTOR: 20650291 (page 14).

[Li+13] Yuan-Fang Li et al. An Ontology-centric Architecture for Extensible Scientific Data Management Systems. *Future Gener. Comput. Syst.* 29(2):(2013), 641–653. ISSN: 0167-739X. http://dx.doi.org/10.1016/j.future.2011.06.007 (visited on 09/15/2017) (page 81).

[LJ20] Martin Liebenberg and Matthias Jarke. Information Systems Engineering with Digital Shadows: Concept and Case Studies. *Advanced Information Systems Engineering*. Lecture Notes in Computer Science. Cham, 2020, ISBN: 978-3-030-49435-3 (page 81).

[LLM20] Anne Laurent, Dominique Laurent, and Cédrine Madera. *Data Lakes*. John Wiley & Sons, 2020. ISBN: 978-1-119-72043-0 (page 39).

[Lós+17] Bernadette Lóscio et al. *Data on the Web Best Practices*. Recommendation. 2017. https://www.w3.org/TR/dwbp/ (page 92).

[LR15] Avraham Leff and James T. Rayfield. Integrator: An Architecture for an Integrated Cloud/On-Premise Data-Service. *2015 IEEE International Conference on Web Services*. 2015 IEEE International Conference on Web Services. 2015, (page 81).

[LS07] Miltiadis D. Lytras and Miguel-Angel Sicilia. Where Is the Value in Metadata? *International Journal of Metadata, Semantics and Ontologies* 2(4):(2007), 235–241. ISSN: 1744-2621. https://www.inderscienceonline.com/doi/abs/10.1504/IJMSO.2007.019442 (visited on 07/22/2022) (page 18).

[LS16] Alice LaPlante and B. Sharma. *Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases*. First edition. Sebastopol, CA: O'Reilly Media, 2016. 1 online resource (1 volume). https://learning.oreilly.com/library/view/-/9781492042518/?ar (visited on 08/28/2021) (pages 39, 40, 95).

[Mas19] M Masnick. *Protocols, Not Platforms: A Technological Approach to Free Speech*. Knight First Amendment Institute. 2019. https://knightcolumbia.org/content/protocols-not-platforms-a-technological-approach-to-free-speech (page 91).

[Mat17] Christian Mathis. Data Lakes. *Datenbank-Spektrum* 17(3):(2017), 289–293. ISSN: 1610-1995. https://doi.org/10.1007/s13222-017-0272-7 (visited on 08/27/2021) (page 39).

[Meh+19] Hassan Mehmood et al. Implementing Big Data Lake for Heterogeneous Data Sources. *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*:(2019) (pages 39, 94).

[Met00] CC:DA Task Force on Metadata. Final Report. 2000. https://www.libraries.psu.edu/tas/jca/ccda/tf-meta6.html (page 31).

[ML16] Cedrine Madera and Anne Laurent. The next Information Architecture Evolution: The Data Lake Wave. *Proceedings of the 8th International Conference on Management of Digital EcoSystems*. MEDES. New York, NY, USA, 2016, ISBN: 978-1-4503-4267-4. https://doi.org/10.1145/3012071.3012077 (visited on 08/26/2021) (page 39).

[MM98] Christine Moorman and Anne S. Miner. Organizational Improvisation and Organizational Memory. *Academy of Management Review* 23(4):(1998), 698–723. ISSN: 0363-7425. https://journals.aom.org/doi/abs/10.5465/AMR.1998.1255634 (visited on 03/08/2022) (page 8).

[Mon+17] Barend Mons et al. Cloudy, Increasingly FAIR; Revisiting the FAIR Data Guiding Principles for the European Open Science Cloud. *Information Services & Use* 37(1):(2017), 49–56. ISSN: 0167-5265. https://content.iospress.com/articles/information-services-and-use/isu824 (visited on 11/12/2021) (page 81).

[Mon09] Alexandre Monnin. Artifactualization: Introducing a New Concept. InterFace 2009: 1st International Symposium for Humanities and Technology. Southampton, United Kingdom, 2009 (page 43).

[Mon12] Alexandre Monnin. The Artifactualization of Reference and "Substances" on the Web. *American Philosophical Association Newsletter (APA) Newsletters* 11(2):(2012). https://hal-paris1.archives-ouvertes.fr/hal-00672301 (visited on 08/25/2022) (pages 45, 46, 49, 93).

[Moo08] Reagan Moore. Towards a Theory of Digital Preservation. *International Journal of Digital Curation* 3(1):(1 2008), 63–75. ISSN: 1746-8256. http://ijdc.net/index.php/ijdc/article/view/63 (visited on 11/25/2021) (page 9).

[Nil10] Mikael Nilsson. From Interoperability to Harmonization in Metadata Standardization: Designing an Evolvable Framework for Metadata Harmonization [Doctoral Thesis]. Report. KTH School of Computer Science and Communication, 2010. https://repository.oceanbestpractices.org/handle/11329/1231 (visited on 07/22/2022) (pages 2, 18, 22–25, 91).

[NIS04] NISO. Understanding Metadata. 2004. https://www.lter.uaf.edu/metadata_files/UnderstandingMetadata.pdf (pages 2, 11, 17).

[NN10] Mikael Nilsson and Ambjörn Naeve. *Metadata Harmonization : A Roadmap for Standardization*. 2010. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-25689 (visited on 07/29/2022). preprint (page 23).

[Not23] Mark Nottingham. *Centralization, Decentralization, and Internet Standards*. Informational. 2023. https://www.ietf.org/archive/id/draft-nottingham-avoiding-internet-centralization-10.txt (page 96).

[NPB03] Mikael Nilsson, Matthias Palmér, and Jan Brase. The LOM RDF Binding : Principles and Implementation. Third Annual ARIADNE Conference, Leuven Belgium, 2003. 2003. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-25685 (visited on 07/29/2022) (page 23).

[OJ19] Boris Otto and Matthias Jarke. Designing a Multi-Sided Data Platform: Findings from the International Data Spaces Case. *Electronic Markets* 29(4):(2019), 561–580. ISSN: 1422-8890. https://doi.org/10.1007/s12525-019-00362-x (visited on 05/18/2021) (pages 40, 81, 91).

[OJW08] E. R. Orme, A. C. Jones, and R. J. White. LSID Deployment in the Catalogue of Life. *BNCOD 2008 Biodiversity Informatics Workshop; Cardiff University, United Kingdom*:(2008). https://www.vliz.be/en/maps-library?module=ref&refid=290608 (visited on 03/23/2022) (page 81).

[OR21] Nadine Ostern and Michael Rosemann. A Framework for Digital Affordances. *ECIS 2021 Research Papers*:(2021). https://aisel.aisnet.org/ecis2021_rp/145 (page 10).

[Ora15] Andrew Oram. *Managing the Data Lake: Moving to Big Data Analysis*. O'Reilly Media, 2015 (page 19).

[OS99] A. M. Ouksel and A. Sheth. Semantic Interoperability in Global Information Systems. *ACM SIGMOD Record* 28(1):(1999), 5–12. ISSN: 0163-5808. https://doi.org/10.1145/309844.309849 (visited on 08/17/2022) (pages 19, 96).

[Ott+18] Boris Otto et al. *IDS Reference Architecture Model. Industrial Data Space. Version 2.0*. Berlin: International Data Spaces Association, 2018. 91 pp. (page 81).

[Pat+05] Manjula Patel et al. *Semantic Interoperability in Digital Library Systems*. UKOLN, University of Bath, 2005. http://delos-wp5.ukoln.ac.uk/project-outcomes/SI-in-DLs/SI-in-DLs.pdf (visited on 08/17/2022) (page 27).

[Pel+16] Thomas Pellissier Tanon et al. From Freebase to Wikidata: The Great Migration. *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Republic and Canton of Geneva, CHE, 2016, ISBN: 978-1-4503-4143-1. https://doi.org/10.1145/2872427.2874809 (visited on 08/23/2022) (page 30).

[Pen+19a] Jan Pennekamp et al. Security Considerations for Collaborations in an Industrial IoT-based Lab of Labs. *2019 IEEE Global Conference on Internet of Things (GCIoT)*. 2019 IEEE Global Conference on Internet of Things (GCIoT). 2019, (page 81).

[Pen+19b] Jan Pennekamp et al. Towards an Infrastructure Enabling the Internet of Production. *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS). 2019, (page 81).

[Pik+19] Matthew Pike et al. Sensor Networks and Data Management in Healthcare: Emerging Technologies and New Challenges. *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). Vol. 1. 2019, (page 113).

[Pol+20] Axel Polleres et al. A More Decentralized Vision for Linked Data. *Semantic Web* 11(1):(2020), 101–113. ISSN: 1570-0844. https://content.iospress.com/articles/semantic-web/sw190380 (visited on 09/30/2021) (pages 36, 38, 84).

[QHV16] Christoph Quix, Rihan Hai, and Ivan Vatov. Metadata Extraction and Management in Data Lakes With GEMMS. *Complex Systems Informatics and Modeling Quarterly* (9):(9 2016), 67–83. ISSN: 2255-9922. https://csimq-journals.rtu.lv/article/view/csimq.2016-9.04 (visited on 08/23/2021) (pages 39, 94).

[Qui+20] Erwann Quimbert et al. Data Cataloguing. *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences: A Reference Model Guided Approach for Common Challenges*. Lecture Notes in Computer Science. Cham, 2020, ISBN: 978-3-030-52829-4. https://doi.org/10.1007/978-3-030-52829-4_8 (visited on 07/21/2022) (pages 18, 19, 40, 95).

[Ram+19] Seeram Ramakrishna et al. Materials Informatics. *Journal of Intelligent Manufacturing* 30(6):(2019), 2307–2326. ISSN: 1572-8145. https://doi.org/10.1007/s10845-018-1392-0 (visited on 11/28/2021) (pages 5, 59).

[Rei+11] Peter Reimann et al. *SIMPL – a Framework for Accessing External Data in Simulation Workflows*. Gesellschaft für Informatik e.V., 2011. ISBN: 978-3-88579-274-1. http://dl.gi.de/handle/20.500.12116/19600 (visited on 11/13/2021) (page 81).

[Ril17] Jenn Riley. Understanding Metadata: What Is Metadata, and What Is It For?: A Primer. 2017. https://www.niso.org/publications/understanding-metadata-2017 (page 19).

[Ros21] Michael Rosemann. Designing Intelligent Systems: The Role of Affordances and Trust (Extended Abstract of Keynote). *Advanced Information Systems Engineering*. 33rd International Conference, CAiSE 2021. Lecture Notes in Computer Science. 2021, (page 10).

[RPC19] Theofanis P. Raptis, Andrea Passarella, and Marco Conti. Data Management in Industry 4.0: State of the Art and Open Challenges. *IEEE Access* 7:(2019), 97052–97093. ISSN: 2169-3536 (pages 5, 59, 81).

[RS13] Armin Roth and Sebastian Skritek. Peer Data Management. *Data Exchange, Integration, and Streams*. Vol. 5. Dagstuhl Follow-Ups. Dagstuhl, Germany, 2013, ISBN: 978-3-939897-61-3. http://drops.dagstuhl.de/opus/volltexte/2013/4294 (visited on 11/12/2021) (page 97).

[RZ19] Franck Ravat and Yan Zhao. Metadata Management for Data Lakes. *New Trends in Databases and Information Systems*. Communications in Computer and Information Science. Cham, 2019, ISBN: 978-3-030-30278-8 (pages 19, 40, 95).

[Saw+19] Pegdwendé N. Sawadogo et al. Metadata Systems for Data Lakes: Models and Features. *New Trends in Databases and Information Systems*. Communications in Computer and Information Science. Cham, 2019, ISBN: 978-3-030-30278-8 (page 19).

[SD20] Pegdwendé Sawadogo and Jérôme Darmont. On Data Lake Architectures and Metadata Management. *Journal of Intelligent Information Systems*:(2020). ISSN: 1573-7675. https://doi.org/10.1007/s10844-020-00608-7 (visited on 09/05/2020) (pages 1, 10, 19, 38–40, 92, 95).

[SO11] Sebastian Schlauderer and Sven Overhage. How Perfect Are Markets for Software Services? An Economic Perspective on Market Deficiencies and Desirable Market Features. *ECIS 2011 Proceedings*:(2011). https://aisel.aisnet.org/ecis2011/110 (page 14).

[Soe09] D. Soergel. Knowledge Organization Systems: Overview. 2009. https://web.archive.org/web/20180721224631id_/http://www.dsoergel.com/UBLIS571DS-08.2a-1Reading4SoergelKOSOverview.pdf (pages 21, 25).

[SP11] Owen Sacco and Alexandre Passant. A Privacy Preference Ontology (PPO) for Linked Data. *WWW2011 Workshop on Linked Data on the Web, Hyderabad, India, March 29, 2011*. Vol. 813. CEUR Workshop Proceedings. 2011. http://ceur-ws.org/Vol-813/ldow2011-paper01.pdf (page 37).

[SPG05] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A Survey of Data Provenance in E-Science. *ACM SIGMOD Record* 34(3):(2005), 31–36. ISSN: 0163-5808. https://doi.org/10.1145/1084805.1084812 (visited on 09/05/2020) (page 81).

[SR14] Guus Schreiber and Yves Raimond. *RDF 1.1 Primer*. 2014. https://www.w3.org/TR/rdf11-primer/ (visited on 08/04/2021) (pages 36, 37).

[Sta11] International Organization for Standardization. ISO/IEC 42010:2007 Systems and Software Engineering – Architecture Description. 2011. https://www.iso.org/standard/50508.html (pages 4, 35).

[Ste75] Interim Report: ANSI/X3/SPARC Study Group on Data Base Management Systems 75-02-08. *Bulletin of ACM SIGMOD* 7(2):(1975), 1–140. http://portal.acm.org/toc.cfm?id=984332 (page 24).

[STW84] M. Schrefl, A. M. Tjoa, and R. R. Wagner. Comparison-Criteria for Semantic Data Models. 1984 IEEE First International Conference on Data Engineering. 1984, ISBN: 978-0-8186-0533-8. https://www.computer.org/csdl/proceedings-article/icde/1984/07271263/12OmNBuL1dC (visited on 07/22/2022) (page 21).

[Tar22] Ben Tarnoff. *Internet for the People: The Fight for Our Digital Future*. Verso Books, 2022. 273 pp. ISBN: 978-1-83976-202-4. Google Books: vGiVzQEACAAJ (page 91).

[Tay99] Arlene G. Taylor. *The Organization of Information. Library and Information Science Text Series*. Libraries Unlimited, Inc, 1999. ISBN: 978-1-56308-498-0 (page 19).

[Tra+18] Luca Trani et al. Establishing Core Concepts for Information-Powered Collaborations. *Future Generation Computer Systems* 89:(2018), 421–437. ISSN: 0167-739X. https://www.sciencedirect.com/science/article/pii/S0167739X17327619 (visited on 07/21/2022) (page 9).

[TVY90] Hideaki Takeda, Paul Veerkamp, and Hiroyuki Yoshikawa. Modeling Design Process. *AI Magazine* 11(4):(4 1990), 37–37. ISSN: 2371-9621. https://ojs.aaai.org/index.php/aimagazine/article/view/855 (visited on 10/20/2021) (page 14).

[UG02] Michael Uschold and Michael Gruninger. Creating Semantically Integrated Communities on the World Wide Web. Semantic Web Workshopz, Co-located with WWW 2002. Honolulu, HI, 2002 (page 32).

[UG04] Michael Uschold and Michael Gruninger. Ontologies and Semantics for Seamless Connectivity. *ACM SIGMOD Record* 33(4):(2004), 58–64. ISSN: 0163-5808. https://doi.org/10.1145/1041410.1041420 (visited on 09/28/2022) (pages 21, 32).

[vdVW08] Hans van der Veer and Anthony Wiles. *Achieving Technical Interoperability - the ETSI Approach (ETSI White Paper No. 3)*. 3rd edition. 2008. https://www.etsi.org/images/files/ETSIWhitePapers/IOP%20whitepaper%20Edition%203%20final.pdf (page 96).

[Vel01] Kim H. Veltman. Syntactic and Semantic Interoperability: New Approaches to Knowledge and the Semantic Web. *New Review of Information Networking* 7(1):(2001), 159–183. ISSN: 1361-4576. https://doi.org/10.1080/13614570109516975 (visited on 07/22/2022) (pages 31, 97).

[Ver20] Ruben Verborgh. Re-Decentralizing the Web, for Good This Time. *Linking the World's Information: A Collection of Essays on the Work of Sir Tim Berners-Lee*. 2020. https://ruben.verborgh.org/articles/redecentralizing-the-web/ (pages 83, 84).

[VV20] Ruben Verborgh and Miel Vander Sande. The Semantic Web Identity Crisis: In Search of the Trivialities That Never Were. *Semantic Web Journal* 11(1):(2020), 19–27. https://ruben.verborgh.org/articles/the-semantic-web-identity-crisis/ (pages 37, 38).

[W81] Shipman D. W. The Functional Data Model and the Data Language DAPLEX. *ACM Trans. Database Syst.* 6(1):(1981), 140–173. https://cir.nii.ac.jp/crid/1573105976297882624 (visited on 07/22/2022) (page 20).

[Wet18] Linda Wetzel. Types and Tokens. *The Stanford Encyclopedia of Philosophy*. Fall 2018. 2018. https://plato.stanford.edu/archives/fall2018/entries/types-tokens/ (visited on 09/04/2022) (page 52).

[WG17] Vocabulary Management WG. Issues in Vocabulary Management (NISO TR-06-2017). 2017. https://www.niso.org/node/17201 (page 28).

[Wil+16] Mark D. Wilkinson et al. The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Scientific Data* 3(1):(1 2016), 160018. ISSN: 2052-4463. https://www.nature.com/articles/sdata201618 (visited on 03/17/2021) (pages 1, 10, 81, 92).

[WM] William David Williams and Kenneth H. Mann. *Inland Water Ecosystem*. Encyclopedia Britannica. https://www.britannica.com/science/inland-water-ecosystem (visited on 05/19/2021) (page 4).

[Wu+15] Dazhong Wu et al. Cloud-Based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation. *Computer-Aided Design* 59:(2015), 1–14. ISSN: 0010-4485.

http://www.sciencedirect.com/science/article/pii/S0010448514001560 (visited on 09/18/2020) (pages 5, 59, 81).

[Zan+14] Andrea Zanella et al. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1(1):(2014), 22–32. ISSN: 2327-4662 (page 113).

[ZC15] Marcia Lei Zeng and Lois Mai Chan. Semantic Interoperability. *Encyclopedia of Library and Information Sciences, Third Edition*. 3rd ed. 2015. ISBN: 978-0-203-75763-5 (page 31).

[ZdB14] Paul Zikopoulos, Dirk deRoos, and Christopher Bienko. *Big Data Beyond the Hype: A Guide to Conversations for Today's Data Center*. UK ed. edition. New York, NY Chicago San Francisco: McGraw-Hill Professional, 2014. 394 pp. ISBN: 978-0-07-184465-9 (pages 39, 95).

[Zen19] Marcia Lei Zeng. Interoperability. *KO KNOWLEDGE ORGANIZATION* 46(2):(2019), 122–146. ISSN: 0943-7444. https://www.nomos-elibrary.de/10.5771/0943-7444-2019-2-122/interoperability-jahrgang-46-2019-heft-2?page=1 (visited on 08/17/2022) (page 32).

[Zie+21] Julian Ziegler et al. A Metadata Model to Connect Isolated Data Silos and Activities of the CAE Domain. *Advanced Information Systems Engineering*. Lecture Notes in Computer Science. Cham, 2021, ISBN: 978-3-030-79382-1 (pages 5, 59, 60, 81).

[ZM19] Marcia Lei Zeng and Philipp Mayr. Knowledge Organization Systems (KOS) in the Semantic Web: A Multi-Dimensional Review. *International Journal on Digital Libraries* 20(3):(2019), 209–230. ISSN: 1432-1300. https://doi.org/10.1007/s00799-018-0241-2 (visited on 07/17/2022) (page 26).

[ZMW20] Amy X. Zhang, Michael Muller, and Dakuo Wang. How Do Data Science Workers Collaborate? Roles, Workflows, and Tools. *Proceedings of the ACM on Human-Computer Interaction* 4:(CSCW1 2020), 022:1–022:23. https://doi.org/10.1145/3392826 (visited on 11/10/2021) (page 81).

[ZQ16] Marcia Lei Zeng and Jian Qin. *Metadata*. 2nd ed. American Library Association, 2016. 584 pp. ISBN: 978-1-55570-965-5. Google Books: V1MungEACAAJ (page 19).