

Chapter 4

Semantic Interoperability Framework for Digital Finance Applications



Giovanni Di Orio, Guilherme Brito, and Pedro Maló

1 Introduction

Data are today the most valuable and precious asset within organizations; however, they also represent an ongoing challenge for businesses since data is continually growing in variety, complexity, as well as fragmentation. As a matter of fact, most of the data collected and possessed by financial organizations reside in a wide array of “siloes” (i.e., fragmented) and heterogeneous systems and databases. Online Transaction Processing (OLTP) databases, Online Analytical Processing (OLAP) databases, data warehouses, and data lakes are only few examples within the data landscape. Furthermore, intensive and heavy data consumption tasks are usually performed over OLAP systems, which lead financial organizations in transferring data from OLTP, data lakes, and other systems to OLAP systems based on intrusive and expensive extract-transform-load (ETL) processes. In several cases, ETLs consume 75–80% of the budget allocated to data analytics while being a setback to seamless interoperability across different data systems using up-to-date data. Beyond the lack of integrated OLTP and OLAP processes, financial and insurance organizations have no unified way of accessing and querying vast amounts of structured, unstructured, and semi-structured data, which increases the effort and cost that are associated with the development of big data analytics and artificial intelligence (AI) systems. Except for data fragmentation, there is also a lack of interoperability across diverse datasets that refer to the same data entities with similar semantics. This is a main obstacle to datasets sharing across different stakeholders and to enabling more connected applications and services that span multiple systems across the financial supply chain. The impacts these aspects

G. Di Orio (✉) · G. Brito · P. Maló
NOVA School of Science and Technology, Caparica, Portugal
e-mail: gido@uninova.pt; guilherme.brito@uninova.pt; pmm@uninova.pt

© The Author(s) 2022
J. Soldatos, D. Kyriazis (eds.), *Big Data and Artificial Intelligence in Digital Finance*,
https://doi.org/10.1007/978-3-030-94590-9_4

have on data management are huge and are forcing both financial and insurance organizations to research, rethink, and apply new strategies and approaches for data management.

Data management can be defined as [1] “the development, execution, and supervision of plans, policies, programs, and practices that deliver, control, protect, and enhance the value of data and information assets throughout their life cycles.” The inclusion of these disciplines triggers a transformation process that allows organizations to become data-driven [2], i.e., to ensure the right usage of data at the right time to support operational business intelligence (BI). The activities related to data management span multiple areas from data architecture and data modeling and design to data governance and security passing through data interoperability and data persistence operations. This chapter focuses on the data integration, data interoperability, and data modeling aspects of data management.

2 Background: Relevant Concepts and Definitions for the INFINITECH Semantic Interoperability Framework

2.1 Interoperability

There is no unique definition of interoperability in the literature since the concept has different meanings depending on the context. As a matter of fact, according to ISO/IEC 2382-01 [3], interoperability is defined as “The capability to communicate, execute program, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.” According to ETSI’s technical committee TISPAN [4], interoperability is “the ability of equipment from different manufacturers (or different systems) to communicate together on the same infrastructure (same system), or on another.” Moreover, EICTA defines interoperability as [5] “the ability of two or more networks, systems, devices, applications or components to exchange information between them and to use the information so exchanged.” Based on these definitions, interoperability is always about making sure that systems are capable of sharing data between each other and of understanding the exchanged data [6]. In this context, the word “understand” includes the content, the format, as well as the semantics of the exchanged data [7]. Interoperability ranges over four different levels [8], namely:

1. Physical/technical interoperability: It is concerned with the physical connection of hardware and software platforms.
2. Syntactical interoperability: It is concerned with data format, i.e., it relates on how the data are structured.
3. Semantic interoperability: It is concerned with the meaningful interaction between systems, devices, components, and/or applications.
4. Organizational interoperability: It is concerned with the way organizations share data and information (Fig. 4.1).

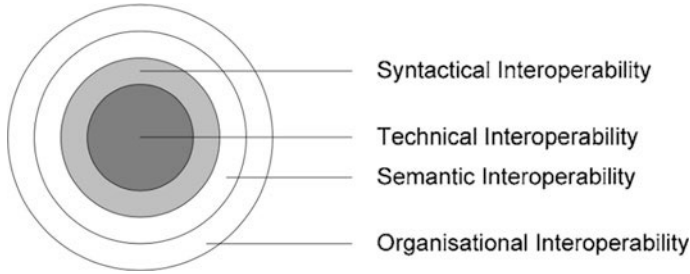


Fig. 4.1 Different interoperability levels according to [8]

2.1.1 Semantic Interoperability

Semantics plays a main role in interoperability for ensuring that exchanged information between counterparts are provided with sense. For computer systems, this notion of semantic interoperability translates in the ability of two or more systems to exchange data between them, by means of precise unambiguous and shared meaning, which enables readily access and reuse of the exchanged data. The concept of Semantic Web [9] was introduced by World Wide Web (WWW) founder Tim Berners-Lee in the 1990s and has been widely used in research and industry contexts. It has also given rise to the development of the concepts of Semantic Web services and more recently of Semantic Internet of Things (IoT) concepts [10–12]. These concepts aim to facilitate collaboration across semantically heterogeneous environments toward contributing to a connected world of consuming and provisioning devices. This connected world can potentially exchange and combine data to offer new or augmented services. However, the accomplishment of this vision is associated with several challenges due to the existence of a variety of standards, legacy systems constraints, and tools. The semantic interoperability process can, therefore, focus on different viewpoints of semantic aspects, such as the exchanged data description or the systems interaction terms. As a prominent example, in IoT systems, the interoperability specifications can be used to define the meaning of a given sensor or IoT device, but also to provide information on the units of its value and on the protocols that can be used to connect and extract the value from the device.

2.1.2 Semantic Models

The provision of semantic information modeling can be implemented in various ways, including key-value, markup scheme, graphics, object-role, logic-based, and ontology-based models [13]. From this set, the key-value type offers the simplest data structure but lacks expressivity and inference. On the other hand, the ontology-based model provides the best way to express complex concepts and interrelations, being therefore the most popular model for elaborating semantic models.

2.1.3 Ontologies

The inherent semantic interoperability features of the Semantic Web have been mostly grounded on the use of ontologies for knowledge representation. In most cases, there exists a top-level ontology (or domain ontology) and multiple sub-domain ontologies, each one representative of a more specific domain. With the use of ontologies, entities can be described in very comprehensive ways [14].

2.1.4 Semantic Annotations

Semantic annotation is the process of attaching additional information to any element of data comprised of some sort of document. Ontologies on their own are not sufficient to fulfill the semantic interoperability requirements needed to enable data readability by machines. This is because of the differences and inconsistencies across different data sources and their ontologies. Semantic annotation has been widely used to fill this gap by creating links between the disparate ontologies and the original sources [15].

2.2 Methodologies for Ontology Engineering

2.2.1 METHONTOLOGY

METHONTOLOGY has been developed by the Ontology Engineering Group at the Universidad Tecnica de Madrid. It is a structured method to build ontologies initially developed in the domain of chemicals [16]. The methodology guides the ontology development process throughout the whole ontology life cycle. It consists of the following main development activities:

- *Specification*: It is concerned with the definition of the objectives of the ontology and of the end users, i.e., it frames the domain.
- *Conceptualization*: It is concerned with developing an initial conceptual representation/model of a perceived view of the application domain. A set of intermediate representations are here used to organize the concepts to be easily understood by both ontology and domain experts.
- *Formalization*: It is concerned with the implementation of a semi-computable model from the conceptual model generated in the previous activity.
- *Integration*: It is concerned with knowledge reuse, i.e., extracting and integrating definitions and concepts from already built ontologies.
- *Implementation*: It is concerned with the implementation of fully computational models using various ontology languages.
- *Maintenance*: It is concerned with any update to the ontology.

Furthermore, as part of the methodology, several orthogonal supporting activities are also identified to manage and support the development ones. These activities span knowledge acquisition, documentation, and evaluation.

2.2.2 SAMOD

The Simplified Agile Methodology for Ontology Development (SAMOD) [17] focuses on designing and developing well-developed and well-documented models from significant domain data and/or descriptions. It consists of three simple and small steps that are part of an iterative process aimed to produce preliminary and incremental results. The three steps can be labeled as:

1. *Test case definition*: This is about writing down a motivating scenario, being as close as possible to the language commonly used for talking about the domain.
2. *Merging current model with modelet*: This merges the modelet included in the defined test case with the current model.
3. *Refactoring current model*: This refactors the current model shared among all the defined test cases.

2.2.3 DILIGENT

The methodology for distributed, loosely controlled, and evolving engineering of ontologies (DILIGENT) [18] is a methodological approach intended to support domain experts in a distributed setting to engineer and evolve ontologies. It is based on rhetorical structure theory, viz., the DILIGENT model of ontology engineering by argumentation. The process comprises five main activities, namely:

1. *Build*: It concerns with the development of ontologies by having different stakeholders, with different needs and purposes that are typically distributed.
2. *Local Adaptation*: It concerns with the usage and adaptation of the developed ontology. By using the ontology, many updates can be necessary due, for example, to new business requirements and/or new arising needs.
3. *Analysis*: It concerns with the analysis of any local request for update. As a matter of fact, local ontologies can be updated, but the shared ontology will be updated only after the analysis of the update request.
4. *Revision*: It concerns with the constant revision of the shared ontology to guarantee the alignment with the local ones.
5. *Local Update*: It concerns with the update of the local ontologies after a new shared ontology is available.

2.2.4 UPON Lite

The lightweight unified process for ontology building (UPON Lite) methodology [19] is a simple, agile ontology engineering approach and/or method that is intended to place the end users and domain experts at the center of the overall ontology building process while avoiding the presence of ontology engineers. Therefore, the main pillars of the process are (i) the adoption of a fully user-centered approach, (ii) the adoption of a social collaborative approach to collect domain expert knowledge to achieve all the steps in the method, and (iii) an ontology building method based on six main activities. The six activities and/or steps of the UPON Lite method are the following (named and/or labeled according to the produced outcome):

1. *Domain terminology*: It is concerned with producing the list of all the fundamental domain terms that characterize the observed domain.
2. *Domain glossary*: It provides the definition and possible synonyms of the domain terms.
3. *Taxonomy*: It concerns with the organization of the domain terms according to an “ISA” hierarchy.
4. *Predication*: It concerns with the identification of those terms that represent properties and/or relations between other terms and/or concepts.
5. *Parthood*: It concerns with the analysis of the structure of the identified concepts and/or entities in order to elicit their (de-)composition hierarchies.
6. *Ontology*: It concerns with the production of the formally encoded ontology.

3 INFINITECH Semantic Interoperability Framework

The INFINITECH Semantic Interoperability Framework is a commonly agreed approach to enable semantic interoperability between applications and services within the INFINITECH platform while defining basic interoperability guidelines in the form of common principles, models, and recommendations. Furthermore, as part of the framework, ontology mapping processes are also considered to establish a common platform to deal with multiple ontologies. The proposed framework has been designed by combining a top-down and bottom-up approach (hybrid approach) as shown in Fig. 4.2. The latter – also called *pilot characterization* – is aimed to describe the specific application domain for each one of the test beds and pilot within the project. The main objective here is the identification, the definition, and the clear description of the context of application in terms of domain terminologies, glossaries, and taxonomies. The former – also called *state-of-the-art (SotA) analysis* – is aimed to identify reference ontologies for considered domain (finance and insurance); these ontologies are not linked to a specific application domain. The main objective here is the identification of a common and above all generic set of core concepts and relationships between them that can be used as top ontology, i.e., the glue between diverse specific domain

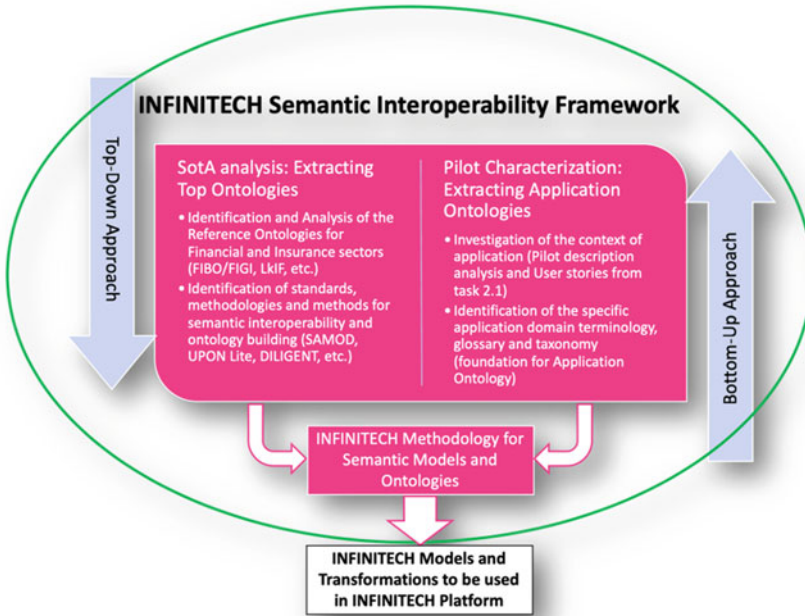


Fig. 4.2 Hybrid approach for interoperability in INFINITECH

ontologies for the same context of application. In both cases, the combination of the results of the *pilot characterization* and *SotA analysis* is used as inputs of the INFINITECH methodology for semantic models and ontologies and used for generating INFINITECH models, as well as baseline for the development of transformers that needs to be used to exploit all the features and full potentiality of the INFINITECH platform. Therefore, the hybrid approach aims firstly to design a high-level normalized and domain-specific enterprise model and secondly to link this model to the business-specific data.

The INFINITECH Semantic Interoperability Framework focuses primarily on the planning/designing and development/implementation of the objectives, infrastructure, and necessary deliverables to ensure interoperability in digital finance applications. The successful execution of the planning/development and development/implementation phases – of the framework – strictly depends on the presence of a data governance layer. As pointed out in [20], data governance is a business-driven process where specific business data representations are aligned to data domain. To do that, several data governance procedures have been carried out, namely (see Fig. 4.3):

- Identifying potential data sources and data owners
- Assigning roles and responsibilities to data management processes

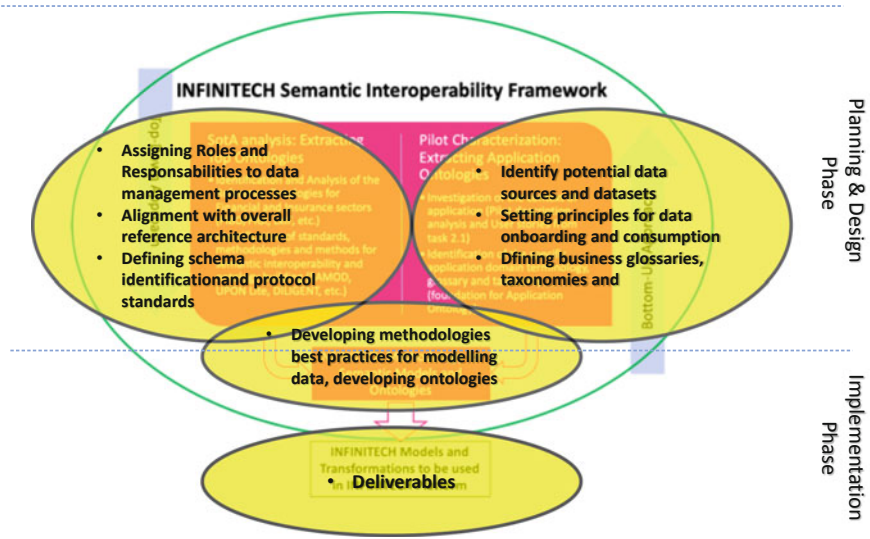


Fig. 4.3 Mapping data governance activities with proposed hybrid approach

- Defining the granularity of the data according to the type of applications needed to deliver it
- Alignment with the overall reference architecture
- Defining schema identification requirements and protocol standards for common data formats
- Developing methodologies and best practices for modeling data, developing ontologies, defining business glossaries, etc.
- Setting principles for data onboarding and consumption

3.1 Methodology for Semantic Models, Ontology Engineering, and Prototyping

Ontologies are the baseline for developing semantic interoperability applications. Ontologies are conceptual models – constituted by interlinked concepts related to a specific domain – of an observed reality. Since ontologies play a fundamental role in INFINITECH while providing the necessary mechanisms for describing test beds and pilot application domain, then a systematic engineering approach is needed to facilitate the design and development of high-quality and, above all, pilot-aligned ontologies to reference top-level ontologies for the domain.

As shown in Fig. 4.4, the INFINITECH methodology for ontology engineering shares terminology, definitions, and activities and/or steps with the SAMOD

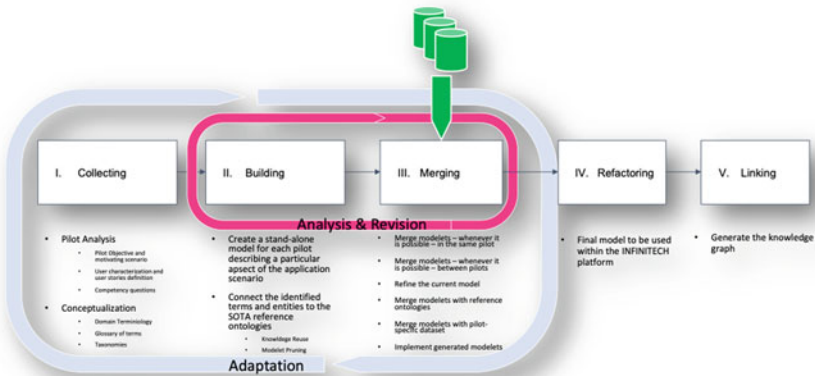


Fig. 4.4 Methodology for semantic models and ontology engineering

methodology. It is an iterative process that is aimed at building semantic models and ontologies. It is organized as a sequence of four sequential steps, namely:

1. *Collecting*. This step collects all the information about the application domain. It involves the following tasks and/or activities:
 - *Pilot Analysis*: Write down the motivating scenario and identify user expectation by writing down *user stories* and clarifying everything by using a set of competency questions (user characterization).
 - *Conceptualization*: Write down domain terminology, glossary of terms, and taxonomies of concepts.
2. *Building*. This step builds a new interoperability test case (*aka modelet*). The *modelet* is a stand-alone model describing the application domain for the considered pilot and/or test bed. The step involves the following tasks and/or activities:
 - Creation of a stand-alone model for the pilot or test bed describing the relevant aspects of the application domain.
 - Connection with the top reference ontology(ies). This activity is aimed to reuse as much as possible already defined concepts, relations, and properties while pruning all the elements that are superfluous.
3. *Merging*. This step refines the generated *modelet* with concepts and relations extracted from reference ontologies for the domain to determine more generic domain ontologies. The step involves the following tasks and/or activities:
 - Merge *modelets* in the same pilot/test bed.
 - Merge *modelets* between different pilots/test beds within the same application domain.

- Refinement of the current modelet.
 - Merge *modelets* with reference ontologies.
 - Implement generated *modelets*.
4. *Refactoring*. This step provides the final ontology and semantic model as conceptual schema to be used within INFINITECH. This model delivers the complete description and characterization of the application domain aligned with reference ontologies while enabling any user of the INFINITECH application to seamlessly access diverse ontologies and thus concrete data.
 5. *Linking*: This step links the refactored models to real data while generating the so-called linked knowledge graph.

Two iteration cycles (analysis and revision and adaptation) are part of the methodology. The analysis and revision iteration (executed essentially during the *building* step) aims at analyzing and reviewing the building process to guarantee the alignment with the domain expert's expectations and requirements. The result of this step and its related iterations is a preliminary model also called *modelet*. The adaptation iteration includes the steps *collecting*, *defining*, and *merging* and aims at refining the generated *modelets* to cope with new knowledge and/or any change in user characterization, user needs, application domain, or, more in general, any change that could directly have an impact on the way domain experts describe their own business and – thus – application domains.

Generated modelets are very specific and targeted conceptual models that need to be filled and populated with dynamic data from typically heterogeneous and distributed resources. Here is where the semantic graphs and/or knowledge graphs play a fundamental role.

3.1.1 Modeling Method

The main result of applying the methodology for semantic models and ontology engineering is an evolving conceptual schema (e.g., ontology) and linked knowledge graph that empowers the INFINITECH platform to access, query, use and process/analyze data and/or information from heterogeneous and distributed sources.

The conceptual schema is determined by using an evolving prototyping (i.e., a foundation of agile software methodologies like DevOps) approach, where it grows up by layers while continuously delivering software prototypes. In particular, the conceptual model is the combination of three layers, according to [2]:

- Top-level ontology: Describes at very high-level concepts of interest for the domain
- Domain ontology: Describes specific concepts typically related to sub-domains of the top-level model
- Application ontology: Describes very specific concepts related to the particular application and scenario

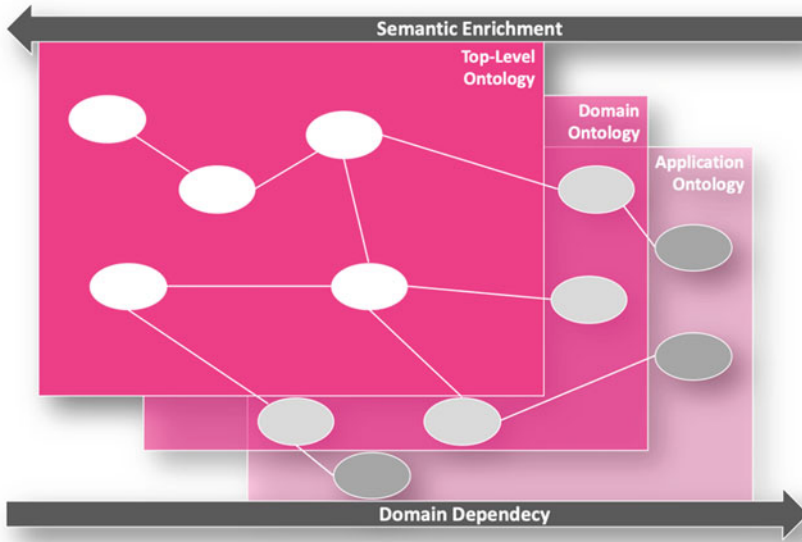


Fig. 4.5 Semantic and ontology modeling method

The layered model allows easy adaptation and extension while enabling for knowledge reuse, i.e., to reuse as much as possible already available ontologies and models. As a matter of fact, this model facilitates the adaptation to various applications as well as new domains (Fig. 4.5).

3.1.2 Envisioned Roles and Functions in Semantic Models, Ontology Engineering, and Prototyping

Several actors are typically involved in the process of defining, specifying, and developing semantic models and ontologies. The ontology engineering process is a collaborative process among several stakeholders. Since the main objective of the INFINITECH methodology for semantic models and ontology engineering is to provide a stakeholder-centric approach, it is necessary to identify the main roles and functions of the distinct actors of the process (see Fig. 4.6). The engineering process starts by having a small group composed by the following stakeholders: domain experts, end users, knowledge, and ontology engineers. The actors assume a distinct role considering the specific step within the methodology. In particular, during the *collecting* step, they assume the role of *data user* since they are essentially individuals who intend to use data for a specific purpose and are accountable for setting requirements. During the *building*, *merging*, and *refactoring*, ontology engineers play the role of *data trustee* since they are responsible for managing the data and

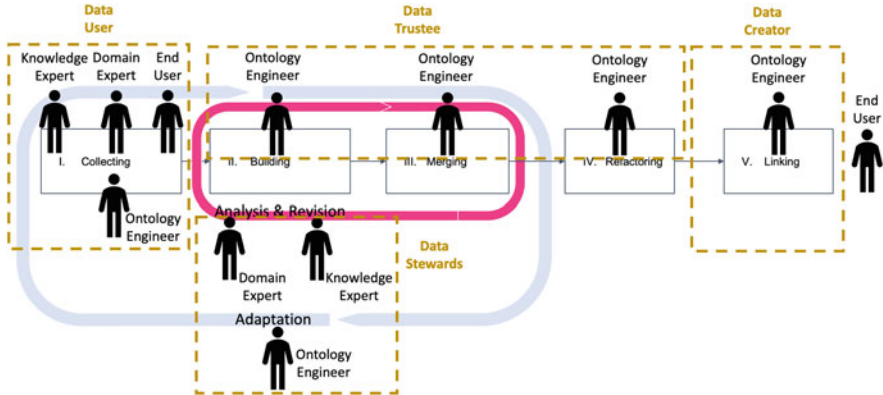


Fig. 4.6 Data governance: actors, roles, and functions

related metadata, data classifications, and conceptual schemas. Furthermore, during the *analysis and revision* process, the actors play the role of *data stewards* since they are responsible in ensuring the data policies and data standards are adhered to. Finally, during the *linking* step, the ontology engineer plays the role of *data creator* by physically creating and linking data to models as defined by the *data trustee*. This data is then ready to be consumed by end users.

4 Applying the Methodology: Connecting the Dots

This section illustrates the INFNITECH methodology for semantic models, ontology engineering, and prototyping, using exemplary data from considered pilots and selected supporting technologies. INFNITECH pilots have typically their own very specific data with different formats, data structure and differently organized. To establish the foundation for interoperability between those pilots, in the same application domain, ontologies are needed. However, most of them have not a well-defined and well-established conceptual model of their own application domain (the so-called application ontology). Furthermore, the usage of reference ontologies (such as FIBO, LKIF, FIGI, etc.) alone becomes practically impossible due to the lack of a connection with the application ontology. Therefore, it is peremptory to provide firstly pilots with application ontologies.

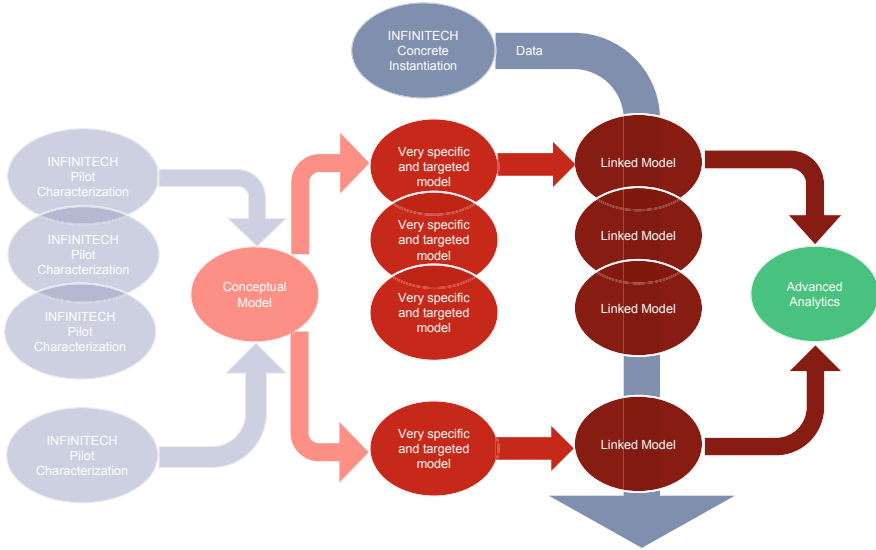


Fig. 4.7 Workflow and main deliverables and their connection

4.1 Workflow and Technological Tools for Validation of the Methodology

The proposed methodology is an iterative process that aims at providing very specific and targeted models to be used by advanced analytic applications (typically outside to the INIFITECH platform). Figure 4.7 shows the main output of the methodology starting from the pilot characterization and how it is connected to the INIFINITECH platform.

The *collecting* and *building* steps are initially used to define and build the conceptual model (*modelet*). The *modelet* needs to be merged, from one side to reference ontologies and from the other side to specific pilot conceptual model, and refactored to provide a very specific and highly targeted model. Finally, the model needs to be linked to real data from the pilot as the result of the *linking* step.

4.2 Collecting

The *collecting* step is the first step of the INFINITECH methodology and is aimed to characterize the application domain by providing three fundamental deliverables, namely (see Fig. 4.8):

- Domain terminology: The complete list of terms that are relevant for the application domain.



Fig. 4.8 *Collecting* step inputs and outputs

- Glossary of terms: The domain terminology enriched with the description of the term as well as possible synonyms. Furthermore, the object process actor modeling language (OPAL) semantic is also used at this stage that provides a first high-level classification of concepts.
- Taxonomy of identified concepts: The list of terms represented/organized into hierarchies according to the “ISA” relationship.

The output of this activity is then used – together with the description of available data – to create a conceptual model around the application domain during the next step: *building*.

4.3 *Building and Merging*

During the *building* step, a semantic approach is used where collected data are analyzed to understand the meaning of the data, as well as to identify the main concepts and relationships between them. The main result is a knowledge graph (see Fig. 4.9) that, in turn, represents the first step toward harmonization and standardization of data assets.

However, the knowledge graph needs to be further analyzed and refined to be connected to reference ontologies. This is done during the *merging* step where FIBO reference ontology is considered considering the domain of application.

At this stage, both pilot-specific and FIBO models have been analyzed to identify:

- Common concepts
- Connections and relations between the two models

The final knowledge graph is shown in Fig. 4.10.

The following FIBO concepts have been used for modeling the final knowledge graph of the INFINITECH real-time risk assessment pilot system:

- *Currency*: Medium of exchange value, defined by reference to the geographical location of the monetary authorities responsible for it.

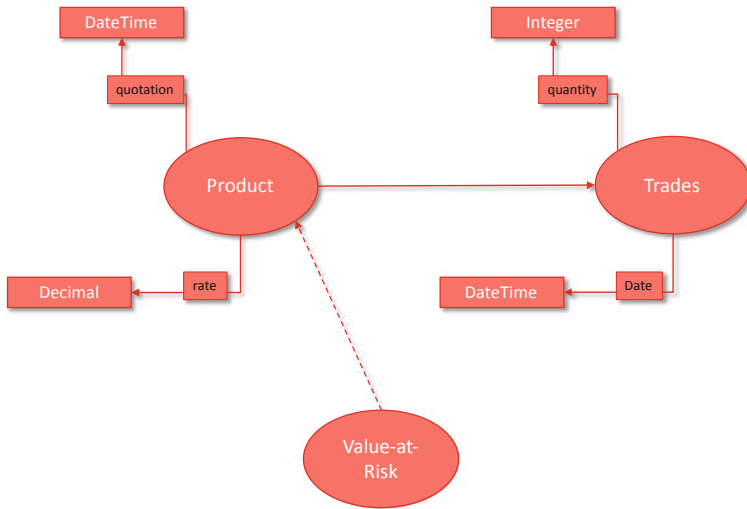


Fig. 4.9 Pilot-specific knowledge graph

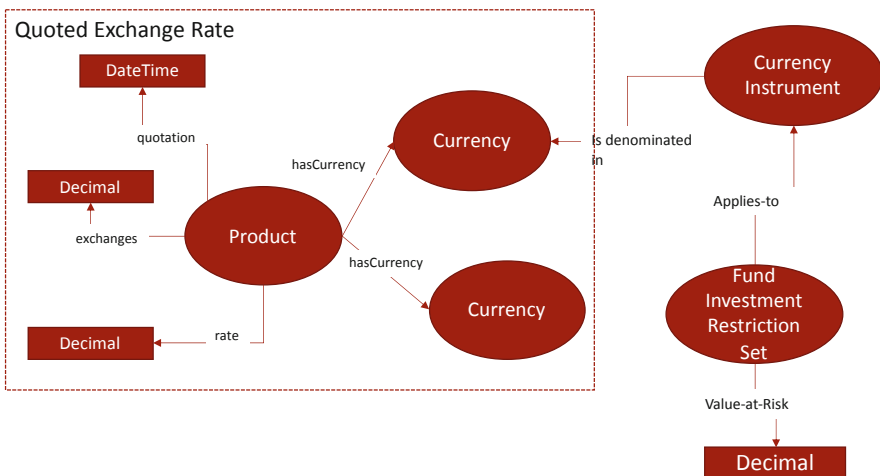


Fig. 4.10 Pilot-specific knowledge graph aligned with FIBO

- *Quoted Exchange Rate*: An exchange rate quoted at a specific point in time, for a given block amount of currency as quoted against another (base) currency. An exchange rate of R represents a rate of R units of the quoted currency to 1 unit of the base currency.
- *Value at Risk*: Measures and quantifies the level of financial risk within a firm, portfolio, or position over a specific time frame.

- *Fund Investment Restriction Set*: Limitations that apply to the fund as a whole, such as risk factors. These are used to determine whether the fund is appropriate for a given type of investor to invest in.
- *Currency Instrument*: Financial instrument used for the purposes of currency trading.

The result of the *merging* step is a connected graph – aligned with FIBO ontology – capable of spanning organizational concepts that are relevant for the selected application scenario and use cases.

4.4 Refactoring and Linking

The *refactoring* and *linking* stages aim at concretely developing and implementing knowledge graph produced after the *merging* stage while also linking it to the pilot-specific real data. Therefore, it is mainly focused on the selection of the model serialization format (RDF, JSON-LD, etc.) and concrete supporting technology for creating linked knowledge graphs. During these steps, selected technology is applied to support the overall process of data harmonization (according to FIBO) and data-sharing and provisioning to any external application that needs to use them. The main result of these steps is a linked knowledge graph that can be queried.

At this point, two technologies have been selected to show the repeatability of the process, regardless of the specific environment deployed within the pilot, namely, (1) [dataworld.com](https://data.world)¹, a cloud-based collaborative data catalog aimed to supply data integration and analysis, and (2) *GraphDB*², an RDF database for knowledge graphs with extended features for providing integration and transformation of data. With both solutions, the execution of the validation stages of the workflow was attempted, with support of other technologies used in INFINITECH, such as *LeanXcale* database for the origin data, *Node-RED* for client application, or *Docker* for deployment. As a result, a simple architecture was designed. The architecture specifies the necessary elements of the system, along with their internal interfaces and communications (Fig. 4.11).

4.4.1 Data Ingestion

The first necessary step for validation of the modeling methodology is to pull the real data from the source into the selected technology. Currently, it is already possible to import data from a wide range of sources (local files, cloud locations, different

¹ <https://data.world>

² <https://graphdb.ontotext.com>

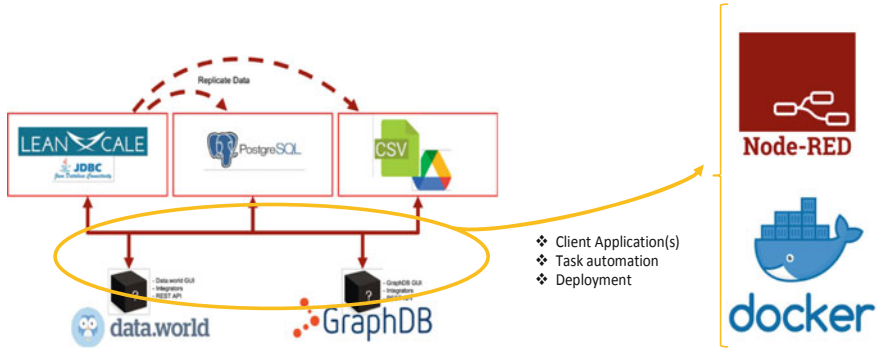


Fig. 4.11 Communication architecture of the validation setup

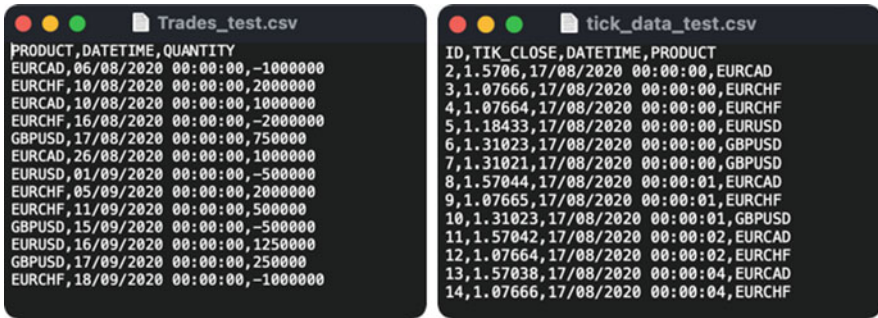


Fig. 4.12 Snippets of the origin datasets (trades and ticks), in CSV format

databases such as PostgreSQL or MySQL using JDBC connectors, etc.) and formats (CSV, JSON Lines, etc.). In Fig. 4.12, a snippet of the datasets used can be seen:

After importing the datasets, the data is then able to be accessed, combined (as seen in Fig. 4.13, where the imported data was preprocessed into a single dataset), and used on the next steps.

4.4.2 Semantic Alignment: Building and Merging

This is the stage where the semantic alignment model obtained by using the modeling methodology is executed. In the example given in Fig. 4.14, the mapping is constructed by using the mapping tool, by associating the items of the imported datasets with the modeled semantics.

On the other hand, the developer of the semantic mappings can, for example, replicate the semantic alignment by using SPARQL queries (Fig. 4.15):

The screenshot shows the OntoRefine interface with a table of 15885 rows. The table has the following columns: id, date_time, product, tik_close, and quantity. The data is organized into groups of 2 rows each, with a 'Show as: rows records' dropdown menu. The first few rows are as follows:

id	date_time	product	tik_close	quantity
1	2020-08-17T00:00:00	EURCAD	1.5706	
2	2020-08-17T00:00:00	EURCHF	1.07686	
3	2020-08-17T00:00:00	EURCHF	1.07684	
4	2020-08-17T00:00:00	EURCHF	1.07683	
5	2020-08-17T00:00:00	GBPUSD	1.35023	750000
6	2020-08-17T00:00:00	GBPUSD	1.35021	750000
7	2020-08-17T00:00:00	EURCAD	1.57044	
8	2020-08-17T00:00:00	EURCHF	1.07685	
9	2020-08-17T00:00:00	GBPUSD	1.35023	
10	2020-08-17T00:00:00	EURCAD	1.57042	
11	2020-08-17T00:00:00	EURCHF	1.07684	
12	2020-08-17T00:00:00	EURCAD	1.57038	
13	2020-08-17T00:00:00	EURCHF	1.07686	
14	2020-08-17T00:00:00	EURCAD	1.57048	
15	2020-08-17T00:00:00	EURCAD	1.57046	
16	2020-08-17T00:00:00	EURCAD	1.57043	
17	2020-08-17T00:00:00	EURCHF	1.07686	
18	2020-08-17T00:00:00	EURUSD	1.08433	
19	2020-08-17T00:00:00	EURUSD	1.07676	
20	2020-08-17T00:00:00	EURCHF	1.07687	
21	2020-08-17T00:00:00	EURUSD	1.08442	
22	2020-08-17T00:00:00	EURCAD	1.57067	
23	2020-08-17T00:00:00	EURCHF	1.07681	
24	2020-08-17T00:00:00	EURUSD	1.08444	
25	2020-08-17T00:00:00	GBPUSD	1.3502300000000002	
26	2020-08-17T00:00:00	EURCAD	1.57071	
27	2020-08-17T00:00:00	EURCHF	1.07683	
28	2020-08-17T00:00:00	EURUSD	1.08446	
29	2020-08-17T00:00:00	GBPUSD	1.35024	
30	2020-08-17T00:00:00	EURCAD	1.57064	
31	2020-08-17T00:00:00	EURCHF	1.07683	
32	2020-08-17T00:00:00	GBPUSD	1.3502	

Fig. 4.13 Imported datasets from the test pilot

The screenshot shows the OntoRefine interface with a semantic mapping configuration. The mapping is defined as follows:

Property	Domain	Range
inf_pilot_2: id	<IRI>	a
rdfs: label	<IRI>	product
fibonacci_acc_cur: hasRateValue	<IRI>	tik_close
fibonacci_qt_qtu: hasNumericValue	<IRI>	quantity
fibonacci_ind_ind: hasQuotedTime	<IRI>	date_time
fibonacci_acc_cur: hasBaseCurrency	<IRI>	baseC
fibonacci_acc_cur: hasDealCurrency	<IRI>	dealRC

Fig. 4.14 Semantic mapping performed with GraphDB mapping tool

4.4.3 Semantic Transformation: Generating a Queryable Knowledge Graphs

Once the mapping is specified, it can be used against the original dataset, to enable the transformation and, thus, to produce the semantic data. This implies that the resulting knowledge graph follows the semantic alignment that is provided by the merging stage. In the specific cases of the adopted frameworks, the user can invoke

SPARQL Query & Update

Editor only Editor and results Results only

```

1 BASE <http://example.com/base/>
2 PREFIX mapper: <http://www.ontotext.com/mapper/>
3 PREFIX fibo-fnd-acc-cur: <https://spec.edmcouncil.org/fibo/ontology/FND/Accounting/CurrencyAmount/>
4 PREFIX fibo-fnd-qt-qtu: <https://spec.edmcouncil.org/fibo/ontology/FND/Quantities/QuantitiesAndUnits/>
5 PREFIX fibo-ind-ind-ind: <https://spec.edmcouncil.org/fibo/ontology/IND/Indicators/Indicators/>
6 PREFIX fibo-ind-fx-fx: <https://spec.edmcouncil.org/fibo/ontology/IND/ForeignExchange/ForeignExchange/>
7 PREFIX inf-spec-ont-pilot-2: <https://spec.infinitech.org/ontology/application/pilot_2/>
8 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
9
10 CONSTRUCT {
11   ?s1 a fibo-ind-fx-fx:QuotedExchangeRate ;
12     inf-spec-ont-pilot-2:CurrencyPairTag ?o_CurrencyPairTag ;
13     fibo-ind-ind-ind:hasQuotationDateTime ?o_hasQuotationDateTime ;
14     fibo-fnd-acc-cur:hasRateValue ?o_hasRateValue ;
15     fibo-fnd-qt-qtu:hasNumericValue ?o_hasNumericValue .
16 } WHERE {
17   SERVICE <rdf-mapper:ontofine:243597023170> {
18     # Columns as variables:
19     # ?c_id, ?c_datetime, ?c_product, ?c_tik_close, ?c_quantity
20     # Metadata as variables:
21     # ?row_index, ?record_id
22     BIND(IRI(mapper:encode_if([inf-spec-ont-pilot-2], ?c_id)) as ?s1)
23     BIND(STROT(?c_product, xsd:string) as ?o_CurrencyPairTag)
24     BIND(STROT(?c_datetime, xsd:dateTime) as ?o_hasQuotationDateTime)
25     BIND(STROT(?c_tik_close, xsd:long) as ?o_hasRateValue)
26     BIND(STROT(?c_quantity, xsd:int) as ?o_hasNumericValue)
27   }
28 }
29

```

Fig. 4.15 Semantic mapping with SPARQL query

```

1 @base <http://example.com/base/> .
2 @prefix fibo-fnd-acc-cur: <https://spec.edmcouncil.org/fibo/ontology/FND/Accounting/CurrencyAmount/> .
3 @prefix fibo-fnd-qt-qtu: <https://spec.edmcouncil.org/fibo/ontology/FND/Quantities/QuantitiesAndUnits/> .
4 @prefix fibo-ind-ind-ind: <https://spec.edmcouncil.org/fibo/ontology/IND/Indicators/Indicators/> .
5 @prefix fibo-ind-fx-fx: <https://spec.edmcouncil.org/fibo/ontology/IND/ForeignExchange/ForeignExchange/> .
6 @prefix inf-spec-ont-pilot-2: <https://spec.infinitech.org/ontology/application/pilot_2/> .
7 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
8
9 inf-spec-ont-pilot-2:2 a fibo-ind-fx-fx:QuotedExchangeRate;
10 inf-spec-ont-pilot-2:CurrencyPairTag "EURCAD";
11 fibo-ind-ind-ind:hasQuotationDateTime "2020-08-17T00:00:00"^^xsd:dateTime;
12 fibo-fnd-acc-cur:hasRateValue "1.5706"^^xsd:long .
13
14 inf-spec-ont-pilot-2:3 a fibo-ind-fx-fx:QuotedExchangeRate;
15 inf-spec-ont-pilot-2:CurrencyPairTag "EURCHF";
16 fibo-ind-ind-ind:hasQuotationDateTime "2020-08-17T00:00:00"^^xsd:dateTime;
17 fibo-fnd-acc-cur:hasRateValue "1.07666"^^xsd:long .
18
19 inf-spec-ont-pilot-2:4 a fibo-ind-fx-fx:QuotedExchangeRate;
20 inf-spec-ont-pilot-2:CurrencyPairTag "EURCHF";
21 fibo-ind-ind-ind:hasQuotationDateTime "2020-08-17T00:00:00"^^xsd:dateTime;
22 fibo-fnd-acc-cur:hasRateValue "1.07664"^^xsd:long .
23
24 inf-spec-ont-pilot-2:5 a fibo-ind-fx-fx:QuotedExchangeRate;
25 inf-spec-ont-pilot-2:CurrencyPairTag "EURUSD";

```

Fig. 4.16 RDF results resulting from the semantic alignment applied to the origin dataset

them to, for example, execute the relevant SPARQL queries, by using the REST API endpoints or any other solution that may be developed for the effect. In the current implementation, either running SPARQL queries or streaming data through HTTP request can fulfill the data transformation, which results on the generation of an RDF format dataset, such as the code snippet shown in Fig. 4.16.

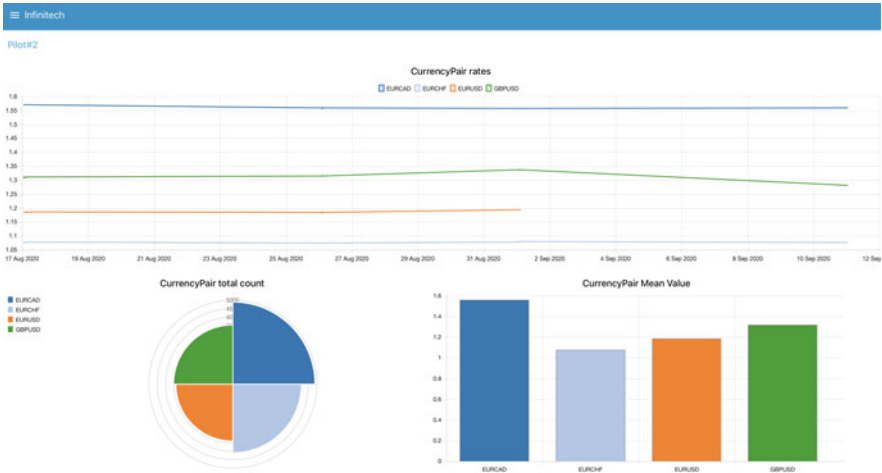


Fig. 4.17 Analytics over RDF data generated from the origin datasets

4.4.4 Data-Sharing/Provisioning

Finally, it is important that end users possess the necessary means to access and further use the generated data. As such, the set of solutions in use allow to directly download the RDF results from transformation process. Moreover, they enable the storage of the results as knowledge graphs which can later be retrieved using the REST APIs or other access points created.

Moreover, RDF compliant client applications can be developed to make use of such capability using the data to execute the desired analytics over the data.

In the scope of this validation process, a simple node.js client application was developed for demonstration, where a few analytic algorithms were applied to the RDF data. The latter data were consumed through one of the REST APIs, as presented in Fig. 4.17.

5 Conclusions

This chapter presented a methodology for generating models that enable the transformation of data to semantically annotated data, in alignment with reference ontologies of financial and insurance sectors.

The outcomes (*modelelets*) of the proposed methodology provide a simpler and more trustable way to generate and transform the original data into semantically meaningful data. In this way, the proposed methodology enhances the interoperability of different sources.

To validate the approach, state-of-the-art technological tools for the processing and management of tabular and RDF data were used. Leveraging these tools, it is demonstrated that the mapping process can be extremely simplified and, also, that it is applicable to different types of available solutions. As a matter of fact, the same results were obtained when applying the methodology into different sources of data or when using other types of mapping technologies.

Our future work involves the application and validation of the methodology at a larger scale, starting from its deployment and use in the digital finance systems that have been developed, deployed, and validated in the scope of the INFINITECH project.

Acknowledgments The research leading to the results presented in this chapter has received funding from the European Union’s funded Project INFINITECH under Grant Agreement No. 856632.

References

1. Cupoli, P., Earley, S., & Henderson, D. (2012). *DAMA-DMBOK2 framework* (p. 27).
2. Strengtholt, P. (2020). *Data management at scale: Best practices for enterprise architecture*. O’Reilly Media Inc.
3. C. Iso and others. (1993). *ISO/IEC 2382-1: 1993 information technology-vocabulary-part 1: Fundamental terms*.
4. ETSI. (2011). *ETSI TS 186 020 V3.1.1 – Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IMS-based IPTV interoperability test specification*. Technical.
5. EICTA. (2004). *EICTA INTEROPERABILITY WHITE PAPER*. [Online]. Available: <http://www.agoria.be/www.wsc/webextra/prg/nwAttach?vWebSessionID=8284&vUserID=999999&appl=Agoriav2&newsdetid=73029&attach=DescFile10.720108001095774198.pdf>
6. Maló, P. M. N. (2013). *Hub-and-spoke interoperability: An out of the skies approach for large-scale data interoperability*. [Online]. Available: <http://run.unl.pt/handle/10362/11397>. Accessed 18 Jan 2016.
7. Martidis, A., Tomasic, I., & Funk, P. (2014). Deliverable 5.3: CREATE interoperability. August 25, 2014.
8. van der Veer, H., & Wiles, A. (2008). *Achieving technical interoperability*. European Telecommunications Standards Institute.
9. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 34–43.
10. Palavalli, A., Karri, D., & Pasupuleti, S. (2016). Semantic internet of things. In *2016 IEEE tenth international Conference on Semantic Computing (ICSC)* (pp. 91–95).
11. Elkhodr, M., Shahrestani, S., & Cheung, H. (2013). The Internet of Things: Vision & challenges. In *IEEE 2013 Tencon-Spring* (pp. 218–222).
12. McIlraith, S. A., Son, T. C., & Zeng, H. (2001). Semantic web services. *IEEE Intelligent Systems*, 16(2), 46–53.
13. Slimani, T. (2013). Semantic annotation: The mainstay of semantic web. *arXiv preprint arXiv:1312.4794*.
14. Andrews, P., Zaihrayeu, I., & Pane, J. (2012). A classification of semantic annotation systems. *Semantic Web*, 3(3), 223–248.

15. Liao, Y., Lezoche, M., Panetto H., & Boudjlida, N. (2011). Semantic annotation model definition for systems interoperability. In *OTM confederated international conferences "On the Move to Meaningful Internet Systems"* (pp. 61–70).
16. Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). *Methontology: from ontological art towards ontological engineering*. Stanford University.
17. Peroni, S. (2016). A simplified agile methodology for ontology development. In *OWL: Experiences and directions—reasoner evaluation* (pp. 55–69). Springer.
18. Pinto, H. S., Staab, S., & Tempich, C. (2004). DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In *Proceedings of the 16th European conference on artificial intelligence* (pp. 393–397).
19. De Nicola, A., & Missikoff, M. (2016). A lightweight methodology for rapid ontology engineering. *Communications of the ACM*, 59(3), 79–86.
20. Allen, M., & Cervo, D. (2015). *Multi-domain master data management: Advanced MDM and data governance in practice*. Morgan Kaufmann.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

