OPINION

Foundational Ontologies, Ontology-Driven Conceptual Modeling and their Multiple Benefits to Data Mining

Glenda Amaral[1] | **Fernanda Baião**[2] | **Giancarlo Guizzardi**[1]

[1]Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, BZ, 39100, Italy

[2]Department of Industrial Engineering, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, RJ, 22451-900, Brazil

Correspondence
Glenda Amaral, Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, BZ, 39100, Italy
Email: gmouraamaral@unibz.it

Funding information
CAPES (PhD grant# 88881.173022/2018-01) and NeXON project (UNIBZ).

For many years, the role played by domain knowledge in all stages of knowledge discovery has been recognized by authors in the field. However, the real-world semantics embedded in data is often still not fully considered in traditional data mining methods and techniques. In this paper, we defend that the quality of data mining results is directly related to the extent that they reflect important properties of real-world entities represented therein. Analysing and characterising the nature of these entities is the very business of the area of Formal Ontology. We briefly elaborate on two particular types of artefacts produced by this area: *Foundational Ontologies* and *Ontology-Driven Conceptual Modeling languages* grounded on them. We then elaborate on the benefits they can bring to several activities in a Data Mining process.

KEYWORDS
Data Mining, Foundational Ontologies, UFO, OntoUML

1 | INTRODUCTION

Data has always been an essential resource supporting decision-making. In recent years, the exponential growth in the volume of data available has challenged organizations to find efficient ways to extract relevant information from their data assets. In this scenario, data mining processes emerge as an essential approach to extracting strategic

knowledge in large collections of data. Fayyad et al. (1996) first coined the term *Knowledge Discovery in Databases (KDD)* as the "nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". Han et al. (2011) further stressed the characteristics of KDD cycle in dealing with massive amounts of data spread in different kinds of data sources. These data sources may span databases, data warehouses, flat files, the Web and other data repositories, data streams dynamically fed from social media, sensors or other devices.

This multidisciplinary field of study emerged during the late 1980s, culminating with the recent - and still increasingly active - *boom* of Data Science lifecycles and methodologies (Priebe and Markus, 2015; Shcherbakov et al., 2014). Despite the evolving terminology that has been applied to this field and its related areas, the knowledge discovery process is essentially an iterative cycle comprising *Problem Understanding*, *Data Pre-processing*, *Data mining*, and *Data Post-processing*. Problem Understanding specifies the problem to be addressed, and the objectives that are pursued by the organization that motivate the knowledge discovery process. Data Pre-processing prepares data for mining, and comprises the activities of: (i) *Data cleaning* - to eliminate noise and inconsistencies in data; (ii) *Data integration* - to combine multiple data sources; (iii) *Data selection* - to filter out data that is considered irrelevant to the problem at hand; and (iv) *Data transformation* - to transform and consolidate data into forms appropriate for mining. Data Mining, in turn, is the fundamental step where computational techniques are applied to extract patterns from pre-processed data. In industry, in media, and in some research fields “the term data mining is often used to refer to the entire knowledge discovery process” (Han et al., 2011), or as a synonym for KDD, given its essential role in the entire enterprise. Finally, Data Post-processing includes *Pattern evaluation* (to select from the discovered patterns the ones that truly represent interesting knowledge, based on the quality metrics applicable for each mining technique) and *Knowledge presentation and interpretation* (“when visualization and knowledge representation techniques are applied to present mined knowledge to users” (Han et al., 2011)).

For many years, the fundamental role of domain knowledge for all stages of knowledge discovery has been recognized by authors in the field (Fayyad et al., 1996; Han et al., 2011). Nonetheless, traditional data mining methods and techniques treat data as merely “sums of attribute values" so that statistics can be calculated on them to then foster the construction of patterns and models. Generally speaking, the *real-world semantics* embedded in the data is not explicitly considered in these approaches. However, as nicely put by Mealy (1967)[1], *“Data are fragments of a theory of the real-world"*. As a consequence, approaches that are oblivious to the real-world semantics of data are limited to the correlations and data relationships that they can discover. In other words, the quality of data mining results is directly related to the extent that they reflect important properties of real-world entities represented therein. In line with this view, a recent trend in data mining research proposes leveraging domain knowledge, especially represented as *ontologies* (Guarino, 1994, 1998; Guizzardi, 2007), to improve the KDD process. This paper is harmonious with this trend. However, differently from the approaches that propose the use of the so-called domain ontologies to support the KDD process, here we focus on discussing how each step of the KDD process can benefit from data grounded on *Foundational Ontologies*. We discuss the role of foundational ontologies not as a substitute for the use of domain ontologies, but as a complementary approach.

Foundational ontologies are philosophically well-founded axiomatic systems of domain-independent categories and their ties (e.g. objects, events, causality, parthood, spatial-temporal connections, dependencies, etc.) that can be used to articulate the representation of phenomena in different material domains (Guizzardi, 2007; Gangemi et al., 2002). Examples of foundational ontologies include DOLCE (Borgo and Masolo, 2009), GFO (Herre, 2010), SUMO (Niles and Pease, 2001) and UFO (Guizzardi, 2005; Guizzardi, G. et al., 2015). As such, they serve as a conceptual basis for capturing the essence of particular domains in reality, and thus are potentially valuable for identifying correlations and other interesting relationships among data reflecting their ontological counterparts.

[1] This paper by Mealy contains the first mention of the term ‘Ontology’ in Computer Science.

The fundamental ontological distinctions embodied in a foundational ontology can be used to improve the quality of the data mining process, mainly when it includes information from multiple sources that may commit to different theories about a particular concept. Let us take the example of two different systems A and B that record information about organ transplants. In this case, A and B may commit to different theories (ontologies) of transplants. We cannot assume that just because the same term (e.g., Transplant) is used in both structures that they mean the same thing. For instance, the relation between Transplant-A and Transplant-B is not one of identity if, for example, the instances of Transplant-A are individual transplants that occur in particular time and space and the instances of Transplant-B are types of transplant. Moreover, even if one takes transplants in senses A and B to both refer to individuals in space and time, they can still refer to *events* in one case (e.g., A), while referring to reified relationships (technically, *relators* - see discussion on the next section), in the other (B) (Guarino and Guizzardi, 2016; Guizzardi et al., 2016). In the former case, the relation between Transplant-A and Transplant-B is one of instantiation, i.e., the instances of Transplant-A are instances of instances of Transplant in the sense of system B (Transplant-B); in the latter case, it is one of manifestation, i.e., instances of transplant in one case are manifestations of properties (e.g., right, obligations, powers) of transplant as a bundle of relational aspects. Finally, even if transplants are events in both senses A and B, they might refer to different formal notions of events. For example, according to sense A, an event could be an entity that only exists in the past, thus having all its properties immutable (Guizzardi et al., 2016) whilst, according to sense B, it could be something that unfolds in time by possibly having mutable temporal parts (Guarino, 2017). As previously mentioned, the ontological distinctions put forth by foundational ontologies can be powerful tools for addressing these challenges.

As we report in section 2 below, we are not the first to investigate the connection between the topics of foundational ontologies and data mining. However, there are two perspectives in which the analysis put forth here is novel. Firstly, as discussed below, most of the existing approaches that leverage on research of foundational ontologies in this area focus on these ontologies *as artifacts*, e.g., using these models as “semantic bridges" or “pivots" for connecting other models (as in the area of ontology matching for schema integration). Here, in contrast, our focus is on the role of foundational ontologies as providing a conceptual toolbox for supporting ontological analysis, meaning explication and negotiation, and conceptual clarification. In fact, most existing approaches that use foundational ontologies in data mining employ lightweight versions of these ontologies (Trojahn, C., Vieira, R., Schmidt, D., Pease, A., Guizzardi, G., 2021), typically coded in inexpressive formal languages such as OWL (Web Ontology Language). These models are drastic simplifications of their corresponding ontologies’ original axiomatizations, and as such, they are much poorer tools supporting the type of analysis advocated here. Moreover, as formal theories, they are not easy to be directly adopted in data mining activities such as domain modeling. This is because they lack proper associated representation languages (i.e., domain models based on these foundational ontologies should be done by either specializing or instantiating these formal theories) and tools. So, a second perspective in which this article diverges from the existing approaches in the literature is that it reports on advances in an area termed *Ontology-Driven Conceptual Modeling*. This area aims at employing foundational ontologies to derive engineering tools for conceptual modeling/data modeling. These include modeling languages, libraries of patterns and anti-patterns, and computational tools for model creation, verification, validation, verbalization, codification and database generation, as well as automated model diagnosis and repair via learning (Guizzardi, G. et al., 2015; Fumagalli et al., 2020). In particular, we focus here on the only foundational ontology we are aware of that provides such a complete ecosystem of technologies and in a way that is familiar to data modelers, namely, the Unified Foundational Ontology (UFO) (Guizzardi, G. et al., 2015), and the UFO-Based Conceptual Modeling language OntoUML (Guizzardi, 2005).

The remainder of this paper is organized as follows. In Section 2, we briefly review existing literature in data mining that refers to Foundational Ontologies. In Section 3, we briefly introduce the reader to the *Unified Foundational*

Ontology (UFO) and OntoUML. In Section 4, we illustrate how some fundamental KDD activities can be improved with the support of ontologically well-founded artifacts such as UFO and OntoUML. Finally, section 5 presents some final considerations.

2 | FOUNDATIONAL ONTOLOGIES AND DATA MINING: AN UNDER-EXPLORED CONNECTION

There are few works in the literature that jointly address the topics of "foundational ontology" and "data mining". However, these proposals do not address the general implications of considering a foundational ontology in the results of applying a data mining technique in specific material domains of interest.

The work of DMOP (Data Mining OPtimization Ontology) (Keet et al., 2015) proposes an ontology for the data mining process domain. The proposed ontology is aligned to the DOLCE foundational ontology; however, the authors do not address the semantic precision of material domain concepts, or how their ontological distinctions impact on the results of the data mining process and its activities.

The works of Khan and Keet (2014) and of Padilha et al. (2012) address the alignment of foundational ontologies, or of domain and foundational ontologies, and the data mining domain happens to be one of the studied domains for applying their proposal. However, they also do not elaborate on the impact of making explicit the ontological foundations of the aligned concepts on the patterns extracted by the mining techniques applied. In (Mascardi et al., 2009) the authors systematically evaluate the use of three foundational ontologies (or upper ontologies, as they name it) as bridges for improving the results of the matching process. They acknowledge that two of the ontologies they evaluated (OpenCyc and SUMO-OWL) were, in fact, large-coverage general-purpose ontologies that include many domain-specific concepts, and the third ontology (DOLCE) was the only “pure upper-ontology” they used. They concluded that the gain from an upper-ontology was limited to the cases where there were domain-specific concepts and that the use of a pure upper-ontology deteriorates the process. This could be explained by the fact that they did not take into account the meta-properties of true ontological nature that characterize the distinctions put forth by DOLCE. Instead, as previously mentioned, they constrained the upper ontology as “semantic bridges” (or as a pivot model) in the matching process. This is also what is observed in the comprehensive survey reported in (Trojahn, C., Vieira, R., Schmidt, D., Pease, A., Guizzardi, G., 2021) on the use of foundational ontologies for schema matching; that is, as reported there, foundational ontologies are used in that area: mainly as an artifact; represented by their weakly-axiomatized lightweight versions; and, hence, not mainly as a tool for ontological analysis, conceptual clarification and meaning negotiation; without the support of proper data modeling tools.

The work of Bleisch et al. (2014) uses a "foundational ontology of causation" to structure the concepts of state and event, as well as a *causality* relation between events, and an *allowance* relation between a state and an event. The authors use the referred ontology to define well-founded patterns and discover causality relations and movement patterns by applying two data mining techniques: association rule mining (ARM) and sequence mining, which extends ARM to incorporate temporal sequences of transactions. The authors justify the use of a foundational ontology to amplify the generality of their approach but do not address the issue of semantic precision in interpreting domain concepts. In any case, their approach may be considered an example of applying some of the ideas we propose here to a very specific context.

Ristoski and Paulheim (2016) present a survey on the usage of Linked Open Data to provide additional knowledge to enhance the value of data mining. The authors discuss how semantic data can be used at the different stages of the knowledge discovery process and analyze how different characteristics of Linked Open Data are exploited by

different approaches, including the role played by ontologies. The vast majority of the approaches analyzed in that work use a custom ontology or reuses an existing domain ontology, however the authors neither mention if these ontologies are built with the support of a foundational ontology nor discuss the benefits this could bring to the KDD process. In a previous work (Ristoski and Paulheim, 2014), the authors introduce a method that exploits hierarchies to reduce the set of features in the pre-processing step of data mining, which has an impact on both the runtime and the result quality of the subsequent processing steps. Similarly, although the semantics existing in hierarchies is explored to improve the pre-processing step, as the hierarchies considered are not based on any upper-level ontology, they do not benefit from the application of the full axiomatization of foundational ontologies as a analysis tool during the mining process.

Similarly, other works in the literature exploit the use of taxonomic structures to improve data mining techniques, also without benefiting from the full axiomatization of foundational ontologies in the process. This is the case of the work from Baralis et al. (2012), who proposed a framework to discover interesting generalized association rules driven by a multiple taxonomy that allows the opportunistic extraction of knowledge at different aggregation levels. Also, Silla and Freitas (2011) surveyed methods addressing the problem of hierarchical classification (which the authors defined as being any classification problem with a class structure of an IS-A hierarchy that is asymmetric, anti-reflexive and transitive, such as taxonomies). The authors even acknowledge the existence of a few cases in which the semantics of the underlying class hierarchy might differ (such as with the Gene Ontology which, from a semantic point of view, is essentially a PART-OF class hierarchy), but as long as the aforementioned properties are satisfied, they are considered as hierarchical classification problems. Therefore, these works do not commit to searching for patterns in data by taking real-world semantics into account.

3 | THE UNIFIED FOUNDATIONAL ONTOLOGY (UFO) AND THE ONTOUML LANGUAGE

The Unified Foundational Ontology (UFO) has been "developed by consistently putting together a number of theories originating from areas such as Formal Ontology in philosophy, cognitive science, linguistics and philosophical logics" to provide formal ontological foundations for conceptual modeling (Guizzardi, G. et al., 2015). DOLCE, GFO and UFO are all centered around the same notions of what is termed the *Aristotelian Square* (Guizzardi, 2005). So, in that respect, they are rather similar. In fact, UFO was created as an extension of the unification of DOLCE and GFO to deal with a number of specific phenomena that arise in Conceptual Modeling. For example, unlike DOLCE, UFO includes a rich *ontology of relations* (Guarino and Guizzardi, 2016). Moreover, unlike DOLCE, which is an ontology of particulars, UFO includes a number of formal distinctions among types of universals (e.g., kinds, phases, roles, mixins - see next section). Both these features have been shown to be fundamental for conceptual modeling Guizzardi (2005). Additionally, unlike GFO, UFO has a theory of relations that is finitely instantiable (Guizzardi, G. et al., 2015), which makes it practical to conceptual modeling applications. Furthermore, unlike DOLCE (but also SUMO, GFO, BFO), UFO is formally connected to a conceptual modeling language (OntoUML). OntoUML was designed such that its modeling primitives reflect the ontological distinctions of its underlying ontology, and its grammar is enriched with semantically-motivated syntactical constrains that mirror UFO's axiomatization. Finally, research shows that UFO is vastly more used in Conceptual Modeling (an area closely connected to the theme of this article) than the aforementioned ontologies (Verdonck and Gailly, 2016). For this reason, in the remainder of this section, we focus our discussion on UFO, briefly explaining a subset of its ontological distinctions that are relevant for the process of knowledge discovery from data, as discussed in Section 4. For a fuller discussion on this ontology, one should refer

to (Guizzardi, 2005; Guizzardi, G. et al., 2015). In any case, we highlight that our focus on UFO here is to make our discussion more concrete with OntoUML domain modeling examples. However, modulo the previously discussed limitations, many of the benefits discussed here can also be achieved with use of other foundational ontologies.

UFO makes a fundamental distinction between *individuals* (particulars), and *types (or universals)*, i.e., patterns of features that are repeatable across individuals. Individuals can be *endurants* (roughly, things or object-like entities) and *perdurants* (roughly, events, occurrences, processes). Within the category of endurants, UFO distinguishes *substantials* and *aspects (also termed moments)*. Substantials are existentially independent objects, such as the Moon, an enterprise, a person, a horse. As for aspects, they are existentially dependent entities, such as: (a) John's capacity to play tennis (which existentially depends on him); (b) a flower's color (which depends on that flower); (c) the marriage between Bob and Alice (which depends on both Bob and Alice). Aspects of types (a) and (b) are termed *intrinsic aspects*. In particular, those of type of (a) are termed *modes*, and those of type (b) *qualities*; finally, those of type (c) are termed *relators*.

An important characteristic of all endurants is that they exist in time keeping their identity, even if changing in a qualitative way (e.g., a flower's color which may change from red to brown while keeping its identity). Modes and qualities can be seen as objectifications of intrinsic properties of endurants. Likewise, relators can be seen as objectifications of their relational properties. Relators are individuals with the power of connecting entities. For example, an Enrollment relator connects an individual playing the Student role with an Educational Institution. As discussed in depth in (Guarino and Guizzardi, 2016), relators represent the material content of domain relations and, hence, they are reponsible for relationships holding between domain entities.

Qualities are intrinsic moments that have the power to connect the entity they qualify with values into certain *value spaces*. Examples of qualities include mass, age, electrical charge and color. UFO relies on the theory of Conceptual Spaces (Gärdenfors, 2004) to assume that for several perceivable or conceivable *quality types* there is an associated *quality dimension* in human cognition. Let us take the example of 'height' and 'mass', which are associated with one-dimensional structures with a zero point isomorphic to the half-line of non-negative numbers. Similarly, 'date' can be associated to a structure (a *quality domain*) formed by three dimensions, named Day, Month and Year. Moreover, these structures can provide ordering for these values, allowing the comparison of qualities associated with the same or equivalent structures.

Following Gärdenfors (2004), UFO distinguishes between integral and separable quality dimensions: "certain quality dimensions are integral in the sense that one cannot assign an object a value on one dimension without giving it a value on the other. For example, an object cannot be given a hue without giving it a brightness value. Dimensions that are not integral are said to be separable, as for example the size and hue dimensions" (Gärdenfors, 2004). Quality domains can then be defined as "a set of integral dimensions separable from all the others" (Gärdenfors, 2004): These form geometrical structures. UFO introduces the category *quality structure* as a general category to the categories quality dimension and quality domain. Finally, the projection of a quality into a quality structure is referred in the literature as a *quale*, or a *quality value* (e.g. '10kg', '40°C', 'blue') (Albuquerque and Guizzardi, 2013). In this paper, the notions of qualities and conceptual value spaces (quality structures) are important because they provide powerful means to calculating similarity among entities, as discussed in the next section.

As previously discussed, endurants can change *in certain ways* while preserving their identity. The sorts of changes an endurant can undergo and still be the same is determined by its *kind*. By a kind, we mean an entity type that necessarily classify their instances, being responsible for their principle of identity. For this reason, all endurants classified by a kind cannot cease to instantiate it without ceasing to exist (e.g., a horse cannot cease to be a horse and keep existing). Kinds can be specialized in other subtypes that also necessarily classify their instances, named *subkinds*. For example, if we take 'Person' to be a kind then some of its subkinds could be 'Man' and 'Woman'.

Endurant kinds and subkinds are also termed *rigid types* as they represent essential properties of objects. There

are, however, types that represent contingent or accidental properties of objects, named *anti-rigid types*. Examples of anti-rigid types are *phases* and *roles*. The former represent properties that are intrinsic to entities, while the latter represent properties that entities have in a relational context, i.e., contingent relational properties.

Kinds, subkinds, phases, and roles are categories of *sortals*. In the philosophical literature, a sortal is a type that provides a uniform principle of identity, persistence, and individuation for its instances (Guizzardi, 2005). In contrast with sortals, *non-sortals* are types that represent properties shared by entities of multiple kinds. A particular type of non-sortal of UFO of interest here is a *roleMixin*. RoleMixins are role-like (i.e., anti-rigid and relationally dependent) types but which can be played by entities of multiple kinds. An example is the roleMixin 'Service Provider', which can be played by both people and organizations.

Over the years, UFO has been applied to analyze and (re)design a multitude of modeling languages and standards. One of these applications, however, stands out, namely the conceptual modeling language OntoUML (Guizzardi, 2005; Guizzardi, G. et al., 2015). OntoUML is a version of UML class diagrams that has been designed such that its modeling primitives reflect the ontological distinctions put forth by UFO (including the ones just discussed), and its grammatical constraints follow UFO axiomatization. In fact, OntoUML is formally a pattern-language whose modeling primitives are *ontological design patterns*, representing UFO's constituting (micro)theories (Ruy et al., 2017).

4 | IMPROVING DATA MINING WITH FOUNDATIONAL ONTOLOGIES

4.1 | Problem Understanding

Problem understanding and background knowledge are paramount for the success of a data mining process, as no algorithm is always better than all the others for all criteria in all situations (Magdon-Ismail, 2000). "For matching suitable algorithms for the mining goal, one needs to know not only the character of each algorithm" (Lin et al., 2006), but also whether an algorithm is appropriate for the problem and data being considered. Moreover, "background knowledge can be incorporated with the induction algorithm and used for evaluating the mined results" (Lin et al., 2006).

One of the main purposes of foundational ontologies is to facilitate *semantic transparency*, i.e., to make explicit the real-world semantics of data (Guizzardi, 2020). In general, the main purpose of a conceptual model (including a domain ontology) is to make explicit the *ontological commitment* made by a given representation artifact (Guarino et al., 2019). In an ontology-based conceptual language such as OntoUML, we go one step further as the modeling primitives of the language make a direct connection to the categories of the underlying foundational ontology (Guizzardi, 2005).

Another aspect related to which ontology-driven modeling languages can contribute to domain understanding is through their mechanism to support complexity management (Guizzardi et al., 2019; Figueiredo et al., 2018). Complex domains require representations that are both large in scale, and rich in subtleties. Languages such as OntoUML are endowed with mechanisms such as modularization, viewpoint extraction, model abstraction and summarization, as well as the breaking down of models in cognitively tractable chunks. In fact, in OntoUML, the modeling elements never occur freely but appear in certain modeling configurations and combined with other modeling elements, thus forming certain modeling patterns. In other words, the modeling primitives of the language are actually patterns, i.e., "higher-granularity clusters of modeling elements that can appear in a model only in particular fixed configurations" (Guizzardi, G. et al., 2015). As previously discussed, "these patterns are of an ontological nature, as they directly reflect the ontological micro-theories underlying the language" (Guizzardi, G. et al., 2015). Besides working as a complexity management mechanism, this characteristic of the language brings more uniformity to its models (which become described in terms of known patterns), thus contributing to improve model comprehensibility and, consequently, to

data understanding (Verdonck et al., 2019).

4.2 | Data Pre- and Post-processing

Data pre-processing refers to the steps applied to make data more suitable for data mining. The major tasks involved in data pre-processing are “data cleaning, data integration, data reduction, and data transformation" (Fayyad et al., 1996). Among them, we argue that data integration is the one that can most benefits from the use of foundational ontologies.

Data Integration focuses on combining data from multiple sources and elucidating data value conflicts. This includes the integration not only of data models but also of data schemas, considering the several levels of heterogeneity, in particular semantic heterogeneity (Ziegler and Dittrich, 2007). Semantic data integration aims to establish correct semantic relationships between data elements represented in different datasets. This, of course, requires understanding the relation in reality between their referents, i.e., the real-world entities they represent (Guizzardi, 2020).

The use of ontologies for data integration purposes has been widely acknowledged to help semantic interoperability between distributed data sources, because the semantics of data included in each data source can be made explicit with respect to an ontology a particular user group commits to. However, without making explicit the foundational ontological category a given domain concept maps to, there is a significant risk of semantic misinterpretations (or false-agreements). Therefore, the use of foundational ontologies have been successfully considered to further increase the precision of the concepts definition, thus reducing ambiguous interpretations. The use of ontology-driven languages (such as OntoUML) and of ontology design patterns (ODP) make explicit the basic ontological categories of the domain notions represented therein and, hence, serve to “identify and/or discard potential alignments between data from different repositories which presumably refer to the same real-world entity" (Padilha et al., 2012). When two data repositories are semantically described through foundational ontologies, it is possible to check whether the equivalence between two classes is correct by, for example, observing whether the basic ontological categories they belong to are compatible.

Take the example of a typical scenario described in (Padilha et al., 2012) and modeled in Figure 1 (dotted lines are potential concept alignments that are further investigated). In this example, a company has two subsidiaries with independent operations; in one of them customers may be private or corporate, while the other subsidiary has only private customers. As explained by Padilha et al. (2012), “although there is a Customer concept in each ontology describing a source repository being integrated, the «roleMixin» stereotype applied to the Customer concept in the left ontology makes it explicit that it is not semantically equivalent (and therefore should not be aligned) to the class Customer in the right ontology, which is stereotyped as a «role»[2]”. Thus, in a data mining application in which data from all subsidiaries should be integrated, taking foundational semantics into account prevents the mistaken alignment of these two classes both of which happen to be termed ‘Customer’ (an alignment that would probably be asserted otherwise). The data integration task would proceed in the same way as the ontology alignment process described in (Padilha et al., 2012), that is, “by searching for the «kind» concept that is the most specific superclass of the «role» concept in the left ontology (that is, Person), identifying the «role» subclass PrivateCustomer and consider it as being equivalent to the «role» Customer in the right ontology (despite their distinct names)”.

Therefore, data integration approaches could benefit from well-founded ontologies in UFO by considering the stereotype of the OntoUML classes within design patterns to prevent the identification of incorrect semantic associations between classes. Once more, these spurious association would probably happen otherwise when using methods such as manual analysis, automatic analysis supported by lexical-based techniques, or even automated analysis and

[2] For simplicity, the relational dependence of the roleMixin and role classes are not made explicit in this figure.

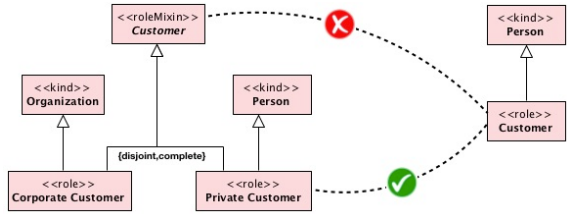

FIGURE 1 Preventing semantic integration problems using well-founded ODPs

mapping techniques based on lightweight domain ontologies.

Data post-processing refers to the steps applied to assess the knowledge (in the form of patterns or models) discovered by the data mining techniques, possibly returning to any of the previous phases of the KDD cycle. The major tasks involved in data post-processing are pattern evaluation, presentation and interpretation. Pattern evaluation will be addressed by means of the metrics presented and discussed for each of the covered data mining techniques in the following Sections. With regard to pattern interpretation, the existence of a well-founded domain ontology may be used to systematically guide the domain expert in understanding and validating the patterns, as well as in mapping them to domain concepts taking into account their ontological meta-properties (Guizzardi, 2005).

4.3 | Classification

Classification is "the process of finding a function (a model, also named a classifier) that describes and distinguishes data classes" (Han et al., 2011), where a class is a categorical label from a discrete and unordered domain, in that each value denotes a category. More specifically, it is a data analysis task to extract a set of deductive rules which are presumed to describe a predefined concept from the domain of discourse, with respect to recurring situations historically observed that are provided as a set of training instances. Once a classifier is learned, it is used to automatically assign a class label to a previously unseen instance, based on the values of its other (predictive) features. Classification is also known as "supervised learning, because the class label of each input training instance is provided" (Han et al., 2011). Due to its broad applicability, relative simplicity and efficacy of several techniques in producing comprehensive models (such as decision trees and deductive rules), Classification is one of the most relevant types of data mining approaches in the knowledge discovery scenario.

The classification problem can be defined as follows. Let X = $X_1, \ldots, X_d$ be a set of d predictive features and L = $l_1,\ldots,l_q$ be a set of q class labels, where q $\geq$ 2. Let D = $(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)$ be a dataset with N instances, where, for the i-th instance, x_i corresponds to a vector ($x_i1, x_i2, \ldots, x_id$), which stores values for the d features in X and each $y_i \in$ L corresponds to a single target class. The goal of the classification task is to learn a classifier from D that, given an unlabelled instance t = (x, ?), predicts its class label y.

Typically, the evaluation of a classifier refers to a subset of classical, well-defined metrics from the literature, such as accuracy (or recognition rate), sensitivity (also named recall or true positive rate), specificity (also known as true negative rate), precision and recall (Han et al., 2011). Most of these metrics are formally defined based on measures that take each class label as a reference, and presumably reflect how good the classifier is at describing this class label. Thus, to assess a classifier with respect to a class label l_i, for each instance we compare the classifier's class label prediction with the instances's known class label. Given that an instance of class l_i is considered a positive instance, while all other instances are considered negative, the following measures are defined: (i) P is the total number of positive instances; (ii) N is the number of negative tuples; (iii) TP (*true positives*) is the number of positive instances

that were correctly labeled by the classifier; (iv) TN (*true negatives*) is the number of negative instances that were correctly labeled by the classifier; (v) FP (*false positives*) is the number of negative tuples that were incorrectly labeled as positive; and (vi) FN (*false negatives*) is the number of positive instances that were mislabeled as negative. For example, when classifying whether a set of Patients have cancer or not, TP is the number of ill patients that were predicted as so by the classifier, while FN is the total of patients that do have cancer but were predicted as healthy by the classifier. Finally, the performance metrics are calculated. Accuracy of a classifier on a given test set is the percentage of test set instances that are correctly classified by the classifier. Thus, *accuracy = (TP + TN) / (P + N)*. Sensitivity is the proportion of positive instances that are correctly identified), given by *sensitivity = TP / P*. Specificity is the proportion of negative tuples that are correctly identified, calculated as *specificity = TN / N*. Precision can be thought of as a measure of *exactness* (i.e., what percentage of tuples labeled as positive are actually such) given by *precision = TP / (TP + FP)*, whereas recall is a measure of *completeness* (what percentage of positive tuples are labeled as such), computed as *recall = TP / (TP + FN) = TP / P*.

We argue that the benefits of making explicit the ontological nature of the domain concepts cover several perspectives of a classification experiment, ranging from the definition of the class to be learned, the selection of relevant features that characterize the training instances, and the potential for results improvement.

With regard to improving quality results, we argue that a classifier, as a model, should also isomorphically correlate to its represented domain of discourse, that is, through a *sound*, *complete*, *lucid* and *laconic* correlation (Guizzardi, 2007). As defined by Kirk and MacDonell (2015), "soundness requires that each modelling construct maps to a domain concept; completeness requires that each domain concept is represented by a modelling construct; lucidity requires that each modelling concept represents at most one domain concept (i.e., there is no construct overload); and laconicity requires that each domain construct is represented by at least one modelling construct". By having the guidance of a proper ontology model, one can appropriately define the class to be learned, as well as the class labels real-world semantics. Non-lucid class definitions, for example, lead to ambiguity (where a class label represents more than one domain concept), which inevitably reduces precision.

Additionally, similarly to the pre-processing phase described in the previous section, the use of well-founded ontological design patterns (ODP) may serve as an important feature to guide the development of a classifier, helping to either choose appropriate class labels, select relevant features or improve performance of classification algorithms. For example, take the Phase ODP as described by Amaral and Guizzardi (2019), consisting of "a phase partition, i.e., a disjoint and complete set of two or more complementary phases that specialize the same sortal type and that are associated with the same *dividing principle* (e.g., gender, life status, developmental state). Phases in UFO are relationally independent, anti-rigid types, defined as a partition of a sortal. This partition is derived based on intrinsic properties of that sortal". Suppose the context of a hospital that is conducting a health campaign, and which applies a classification technique to identify patients that are likely to have cancer. Knowing that the "has cancer" class to be learned is essentially a Phase of a Patient, will guide the definition of the class labels corresponding to each of the phases in the partition, and also guide the selection of the features which are relevant to describe all the concepts involved in the Phase ODP. The use of ODP may also improve the performance of the classification algorithm, by pruning the search space of classification rules based on the ODP structural restrictions.

4.4 | Clustering

Clustering is "the process of partitioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters" (Han et al., 2011). Cluster analysis can reveal previously unknown groups within the data, as the "partitioning is not performed

by humans, but by the clustering algorithm" (Han et al., 2011). "Clustering techniques apply when there is no class to be predicted but the instances are to be divided into natural groups. The clusters rely on the values of a set of pre-selected characteristics for each object in the dataset, and the set of discovered clusters presumably reflect some mechanism that is at work in the domain from which instances are extracted, a mechanism that causes some instances to bear a stronger resemblance to each other than they do to the remaining instances" (Witten et al., 2016).

Effective clustering maximizes similarities within the cluster and minimizes similarities across clusters (Chen et al., 1996). According to (Han et al., 2011), "because a cluster is a collection of data objects that are similar to one another within the cluster and dissimilar to objects in other clusters, a cluster of data objects can be treated as an implicit class".

In this article we take the position that foundational ontologies and the process of *ontological analysis* supported by them serve as a fundamental support for establishing grouping criteria and similarity calculation, reducing the possibility of creating groups of objects that do not reflect genuine real-world regularities. Because they serve as a basis for identifying the *essence* of entities of a given kind (Guizzardi et al., 2019), foundational ontologies are potentially valuable for identifying similarities in clustering process that are not merely accidental. In this direction, the identification of the foundational categories from which the concepts are derived, makes it possible to determine their nature, thus elucidating the differences between, for example, objects and events, dependent and independent entities, kinds of things and their roles, among others. The ability to make explicit these distinctions may help to prevent incorrect associations during clustering.

In the sequel we elaborate on some advantages from a clustering point view of grounding data on a foundational ontology such as UFO.

By virtue of the aforementioned characteristics of the representation of qualities in UFO, "once the semantics behind qualities, its values and value spaces are made explicit, it is possible to compare qualities, to constrain formal relations based on the properties of a value space (e.g., John being-older-than Peter), to establish mappings of values among different value spaces and to calculate similarity among entities based on their qualities" (Albuquerque and Guizzardi, 2013). In UFO, a quality type can be associated with several distinct quality structures. Let us take the example of the quality type color, which can be associated both with the RGB color space and with the HSB color space. Once all quality structures associated to a quality type are compatible, it is possible to establish equivalence relations among the regions of these structures, even when those relations are not made explicit *a priori*. This allows the conciliation of different conceptualizations based on the structure of quality types. For instance, two quality regions can be considered equivalent iff the same quality values are approximated by both regions. Following this distinction, the color blue in HSB {Hue = 240, Saturation = 100, Brightness = 100} can be defined as equivalent to the color blue in RGB {Red = 0, Green = 0, Blue = 255}, because these quality regions approximate the same quality value. Thus, clustering algorithms can benefit from the equivalence of quality regions to group similar objects.

The characteristics of *integral quality dimensions*, according to which "one cannot assign an object a value on one dimension without giving it a value on the other one" (Gärdenfors, 2004) is an important issue to be considered during clustering processes, as it generally does not make sense to analyse these attributes separately when clustering similar objects. For example, one should not consider the hue value for an object's color without considering also the saturation and brightness values.

Another case is the occurrence of second-order types categorizing instances of quality type, considering possible regions of the associated quality structure. In the example provided in Carvalho et al. (2017), the authors propose the definition of a second-order type, named *color type*, "categorizing color according to selected regions of a color domain, having instances such as 'Blue-Toned Color' and 'Green-Toned Color' ". Regarding this example, the authors state that "since each instance of color type determines a region of the color domain, its instances (i.e., instances of

color) always have values for quality dimensions within the specified region". The ability to define second-order types categorizing instances of quality type makes it possible to specify important relations between these instances. For example, a clustering algorithm can identify that a color type is similar to another one (eg. 'Yellow-Toned Color' is similar to 'Orange-Toned Color', while 'Yellow-Toned Color' is not similar to 'Blue-Toned Color').

Finally, the representation of concepts using *generalization sets* provides considerably more domain information than the specification of delimited sets of possible values for attributes. For example, in UFO it is possible to create a generalization set to specialize an entity into subkinds or phases. For example, *phases* are relationally independent universals defined as a partition of a sortal. This partition is derived based on an intrinsic property of that universal (e.g., Child is a phase of Person, instantiated by instances of persons who are less than 12 years old). In the context of clustering algorithms, the specialization of an entity in phases, instead of modeling its situation as a boolean attribute to represent its status, provides far more domain information, which can be used to support both the clustering process and the cluster quality evaluation. Let us take the example of the specialization of 'Organization' in two phases ('Active Organization' and 'Extinct Organization') instead of modeling the situation of the organization as a boolean attribute (e.g., 'active: yes or no?'). Clustering data into active and extinct organizations provides far more information about the nature of the involved data objects than grouping data based on a boolean status property.

4.5 | Association Rule Mining

"Association rules mining searches for recurring relationships in a given dataset, in order to discover interesting associations and correlations between item sets" (Han et al., 2011). Let us consider the case of a sales dataset. The mining of interesting associations can identify, for example, groups or sets of items that are likely to be purchased together. "Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data classification, clustering, and other data mining tasks" (Han et al., 2011).

Similar to the sales dataset previously mentioned, databases often store data resulting from business processes, which correspond to physical observable events. In the sales dataset, for example, each sale can be seen as an event that corresponds to an instance in the dataset. Foundational ontologies, such as DOLCE (Borgo and Masolo, 2009), GFO (Herre, 2010) and UFO (Guizzardi, 2005) make a fundamental distinction between endurants and events (perdurants). Guarino and Guizzardi (2016), based on the ontological foundations about events and their relations, propose a view in which "events emerge from scenes as a result of a cognitive process that focuses on *relators* and *modes*: endurants such as relators and modes are therefore the focus of events, which in turn can be seen as manifestations of these endurants". In the light of this discussion, a reasonable approach would be to analyse the business events that give rise to data in order to identify the endurants that are their focus. Since these endurants are typically relators, they can be potential indicators of interesting associations between item sets.

A further important aspect, related to the association rule mining process, is the significant number of discovered rules, which makes it very difficult to select and identify the interesting ones. According to Marinica et al. (2008) "in a database, there exist, in most of cases, relations between items that we consider obvious or that we already know". These relations are not interesting under the perspective of knowledge discovery and should be removed from the set of discovered rules. When data is modeled grounded on foundational ontologies, much of the semantics about the real-world is made explicit, thus making it possible to identify associations between item sets that correspond to previous knowledge about the domain. These associations can be removed from the set of discovered rules, as they do not constitute new knowledge. Additionally, models based on foundational ontologies make it easy to identify the modeling patterns used to represent data. The relations that compose these patterns should be analysed as they may

not correspond to new knowledge, but instead to solutions to known recurrent problems. In this case, the association rules resulting from these relations should also be filtered during the post-processing phase, thus reducing the number of discovered rules.

As previously mentioned, ontologies have been acknowledged as a useful conceptual tool for the improvement of semantic expressiveness of information stored in databases (Guizzardi, 2005). When it comes to mining, the semantic regarding data objects and their relations can provide some context for the discovered associations, which can help to identify if a potential association rule makes sense. It is especially important for detecting associations that reflect statistical dependence between item sets rather than some genuine ontological connection.

Let us take an example extracted from (Lapuschkin et al., 2019), in which a model "trained to distinguish between 1000 categories, has not learned dumbbells as an independent concept, but associates a dumbbell with the arm which lifts it". Due to the occurrence of an arm in the great majority of dumbbell images, the algorithm derived this kind of apparent or illusory association, even though semantically the two objects are completely distinct and their connection is merely accidental. In cases like this, in which algorithms do not use features that provide real world semantics, but are based only in statistical information, the model can generate undesired associations between item sets, also known as spurious correlations. Foundational ontologies, like UFO, comprise theories about a number of fundamental concepts like object types, properties, relations, part-whole relations, among others. In the case of the aforementioned example, according to the theories put forth by UFO, the dumbbell is an existentially independent object that has a uniform principle of identity, while the arm is a part that plays a particular functional role, contributing in specific ways to the functionality of a whole, which is the body. The body, in turn, is another existentially independent object that has a a principle of identity, distinct from the dumbbell, and which has a principle of unity that binds together its parts. Following this principle of unity, among its parts we have the arm but not the dumbbell. Again, by making explicit the ontological nature of the information stored in the dataset, foundational ontologies can reduce the creation of unwarranted accidental associations between item sets, which do not reflect real-world connections. As a consequence, it can help to prevent the mining of misleading association rules.

Table 1 summarizes some benefits of grounding data on foundational ontologies to the different steps of the KDD process.

5 | CONCLUSIONS

This paper provides a brief introduction to Foundational Ontologies (philosophically well-founded domain-independent formal theories) as well as to Ontology-Driven Domain Modeling Languages based on these Foundational Ontologies. The objective is to raise awareness in the Data Mining community of the benefits these artefacts can bring to several KDD activities including Problem Understanding, Data Pre-Processing and Post-Processing (pattern evaluation) and Data Mining, in particular for Classification, Clustering, and Association Rule Mining techniques, thus including both supervised and unsupervised learning approaches. We focused on these techniques due to their broad applicability in the Data Mining literature, although some of our proposed insights and discussions also apply to regression techniques and to other techniques that follow the reinforcement learning paradigm, as well as to pattern interpretation, which may be addressed in future work.

The benefits discussed in this work are grounded on two complementary aspects. On one hand, if "Data are fragments of a theory of the real-world", uncovering the nature of the real-world entities represented in the data is fundamental in finding truthful, stable and informative correlations and groupings therein. On the other hand, analyzing, characterizing, and making explicit the nature of these entities is the very business of Formal Ontology for

centuries. Although the area of Ontology-Driven Domain Modeling is supported by a multitude of methodological and engineering tools (Guizzardi, G. et al., 2015), research is still needed for seamlessly incorporating these tools into practical KDD methods and techniques.

TABLE 1 Summary of the benefits of Foundational Ontologies and Ontology-Driven Domain Modeling Languages to each step of the KDD process

KDD step	Benefit
Problem Understanding	Semantic transparency Complexity management mechanisms for complex domains Data models are more uniform
Data Pre-processing	Semantic interoperability Ontological commitments made explicit
Classification	Systematic guidance in the development of classifiers Increasing classification precision
Clustering	Higher probability of clusters that reflect genuine real-world categorizations Similarity calculation grounded on ontological foundations Easier to identify similarities that are not accidental Preventing unwarranted associations Expressive semantic support to the clustering process and to cluster quality evaluation
Association Rule Mining	Easier to identify relators (indicators of interesting associations between item sets) Easier to identify relations that are not interesting to knowledge discovery Detection of associations that reflect statistical dependence between item sets rather than genuine ontological connections Lower probability of creating associations that do not reflect real-world semantics Easier to identify spurious correlations
Data Post-processing	Improved understanding of the patterns discovered Systematic guidance in the validation of the patterns discovered Correspondence between the patterns discovered and the domain is grounded on ontological meta-properties

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

references

Albuquerque, A. and Guizzardi, G. (2013) An ontological foundation for conceptual modeling datatypes based on semantic reference spaces. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, 1–12. IEEE.

Amaral, G. and Guizzardi, G. (2019) On the application of ontological patterns for conceptual modeling in multidimensional models. In *European Conference on Advances in Databases and Information Systems*, 215–231. Springer.

Baralis, E., Cagliero, L., Cerquitelli, T. and Garza, P. (2012) Generalized association rule mining with constraints. *Information Sciences*, **194**, 68–84.

Bleisch, S., Duckham, M., Galton, A., Laube, P. and Lyon, J. (2014) Mining candidate causal relationships in movement patterns. *International Journal of Geographical Information Science*, **28**, 363–382.

Borgo, S. and Masolo, C. (2009) Foundational choices in DOLCE. In *Handbook on ontologies*, 361–381. Springer.

Carvalho, V. A., Almeida, J. P. A., Fonseca, C. M. and Guizzardi, G. (2017) Multi-level ontology-based conceptual modeling. *Data & Knowledge Engineering*, **109**, 3–24.

Chen, M.-S., Han, J. and Yu, P. S. (1996) Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, **8**, 866–883.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) From data mining to knowledge discovery in databases. *AI magazine*, **17**, 37–54.

Figueiredo, G., Duchardt, A., Hedblom, M. M. and Guizzardi, G. (2018) Breaking into pieces: An ontological approach to conceptual model complexity management. In *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, 1–10. IEEE.

Fumagalli, M., Sales, T. P. and Guizzardi, G. (2020) Towards automated support for conceptual model diagnosis and repair. In *International Conference on Conceptual Modeling*, 15–25. Springer.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. and Schneider, L. (2002) Sweetening ontologies with dolce. In *International Conference on Knowledge Engineering and Knowledge Management*, 166–181. Springer.

Gärdenfors, P. (2004) *Conceptual spaces: The geometry of thought*. MIT press.

Guarino, N. (1994) The ontological level. *Philosophy and the cognitive sciences*.

— (1998) *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, vol. 46. IOS press.

— (2017) On the semantics of ongoing and future occurrence identifiers. In *International Conference on Conceptual Modeling*, 477–490. Springer.

Guarino, N. and Guizzardi, G. (2016) Relationships and events: towards a general theory of reification and truthmaking. In *Conference of the Italian Association for Artificial Intelligence*, 237–249. Springer.

Guarino, N., Guizzardi, G. and Mylopoulos, J. (2019) On the philosophical foundations of conceptual models. In *29th International Conference on Information Modeling and Knowledge Bases (EJC'19),Lappeenranta, Finland*.

Guizzardi, G. (2005) *Ontological foundations for structural conceptual models*. Telematica Instituut / CTIT.

— (2007) On ontology, ontologies, conceptualizations, modeling languages, and (meta) models. *Frontiers in artificial intelligence and applications*, **155**, 18.

— (2020) Ontology, ontologies and the "i" of fair. *Data Intelligence*, **2**, 181–191.

Guizzardi, G., Figueiredo, G., Hedblom, M. M. and Poels, G. (2019) Ontology-based model abstraction. In *IEEE 13th International Conference on Research Challenges in Information Science (RCIS 2019), Brussels, Belgium*.

Guizzardi, G., Guarino, N. and Almeida, J. P. A. (2016) Ontological considerations about the representation of events and endurants in business models. In *International Conference on Business Process Management*, 20–36. Springer.

Guizzardi, G. et al. (2015) Towards Ontological Foundations for Conceptual Modeling: The Unified Foundational Ontology (UFO) Story. *Applied ontology*, **10**.

Han, J., Pei, J. and Kamber, M. (2011) *Data mining: concepts and techniques*. Elsevier.

Herre, H. (2010) General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In *Theory and applications of ontology: computer applications*, 297–345. Springer.

Keet, C. M., Ławrynowicz, A., d'Amato, C., Kalousis, A., Nguyen, P., Palma, R., Stevens, R. and Hilario, M. (2015) The data mining optimization ontology. *Journal of web semantics*, **32**, 43–53.

Khan, Z. C. and Keet, C. M. (2014) Feasibility of automated foundational ontology interchangeability. In *International Conference on Knowledge Engineering and Knowledge Management*, 225–237. Springer.

Kirk, D. and MacDonell, S. (2015) An ontological analysis of a proposed theory for software development. In *ICSOFT*, 155–171. Springer.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W. and Müller, K.-R. (2019) Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications*, **10**, 1096.

Lin, M.-S., Zhang, H. and Yu, Z.-G. (2006) An ontology for supporting data mining process. In *The Proceedings of the Multiconference on" Computational Engineering in Systems Applications"*, vol. 2, 2074–2077. IEEE.

Magdon-Ismail, M. (2000) No free lunch for noise prediction. *Neural computation*, **12**, 547–564.

Marinica, C., Guillet, F. and Briand, H. (2008) Post-processing of discovered association rules using ontologies. In *2008 IEEE International Conference on Data Mining Workshops*, 126–133. IEEE.

Mascardi, V., Locoro, A. and Rosso, P. (2009) Automatic ontology matching via upper ontologies: A systematic evaluation. *IEEE Transactions on knowledge and data engineering*, **22**, 609–623.

Mealy, G. H. (1967) Another look at data. In *Proceedings of the November 14-16, 1967, fall joint computer conference*, 525–534. ACM.

Niles, I. and Pease, A. (2001) Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, 2–9.

Padilha, N. F., Baião, F. and Revoredo, K. (2012) Ontology alignment for semantic data integration through foundational ontologies. In *International Conference on Conceptual Modeling*, 172–181. Springer.

Priebe, T. and Markus, S. (2015) Business information modeling: A methodology for data-intensive projects, data science and big data governance. In *2015 IEEE International Conference on Big Data (Big Data)*, 2056–2065. IEEE.

Ristoski, P. and Paulheim, H. (2014) Feature selection in hierarchical feature spaces. In *International conference on discovery science*, 288–300. Springer.

— (2016) Semantic web in data mining and knowledge discovery: A comprehensive survey. *Journal of Web Semantics*, **36**, 1–22.

Ruy, F. B., Guizzardi, G., Falbo, R. A., Reginato, C. C. and Santos, V. A. (2017) From reference ontologies to ontology patterns and back. *Data & Knowledge Engineering*, **109**, 41–69.

Shcherbakov, M., Shcherbakova, N., Brebels, A., Janovsky, T. and Kamaev, V. (2014) Lean data science research life cycle: A concept for data analysis software development. In *Joint Conference on Knowledge-Based Software Engineering*, 708–716. Springer.

Silla, C. N. and Freitas, A. A. (2011) A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, **22**, 31–72.

Trojahn, C., Vieira, R., Schmidt, D., Pease, A., Guizzardi, G. (2021) Foundational ontologies meet ontology matching: A survey. *Semantic Web Journal (accepted, forthcoming)*.

Verdonck, M. and Gailly, F. (2016) Insights on the Use and Application of Ontology and Conceptual Modeling Languages in Ontology-Driven Conceptual Modeling. In *Proc.35th ER*.

Verdonck, M., Gailly, F., Pergl, R., Guizzardi, G., Martins, B. and Pastor, O. (2019) Comparing traditional conceptual modeling with ontology-driven conceptual modeling: An empirical study. *Information Systems*, **81**, 92–103.

Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2016) *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Ziegler, P. and Dittrich, K. R. (2007) Data integration—problems, approaches, and perspectives. In *Conceptual modelling in information systems engineering*, 39–58. Springer.