Data & Knowledge Engineering 101 (2016) 1–23

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

An analysis of ontologies and their success factors for application to business

Christina Feilmayr, Wolfram Wöß *

Johannes Kepler University Linz, Institute for Application Oriented Knowledge Processing (FAW), Altenberger Straße 69, 4040 Linz, Austria

A R T I C L E I N F O

Article history:
Received 22 June 2014
Received in revised form 24 November 2015
Accepted 30 November 2015
Available online 23 December 2015

Keywords:
Ontologies
Data models
Ontology engineering
Data modeling

A B S T R A C T

Ontologies have been less successful than they could be in large-scale business applications due to a wide variety of interpretations. This leads to confusion, and consequently, people from various research communities use the term with different – sometimes incompatible – meanings. This research work analyzes and clarifies the term *ontology* and points out its difference from taxonomy. By way of two business case studies, both their potential in ontological engineering and the perceived requirements for ontologies are highlighted, and their misuse in research and business is discussed. In order to examine the case for applying ontologies in a specific domain or use case, the main benefits of using ontologies are defined and categorized as technical-centered or user-centered. Key factors that influence the use of ontologies in business applications are derived and discussed. Finally, the paper offers a recommendation for efficiently applying ontologies, including adequate representation languages and an ontological engineering process supported by reference ontologies. To answer the questions of *when* ontologies should be used, *how* they can be used efficiently, and *when they should not be used*, we propose guidelines for selecting an appropriate model, methodology, and tool set to meet customer requirements while making most efficient use of resources.

1. Introduction

Computer science and software engineering are relatively recent disciplines compared to other sciences and philosophy. The field continues to evolve as languages mature, representations develop, and the ability to devise solutions to new challenges increases with more advanced software and hardware systems. Artificial intelligence (AI) has been a significant challenge for computer scientists, and while the community has yet to develop "true" human-level machine intelligence, the pursuit of AI has led to the development of knowledge representation and semantic relationships [1,2]. An *ontological representation* allows modeling meaning in systems that are to be implemented using a programming language and a database schema. Unlike general software development approaches, such as object-oriented programming models, which enable the transformation of a model into a useful software artifact [3], an ontological model allows software to be generated that can evaluate semantic relationships, validate statements made within a domain of knowledge, and provide much richer rules for information management [4]. As required in artificial intelligence, an ontological model allows known facts and/or assumptions to be used to derive a conclusion or to make inferences (i.e., reasoning).

However, although ontological engineering has been applied for decades, there are still very few truly ontology-based systems that exploit all the benefits of an ontology-based approach (i.e., reasoning) and do much more than classify knowledge into convenient categories. This paper does not aim to provide a broad discussion of the term *ontology* merely by comparing the semantic differences between several definitions. Rather, the remainder of this paper focuses on the analysis of ontologies and their

* Corresponding author. Tel.: +43 732 2468 4184; fax: +43 732 2468 4200.
E-mail addresses: christina.feilmayr@jku.at (C. Feilmayr), wolfram.woess@jku.at (W. Wöß).

http://dx.doi.org/10.1016/j.datak.2015.11.003
0169-023X/

frequent misuse in research and business and examines why ontologies have not been as successful as they could be in large-scale business applications. Ontological engineering is discussed in detail and compared with both entity-relationship modeling for conceptual database design and software engineering for software design in order to identify their proven benefits and best practices and, whenever possible, to adapt them to ontological engineering.

This paper seeks to clarify the benefits of an ontology-based approach. First, reasons for the frequent misuse or rare use of ontologies in research and business are discussed to examine when ontologies should be used, how they can be used efficiently, and when they should not be used. Guidelines to support the decision for a correct model, methodology, and tool set to meet the project specifications and the customer requirements are proposed. For this purpose, the resources available should be used most efficiently as it is widely established in software engineering to clearly identify which approach to take under given circumstances with a high degree of confidence. Considering the benefits of a clear and machine-interpretable basis for *meaning* in a system built on ontologies, this is a great opportunity to improve knowledge management and decision making in software systems. Another important benefit of an ontological approach, as discussed further below, is that concepts, their meaning, and their relationships can be shared [5]; this makes possible a clear and unambiguous agreement between a larger number of participants and offers new opportunities in terms of data interchange and data interpretation by machines.

Section 2 provides an overview of ontologies and their background, including a brief discussion of their benefits and a differentiation from taxonomies, and concludes that the general understanding of ontologies is low. This has an impact on their uptake and further deployment, including reuse. In Section 3, case studies from industry and research (i) point out their potential for ontological engineering and their perceived requirements for ontologies and (ii) identify reasons for the misuse of ontologies in research and business. Section 4 discusses in detail the benefits of appropriately used ontologies. Further, it provides a comparative classification of other existing models and technologies, whose relations to ontologies to support the decision for the best-suited methodology or method for each case. Ontological modeling is compared to other conceptual modeling approaches in Section 6, which leads to the question whether ontological engineering is a new development or just a new term for something that already exists. Examining the field of software engineering reveals a number of ontological engineering developments that have trailed software engineering process models. As demonstrated, the software engineering community has already dealt with most of the process models and engineering challenges, and while there are more subtleties and abstractions in the area of ontological engineering, a main goal of this paper is to motivate ontological practitioners to consider adopting some of software engineering's successful techniques, including the increasing reuse of existing ontologies by applying reference ontologies. Finally, ontological engineering is motivated as a separate and valuable discipline. The discussion concludes with (i) indicators and recommendations for choosing appropriate models and tools and (ii) a critical analysis of ontologies and their application in business in order to improve the maturity and capabilities of ontological engineering. This process must be accompanied by a mature community understanding of what is being discussed and, most importantly, when a project is or is not ontologically based. In order to provide a recommendation for efficient application of ontologies in a (business) project, the following key influencing factors and requirements were identified as being relevant:

- Requirement for sharing
- Semantic expressiveness
- Complexity of the universe of discourse
- Size of the sharing community (ontology stakeholders)

Table 1
Definitions of and references for important terms related to ontologies.

Term	Definition	Reference
Ontological representation	An ontological representation is used to represent defined knowledge. The term *ontology* is frequently used as a short form of *ontological representation*.	Our definition.
Conceptual model	A conceptual model focuses on capturing and representing certain aspects of human perceptions of the real world.	[6]
	A conceptual model is intended to capture knowledge about a real-world domain.	[7]
Ontological model	The term ontological model is often used synonymously with the term ontology.	[1]
	An ontological model is created to analyze the meaning of common conceptual modeling constructs.	[7]
Knowledge framework	A knowledge framework supports the analysis of an area of knowledge. Within this framework, features of an area are identified in the form of a specific terminology and concepts that shape that area of knowledge. At a minimum, one considers the four basic processes of creating, storing/retrieving, transferring, and applying knowledge.	Accords with the definition of [8]
Knowledge representation	(i) Most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it. (ii) A set of ontological commitments. (iii) A fragmentary theory of intelligent reasoning. (iv) A medium for pragmatically efficient computation. (v) A medium of human expression, i.e., a language in which we say things about the world.	[9]
Semantic relations	Semantic relations are relations between concepts. A relationship covers associations between concepts that go beyond hierarchical ones; thus, they are conceptually associated to such an extent that the link between them should be made explicit.	Adapted from [10]

The most important concepts, their definitions, and meanings used and discussed in this research paper are introduced in Table 1. Based on comprehensive research, we define an ontology by extending the definition by Studer et al. [1] with the key influencing factors which constitute the main benefits of ontologies (cf. Section 4.2.):

> *An ontology is a formal, explicit specification of a shared conceptualization that is characterized by high semantic expressiveness required for increased complexity*.

2. Information technology following philosophy

The general misuse of ontologies can be explained by (i) many existing, sometimes conflicting, definitions; (ii) insufficiently precise specifications of semantic technologies; and (iii) the existence of complex modeling processes that are too abstract or too complex. Since the term ontology is used with a variety of meanings, some are accurate, it has become increasingly difficult to identify who has or has not using ontologies correctly. The semantic web [11] with its characteristics and goals seems to be the most prominent promoter of ontologies and, potentially, also of all the inconsistencies in their understanding and application. This section highlights important limitations that prevent ontologies from becoming successful.

The term *ontology* is derived from philosophy and describes the study of the nature of being, existence, or reality in general, as well as the basic categories of being and relations [12]. Ontology deals with whether a certain thing exists or can be said to exist. Computer scientists have borrowed the philosophical term *ontology*, which has become established but also evolved and changed its meaning over time.

One of the first computer science definitions was given by Neches et al. [13], "An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary." According to Gruber [5], "An ontology is an explicit specification of a conceptualization." Guarino and Giaretta [14] presented a broad discussion of possible interpretations of the term ontology, concluding with a more formal notion of the "conceptualization," and addressed Gruber's definition to determine exactly which interpretations were consistent with which definitions. They found that the use of the term ontology depends on whether both parties using it have already decided upon a degree of expressiveness or a shared conceptualization. An ontology was defined by Borst [15] as a "formal specification of a shared conceptualization." This definition additionally required that the conceptualization should express a view shared by several parties (a consensus rather than an individual view) in a (formal) machine-readable format. Studer et al. [1] merged the definitions by Gruber and Borst: "An ontology is a formal, explicit specification of a shared conceptualization."

Berners-Lee, Hendler, and Lassila [11] published the original "semantic web" paper, introducing a "web of data" that enables machines to understand the semantics (i.e., meaning) of information on the world wide web (WWW). They introduced ontologies as the third basic semantic web component, covering a taxonomy and a set of inference rules. In the following years, the meaning of the terms *semantic web*, *semantics*, and *ontologies* mutated through misuse to become predominantly buzzwords. At the same time, the application of inference rules, which are used for automated reasoning, shifted ontologies from their field of origin, AI, into the focus of a broad audience in the WWW (cf. Section 2.1). As ontologies became more useful, overuse and misuse of the term rendered it more difficult to assess who was or was not using true ontologies.

A number of further definitions of ontology exist within computer science. Frequently changing definitions may indicate a lack of understanding in the field or an inability to communicate effectively and share an understanding. Whatever the reason, introducing multiple definitions of one concept, in particular of a complex one such as ontology, leads to confusion and causes people from various research communities to use the term with different, partly incompatible meanings. It seems paradoxical that the core term of a novel field of research that aims to reduce ambiguity about the intended meaning of symbols is understood and used so inconsistently [16].

Given these definitions and contradicting interpretations of ontology, the question arises of why ontologies are so popular in information technology and its research fields, but not yet widely applied in business. The following sections discuss driving factors that are responsible for the increasing importance of ontologies and address key issues when applying ontologies.

2.1. The semantic web

The semantic web, a key application of ontologies, includes a family of standards, patents, and languages. One of them is the resource description framework (RDF),[1] a metadata model that allows statements about resources to be made. However, one of the most obvious advantages of the semantic web – the ability to search through documents that support RDF mark-up – appears to be losing importance.

Whereas the world wide web is a medium of documents for people, the semantic web addresses data and information that can be processed automatically by providing rich and extensible metadata. Improving information retrieval with semantically enriched search functions is seen as a major advantage of the semantic web [17]. However, the demand for optimized search functions remains high. While, in the past we sought to increase the number of pages or documents we could find (i.e., to

[1] http://www.w3.org/RDF/ [December 07, 2015].

improve *recall*), the huge number of documents now available from the global data corpus requires us to shift the focus to how to find the most relevant documents (i.e., to improve *precision*). Thesauri or extensive synonym lists are not adequate for this task because they increase recall but reduce precision – contrary to our goal.

Adding metadata provides embedded mark-up on web pages that increases hit rates in response to customer queries and improves document management. This is a straightforward mining exercise to generate the largest number of synonyms that can legitimately be matched to search engine queries. Most importantly, while this uses parts of semantic web technology, there are no actual "semantics" that are embedded with these synonym search terms, and therefore this is a misleading use of the term semantics. True semantic technology enables the expression of both data and rules for reasoning about data and allows rules from existing knowledge-representation systems to be exported to the web [11]. RDF triple sets (subject–predicate–object) can be defined to express relations between documents and their meanings. RDFa adds a set of attribute-level extensions to HTML, XHTML, and various XML-based document types for embedding rich metadata within web documents. The RDF data model mapping enables embedding of RDF subject–predicate–object expressions within XHTML documents. This flexible, domain-independent language is already included in some search engines, such as Google [18], but is limited to a subset of possible domains. RDFa clearly separates the linguistic semantics of the content piece and the (to the user) hidden semantics added by the content creator. However, these semantics are still locally contextualized based on the knowledge of the creator and the assumed scope of the reader. If information is to be compared or combined across schema borders, it must be possible to express some kind of common meaning. This *shared conceptualization* leads directly to a requirement for ontologies.

All definitions of *ontology* we have described so far are not what is being used in the real-world search engine case. Search engines use an approach that is much closer to a *standardized vocabulary* or, in its most expressive form, a hierarchical set of vocabulary classes that provide wider match areas, a thesaurus. What sets ontologies apart is that they enable precise matching of concepts, thus going beyond analyzing grammar and context to resolve language ambiguities such as words with multiple meanings (homonyms) or different words for the same meaning (synonyms).

If a semantic matching technology is to be incorporated into the mass-synonym case, then – notionally – an ontology is required on both ends of the search mechanism (not just on the provider web site) in order to allow the user and search query provider to return the most precise matches through automated agreement. In practice, however, almost all search engines are based on single-sided-semantics. Google search, for example, is believed to make extensive use of ontologies to identify likely term matches that, in conjunction with exhaustive search term matching on the provider web site, return the largest possible range of matches. Again, this enhances recall, but at the cost of precision. From the performance perspective, while a high recall mechanism arguably reduces the need for a human to search synonym space, this advantage is negated by having to search many – potentially inaccurate – returned search items, which is frustrating and time-consuming.

The real benefit of the semantic web is realized by collecting web content from diverse sources, processing this data, and exchanging and distributing the results. The linked open data (LOD)[2] project is a successful scenario of semantic web approaches. Linked data uses the web to connect data by semantic relationships and build upon standard technologies such as HTTP, RDF, and URIs. Linked open data is becoming increasingly important in state-of-the-art information and data management [19]. In order to fully benefit from LOD, information and data must be put into a context that creates new knowledge and enables powerful applications [19]. However, the way from "information is available on the web" to "data is linked to other data to provide context" is long because for open data, it is essential that it is complete, provides a high level of granularity, and is timely, accessible, machine-processable, license-free, and permanent. The LOD cloud increases continuously (2010[3]: 26,930,509,703 triples, 203 data sets; 2011[4]: 31,634,213,770 triples, 295 data sets; 2014[5]: 900,129 documents describing 8,038,396 resources, 1014 data sets), and knowledge transfer has partially improved. In 2011, 190 (64.41 %) data sets mapped to a proprietary vocabulary (35.59 % used a non-proprietary vocabulary), where terms were not defined in the same top-level domain, 15 (7.89 %) of 190 data sets provided mappings to other vocabularies for their terms, and 154 (52.20 %) of 295 data sets did not provide provenance information (data about the origin of the information to assess data quality). In 2014, 241 (23.17 %) data sets used proprietary vocabularies, and nearly all (99.87 %) data sets also used non-proprietary vocabularies. In total, 19.25 % of all proprietary vocabularies were fully dereferenceable. However, 72.75 % of all proprietary vocabularies were not dereferenceable at all. The remaining vocabularies, which amounted to 8.00 % of all proprietary ones, were partially dereferenceable, which means that a definition could be retrieved for some – but not for all – terms. Of all data sets, 35.77 % used some provenance vocabulary. The number of data sets that set RDF links which pointed to numerous other data sets remained relatively small, while many data sets linked to only a few other data sets. Remarkably, most context information and cloud resources were retrievable by URLs, and web applications could use linked data by means of standard web services. The LOD cloud is considered to be domain-independent, but it provides mostly content from the media, geographic, publications, user-generated content, government, and life-sciences domains.

Most importantly, machine-interpretable web content promotes web information integration because even systems that have not been expressly designed to work together can transfer data to each other if it is annotated with semantics. The semantic web, through its goals and standardized nature, supports a strong middleware focus in the architecture, but it assumes that the middleware is automated. Humans should not be the primary middleware that shoulders the burden of semantic alignment between systems.

[2] http://linkeddata.org [December 07, 2015].
[3] http://lod-cloud.net/state/2010-10-19/ [December 07, 2015].
[4] http://lod-cloud.net/state/ [December 07, 2015].
[5] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/ [December 07, 2015].

The factors described above and their relevance to the semantic web explain why ontologies have became so popular:

- The requirement for sharing is very important in the semantic web.
- The semantic web is highly complex due to the open-world assumption and comprises a large number of objects and relationships.
- The semantic web therefore requires powerful concepts and languages to reach the necessary semantic expressiveness.
- The size of the sharing community (participating organizations) is huge.

Ontologies promise to achieve interoperability (i) between multiple representations of reality (e.g., a data model) residing inside computer systems and (ii) between such representations and reality – namely, human users and their perception of reality [16]. Consequently, the semantic web with its characteristics and goals seems to be the most prominent promoter of ontologies and, potentially, also of all the inconsistencies in understanding and applying them.

2.2. Ontologies are used as a one-size-fits-all solution

In order to become ubiquitous, a technology must be implemented and then delivered such that it is either considered to be indispensable and hence worth any integration effort, or seamlessly integrated and hence invisible from an effort perspective. However, ontologies cannot be considered to be either entirely indispensable or entirely without effort. Consequently, the question arises of why ontologies are ubiquitous in information technology research but the industrial application that raises ontologies to a necessity in business and in the industrial community remains elusive. We must assess whether ontologies have the potential to meet the most important requirements for a successful technology; namely, that it

- can save time and/or money,
- is easily understood and passed on,
- is sufficiently widely used, and
- is, effectively, ubiquitous.

While ontologies (or rather taxonomies) are used *within* a number of applications, it is rare for users to be directly exposed to, or required to interact with, ontologies. The semantic web, whether a powerful application for a web of data or for semantics-assisted search, has remained a niche technology to the extent that it has failed to gain widespread attention or to be seen as relevant to mainstream groups. The life sciences are one area where ontologies are pervasive and part of research [20]. However, as discussed below, the reason for their success in the life sciences is not due to the advantages of computer-based ontologies in the pure sense but because this is a natural extension of the existing application of ontologies and taxonomies within the life sciences. Technologies that are analogous to existing practice are more readily adopted by a community, as the *affordance* (i.e., the quality of a technology or item that allows a user to interact with it successfully) of computational taxonomies is effectively the same as that of the tree of life or of genome sequences. Application domains with no analogue require a user to develop a new understanding of how to employ it correctly.

One of the important limitations that prevent ontologies from becoming successful is the fundamental requirement of a minimal (shared) ontological commitment of the knowledge stakeholders. The stakeholders must have, and must be able to agree upon, a shared understanding of the fundamental terms [21]. Especially for ontologies intended to support large-scale interoperability, it is important to be well founded in the sense that the basic primitives they are built on are sufficiently well chosen and axiomatized to be generally understood [22]. Reaching such an agreement in conceptual alignment normally requires human interaction, primarily during the design phase, and hence requires an additional investment in earlier phases to make information machine-interpretable.

Ontologies have to fulfill a central function if they are intended to be valued as effective and successful. This core function enables communication between human and machine or even inter-machine and inter-human communication and is therefore one of the main reasons why ontologies should be applied frequently. However, ontologies are not the one-size-fits-all solution they are presented as in a large number of current research approaches. Considering the key characteristics of a successful technology, the concluding questions arise, which are discussed in the following sections.

- Is there truly a shared understanding of what ontologies are, given how many definitions are already in use?
- Assuming that identity and nominative concerns have been addressed, for what kinds of complexity and application are ontologies appropriate?

2.3. Ontologies evolving from a buzzword to business applications

It is obvious from the large number of different definitions of ontology currently in use that very few computer scientists can understand the term correctly and consistently, identify how ontologies can be applied, and correctly assert the list of benefits that can be derived from ontologies. This feeds misunderstandings between researchers in different fields and between academic research and business/industrial use. Ontologies can be specified by using only informal means, such as UML class diagrams [23], entity-relationship models [24], and semantic nets, whereas conceptual entities in ontologies can also be defined mainly by formal means, for instance, by using axioms to specify the intended meaning of domain elements [16]. This uncertainty leads to a broad spectrum of models, concepts, and specifications being interpreted and published as ontologies. Originally, and as evidenced by the previous references, the term ontology conveyed a considerable weight and importance. However, it has now become almost some kind of warning signal due to its being overused increasingly vaguely and inaccurately. This is often the path by which a perfectly acceptable and well-defined word becomes a hyped buzzword that raises expectations which are never fulfilled.

The unfulfilled promise of early academic technologies and research is a specter that haunts many new technologies. For instance, the object-oriented database initiatives of the late 20th century never made it from the academic to the economical field. After initial hype, which advertised object-oriented database management systems as the database technology of the future that would supplant relational database management systems (RDBMSs), further development declined. None of these predictions has come to pass; relational databases remain the most widely used databases, and object-oriented databases are increasingly rare. Nevertheless, among others, Oracle added some object-oriented concepts to RDBMSs, thus offering a compromise solution for the majority of purely relational applications and supporting some research projects and certain businesses with an interest in object-oriented databases. However, this is not the path that leads to ubiquitous adoption. Learning from the experiences associated with the over-promotion and over-promising of object-oriented databases and the recent developments of ontologies, the following requirements can be derived for the successful, beneficial integration of ontologies into business applications.

- The terms *ontology* and *semantics* must be clearly defined. For example, some businesses have placed static search tags onto every web page and refer to this as a "semantic" application [25,26].
- Guidelines should be provided that help to answer the questions of *when* ontologies should be used, *how* they can be used, and *when they should not be used*. Use cases must move beyond the simple pizza and wine/food examples published as part of the W3C standards to provide exemplars that businesses can understand and apply immediately. They must have sufficient scope to be helpful.
- The key differences between applying ontologies in research and in business must be recognized. Researchers aim to share their knowledge data structures and instance data in order to demonstrate the usefulness of their work and to facilitate scientific interaction. Businesses are far more likely to keep their information in house instead of sharing their developed resources widely. Thus, we call for an informal answer to the question "Is the considered implementation open-world or closed-world?" The business-centric representation may be considered as closed-world if knowledge is expected to be shared only internally. Conversely, the research standpoint is one of sharing and large-scale collaboration.
- Considering the application requirements – the required degree of expressiveness, the representational model, and the required degree of sharing – an adequate representation language (knowledge interchange format, resource description framework, OWL, the web ontology language, etc.) must be selected. The most suitable representation language minimizes the potential mismatch between the user requirements and the final representation.

An example that demonstrates some of these challenges is the discussion concerning RDF versus OWL (more specifically, taxonomy versus ontology), which cannot be resolved by drawing a line and labeling one side "no explicit semantics required" and the other "explicit semantics required." Meaning is also conveyed by structure, and any reasoning requires structural semantics to identify additional and missing elements.

2.4. Taxonomy versus ontology

Different understandings and interpretations of the terms taxonomy and ontology are in use, mainly due to the evolution of the term ontology (as mentioned above). The trend seems to be toward a more frequent use of complex term ontology without further qualifications [27]. An example in which these terms are confused is the Mondeca Tourism Ontology[6] – which is in fact a taxonomy that takes its concepts from the World Tourism Organization (WTO) thesaurus and is managed by Mondecas Intelligent Topic Manager. Accordingly, in a significant number of publications the Mondeca tourism taxonomy is erroneously cited and represented as an ontology (e.g., [28]). In order to clarify the terms and their interpretation, this section briefly compares taxonomies with ontologies as representative example for applying or not applying the most appropriate representation model for a specific use case.

A *taxonomy* is a hierarchical classification of entities within a domain, with generalization/specialization relationships and subtype inheritance based on the very general meaning of "is-a" (one class "is-a" superclass of another). In a well-defined taxonomy, entities are assigned to unambiguous and mutually exclusive classes and subclasses [29]. A typical application scenario for a taxonomy are locations in the tourism domain, describing that a town is in a state, a state is in a country, and a country has several regions.

An *ontology* is broader in scope than a taxonomy, as it contains a full specification of a domain, including relationships, such as "composition," "if-then-else," "and," "or," "not," "equal," and the clear distinction between schema and instances [29]. In addition, an ontology supports reasoning to deduce new classifications. Nevertheless, the backbone of an ontology consists of a generalization/specialization hierarchy of concepts, that is, a taxonomy [22]. Since an ontology is more comprehensive and includes additional concepts compared to a taxonomy [30], the creation and establishment process is more complex; for example, relationships must be specified explicitly in a manner that supports reasoning. Noy and McGuiness [4] introduced a development methodology and clearly showed the steps necessary to systematically build an ontology:

1. Determine the domain and scope of the ontology.
2. Consider reusing existing ontologies.
3. Enumerate important terms in the ontology.

[6] Mondeca Intelligent Topic Manager [December 07, 2015].

4. Define the classes and the class hierarchy.
5. Define the properties of classes and slots.
6. Define the facets of the slots.
7. Create instances.

Step 4 of this iterative process concerns the creation of a taxonomy, which is often considered to be a straightforward task based on available exemplars and existing knowledge. Ontology development extends this single step by adding the definition of properties, facets, and instances in order to allow inference. As a result of this close connection, a taxonomy is also described as a "simple ontology" in the literature [27]. In addition to this methodology, extensive testing is required to ensure a correct inferential span. Building a complex structural and relational ontology model – a time-consuming and arduous iterative development process – is therefore a waste of time and resources when only a taxonomy is needed.

3. Case studies

In order to further emphasize the importance of target-oriented investment regarding model, representation, and resources, this section introduces two case studies from our portfolio of project collaborations. Both case studies are analyzed in terms of their potential in ontological engineering, and the perceived requirements for ontologies are highlighted considering the following factors:

- Characteristics of the target domain and the stakeholders involved.
- Reasons for applying or refraining from applying ontologies within a specific domain along with identified benefits and drawbacks.
- Conclusions and lessons learned from the results of the projects.

The first case study "manufacturing industry" points out current requirements of manufacturing enterprises in terms of data integration by analyzing the results of a 5-year collaboration with the automotive industry. An ontology-based development is not considered suitable for efficiently establishing homogeneous data views, data integration, and workflows. The second case study "semantic web in tourism" discusses a business sector where several initiatives and Europe-wide projects developed a set of diverse ontologies. Either they did not succeed in completely replacing existing well-established methods or the projects failed to build sufficient consensus in the tourism community to specify a common ontology in order to support interoperability and information integration.

3.1. Manufacturing industry

This case study identifies the requirements for computer-aided manufacturing with the aims of product quality optimization and workflow efficiency and illustrates why ontologies are not (yet) of interest to the manufacturing industry. In the last two decades, an increasing number of company divisions benefitted from widespread support of various information systems that, ideally, provide an exact representation of the distinct processes of a specific business sector (e.g., construction, or manufacturing). Such information systems typically store their data in a custom-built, dedicated database, separate from other existing databases that are already in use. After some time, such a scenario develops into a distributed and heterogeneous information system landscape within an enterprise. Integrating distributed local data into a central database prevents redundancies and inconsistencies, and additionally provides homogeneous data structures. However, due to the tight coupling between software and corresponding data structures, the data is strongly tied to a data schema. A central database scenario is not realistic for most companies since the necessary software adaptation would be both time- and cost-intensive. Alternatively, data interchange mechanisms between several information systems can be established. In a manufacturing enterprise, the product life cycle with its involved information systems must ultimately establish such data interchange mechanisms to support the entire process chain. Moreover, business logic that overlaps the information system borders must be implemented on the software side. These challenges grow with the size of the company and the number of active information systems and are a significant burden for large companies or group companies. In most cases, manufacturing enterprises still lack support of the *entire process chain*, as we have observed in several industrial project collaborations.

The automotive industry is a representative example. If the final product (e.g., car) causes a problem, the customer visits a car repair company, where the defect is detected and stored in a specific defect database. Accumulated data is transferred to the manufacturer's quality management division, where the experts try to find out whether there is a general problem in the production process or a specific problem with the individual vehicle. In the time span between detecting the defect and identifying its possible cause in the production, a (very large) number of faulty components may be produced. Consequently, in order to save costs, it is essential that the time between fault detection at the car repair company and production improvements on the manufacturer's side is as short as possible. To optimize this process, the quality management must first identify the relevant attributes of the detected car defect. In the next step, affected bills of materials must be delimited (in this step a determination is not yet possible) and, in combination with the recorded time of technical changes within the subset of delimited bills of materials, the responsible production step can be identified [31].

This analysis procedure is representative of many other business applications. It requires interaction and cooperation of several domain experts across different divisions and is thus a manual knowledge alignment problem with temporal and geographic constraints that uses humans as middleware. Furthermore, data is not available on time, potentially because export and import procedures are too inefficient. At this time, in the majority of companies, data transformation, data assignment and combination, and data analysis are performed manually because an integrated process view supporting the *entire process chain* of the specific company has not yet been implemented. Even once all the data manipulation has been carried out, interconnection of the identifying primary keys in different information systems is not always possible, since for instance, the identifier of the final product (e.g., car) has no reference to the bill of materials in the production database. Bills of materials are managed at the type level, whereas final products are managed at the instance level. A shift from type level to instance level is also triggered by new manufacturing requirements, for example, when carbon-fiber-reinforced plastic parts are used. In addition to considering the bills of materials, the process parameters (temperature, pressure, etc.) must be obtained; they must be stored during the manufacturing process to ensure reproducibility and to improve the cooperation between development and manufacturing divisions. A requirement for tracking introduces increasing load to the production process in terms of monitoring and early alignment of data to the workflow. Hence, the focus is on traditional data management and database challenges.

For quality optimization purposes, manufacturing companies must bridge the gaps between different information systems in order to be able to establish analytical data processing of historical data that supports reactive adaptations and decisions. For a far-reaching optimization, which considers also the increasing number of parameters during the development and manufacturing processes, more active data analysis is necessary for cause and effect forecasts. The aspects discussed above are reflected in the following optimization requirements we derived, where requirements 1–3 are collaboration tasks and requirements 4 and 5 are common data engineering tasks:

1. Global consideration of workflows
2. Defragmentation of workflows
3. Integrated information flows
4. Data preparation, data cleansing
5. Establishment of a global metadata model
    a. Mapping of incompatible identifiers
    b. Harmonization (syntax, scale, etc.)
    c. Aggregation

Semantic integration has low priority because intra-company integration is based on structured information in a mostly closed world. The degree of sharing corresponds to the degree of necessity of semantic expressiveness. There is no real requirement for sharing, as we have already concluded. Using ontologies is efficient when formal models are needed that aim to compile and classify information and resources in several knowledge domains such as the (semantic) web. In such an open-world scenario, ontologies support shared understanding in a domain of knowledge that may be used as a unifying framework to handle interoperability, reuse, sharing, and mismatching terms, as required in business-to-business integration and the life sciences [32–34].

In contrast, data quality improvement, as needed for optimization purposes in manufacturing companies, requires homogeneous, consistent views of distributed information that focus on syntactic and structural heterogeneities. Such a homogeneous view is a global schema resulting from integration of individual schemas, each of which represents one distributed information system. Representing a database schema as an ontology is inefficient because it

- is already an abstract model of the real world,
- is explicitly available,
- is machine-readable (data dictionary),
- does not require reasoning in order to derive knowledge,
- is typically developed for a limited number of applications, whereas ontologies represent a consensus of a larger number of partners modeled by a set of experts in a specific domain.

Businesses such as those in the automotive industry have no interest in sharing business processes that are difficult and expensive to develop because this may, in their opinion, reduce their business advantage without a real benefit. Thus, while these businesses would benefit from using an ontology for their internal, intra-business communication, the closed-world model will defeat any attempts to create a more generic – or open – ontology for broader information exchange.

3.2. Semantic web in tourism

Travel and tourism is commonly known as an information-intensive domain where online information plays an important role. Since the web is no longer only an information source but more user-centered, users can express opinions about their preferences, rate different places (e.g., hotels, bars, visitor attractions), take part in social networking, and contribute to the formation of a user-centric data corpus that may be viewed alongside business-provided web elements. The tourism sector is characterized by a large number of key players who are concerned, for instance, with transport, accommodation, gastronomy, tourism services (such as leisure facilities), and tourist destination management (and their cultural offers). Each player has different information requirements and a specific perspective on the overall tourism domain. Hoteliers are concerned with issues that affect their room bookings and return rate, whereas restaurateurs deal with food fads, gastro tourism trends, and table reservations. Consequently, to enable decision making and action taking, the most effective technology must be used to generate, process, and apply this vast amount of information.

Today's information management solutions for the complex tasks performed by tourism intermediaries are still at an early stage from a semantic point of view. Nevertheless, information technologies such as the semantic web and web 2.0 play a central role in the tourism domain. The emergence of ontologies has enhanced web-based tourism information systems, which are highly valued by researchers and key players in the tourism domain as a mainstay of the application of semantic description technologies. The rationale for establishing ontologies in tourism information systems is threefold. First, ontologies are used to compensate for the interoperability problem associated with aligning and integrating heterogeneous data sources. The tourism domain is characterized by a large set of different information systems with different scopes, technologies, architectures, and information structures [35]. Ontologies are expected to support the integration of the different information subdomains of several key players. Second, ontologies are a formal basis for establishing recommender systems, inferential statistics, and analysis, and for deriving new knowledge from recorded information (cf. data mining, opinion mining, and sentiment analysis). Third, key players in the tourism domain want to apply ontologies for higher-quality information search operations for automatic discovery, negotiation, and adaptation/personalization of tourism services [36]. In summary, the most relevant benefit of ontologies in tourism is a shared conceptualization that is explicitly available and formally defined. With this basis, information can be further processed in recommender systems, inference systems (e.g., Apache Jena[7]) and knowledge management systems.

Further application scenarios for tourism ontologies arise from these initial motivational factors for establishing ontologies and semantic technologies in the tourism domain. The latest trends are recommender systems, human–computer interaction, ubiquitous computing, mobile technologies, search systems, location-based services, social media, and system integration. Research groups analyze the system interaction and behavior of users when searching for leisure information and planning travels. Furthermore, decision-making systems have been developed based on contextual information to offer personalized travel planning. Ontology-based travel planning, user profiling, and modeling contextual information are becoming increasingly popular.

3.2.1. Standards

Industry, academia, and several collaborative projects have created different standards, catalogues, taxonomies, and ontologies that should help to manage heterogeneous tourism concepts and their data. However, a semantic unification for sharing information among various participants (e.g., an alignment of the knowledge requirements of a hotel restaurant manager with those of the hotel general manager) is still missing. A prerequisite for developing an ontology in the field of tourism is an analysis of existing standards (e.g., accommodation or hotel classifications) for terms and classifications, which are summarized in the following list:

- Accommodation Facility Classification (Deutscher Tourismusverband e.V.) is a classification system for accommodation facilities with the aim of a more precise product placement and thus of increasing sales opportunities. It includes most of the terms concerning room setup and service [37].
- German Hotel Classification (Deutsche Hotel- und Gaststättenverband, DEHOGA) [38] is applied to define the criteria on whose basis an accommodation facility is given its star rating.
- Tourism services – hotels and other types of tourism accommodation – Terminology (ISO 18513) [39]. This European Standard contains definitions of a number of terms commonly used in the tourism industry. The standard is designed to facilitate understanding between users and providers of tourism services. Assisting consumers in making an informed choice of tourism service has the potential to increase the likelihood of expectations being met and of enhanced satisfaction. The tourism industry also benefits from better-informed consumers. This standard is also intended to be of value to those developing other tourism and travel standards.
- Thesaurus on Tourism and Leisure Activities (World Tourism Organization, WTO) [40] is a tourism terminology guide at an international level for the standardization and normalization of a common indexation and research language.
- XML Schema Documents (Open Travel Alliance, OTA) [41] provides typical concepts for describing events and activities in the travel sector.
- An extension of GoodRelations (a standard vocabulary for the commercial aspects of offers) [42] termed ACCO provides an accommodation ontology for hotels, vacation homes, camping sites, and other forms of accommodation for e-commerce [43].

3.2.2. Application ontologies, domain ontologies, and taxonomies

In addition to the existing (classification) standards, a large number of application ontologies (or taxonomies) have been developed to cover either the entire tourism domain or some relevant subareas. Representative examples are QALL-ME ontology [44], Hi-Touch [45], DERI e-Tourism ontology [46], TAGA [47], GETESS [48], EON-Travelling [49], OnTour [50], ebSemantics [51], and AuSTO [52]. Most of them provide similar concepts and hierarchies that describe a set of typical tourism objects, such as popular attractions, food and service, accommodation, transportation and infrastructure, tourism events (e.g., music festivals), and tourism destinations (e.g., national park or lake region) or consider only some subdomains. For example, ebSemantics defines individual ontologies for accommodation, event, and gastronomy, each of which is used in different application scenarios. Although standardizing the main vocabulary as well as taxonomies and ontologies is a common requirement, no single actor has the potential to impose a commonly accepted standard vocabulary/taxonomy/ontology due to its degree of complexity [35]. In the EU project HarmoNET (Harmonisation Network for the Exchange of Travel and Tourism Information), the project partners defined the Harmonise ontology [53,54], addressing interoperability problems in the tourism field with a focus on data exchange. The Harmonise ontology (IMHO) includes concepts for events and

[7] https://jena.apache.org/index.html [December 07, 2015].

accommodations in order to allow modeling and saving concepts of transaction data. The tourism harmonization network additionally provides mapping rules for transformation. Nevertheless, Harmonise did not achieve the necessary level of acceptance and therefore did not evolve into a standard in the tourism domain.

3.2.3. Tourism ontology and taxonomy design issues

Despite these standards and ontology applications as well as attempts to achieve a unification of tourism concepts (cf. Harmonise ontology), there are still open issues. Whereas in other application domains a simple taxonomy or a vocabulary is sometimes misnamed "ontology," in the tourism domain existing true ontologies do not pass the validation and verification phase in many cases because they have specification deficits, for example, concerning domain and range or first-order logic definitions required for reasoning. Most of the ontologies mentioned are individual application ontologies that are tightly coupled with the source code of the application and therefore cannot be used directly as a shareable model in the semantic web. This weakness also hinders further development and re-engineering processes in the ontology engineering life cycle.

The deficits of application ontologies in tourism may result from incomplete ontology engineering phases and insufficient ontology evaluation. For example, competency questions that should be answered by using the ontology are not considered to a sufficient extent. Assuming that ontologies are misused for individual application-specific solutions while their important benefit of representing a consensus of a large number of stakeholders is ignored, the question arises of whether the tourism domain really needs ontologies.

1. *Are ontologies necessary to cope with interoperability problems, as is the intention of Harmonise?*
   Yes, but there are many challenges that make the unification of a vocabulary difficult: (i) existing cultural and linguistic differences, for example, hotel categorization expressed by different star rating systems in different countries and (ii) tourism organizations defend their interests represented in customized taxonomies and ontologies. In addition, there are country-specific ratings, such as flowers (one to five) to rate Austrian homesteads, and gentians (one to five) for private rental holiday flats. The heterogeneity of the tourism sector makes the process of developing and maintaining a common tourism ontology that covers the entire tourism market, including geographical-, temporal-, and user-related information, very tedious, and requires an agreement on a shared vocabulary between the different tourism organizations and their key players. Hence, in order to cover the semantic space of the tourism domain and to facilitate interoperability between the different tourism services, a *set of ontologies* may be required. However, these ontologies should not exist autonomously and independently but should be complemented by a core domain ontology, as proposed by Stuckenschmidt and van Harmelen [55]. In detail, the core ontology should contain the common vocabulary of the tourism sector and should be extendable in a modular way by other ontologies, for example, for modeling time, location, or user context. Further information concerning the approach of modularized ontologies in the tourism domain is given by Feilmayr et al. [36].
2. *Is there a requirement for personalization ontologies (trip planning, recommendation) such as TAGA, GETESS, Hi-Touch?*
   This question immediately raises the follow-up question of why machine-learning algorithms (such as the k-nearest neighbor approach or collaborative filtering methods that are based on collecting and analyzing large amounts of information about user behavior) are not considered sufficient. A demand for such ontologies exists, but in many cases the aims of tourism projects are reached more efficiently if – in place of an ontology – well-established methods are applied such as data mining methods (i.e., classification and association rules, which are also well-established on textual data), information retrieval methods that provide flexible access to information, and several well-tried mathematical models (e.g., models for fuzzy retrieval, the vector space model for representing information, latent semantic indexing/analysis for dimension reduction and efficient feature selection, or latent Dirichlet allocation for classifying and identifying new, previously unknown information in documents).
3. *Is there a requirement for ontologies that supporting complex search queries such as ebSemantics, DERI e-Tourism, and QALL-ME?*
   On the one hand, ontologies support the production of more semantically enriched and accurate web content and thus also of query results, but on the other hand this approach requires initial content preparation, which is time- and cost-intensive. First, information provided on different web sites must be annotated with ontology concepts, and due to heterogeneous resources, wrappers must be developed for each resource. Second, ontology mappings must be defined, and third a standard vocabulary for the tourism domain must be established.
   Search engines benefit from ontologies regarding term matching (cf. preparing data for the normal web) and retrieval of hidden content in the "deep web," which is not indexed by standard search engines. Further, applying, for example, thesauri to imprecisely specified queries with unclear intention leads to more accurate results than non-semantics-based searches, which are more likely to retrieve false positives and negatives.
   Despite their advantages, efficient indexing methods and common mathematical retrieval models do not guarantee any depth for a search unless the deep search terms are included in the content at indexing time. They will not be fully substituted by ontologies due to the initial preparation effort and complexity of the ontology approach.

The tourism domain is ideally suited to gain benefit from the potential of ontologies. Considering the indicators mentioned in the introduction, the tourism domain has a strong requirement for sharing and semantic expressiveness, it is characterized by a high degree of complexity, and the community is large. As highlighted in the case study, the size of the community is an obstacle to the specification of a tourism domain ontology due to national or cultural differences. The star categorization of hotels, for instance, is not a technological barrier but a business conflict because it is hard to convince hoteliers of one country to specify their 4-star hotels as being equivalent to the 3-star hotels of another country. The Harmonise project [53,54] is a representative example of a large, long-term investment that failed to align national or project partner interests on a larger scale. Nevertheless, modularized ontologies and the application of machine-learning methods for a higher degree of alignment appear promising.

3.3. Case studies—conclusions and lessons learned

The selected case studies represent two extremes: a closed-world industry and the mainly open-world tourism business. In the manufacturing industry, semantic integration has low priority because intra-company integration is based on structured information in a mostly closed world. The degree of sharing corresponds to the level of need for semantic expressiveness. There is no real requirement for sharing except in group companies, which could develop a use case for ontologies in the near future. Without the additional requirement to align concepts with a larger community, ontological representations risk being faint shadows of existing business practices. Ontologies in this use case are neither visionary nor agents of change and, unless mandated at a higher level, may not see extensive use or be maintained or developed. This will not provide strong motivation for widespread adoption of ontologies.

In contrast, in a mainly open world such as the semantic web or tourism web sites, ontologies support shared understanding in a domain of knowledge that may be used as a unifying framework to cope with interoperability, reuse, sharing, and mismatching terms, as required for business-to-business integration. The tourism industry and some other businesses wish to share data but cannot easily agree on alignment or are so focused on their “unique” approaches that their sharing is limited by the lack of alignment, and multiple semi-clones of the concept representations with little reusability or efficiency are therefore produced. Hence, such business sectors do not promote the use of ontologies.

A further business model, which we have not discussed so far, is efficient and effective ontology generation based on global sharing with an open view to adopting other processes in which the ontologies generated form the basis of a large volume of interactions as part of an active and involved knowledge community. While this ideal is the desired goal, the question arises of where this business, or family of businesses, can be found outside scientific research communities. If the goal is to provide a motivator for widespread industrial adoption of ontologies, it must be accepted that it is the business perspective that is paramount rather than an idealized view of unrestricted global information sharing. Consequently, some perceived benefits of ontologies in association with certain data operations or sharing activities must be promised. Moreover, it must be ensured that designers understand precisely what these are before they proceed.

Table 2 summarizes domain characteristics of the case studies that might be considered when dealing with ontologies in these or similar domains. Barriers encountered due to specific domain characteristics can often be reduced to social issues, for instance, the tourism domain failing to find a consensus or the industry requiring that established schemas remain unchanged. Aside from technical barriers, these social parameters must be considered so that ontologies can be used to advantage. The potential benefits listed in Table 2 address specific domain characteristics and serve as a starting point for a detailed discussion in Section 4.

4. Benefits of ontologies

In order to apply ontologies successfully, they should be used efficiently and for a sound reason. The term ontology is more than a buzzword, and it should certainly be more than an incorrect synonym for taxonomy. The following key questions assist in making the correct decision as to whether ontologies should be used:

- What are the benefits of ontologies as concepts?
- Is reasoning their main advantage?
- What are the benefits gained from applying ontologies?

Benefits cannot be considered solely from a technical point of view, as purely technical arguments are not sufficient to convince businesses to adopt a particular technology. Additionally, arguments must be provided that allow decision makers to clearly identify benefits for their own organizations. User benefits are not merely external perceptions of what users may want; they are grounded in surveys carried out across a target user group of this technology. Comparing benefits due to technical superiority

Table 2
Domain characteristics, barriers encountered, and potential benefits of applying ontologies.

Domain	Characteristics	Potential benefits gained from ontologies
Industry (closed world)	- Closed world, less demand for sharing. - Schemas have already been established. - In most cases, support of the entire process chain is still missing.	- Support internal business communication. - Provide homogeneous, consistent views of distributed information sources. - Automated data transformation, data assignment and combination, and data analysis.
Tourism (open world)	- Large and heterogeneous community with cultural and linguistic differences. - Requires a shared understanding in a domain of knowledge and a unifying framework. - Existing application-specific ontologies (some of which have specification deficits).	- Support alignment and integration of heterogeneous information sources. - Formal basis for establishing recommender systems, inferential statistics, and analysis. - Improve search quality with automatic discovery and personalization.

with what users consider to be important may provide compelling arguments that would lead people to decide to use ontological or taxonomic technology correctly. From a *technical point of view*, applying ontologies may be beneficial for several reasons. The first four benefits listed refine the abstract definitions of ontology benefits by Gruninger and Lee [56], and the remaining ones emphasize important characteristics according to [21,57] and our research experience.

- *Communication.* An ontology enables communication between (i) implemented computational systems, (ii) between humans, and (iii) between humans and implemented computational systems [56] and therefore fulfills the strong demand for reducing machine translation steps. Dual readability is the key justification for the use of formats such as OWL, although new problems may arise, for instance, natural language problems such as unambiguity or the handling of synonyms and homonyms. This characteristic implicitly facilitates *knowledge transfer* [57].
- *Inference.* An ontology enables computational inference, which is useful for deriving implicit facts. Inference allows automatic determination of the class hierarchy, classification of instances, consistency checking within an ontology, and query execution considering rules and axioms [56,58].
- *Knowledge organization. Domain analysis* is necessary to make domain assumptions explicit and to share an understanding of the information structure. Ontologies are also means of structuring and organizing knowledge, not only data [57].
- *Reusability* [56]. Ontologies enable, on the one hand, reuse of domain knowledge and, on the other, integration of a new knowledge caucus built upon existing knowledge. OWL widely supports this task and provides inheritance.
- *Standardization* aims for a uniform language that enables protocols. The implicit bootstrapping problem is that everyone must agree to an initial *lingua franca* in order to be able to standardize around it [5].
- *Identification.* A unique identifier, for example, the Internationalized Resource Identifier (IRI) [59] concept, formerly URI, uniquely identifies the meaning of concepts in a given domain of interest. IRIs enable cross-ontology references, which support reuse and interoperability between ontologies [60]. Furthermore, the existence of IRIs allows *reification* – tying concepts to physical items or real-world concepts. Without reification, there is no real grounding, and thus the ontology cannot be applied to businesses.
- *T-Box/A-Box separation.* Ontologies clearly separate between an ontological schema and its instances. OWL DL requires separation of classes, instances, properties, and data values, whereas in OWL Full classes can be instances or properties at the same time. In addition, it is possible to distinguish between aspects of ontologies that fall within DL boundaries and those that do not [61].
- *Constant evolution.* From a business perspective, ontology stores allow agile schema management during application runtime, which is supported by the graph-based data model, in contrast to relational databases [58].

This list of benefits mainly answers the first question but also reveals that inference (i.e., reasoning) is not the only advantage. However, this does not make selecting an appropriate modeling approach easier because also a taxonomy and an entity-relationship model offer these benefits to some extent (with the exception of inference/reasoning). Considering the key influencing factors for using ontologies as identified in the introduction, the benefits of applying ontologies can be summarized as depicted in Table 3. Taking into account these listed benefits and considering the second question, further benefits of applying ontologies – aside from inference/reasoning – are semantic expressiveness and the ability to transfer complex knowledge. In addition to this technical-centered view of ontologies, user surveys (2006–2014) of academic and industrial research institutes and businesses have identified the following benefits of a *user-centered view* of ontologies.

- *Enabling knowledge sharing* between users (cf. knowledge transfer). Cardoso [62] revealed in his study that the main purpose of using ontologies is to share an understanding of the structure of information among people or software agents, and thus models can be interpreted by humans and computers. Pellegrini [63] showed that users think that the aim of using social software is quick access to information and knowledge. Warren et al. [64,65] confirmed the trend that users want to share knowledge (e.g., between individuals in an organization) and to provide common access to heterogeneous data (e.g., a common schema for data access)
- *Making domain assumptions explicit* (cf. domain analysis) [62] and thus offering knowledge base schemas for storing and retrieving information [64,65].
- *Providing a domain specification as a result of an intensive domain analysis* (cf. domain analysis) [62], similar to conceptual modeling, e.g., defining a domain formally [64,65].

Table 3
Main benefits of ontologies.

Benefit	Explanatory notes
Sharing principle	Even though sharing is also covered by taxonomies, the complexity of shared and transferred knowledge supported can be increased significantly by the strength of the semantic expressiveness of ontologies.
Semantic expressiveness	All other known models, such as taxonomy and entity-relationship model, are limited in their semantic expressiveness. Ontologies allow more expressive models to be designed to capture complex real-world concepts by considering their semantics.
Complex models	The definition of ontological relationships, alongside other specific ontology design features such as reasoning, enables the creation of complex and expressive models.
Size of sharing community	Complex models are provided for a broader range of applications, and the *size of the sharing community* (ontology stakeholders) is less restricted than, e.g., for entity-relationship models.

- *Clear separation of operational and domain knowledge* [62].
- *Enabling knowledge reuse* (cf. reusability) [62]. Simperl et al. [66,67] discussed in detail ontology reuse issues in ontology engineering with an inconsistent result. The inconsistent and contradictory result of these studies is that when an ontology is created, expectations and notion of ontology reuse are often associated with informative materials consulted by domain experts. Knowledge engineers often expect domain specific documents consulted by experts as starting point for ontology reuse. This is contradictory to the original purpose of ontology reuse, which aims on the integration of other ontologies or ontology components.

The main advantages of applying ontologies from a user-centric viewpoint can be summarized as (i) intensive analysis of the users' domain in order to (ii) subsequently share the structured knowledge gained within the user community.

4.1. Choosing the most appropriate model

The differences between the benefits of the technical and the user-centric point of view reveal some interesting aspects that explain why ontologies are in many cases misused. Hepp [21] cited multiple reasons why individuals refrain from building ontologies, including unresolved technical limitations, and stated more precisely by Hepp [16]: "It must be said, though, that the broad promises of the early wave of ontology research were too optimistic, because the advocates had ignored the technical difficulties of (1) providing ontologies of sufficient quality and currency, (2) of annotating source data, and (3) of creating complete, current, and correct mappings – and did mostly not compare the costs and benefits of ontologies over their lifespan." Fig. 1 builds on knowledge obtained from different literature sources discussed in the previous paragraph (cf. [5,16,21,33,56–58,62–68]), subsumes both the technical and user-centric views, and shows their intersection.

It is particularly noteworthy that inference, a technical key benefit, is not in the list of the identified main advantages. The reason for this may be the inherent complexity of the entire area of reasoning – ironically, the cognitive load imposed by considering the complexity of a technology that would reduce cognitive load constitutes a barrier. Users would have to be aware of the complexity of the reasoner, its technology, and the ability of a reasoner to substitute for missing experts in this area. Due to the frequent misuse of ontologies as a taxonomic tool, their capacity for inference is rarely used.

The other benefits identified, such as the complexity and the semantic expressiveness of the models, are apparently not clear to the user. The two different views of ontologies can therefore be attributed to the fact that the definition of the term ontology is not clear, which results in different perceived benefits from using ontologies in businesses/applications. Businesses obviously want to achieve maximum profit when using/providing ontology-based applications/services. In order to achieve this ambitious goal, the following points – which should ultimately result in appropriate applications/usage of ontologies – must be considered, and, moreover, the (user) requirements must be fulfilled so that the full benefits of ontologies can be realized.

4.1.1. Expressiveness versus decidability

A survey of published papers on ontologies revealed that a large proportion deals in fact with taxonomies (cf. tourism case study). Many of the so-called ontologies are not validated, fully expressed, or utilized in the appropriate manner. When true ontologies are used, discussions often revolve around the choice of ontology language in terms of expressiveness versus decidability. However, the decidability limitations of ontological representations such as OWL Full are not relevant if the ontology is not intended for reasoning or inference. Thus, if the need for expressiveness or decidability is understood in the context of the required reasoning support level, it is possible to successfully navigate the requirements and arrive at a sufficiently rich and expressive taxonomic language for a given project.

4.1.2. Accuracy and complexity

However, if decidability is truly required, expressiveness is strongly limited, and it is very difficult to transform ontologically limited systems into complete ontological frameworks. Given that the goal is the explicit capturing of a domain, its entities, and their relationships, any movement toward a partial solution that is less than truly expressive risks reducing model accuracy. We propose a cartographic analogue for illustration: Consider the path taken to move from one city to another, for example, from Linz to Vienna. There are many maps that could be used to locate these cities and the paths between them. The highest-level representation is, of course, the globe itself (cf. Fig. 2). To gain a two-dimensional representation suitable for determining distances, a

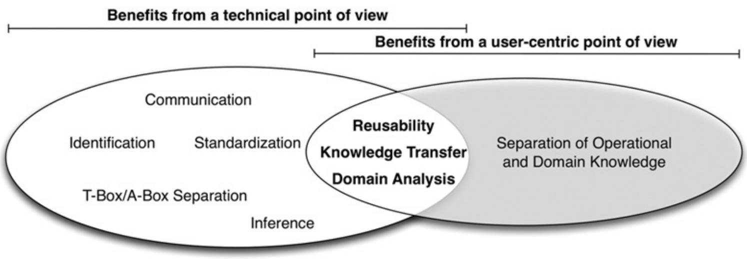

Fig. 1. Venn diagram of the technical- and user-centered views.

Fig. 2. Increasing the complexity of the model to achieve the required accuracy.

Mercator projection may be applied. This simple transformation into a two-dimensional model introduces well-known distortions to the surface of the globe. If the goal is to drive from Linz to Vienna, this projection is still not sufficiently accurate and requires a road map with structures overlaid on the 2D projection. If hiking is the preferred activity, then further details are required; trail maps for those areas where footpaths were not identified as part of roads must be provided. At each stage of this process, the level of detail increases, as does the complexity of the model, until the desired level of detail is reached. Generally, to hide real-world complexity, abstraction mechanisms of models are used which, in this case, visualize the complexity of atoms and atomic interactions that provide the paths upon which to walk or to drive at a level of detail appropriate for navigation.

A desire for accuracy must be tempered with a willingness to initially invest the time required for producing an accurate model. Using an overly complex model, for instance, trail maps for a driving trip, is inefficient, as the level of focus may be wrong, which leads to inefficient transitions from one part of the model to another. The trail example is pertinent here, as trail maps cover 20–30 km, rather than the 100–300 km scale that is needed to drive from Linz to Vienna. Hence, the desired accuracy of the model must be synchronized with the level of detail required.

What impact does the careful consideration of expressiveness, decidability, accuracy, and complexity have on the study of ontologies and taxonomies? If users do not correctly define their modeling requirements, then either the model is too simple and captures too little or it is far too complex and wastes resources for no real benefit, which ultimately results in misuse of the models available. In computational terms, a complex model is far more daunting and may not be tractable with the technology available. Again, when a system is modeled using an ontology or taxonomy and the full expressiveness is not captured in the first pass, a less expressive system cannot be used to reconstruct the master domain, but the knowledge engineer must resurvey the original. A domain analysis and a detailed conceptual design phase reduce the probability of existing models being misused and of a misleading design of the domain (assuming that the level of expressiveness is chosen appropriately). Inaccurate expressiveness may lead to omission of important details or wasting time on trivialities.

Data modeling [69] faces the same challenge. If in the conceptual design phase modeling requirements are insufficiently defined, it is nearly impossible to build a model at an appropriate abstraction level. Correct use of the abstraction concept requires long-term experience and expert knowledge. For example, there are two ways of abstracting a hotel room: A more abstract view is booking (expected date of arrival/departure, number of persons, extra bed for kids y/n, pets y/n), where a hotel room is part of a process. A more specific abstraction is the inventory view, where each individual piece of furniture and interior equipment (e.g., mini bar, TV, hair dryer) is relevant to the data model, resulting in a more realistic description of a hotel room. The outcomes are two different abstractions, which are two different models of the same real-world object "hotel room." An appropriate abstraction level is also crucial for ontology design: There is no reason to reinvent the wheel; we recommend the use of data modeling as best practice for ontological modeling.

4.2. Realizing the full benefits of ontologies

This section addresses the question what is required to realize the full benefits of ontologies. Users are not always aware of the differences between existing technologies (e.g., between ontologies and taxonomies, as the tourism case study indicates), or at least they are not able to adequately express or understand the differences between these technologies the way they are currently presented [27,28]. To improve user understanding, the following factors assist in identifying ways to increase the ubiquity of ontologies and to satisfy user requirements of ontologies, which will lead to greater acceptance.

- *Clear definition*. Inference mechanisms, semantic expressiveness, and model complexity are the most important benefits of applying ontologies. Thus, the community must develop both understanding of and practice in the application of ontologies in order to recognize these aspects as obvious benefits. Accordingly, the benefits must be clearly reflected in the common definition of an ontology.
- *Specification and design methodology*. Data modeling and software engineering offer a specification and design methodology that is suited to providing solutions that are sufficiently intricate without being overly complex. When, for instance, a software engineering design becomes too complex, design documents become overly complicated, pathways become choked and hard to manage, and the risk of project failure increases. Similarly, ontological engineering might attempt to produce an overly complex solution because the initial focus on the necessary requirements is missing. The previously discussed case study from the tourism domain (cf. Section 3.2, tourism ontology and taxonomy design issues 1 and 2) clearly shows that it is difficult to establish a unified vocabulary while taking into account different interests. In combination with inappropriately applied technologies, modeling complexity increases significantly. This can also be inferred from the ontology engineering user study carried out by Simperl and Tempich [66].
- *Visibility*. When the underlying technology is not transparent, users must be trained to work with the visible framework. A successful ontological basis for a project provides all advantages of semantic classification without the burden of users having to interact with the ontological underpinnings.
- *Experts*. Consultation of experts is an essential domain-modeling step; consequently, if a key expert is missing, only a partial model can be formed.

Based on the user requirements discussed, we extend the definition of Studer et al. [1] in order to highlight the main benefits. Thus, an ontology is a formal, explicit specification of a shared conceptualization that is characterized by high semantic expressiveness required for increased complexity.

An application of an ontological engineering method/methodology is essential to increase the chances of success, as is standard in data modeling and software engineering (cf. Section 6). Prior to ontological engineering, selecting an appropriate model for a specific use case is important. Section 5 therefore discusses the fundamental differences between current models and classifies them along the dimensions expressiveness, requirement for sharing, complexity, and representation, and finally answers the questions of when and how a specific model should be used.

5. Guidelines for model selection

Providing guidelines to support the decision for the most appropriate model requires discussing the differences between existing modeling languages and modeling concepts. Several approaches to classifying types of ontologies have been introduced (cf. [16]), and various models have been discussed [70–72]. Hepp [16] classified ontology projects – but not ontologies – according to the following six characteristics: (i) expressiveness, (ii) size of the relevant community, (iii) conceptual dynamics in the domain, (iv) number of conceptual elements in the domain, (v) degree of subjectivity in a conceptualization of the respective domain, and (vi) average size of the specifications per element. Since this publication also discussed fundamental problems in building and using ontologies in business applications, there is a strong focus on ontologies themselves as modeling languages.

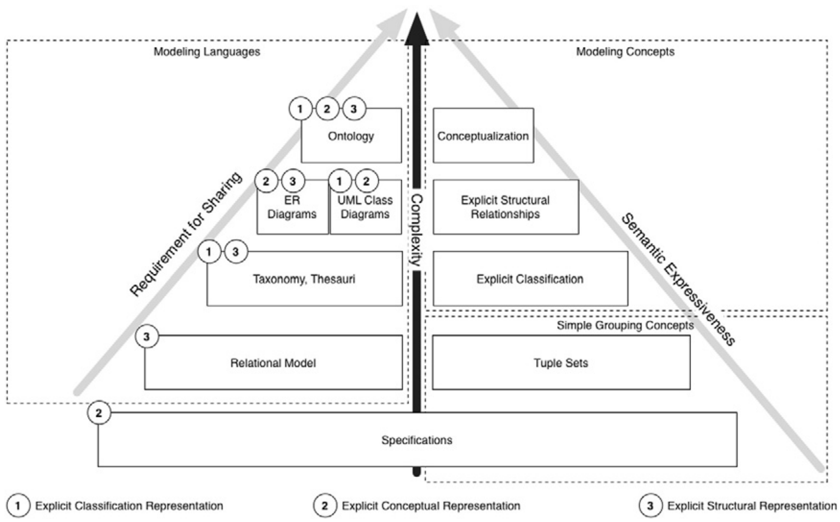

Fig. 3. Relationship and representation classification of existing models.

However, the author provided no recommendations for alternative technologies (or modeling languages/concepts) for the case that ontologies are not an appropriate solution to the problem.

For this reason, we introduce a more abstract classification that characterizes differences in technologies. By any definition, taxonomies deal with structural relationships and ontologies focus on conceptual relationships. Drawing a pyramid-shaped figure with these two relationship criteria, as shown in Fig. 3, provides a general classification of technologies. Being the most explicit representation of the classification and structural relationships available, ontologies are at the top. Specifications which do not have a machine-interpretable representation or explicit structure or classification requirements are arranged at the bottom level. Concept hierarchies with sub-classifications are represented by taxonomies, similarity, and synonymy relations are represented by thesauri, and explicit structural relationships combined with object-oriented classes can be represented by UML.

While this classification initially appears to be suitable, an explicit conceptualization, which is a characteristic of strong data models, is missing. It is not sufficient to express that, since concept c_1 is similar to concept c_2, and therefore c_2 can be classified into a specialization of c_1, as this results in a sub-classification that could be represented by a taxonomy or an UML class generalization. Expressing conceptual relationships, which form the basis for developing an expressive ontology, must additionally be enabled. Thus, besides the classification of model languages and model concepts, a layer for explicit representation is required. The classification of different technologies into *explicit conceptual*, *explicit structural*, and/or *explicit classification representation* is indicated by numbered circles in Fig. 3.

Fig. 3 shows the relationship between existing technologies in three dimensions: the requirements for sharing and semantic expressiveness (grey arrows) and model complexity (black arrow in the center). Further, the figure clearly separates the languages on the left-hand side from the concepts on the right-hand side, which are subdivided into simple grouping concepts (bottom right in Fig. 3) and true modeling concepts, which extend these simple groups with relationship specifications and more mature conceptual arrangements. The overall classification recognizes the discrete nature of the technologies that must implicitly be accepted in order to choose one over another. Additionally, the classification indicates the level of effort required to cope with the levels of complexity (ontologies require more effort). However, the benefit is that the knowledge exists as a consistent and rigorous structure in a conceptually explicit form that can be shared and is semantically expressive rather than as a purely structural requirement. Given the additional overheads required for semantic expressiveness and rigor in contemporary ontology languages, ontologies require far more effort both structurally and intellectually to truly capture the system knowledge than does capturing the same structure taxonomically. While the effort required to achieve a given level of modeling cannot be explicitly quantified without further research, a list of considerations is provided that should be addressed to assist in selecting the least complex and most expressive technology required for a given application:

- *Degree of sharing.* Ontologies offer most benefit when assisting in the sharing and reuse of representations between a larger number of organizations. Outside of company borders, there is no easy way to communicate disparate requirements and align different organizations without an ontology. Thus, as the need for sharing increases, the likelihood that ontological engineering is required also increases.
- *Number of objects.* With a small number of objects in the use case, the effort spent on a complex modeling process is unlikely to be rewarded with any cost or resource savings. However, managing large numbers of objects is complex and probably involves a large number of relationships that come with a variety of conceptual modeling problems. A growing number of objects increases the likelihood that a greater investment of effort in a more complex modeling solution will be rewarded.
- *Degree of reuse.* The effort invested in modeling is not trivial and grows as progress is made up through the complexity layers. If existing modules can be reused several times, the costs of producing a more complex system can be decreased.

Based on our definition of ontology given in Section 4.2, the combination of ontology characteristics allows a targeted selection of an appropriate modeling language for a specific problem. The classification in Fig. 3 and the following requirements are guiding principles for the modeling language decision.

- *Sharing.* Evaluation of the requirement for sharing.
- *Complexity.* Evaluation of the complexity level and the number of objects required.
- *Semantic expressiveness.* Evaluation of the desired degree of semantic expressiveness.
- *Reuse.* Assessment of the demand for reuse.
- *Representation.* Selection of an appropriate representation (classification, structural and/or conceptual).

The framework for measurement and comparison provided allows justifiable selection of one technology over another and answering the question of when ontologies should be used and how they can be used. In summary, the ideal candidate for ontological development is a highly shared, large, and re-usable system. This is, for example, the reason why ontologies are so popular in bioinformatics – genome ontologies meet all of these requirements [20].

However, the increased model complexity necessitates a defined and, in particular, a structured ontology design process that is easy to understand for both experts and non-experts. Thus, aspects critical to the modeling of ontologies must be examined. In the next section, ontological engineering is compared with other modeling processes, especially software engineering, to identify the benefits of these approaches and adapt some of them for ontological engineering with the aim to increase user acceptance. Further, the next section discusses reasons why reuse is practiced widely in software engineering but not in ontology engineering. In order to address this shortfall, we introduce reference ontologies, and propose a new procedure for reusing existing ontologies.

6. Ontological engineering supported by reference ontologies

A review of important facts in the context of ontologies reveals that many other modeling approaches, such as entity-relationship modeling [24] and UML [23], also enable analysis, structuring, and organization of a domain of interest [73,74]. Since in software engineering structured procedures are indispensable for software development, ontological development also requires an appropriate procedure which – provided that a suitable model is selected and correctly employed in the engineering process – leads to a formal, correct, and satisfactory ontological model. These similarities allow ontology engineering not only to be compared to conventional conceptual modeling techniques but also to software engineering approaches.

6.1. Comparison of ontological engineering with conventional conceptual modeling

User surveys [16,21,57,62,66] not only summarize the benefits of ontologies from a user-centered view but also identify obstacles to modeling and using ontologies, which are as follows:

- High complexity of the modeling process and the models themselves
- Sufficient time and effort must be spent on modeling the domain knowledge, especially on considering reasoning and prediction aspects in the design
- Increased demand for project resources (domain experts, knowledge engineers, and ontology experts, which are often not available)
- No or little tool support in guiding the design process of the knowledge framework,
- Knowledge and awareness of structured procedures is almost non-existent.

Other data modeling approaches, for instance, for modeling a data mining application, must also deal with challenges that require an in-depth analysis of the domain and therefore investment of time and human resources, including domain experts, knowledge engineers, and data mining experts. This applies equally to establishing ontologies, as can be derived from the surveys of comparative resource investment required to implement ontological and conventional data modeling solutions, such as entity-relationship diagrams [24] and UML class models [23]. Because these challenges are similar to those of conventional data models, the modeling approaches of the latter can be analyzed and compared to existing ontological engineering processes documented in various literature sources [33,56,70,75,76]. This analysis enables the identification of the main benefits of conventional data modeling approaches. We adopt the benefits of these approaches in order to overcome the weaknesses of current ontology engineering approaches. In this context, two characteristics are particularly important: the simplicity and the minimalistic representation of the models themselves.

6.1.1. Simplicity

One main advantage of entity-relationship modeling is that the models generated are easy to understand, and therefore they can aid communication between several technical and non-technical stakeholders in a company. Quality criteria such as expressiveness, easy applicability, minimalism, and a theoretical basis are the reasons why the entity-relationship model has survived since 1976. For ontological design, no conceptual model similar to the entity-relationship model is available [77]. When the benefits that could be gained from them are not understood, ontologies risk being seen as too abstract to be implemented and too distant from useful modeling. However, the modeling process itself must also be easy to understand and couched in terms that both technical and non-technical users are able to understand easily.

6.1.2. Representation

In ontologies, the focus is on a knowledge framework that classifies new entities of known taxonomic components and extends – in an internally consistent manner – new entities that were not defined at the original time of writing. Beyond that, ontologies provide, in contrast to conventional modeling techniques, (i) a consistency check, (ii) inference of new classes (reasoning), and (iii) a predictive analysis (prediction). A simple notation such as the Chen Notation for entity-relationship modeling [24] would probably increase use in business scenarios. We have therefore started to develop a graphical notation for ontological engineering, which includes representations for classes, data type properties (similar to attributes of an entity), object properties (similar to relations between entities, but with an explicit direction for determining domain and range), and class axioms/restrictions. In addition, simple rules for transforming the ontology model into a formal ontology language (such as OWL DL, RDF, and Manchester syntax) are proposed. These simple rules are similar to the transformation rules required for transforming an entity-relationship model into a relational model as used in the field of relational databases.

Factors in increasing user acceptance are therefore simplicity of the modeling language and an easy-to-understand representation of the domain knowledge to be modeled. That ontologies are not yet ubiquitous is thus not only due to an unsuitable selection of modeling technology but is rather caused by (i) lack or unavailability of knowledge about (the handling of) ontology modeling processes, (ii) lacking support of modeling processes due to missing state-of-the-art modeling tools, and (iii) missing long-term experience (cf. entity-relationship modeling has been more or less state of the art since Chen [24] introduced it). Surveys clearly show that (i) the majority (60–80 %) do not use an ontology engineering methodology for modeling their ontologies, (ii) only some of the development environments, such as NeOn,[8] support a design methodology for ontologies and (iii) the

[8] http://www.neon-project.org/nw/Ontologies [December 07, 2015].

benefits of using a methodology are obviously not clear to all [62,63,66]. Incorrectly applied software engineering methodologies can lead to poorly designed and hard to maintain software. Misusing or not using an ontological engineering methodology has a similar negative impact, resulting in overly restricted or complex ontologies for a specific knowledge-based application. Furthermore, ontological modeling would become more popular if well-established procedures were provided in the process model, as it is the case in software engineering models. The next sub-section compares ontological engineering methods with software engineering methods to determine possible reasons for the low acceptance of state-of-the-art models for ontological engineering.

6.2. Comparison of ontological engineering with software engineering

Gómez-Pérez et al. [70] published a comprehensive work on ontological engineering methods and methodologies. Although tutorials have been carefully constructed to lead by example, ontological engineering is not a well-defined discipline because insufficient effort has been made to communicate with non-technical users. Software engineering, in contrast, is a well-established discipline adopted in practice that has changed the way software is produced worldwide. The question arises whether software engineering experience can be adapted for ontological engineering. There are two promising areas where an ontological engineering discipline can be of assistance.

- Bridging the gap between the model and its application.
- Supporting ontology reuse, thus taking into account their resource-intensive production process.

To put this into context, Fig. 4 illustrates the similarities between software and ontological engineering methods/methodologies, in particular between the waterfall model, (a sequential software development process that was one of the first software engineering approaches) and the approach by Uschold and King [76] (the first method for building ontologies, introduced 25 years after the waterfall model). Fig. 5 shows the V-model [78–80] as an extension of the waterfall model whose roots can be traced back to the 1980s. The right side of the V represents the integration of parts and their verification, which leads to the implementation. In the ontological engineering area, METHONTOLOGY [75], a methodology for ontological development and all-encompassing representation of a development lifecycle, was developed by the Ontology group Universidad Politécnica de Madrid in 1997. Similar to the V-model, METHONTOLOGY also includes verification and validation activities termed management and support activities, although it was developed years later.

Ontological engineering also provides contemporary approaches that are similar to agile software engineering methodologies. One example is eXtreme ontology design (XD), the main principles of which are as follows:

- Understanding the task and expressing it by means of competency questions
- Reusing solutions, such as ontology design patterns,[9] to evaluate the result against the task

Whereas software engineering has established well-defined models and well-known use cases, these exist to a lesser extent – or do not inspire the same confidence or discipline – in ontological engineering. The tourism case study shows how ontologies are used inappropriately and ineffectively due to this lack of application knowledge and a proper discipline. The solution can be found in supporting the conceptual modeling process, including accompanying documentation of best practices, in order to eliminate the following concerns of the industry regarding use and deployment of ontologies.

- *Missing powerful applications.* The community needs a powerful application that works on top of existing systems and, whenever possible, does so with minimal additional work for the user. Powerful applications need to be attractive to the largest possible set of potential users.
- *Missing business applications*. Ontologies are more often applied in research prototypes than in commercial applications [16,21,63].
- *Missing cost-benefits analysis*. Persuasive drivers should convince businesses to adopt ontological engineering.
- *Quality of ontologies*. Evaluating the correctness of ontologies must be possible. Ontologies must be reliable in terms of classification and consistency of the relationships that they provide.

6.2.1. Bridging the gap between a model and its application

Choosing the most appropriate model, be it an entity-relationship model, an UML-based model, or an ontology, is a key decision that has a significant impact on the resources required and the success of the final outcome. Ontological engineering must provide a clear decision process for selecting an appropriate model, methodology, and tool set to meet user requirements with the most efficient use of resources. A conceptual model cannot be implemented directly. This is frequently ignored in the development process, where the model is defined and also populated by using one of the established tools while remaining strictly conceptual on a formal level. However, in a subsequent step the model must be transformed into an implemented application. Ideally, ontological engineering should have a corresponding set of tools (or guidelines for tool production) that lead to semi-automated translation of models into applications.

Almost every ontology engineering method or methodology comes with a specification of its proposed conceptual modeling step [70]. However, in contrast to in comparison to the conceptual modeling phase in relational database engineering (using

[9] The term "design patterns" is borrowed from the area of software development. Design Patterns by Alexander [81] vs. Ontology Design Patterns by Presutti and Gangemi [82] as part of the NEON project.

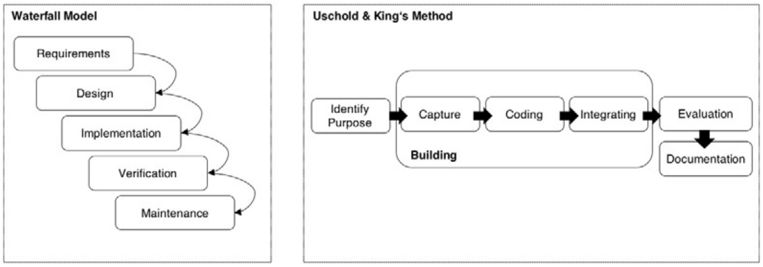

Fig. 4. Waterfall model compared with the first ontology engineering method.

the entity-relationship model), state-of-the-art ontological engineering tools neither support this conceptual phase nor provide some quasi-standard notation for visualization as entity-relationship modeling does. The machine-interpretable nature of a set of modeling solutions naturally leads to automated application-development tools or integrated development environments, which can greatly assist developers.

6.2.2. Reuse in ontological engineering

Reuse is one of the key principles in software engineering. All the effort expended to build a component should not be wasted and components reused as often as possible to achieve the maximum benefit and to minimize costs. The following reasons why reuse is not a dominant driver in ontological engineering are identified.

- Existing ontologies are often built for a particular purpose from the perspective of a single business. Many businesses consider their processes and focus to be unique. Hence, ontology development is carried out in isolation, with each business reinventing the wheel, as shown in the manufacturing industry case study. At the same time, naïve ontologists may search for a central conceptual exchange or "world ontology." There is no "world ontology," but in a shared domain of interest, there is sufficient motivation and vocabulary to establish highly reusable ontologies.
- Existing tools favor small-scale, application-oriented ontologies [83].
- Lack of tools and concerns over return on investment tend to get in the way of building large, comprehensive ontologies. People build what they need now and at the lowest acceptable level of expressiveness rather than create a representational model that can be further extended in the future to cope with new requirements [67]. A similar behavior can be observed in software engineering approaches, especially in agile development methodologies.
- In some cases, people are convinced of the value of their particular representation of data or want to develop an innovative approach; thus, their ontologies must be built from scratch using concepts that make sense within their conceptual context, but without considering pre-existing solutions. This behavior is discussed in the tourism case study.

Thinking about reuse leads naturally to the question of how ontologies are developed. Up to 80 % of survey respondents were not using an ontological engineering methodology [62,63,66]. Ontology reuse has already been identified as a problem in many cases because there is no real mechanism for detecting relevant reusable ontologies. Ontological repositories have been established, such as the Watson semantic web explorer,[10] which is also available as a plug-in to the NeON ontology development environment. However, Watson search is restricted to known and shared ontologies and provides only a keyword-based search mechanism. Even with format-aware extensions to enable keyword search in different syntactic constructs, Watson has two major drawbacks. First, an ontology is identified only if the keyword matches with the keyword annotated by the ontology creator, and second the required ontology must be made available in advance. Especially when dealing with complex ontologies, which are expensive to create appropriately, reuse gains a significant return on investment. Nevertheless, less than 50 % of new ontologies are reused within the sets of existing ontologies [66,67]. While an ontology built for highly specific purposes may not be reusable, many ontologies, including those described in the case studies, could be reused but are not. Missing collaborative tools for ontological development can lead to the building of smaller and more application-oriented ontologies at the expense of standardized (and comprehensive) ontologies. Of course, with a shared domain of interest comes a shared requirement to establish a vocabulary. In order to propagate reuse in ontological engineering, similarities to software engineering should be identified and adapted to the specific aspects and demands of ontological engineering. In both software and ontological engineering, there is an overhead in reusing components, even if they can be located easily. This increased workload can be explained by the following three factors.

1. The time it takes to find the component in some kind of repository.
2. The time it takes to integrate the component with the new software/model.
3. The time it takes to convince all interested parties that the new problem is not so special or unique that the software/model must be developed from scratch.

[10] http://watson.kmi.open.ac.uk/WatsonWUI/ [December 07, 2015].

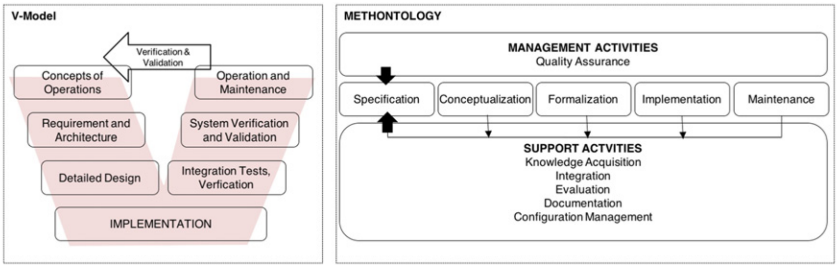

Fig. 5. V-model compared with an ontology engineering methodology.

As mentioned above, the Watson ontology search engine requires the correct (matching) search term in order to find an ontology or component and to put it at the top of the result set. The considerable time taken to do this is wasted if the component is not found (cf. factor 1). A common business and software development view is that processes employed within a given organization or program are somehow superior to – or different from – those of its competitors. Thus, factor 3 may be surprisingly important, and the motivation for widespread reuse wanes if the reuse burden is too high.

Reuse is practiced in software engineering but not yet in ontological engineering. This may be because languages used in software engineering have an extensive community or commercial support, in-built support for modular development, and rich and extensive application programming interfaces (APIs). These APIs provide well-controlled access to language-specific libraries and the libraries, in turn, form the core of a strong reuse platform. For example, very few Java practitioners will write their own linked list implementation unless required for performance reasons, as one is defined in the API. Similarly, there is no need to rewrite core mathematical functions for most modern languages because these are also provided in the libraries and accessed through the APIs. Software engineering has the advantage that reuse components are automatically available to any developer who uses a given programming language. Ontological engineering does not have a notion of the simple API signatures that facilitate a shared library as in software engineering. A partial analogue exists in ontological engineering, as the well-defined ontologies and namespaces that define the standards of OWL, RDF-S, RDF, XML-S, and XML provide a common core of understanding. However, no single API exists that is compatible with all ontologies. Expressing a given concept requires locating the correct ontology, mandating its use and sharing it correctly.

To address this shortfall, using a set of API-equivalents in ontological engineering, termed *reference ontologies*, is proposed. Reference ontologies are shared domain ontologies to which other ontologies can be mapped and which are (or can be) themselves related to other reference ontologies. A reference ontology should help knowledge engineers to select an adequate and suitable ontology for the application they have in mind [84]. In software engineering a similar approach, termed reference models, can be found [85]. Reference models are often applied in financial management and administrative organization software and provide a straightforward approach to maximizing reuse of the correct implementation and minimizing the effort required to meet the specification. The process comprises the following steps:

1. Assume that all businesses follow a similar model and build a standard software system that uses this assumption to provide the tools this standard business will need.
2. Prepare for the eventuality that a specific business will need existing models to be reengineered to provide an exact match to their business processes.
3. Employ programming staff that will perform the required customization on demand.
4. Tailor the software to the business, reusing most of the existing code base.
5. Finish implementation, scan the new code for anything useful that can be reused elsewhere, and integrate this back into the core product.

This model is an excellent representation of profitable software engineering. For ontologies, we propose an analogous process where reference ontologies are established for a number of domains, highlighted in Table 4. From a top-down view, such

Table 4
Adapted engineering process for software reuse in ontological engineering.

Ontology reuse process step	Description
1. Recognition of similarities	Assume that certain key areas in different domains can be represented similarly, and then build a set of reference ontologies that describe these key concepts or families of concepts. Initially, reuse is supported.
2. Generation of clear concepts	Different views of specific businesses or application areas may require rearrangement of concepts. It must therefore be possible to decompose any high-level concept into smaller, effectively indisputable concepts.
3. Process for customization	Develop a process for customization that allows rearrangement of reference ontologies in order to compose a new ontology and promote the existence of this new reference ontology. The result is an ontology view similar to encoded database views.
4. Customization	Customize the reference ontologies by using extensibility and mutability support within the existing ontology language to encode process-specific information. The final result will probably not be a reference ontology, as some of these additions will be too specific to reintegrate.
5. Integration of modifications	Finalize the implementation and integrate any useful modifications back into the parent reference ontology.

reference ontologies can be regarded as a common agreement on sets of statements that can be asserted about the nature of the knowledge they describe, similar to the software engineering concept of patterns. From a bottom-up view, small reference ontologies may be considered to be accepted building blocks for larger reference ontologies. The ontological engineering process, introduced in Table 4, depends on sharing, and given the issues concerning businesses and proprietary information, it must be accepted that while some businesses will share their ontologies and their modifications, step 5 “integration of modifications” may not be a universally accepted step.

In this section, entity-relationship modeling for conceptual database design and software engineering for software design are discussed and compared to ontological engineering. Both approaches are widely accepted by a large community and are recognized as being valid and standard practice. The goal is to remove the obstacles that are currently preventing ontologies from being applied by a broad community, as is standard in entity-relationship modeling and software engineering. One of the main obstacles is reuse of existing components. Analyzing and comparing ontological engineering with software engineering methods revealed why reuse is widely practiced in software engineering but not yet in ontological engineering. Therefore, we propose the adoption of well-established reference models, use of reference ontologies, and an approach to applying them in practice.

7. Conclusion

In this paper, we have clarified the term ontology and have given reasons why ontologies have not been as successful as they could be in large-scale business applications. A critical analysis of existing research and business applications revealed frequent misuse of ontologies. This paper covers four main contributions to overcoming most of the deficits discussed.

1. An extended definition of the term ontology and identification of benefits gained from a targeted application of ontological engineering.
2. Discussion of two case studies (i) to highlight the potential of ontological engineering and (ii) to illustrate the misuse of ontologies in research and business applications.
3. Categorization and classification of existing modeling approaches combined with guidelines for selecting the most appropriate model for a new use case or application.
4. Development of an ontology engineering process that supports reuse by adapting established software engineering processes.

The general misuse of ontologies can be explained by (i) many existing, sometimes conflicting, definitions of the term ontology, (ii) insufficiently precise specifications of semantic technologies, and (iii) the existence of long and arduous iterative modeling processes that are very abstract and complex. This research work points out the main benefits of ontologies that should be reflected in the definition of the term in order to make them transparent. Based on the existing definition given by Studer et al. [1] and the analyzed benefits, we define an ontology as *a formal, explicit specification of a shared conceptualization that is characterized by high semantic expressiveness required for increased complexity*.

The case studies from the manufacturing industry and the tourism domain are key examples in which ontologies were not applied or were applied unsuccessfully. They clearly indicate barriers encountered in real-world applications. Thus, in an industrial environment, preference must be given to adapting ontologies to business-specific needs instead of the idea of open shared data. The case study of the tourism domain clearly indicates potential for applying ontologies, for example, the large, distributed community requires sharing. However, at the same time the community fails to agree on a standardized vocabulary, which is an obstacle to the application of ontologies.

In addition to the extended definition of ontology (i) guidelines for answering the questions of when ontologies should be used, how they can be used and when they should not be used were introduced, and (ii) adequate representation languages were discussed – both are fundamental prerequisites for achieving maximum benefit from the application of ontologies. A classification that characterizes the differences and relationships between alternative technologies (or modeling languages/concepts) is therefore introduced. We provide support in selecting an appropriate modeling language for a specific problem by considering the requirements for sharing and complexity (e.g., real-world abstraction versus desired level of detail, large number of objects and relationships, and inference capabilities), semantic expressiveness and reuse, as well as an attribution to a classification and a structural and/or conceptual representation. The classification indicates the effort required and allows a justifiable decision to be made for one technology over another.

We also compare ontological engineering to other modeling processes, especially software engineering, in order to identify the benefits of the latter and to adopt some of them for the former with the aim of increasing user acceptance. In addition, reasons are discussed why reuse is widely practiced in software engineering but not to the same extent in ontology engineering. To address this shortfall, reference ontologies are introduced and a new process for reusing existing ontologies is proposed.

Choosing an appropriate model, be it an entity-relationship model, a UML-based model, or an ontology, is a critical decision with a direct impact on the resources required and the ultimate success of the project. Therefore, ontological engineering must provide a clear decision process to select the correct model, methodology, and tool set to meet user requirements while using resources most efficiently. The resulting conceptual model must then be transformed into an executable schema before it can be implemented and applied in a system. Ideally, ontological engineering should have an associated set of tools (or guidelines for tool production) that lead to semi-automated transformation of models into applications. A simple notation for ontological engineering, similar to the Chen Notation [24] for entity-relationship modeling, has the potential to increase uptake in business. Further work, which we have already started, therefore concentrates on the development of a graphical notation for ontological engineering. The notation includes representations of classes, data type properties, object properties, and class axioms/restrictions.

In addition, simple rules for transforming an ontology model into a formal ontology language will be proposed. These simple rules are similar to transformation rules required for transforming an entity-relationship model into a relational mode, as used in the field of relational databases.

References

[1] R. Studer, R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, Data Knowl. Eng. 25 (1-2) (1998) 161–198.
[2] S.J. Russel, P. Norvig, Artificial Intelligence—A Modern Approach, in: Alan Apt (Ed.) Prentice-Hall, Inc, 1995.
[3] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object-Oriented Modeling and Design, Prentice-Hall, Inc., 1991
[4] N.F. Noy, D.L. McGuiness, Ontology Development 101: A Guide to Creating Your First Ontology, 2001.
[5] T. Gruber, A translation approach to portable ontology specifications, Knowl. Acquis. 4 (2) (1993) 199–220.
[6] M.L. Brodie, On the development of data models, in: M.L. Brodie, J. Mylopoulos, J.W. Schmidt (Eds.), On Conceptual Modeling, Springer, New York 1984, pp. 19–47.
[7] Y. Wand, V.C. Storey, R. Weber, An ontological analysis of the relationship construct in conceptual modeling, ACM Trans. Database Syst. 24 (4) (1999) 494–528.
[8] M. Alavi, D.E. Leidner, Review: knowledge management and knowledge management systems: conceptual foundations and research issues, MIS Q. 25 (1) (2001) 107–136.
[9] R. Davis, H. Shrobe, P. Szolovits, What is a knowledge representation? AI Mag. 14 (1) (1993) 17–33.
[10] V.C. Storey, Understanding Semantic Relationships, The VLDB Journal, 2 (4), Springer, 1993 455–488.
[11] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American MagazineMarch 26 2008.
[12] H. Stuckenschmidt, Ontologien: Konzepte, Technologien und Anwendungen (German), Springer, 2011 1–274.
[13] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, W.R. Swartout, Enabling Technology for Knowledge Sharing, AI Mag. 12 (3) (1991) 36–56.
[14] N. Guarino, P. Giaretta, Ontologies and Knowledge Bases: Towards a Terminological Clarification, Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, in: N. Mars (Ed.) IOS Press, Amsterdam 1995, pp. 25–32.
[15] W. Borst, Construction of Engineering Ontologies(PhD Thesis) Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands, 1997.
[16] M. Hepp, Ontologies: State of the Art, Business Potential, and Grand Challenges, Ontology Management: Semantic Web, Semantic Web Services, and Business Applications, in: M. Hepp, P. De Leenheer, A. de Moor, Y. Sure (Eds.), Springer 2007, pp. 3–22.
[17] C. Mangold, A survey and classification of semantic search approaches, Int. J. Metadata Semant. Ontol. 2 (1) (2007) 23–34.
[18] D.L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: a web-scale approach to probabilistic knowledge fusio, Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14), New York 2014, pp. 601–610.
[19] F. Bauer, M. Kaltenböck, Linked Open Data: The Essentials—A Quick Start Guide for Decision Makers, http://www.reeep.org/LOD-the-Essentials.pdf (December 07, 2015).
[20] O. Bodenreider, Ontologies and data integration in biomedicine: success stories and challenging issues, Proceedings of the 5th International Workshop on Data Integration in the Life Sciences (DILS'08), Springer 2008, pp. 1–4.
[21] M. Hepp, Possible ontologies: how reality constrains the development of relevant ontologies, Internet Comput. 11 (1) (2007) 90–96.
[22] N. Guarino, D. Oberle, S. Staab, What Is an Ontology? in: Springer, S. Staab, R. Studer (Eds.), Handbook on Ontologies, International Handbooks on Information Systems, Springer 2009, pp. 1–17.
[23] Object Management Group OMG, UML 2.5 Specification, http://www.omg.org/spec/UML/2.5/Beta2/PDF/ (December 07, 2015).
[24] P. Chen, The entity-relationship model—toward a unified view of data, ACM Trans. Database Syst. 1 (1) (1976) 9–36.
[25] M. Hori, Semantic annotation for web content adaptation, Spinning the Semantic Web2003 403–426.
[26] HTML5 Semantic Elements, http://www.w3schools.com/html/html5_semantic_elements.asp (December 07, 2015).
[27] R. Van Rees, Clarity in the usage of the terms ontology, taxonomy and classification, CIB Rep. 284 (432) (2003) 1–8.
[28] K. Prantner, Y. Ding, M. Luger, Z. Yan, C. Herzog, Tourism ontology and semantic management system: state-of-the-art analysis, Proceeding of the IADIS International Conference 2007, pp. 111–115.
[29] G. Dragan, D. Dragan, D. Vladan, Model Driven Engineering and Ontology Development, 2nd edition Springer, Berlin, 2009.
[30] P. Lambe, Organising Knowledge: Taxonomies, Knowledge and Organisational Effectiveness, Chandos, 2007.
[31] T. Leitner, W. Wöß, Optimizing early detection of production faults by applying time series analysis on integrated information, Int. J. Adv. Syst. Meas. 8 (1&2) (2015).
[32] C.J. Janse, H. Kroeze, A. van Wigcheren, et al., A genotype and phenotype database of genetically modified malaria-parasites, Trends in Parasitol. 27 (1) (2010) 31–39.
[33] M. Uschold, M. Gruninger, Ontologies: Principles, Methods and Applications, Cambridge University Press, 1996.
[34] M. Vujasinovic, N. Ivezic, E. Barkmeyer, Z. Marjanovic, Semantic B2B-integration using an ontological message metamodel, Concurr. Eng. 18 (3) (2010) 219–232.
[35] H. Werthner, F. Ricci, E-commerce and tourism, Mag. Commun. ACM 47 (12) (2004) 101–105.
[36] C. Feilmayr, C. Grün, R. Barta, B. Pröll, H. Werthner, Covering the semantic space of tourism: an approach based on modularized ontologies, Proceeding of the 1st Workshop on Context, Information and Ontologies (CIAO'09), ACM, New York, 2009.
[37] Deutscher Tourismusverband: Classification of Tourism Offers, http://www.deutschertourismusverband.de/klassifizierung.html (December 07, 2015).
[38] Deutsche Hotelklassifizierung, http://www.hotelsterne.de/userfiles/Files/downloads/de/Deutsche_Hotelklassifizierung_2010-2014.pdf (December 07, 2015).
[39] ISO 18513:2003, Tourism Services – Hotels and Other Types of Tourism Accommodation – Terminology, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=31812 (December 07, 2015).
[40] World Tourism Organization, Thesaurus on Tourism & Leisure Activities, World Tourism Organization, Madrid, Spain, 2001.
[41] Open Travel, http://www.opentravel.org/Specifications/OnlineXmlSchema.aspx (December 07, 2015).
[42] M. Hepp, GoodRelations: an ontology for describing products and services offers on the web, Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW'08), Springer LNCS, 5268 2008, pp. 332–347.
[43] Accommodation Ontology Language Reference and Accommodation Ontology Meta Data, http://ontologies.sti-innsbruck.at/acco/ns.html (December 07, 2015).
[44] S. Ou, V. Pekar, C. Orasan, C. Spurk, M. Negri, Development and alignment of a domain-specific ontology for question answering, in: European Language Resources Association (ELRA) (Ed.), Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), 2008 (Marrakech, Morocco).
[45] Mondeca, Semantic Web Methodologies and Tools for Intra-European Sustainable Tourism, White PaperSeptember 2004.
[46] DERI, OnTour Ontology, http://e-tourism.deri.at/ont/index.html (December 07, 2015).
[47] TAGA Ontology, http://taga.sourceforge.net/owl/index.html (December 07, 2015).
[48] S. Staab, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, B. Wrenger, GETESS—searching the web exploiting German texts, Proceedings of the 3rd International Workshop on Cooperating Information Agents (CIA'99), Springer, Berlin, 1999.
[49] EON Traveling Ontology, http://www.eurecom.fr/~troncy/DOE/ontologies/EON-TravellingOntology-v0.2.oil (December 07, 2015).
[50] J. Cardoso, Developing an OWL ontology for e-tourism, semantic web services, in: J. Cardoso, A. Sheth (Eds.), Processes and Applications, Springer, New York, 2006.
[51] ebSemantics Ontology, http://www.semantic-web.at/news/ebsemantics-semantic-web-enabled-ecommerce (December 07, 2015).
[52] N. Sharda, R. Jakkilinki, M. Georgievski, M. Ponnadal, Intelligent visual travel recommender systems model for e-tourism websitesTechnical report in: Gold Coast (Ed.) Sustainable Tourism CRC, Qld, 2008.

[53] CEN Workshop Agreement, CEN Workshop on Harmonization of Data Interchange in Tourism—WS/eTOUR, Draft for Public Review, 2009.
[54] O. Fodor, H. Werthner, Harmonise—A Step Towards an Interoperable E-Tourism Marketplace, Int. J. Electron. Commer 9 (2) (2005).
[55] H. Stuckenschmidt, F. van Harmelen, Information Sharing on the Semantic Web, Springer, Berlin, 2005.
[56] M. Gruninger, J. Lee, Introduction: ontology applications and design, Commun. ACM 45 (2) (2002).
[57] T. Bürger, E. Simperl, Measuring the benefits of ontologies, Proceedings of the On the Move to Meaningful Internet Systems Workshops, OTM Confederated International Workshops and Posters, ADI, AWeSoMe, COMBEK, EI2N, IWSSA, MONET, OnToContent + QSI, ORM, PerSys, RDDS, SEMELS, and SWWS 2008 (OTM '08), Springer LNCS, 5333 2008, pp. 584–594.
[58] D. Oberle, How ontologies benefit enterprise applications, Semantic Web 5 (6) (2014) 473–491.
[59] Internationalized Resource Identifiers (IRIs), http://tools.ietf.org/html/rfc3987 (December 07, 2015).
[60] T. Menzies, Cost benefits of ontologies, Intelligence 10 (3) (1999) 26–32.
[61] OWL Web Ontology Language, http://www.w3.org/TR/owl-features/ (December 07, 2015).
[62] J. Cardoso, The semantic web vision: where are we? IEEE Intell. Syst. (2007) 22–26.
[63] T. Pellegrini, Semantic Web Awareness 2009—A Comparative Study on Approaches to Social Software and the Semantic Web, Research Report, http://www.corporate-semantic-web.de/tl_files/pub/CSW_Infotag_20090320/SemWebBarometer_2009_Report.pdf2009 (December 07, 2015).
[64] P. Warren, P. Mulholland, T. Collins, E. Motta, Using ontologies: understanding the user experience, Proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2014), Springer, Linköping, Sweden 2014, pp. 579–590.
[65] P. Warren, Ontology users' survey—summary of results, Technical Report, Knowledge Media Institute, The Open University, 2013.
[66] E. Simperl, C. Tempich, Ontology engineering: a reality check, Proceeding of the On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006 (OMT '06), Part I, Springer LNCS, 4275 2006, pp. 836–854.
[67] E. Simperl, Reusing ontologies on the semantic web: a feasibility study, Data Knowl. Eng. 68 (10) (2009) 905–925.
[68] J. Cardoso, M. Hepp, M. Lytras, The Semantic Web—Real-World Applications from Industry, Springer, 2008.
[69] R. Elmasri, S.B. Navathe, Fundamentals of Database Systems, 7th edition Pearson/Addison-Wesley, 2015.
[70] A. Gómez-Pérez, M. Fernandez-Lopez, O. Corcho, Ontological engineering with examples from the areas of knowledge management, e-commerce and the semantic web, Advanced Information and Knowledge Processing2004.
[71] O. Lassila, D. McGuinness, The role of frame-based representation on the semantic web knowledge systems laboratory, Report KSL-01-02, Stanford University, 2001; also appeared as Linköping Electronic Articles in Computer and Information Science 6 (5), Linköping University, 2001.
[72] C. Roussey, F. Pinet, M. Ah Kang, O. Corcho, An introduction to ontologies and ontology engineering, ontologies in urban development projects, in: G. Falquet, C. Métral, J. Teller, C. Tweed (Eds.), Advanced Information and Knowledge Processing, Springer, London, 2011.
[73] J. Fong, Mapping extended entity relationship model to object modeling techniques, ACM Sigmod Rec. 24 (3) (1995).
[74] R. Juric, The UML rules, ACM SIGSOFT Softw. Eng. Notes 23 (1) (1998).
[75] M. Fernández-López, A. Gómez-Pérez, N. Juristo, METHONDOLOGY: from ontological art towards ontological engineering, Proceedings of Symposium on Ontological Engineering of AAAI, 1997.
[76] M. Uschold, M. King, Towards a methodology for building ontologies, in: D. Skuce (Ed.), Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI'95) 1995, pp. 6.1–6.10 (Montreal, Canada).
[77] M. Jarrar, J. Demeyer, R. Meersman, On using conceptual data modeling for ontology engineering, Journal on Data Semantics (special issue on best papers from the ER, ODBASE, and COOPIS 2002 Conferences) and Springer LNCS, 2800 2003, pp. 185–207.
[78] Das V-Modell, (German) http://v-modell.iabg.de (December 07, 2015).
[79] Clarus Concept of Operations. , Federal Highway Administration (FHWA), 2005 Publication No. FHWA-JPO-05-072.
[80] P. Rook, Controlling software projects, IEEE Softw. Eng. J. 1 (1) (1986) 7–16.
[81] C. Alexander, Notes on the synthesis of form, Harv. Univ. Press 5 (1964).
[82] A. Gangemi, V. Presutti, Ontology design patterns, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, International Handbooks on Information Systems, Springer 2009, pp. 221–243.
[83] A.L.N. Escórcio, J. Cardoso, Editing tools for ontology creation, chapter IV in semantic web services: theory, tools and applications, IGI Glob. (2007) 71–95.
[84] J.C. Arpírez, A. Gómez-Pérez, A. Lozano, S.H. Pinto, Reference ontology and (ONTO)^2 agent: the ontology yellow pages, Knowl. Inf. Syst. 2 (4) (2000) 387–412.
[85] NIST, ECMA, reference model for frameworks of software engineering environment, 3rd edition, NIST Special Publication, 500, 1993 1–211.

Dr. Christina Feilmayr is a senior scientist since 2008 at the Institute of Application Oriented Knowledge Processing (FAW), Johannes Kepler University Linz. Her PhD was awarded in 2014 from the Johannes Kepler University and was in the area of information extraction and deals with completing information extraction results via data/text mining methods. Her research interests are in information extraction, natural language processing, text and data mining, and ontology engineering.

Prof. Dr. Wolfram Wöß is an associate professor for applied computer science at the Johannes Kepler University Linz (JKU), Austria, since 2002. Between 1993 and 2002, he was a university assistant at the Institute for Application Oriented Knowledge Processing (JKU Linz, Austria). Wolfram Wöß managed several research and development projects and supervised a number of PhD students.His research interests include intelligent information systems, integrated information systems, semantic information integration, ontologies, semantic web, knowledge-centered systems, information engineering, data modeling, data science, data mining, data quality in information systems, business intelligence, and e-business systems.