A Critical Reflection on Ontologies and their Applications in Business

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Nickolas Falkner[a,], Christina Feilmayr[b] and Wolfram Wöß[b]
[a] *School of Computer Science, The University of Adelaide, nickolas.falkner@adelaide.edu.au, Adelaide, Australia*
[b] *Institute of Application Oriented Knowledge Processing, Johannes Kepler University Linz, {christina.feilmayr, wolfram.woess}@jku.at, Linz, Austria*

Abstract. In recent history, the term ontology has been used as if conveyed a great deal of weight and importance when, in many cases, the term has been used incorrectly. This diffusion of meaning is often the path by which a perfectly acceptable and well-defined word becomes a buzzword, reduced in meaning and a warning to readers that poor science is ahead. Frequently, a hyped buzzword will lead the reader to form expectations that are never fulfilled. This research work provides a critical reflection on ontologies, their frequent misuse in research and business applications, and concerns aspects why ontologies have not been successful in large-scale business applications until now. When a definition that changes over time, as is the case for ontologies, this may be indicative of a lack of understanding in the field, or an inability to effectively communicate and share a common understanding. Whatever the reason is for this case, introducing numerous definitions for one concept, especially a complex concept, leads to confusion and, consequently, people from various research communities can (and do) use the term ontology with different, partly incompatible meanings in mind. The general misuse of this semantic technology can be explained by (i) the many existing, sometimes conflicting, definitions; (ii) too imprecise a specification of semantic technologies; and (iii) the existence of complex modeling processes that are too abstract or too complex. Because the term is used with a variety of meanings, only some of which are accurate, it has become increasingly difficult to discover who was or was not truly using ontologies. However, the relatively rare use in business applications can be attributed (i) to unknown (or unavailable knowledge about handling of) ontology modeling processes; (ii) to the lacking support of the modeling process because of missing state-of-the-art modeling tools; and (iii) to a lack of mature experience, developed over time. If we are to clearly demonstrate the benefits of an ontological approach, we must first clearly define what ontologies are and exactly when they are in use. Therefore, the causes for the general misuse and rare use of ontologies must be identified. Furthermore, it has to be clearly defined what an ontology is and guidelines are necessary to answer the question as to *when* ontologies should be used, *how* they can be used and *when they should not be used*. Finally, adequate representation languages should be applied to the problem and the ontology design process should be easy to understand and couched in terms that technical and non-technical users alike can understand. The main focus of this research work is to provide clear decisions to select a correct model, methodology and toolset to meet user requirements with the most efficient use of resources.

Keywords: Ontologies, Data models, Ontology Engineering, Data Modeling

1. Introduction

Computer science and software engineering (SE) are relatively recently developed disciplines, by the standards of other sciences and philosophy. Within computer science, the field itself is continuing to evolve as languages mature, representations develop and the ability to undertake solutions to new challenges is enhanced through more advanced software and hardware techniques. Artificial intelligence (AI) has been a significant challenge area for computer scientists over time and, while the community is yet to develop a "true" human-level machine intelligence, the pursuit of AI has led to the development of important work knowledge representation and the abstract relationship of semantic relationships. This work, in *ontological representation*, allows the modelling of meaning in systems that are to be constructed in software. Much as general approaches in software, such as object-oriented programming models, allows the transformation of a model into a useful software artefact, an ontological model allows the production of software that can evaluate semantic relationships, to determine if statements made within a domain of knowledge are valid and to provide much richer rules for the organization of information.

However, despite work in ontological engineering having taken place over decades, there are very few truly ontologically based system that are employing all of the benefits of the approach, are clearly identified as being reasoning support systems and do much more than classify knowledge into convenient categories. More significantly in terms of on-going process maturity and education, there is no "Gold Standard" of describing or producing such a representation that can be clearly indicated as the correct method to undertake. Contrast this with the field of software engineering, where methodologies are widely discussed and understand and it is possible to clearly identify which approach to take under given circumstances, with a high degree of confidence. Given the benefits of establishing a clear, and machine interpretable, basis for meaning in a system, this is a wasted opportunity to greatly enhanced knowledge management and decision making in software systems. One of the other great benefits of an ontological approach, as will be discussed below, is that meaning can be shared providing that there is clear and unambiguous agreement as to what form such a comparison or exchange takes. With so much variation in production, process and representation, it is of little surprise that a well-established, mature, and semantically sound comparison and exchange mechanism does not yet exist.

The goal of this paper is to identify, through a discussion of case studies and a comparison with the mature area of software engineering, that ontological engineering is an immature area, but that there are a number of important forward steps that can be made by clearing defining the terms, adopting process models from other areas and learning from their mistakes, and seeking to communicate to the community when they are and they are not using an ontology. It is impossible to advance a community of practice if the community does not clearly define what their shared practice is and has no clear pathway forward or improvement. As will be demonstrated, the software engineering community have already dealt with most of these problems and, while they are more subtleties and abstractions in the area of ontological engineering, the final goal of this paper is to motivate ontological practitioners to consider adopting some of SE's successful techniques, including the increasing reuse of existing ontologies. If mechanisms can be built that allow a robust and reliable classification of meaning to support reuse, then ontological engineering has taken a giant leap forward.

In this paper, the background of ontologies is provided, leading to three case studies from industry and research, to provide a background to the problem and illustrating that, while there are clear definitions available for these technologies, the general understanding of ontologies is low and that this has an impact upon uptake and further deployment, including reuse. The benefits of ontologies are introduced, leading to a discussion of models and representation of real systems inside an ontological model. In the discussion of modelling, it is a natural progression to compare ontological modeling with other conceptual modeling and, through the mechanism of comparison, the discussion moves to one of the other areas in computer science in which modeling is paramount: software engineering. Inspecting the field of software engineering quickly reveals that a number of the developments in ontologically engineering have been trailing SE - and that this means that it may be possible to shortcut a longer developmental cycle by "jumping to the end" rather than having to laboriously replicate the steps that SE has already taken. But this raises the question: "Is ontological engineering actually an area?" The next section motivates the area of ontological engineering as a separate (and valuable) discipline but emphasizes the practice of reuse as a clear indicator of successful practice. The penultimate section addresses how the knowledge that has been identified in the paper can be used to choose better models and suitable tools that will rapidly improve the maturity and capability of ontological engineering.

Ontologies are more than just buzzwords and, with sound definitions and guidelines for modeling and tool selection, these important technologies can become more widespread and integrated into projects that will benefit from them. But this must also be accompanied by a mature community understanding of what is being discussed and, most importantly, when a project is not ontologically-based.

2. Information technology following philosophy

The term ontology derives from philosophy and is the study of the nature of being, existence or reality in general, as well as the basic categories of being and relations. An ontology deals with whether or not a certain thing exists or can be said that to exist. There are several definitions of the word ontology, the first referring to a systematic explanation of being. Computer scientists have borrowed the philosophical term "ontology", the study of the nature of being and the basic categories of being and relations. In the last decade, in which ontologies have become established in computer science, the meaning of the word ontology has changed and evolved.

One of the first computer science definitions was given by Neches (1991), "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." Gruber (1993) refined this with "An ontology is an explicit specification of a conceptualization" [20].

Guarino (1995) presents a broad discussion of possible interpretations of the term ontology concluding with a more formal notion of the "conceptualization" [22] and addressed Gruber's definition within this in order to determine exactly which interpretations were consistent with which definitions. Guarino identified that the term ontology depends on whether both of the parties using the ontology have already decided upon a degree of expressiveness or a shared conceptualization.

Borst (1997) defined an ontology as a "formal specification of a shared conceptualization" [4]. This definition additionally required that the conceptualization should express a shared view between several parties, a consensus rather than an individual view. Borst also required that such conceptualization has to be expressed in a (formal) machine readable format. Studer (1998) merged Gruber and Borst stating that: "An ontology is a formal, explicit specification of a shared conceptualization." [40]

There are a number of further definitions for ontologies, as they are used within computer science. A definition that changes over time may be indicative of a lack of understanding in the field, or an inability to effectively communicate and share understanding. Whatever the reason, introducing numerous definitions for one concept, especially a complex concept, leads to confusion and, consequently, people from various research communities can (and do) use the term ontology with different, partly incompatible meanings in mind. In fact, it is paradoxical that the seed term of a novel field of research, which aims at reducing ambiguity about the intended meaning of symbols, is understood and used so inconsistently [24].

This paper does not aim to conduct a broad discussion of the term ontology merely by comparing the semantic differences of several definitions. Instead, the remainder of the paper focuses on a critical reflection on ontologies and their applications in business. The paper therefore includes a discussion of the specific characteristics of ontologies and their benefits by reflecting business demands. In order to provide a recommendation for efficiently using ontologies in a (business) project the following factors are identified as relevant:

- requirement for sharing,
- semantic expressiveness,
- complexity of the universe of discourse,
- and size of the sharing community (ontology stakeholders).

On the basis of the discussed definitions and the incompatible interpretation of ontologies the question may arise why ontologies are that popular in information technology and its research fields. And consequently, someone can ask why ontologies are not yet widely spread in the business world.

Tim Berners-Lee, James Hendler and Ora Lassila in 2001 published the initial "Semantic Web" paper introducing a "web of data" that enables machines to understand the semantics, or meaning, of information on the World Wide Web [3]. In this paper, ontologies are introduced as the third basic component of the semantic web, covering a taxonomy and a set of inference rules. In the following years semantic web, semantics, and ontologies, rather than being specific technical terms with well-defined meanings, mutated through misuse to become predominantly buzzwords while, at the same time, the proposed application of inference rules used to conduct automated reasoning lifted ontologies from their originating artificial intelligence field to a broad focus of interest in the WWW. As ontologies became more useful, the overuse and misuse of the term rendered it more difficult to discover who was or was not truly using ontologies.

2.1 The semantic web

The semantic web, a key application of ontologies, now has a family of standards, patents and languages. One of these, the Resource Description Framework

(RDF)[1], a meta data model that allows to make statements about resources. However, one of the most obvious uses of the semantic web, the ability to search through documents that support RDF mark-up, appears to be fading.

Whereas the WWW is a medium of documents for people, the semantic web addresses data and information that can be processed automatically by providing rich and extensible meta data. Improving information retrieval with semantically enriched search functions is seen as one major advantage of the semantic web. But why do we need to optimize the way that we search? While, in the past, we sought to increase the number of pages or documents that we could find, and therefore improve *recall*, the huge amount of documents now available in the global data corpus requires a change in focus to the question of how to get the most relevant documents, thus, improving *precision*. Thesauri or extensive synonym lists are not adequate for this task because they increase recall but dilute precision: the opposite of the goal.

Adding meta data provides embedded mark-up on web pages that increases hit rates in response to customer queries and enhances document management. This is a straightforward mining exercise to generate the largest number of synonyms that can legitimately be matched for queries generated by search engines. Most importantly, while this uses parts of semantic web technology, there are no actual “semantics” that are embedded with these synonym search terms and this is, therefore, a misleading use of the term semantics. True semantic technology allows the expression of both data and rules for reasoning about data, allowing rules from existing knowledge-representation systems to be exported onto the web [3]. RDF triple sets (subject, predicate, object) can be used to express relations between documents and their meaning. RDFa, attribute level extensions to HTML that allow the transport of meta data in an XML-derived language. This flexible and domain-independent language is already in use in some search engines, such as Google, but for all that it 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. But these semantics are still locally contextualized based on the knowledge of the creator and the assumed frame of the reader. If information should be compared or combined across schema borders we must be able to express some kind of common meaning, and this is known as shared conceptualizations that, as has been discussed, leads us directly to a requirement for ontologies.

Once again, all of the previous definitions of ontology are not what is being used in the real world search engine case. The search engines are using an approach that is far 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 their ability to allow us to perform precise matching of concepts, rising above analyzing the grammar and the context to resolve language ambiguities such as words with multiple meanings.

If a semantic matching technology is to be incorporated into the mass-synonym case then, notionally, an ontology is required at 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 practical use, however, almost all search engines are single-sided-semantic. 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, provides the searcher with the largest possible range of matches. Again, this enhances recall but at the cost of precision. From a performance perspective, while a high recall mechanism arguable reduces human need to search synonym space, this advantage is negated by the act of searching through 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, by processing this data and by exchanging and distributing the results. The linked open data project (LOD)[2] is such a successful scenario of semantic web approaches. Linked data uses the web to connect data and builds upon standard technologies such as HTTP, RDF and URIs. Linked open data is becoming increasingly important in the field of state-of-the-art information and data management [2]. Bauer et al. [2] stated that if someone wants to fully benefit from linked open data then it is crucial to put information and data into context that creates new knowledge and enables powerful applications. 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,

[1] http://www.w3.org/RDF/, last visited: July 01, 2014

[2] http://linkeddata.org, last visited: July 01, 2014

it provides a high level of granularity, is timely, accessible, machine-processable, license free and permanent. The linked open data cloud increases continuously (2010[3]: 26,930,509,703 triples, 203 datasets, 2011[4]: 31,634,213,770 triples, 295 datasets), but knowledge-transfer is suboptimal because only 191 of the current 295 datasets map to a non-proprietary vocabulary, where terms are not defined in the same top-level domain, only 15 datasets provide mappings to other vocabularies for their terms and 154 datasets do not provide provenance information (data about information's origin to assess data quality). Notable positive is that most of cloud resources and context information is retrievable by a URL and web applications can make use of linked data by standard web services. The linked open data cloud is regarded to be domain-independent, however the cloud provides mostly content from the media, geographic, publications, user-generated content, government and life-sciences domain.

Most importantly, machine-interpretable web content promotes web information integration because even systems that were not expressly designed to work together can transfer data among themselves when data comes with semantics. The semantic web, by its goals and standardized nature, supports a strong middleware focus in the architecture but the presumption is that this is an automated middleware. Human beings should not be functioning as the primary middleware in these systems to shoulder the semantic alignment burden between two systems.

Reflecting again the above mentioned factors and their relevance for the semantic web explains why ontologies became as popular:

- the requirement for sharing is very important in the semantic web,
- the complexity of the semantic web is really high
- therefore, the semantic web requires powerful concepts and languages to reach the necessary semantic expressiveness,
- and, finally, the size of the sharing community is huge.

Ontologies promise to achieve interoperability between multiple representations of reality (e.g. a data model) residing inside computer systems, and between such representations and reality, namely human users and their perception of reality [24]. Consequently, the semantic web with its characteristics and goal seems to be the most prominent promoter for ontologies and, potentially, also for all the inconsistencies of understanding and applying them.

2.2 Are ontologies a one size fits all solution?

In several observed international conferences and workshops the topics that are on interest and are submitted for publication have not changed significantly in the last years. Papers are submitted, at similar levels, to the same topics and the same problems are once again considered and "solved", only for the same problems to be revisited as novel and open in subsequent papers.

In order to become ubiquitous, a technology has to be implemented and then delivered in a way that the technology is either considered to be indispensable, and hence worth any effort of integration, or seamlessly integrated, and hence invisible from an effort perspective. Revising the load discussion from earlier, the total perceived load is a combination of cognitive and kinesthetic elements: adding both manual effort and intellectual requirement quickly renders a technology unusable. Ontologies cannot be considered to be either indispensable or effortless. Consequently, the question is why on the one hand ontologies are ubiquitous in information technology research and, on the other hand, the definitive industrial application, that defines ontologies as a necessity in the business and industrial community, is still missing. Some important parameters for a successful technology are:

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

The greatest mark of success is when a given application of this technology is so successful and so pervasive that it becomes part of the fabric of daily life – it is unimaginable that it would not have been created. An example is the World Wide Web, where the formal, research-focused Internet of the time, with small groups of users, transformed into a ubiquitous, universal access model with large user groups and no significant technical barrier to entry.

While ontologies (and taxonomies) are used *within* a large number of applications, it is rare for users to be directly exposed or required to interact with the ontology. The semantic web, if it is a powerful application for a web of data or for semantic-assisted search, has remained a niche technology to an extent as it has so far failed to make a substantial dent in

[3] http://lod-cloud.net/state/2010-10-19/, last visited: July 01, 2014
[4] http://lod-cloud.net/state/, last visited: July 01, 2014

popular consciousness or to be seen as relevant to mainstream groups.

One area where ontologies are, as ontologies, pervasive and part of research, is the bioinformation sciences. However, as will be discussed later, the reasons for success in biological science is not due to the advantages of computer-based ontologies in a pure sense, but because this is a natural extension of the existing use of ontologies and taxonomies inside the biological sciences. Technologies that are analogous to existing practice are easier to adopt within a community as the *affordance*, the quality of a technology or item that allows a user to interact with it successfully, of computationally taxonomies is effectively the same as the tree of life or genome sequences. Application domains with no analogue require a user to develop a new understanding of how to correctly use it.

The potential of applying ontologies has already been discussed. One of the important limitations, which prevent ontologies from becoming successful, is the base requirement of a minimal (shared) ontological commitment from the knowledge stakeholders. The stakeholders must have and must be able to agree upon a common understanding of the primitive terms. Especially for those 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 [23]. 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 to be seen as effective and successful. This core function is the facilitation of communication between human and machine, or even for facilitating inter-machine and inter-human communication. That is one of the main reasons why ontologies should be applied frequently, because (semi-) automatic communication is enabled.

Two concluding questions arise: Do we already have a truly common understanding of what ontologies are, given how many definitions are already in use? Having addressed the identity and nominative concerns, for what kind of complexity and application are ontologies appropriate? Both questions are discussed in the further sections.

1.1. Ontologies: a buzzword or something really useful for business applications?

It is obvious, from the large number of definitions of ontologies currently available that very few computer scientists can correctly and consistently understand the term, identify how ontologies can be applied, or correctly assert the list of benefits that can be derived from ontologies, given the lack of agreement over the many definitions. These are the roots that feed misunderstandings between research in different fields and between academic research and business/industrial use. Ontologies can be specified using only informal means, such as UML class diagrams, entity-relationship models, or semantic nets, whereas conceptual entities in ontologies can also be defined mainly by formal means, e.g., by using axioms to specify the intended meaning of domain elements [24]. This uncertainty leads to a broad spectrum of models, or concepts, or specifications, which are interpreted and published as ontology.

Originally, and as evidenced by the previous references, the term ontology conveyed a great deal of weight and importance. However, now it has almost become a warning word due to overuse and an increasingly vague and inaccurate use of the term. This, unfortunately, is often the path by which a perfectly acceptable and well-defined word becomes a buzzword. In many cases a hyped buzzword raises expectations that are never fulfilled.

The unfulfilled promise of early academic technologies and research investigations is a specter that hovers over many new technologies. Considering a previous example, the object-oriented databases initiatives of the late 20th Century never moved from the academic to the economical field. After initial hype, which stated that object-oriented database-management systems would soon become the primary database technology and supplant relational database-management systems, further development faded away. Today’s reality is very different and none of these predictions have come to pass. Relational databases are still by far the most widely used databases, and object-oriented databases are increasingly rare. Nevertheless, Oracle added some object-oriented concepts to their RDBMS, thus, offering a compromise for the majority of pure relational applications and some research projects and certain businesses having an interest in object-oriented databases. This is, however, not the path that leads to ubiquitous adoption.

Learning from the experiences associated with the over-promotion and over-promising associated with object-oriented databases and at the same time considering the discussed developments concerning ontologies, the following requirements can be derived in order to integrate ontologies into business applications, and thus, providing a benefit from their advantages.

- It has to be clearly defined what an ontology is. Additionally, the term semantics has to be clarified. For example, some businesses have placed some static search tags onto every web page and are referring to that as “semantic” application.
- Guidelines should help to answer the question as to when ontologies should be used, how they can be used and when they should not be used. These published use cases have to move beyond the simple delights of the Pizza and Wine/Food examples, provided as part of the W3C standards, to provide exemplars that business can understand and immediately apply, but with sufficient scope to be more than highly-focused items. Use cases have to be useful.
- As part of this understanding, the key differences in the use of ontologies between researchers and business have to be recognized. Researchers plan to share their knowledge, structures and instance data, to demonstrate the usefulness of their work and to facilitate scientific interaction. Businesses are far more likely to keep their information in-house and be very unlikely to share their developed resources widely, if at all, if this exposes core business practices. Thus, we are demanding an informal answer to the question “Is the considered implementation open-world or closed-world?” If a business expects only to share the representation internally, and to answer all questions from within there, it is possible to denote lack of knowledge as a false value.
  Hence, the business-centric representation may be considered to be closed-world. Conversely, the research standpoint is one of sharing and large-scale collaboration.
- Adequate representation languages have to be used. Many different ontology languages that have been produced, including Cyc, KIF and OWL, among the best known. Each language may be chosen based on the expressiveness supported, the representational model, and the degree of sharing that they support. Languages must be chosen to minimize the mismatch between the needs of the user and the final representation. For example, a closed-world, unshared model may use a very different representation to an open-world, globally-available model.

An example, that demonstrates some of the challenges, is the scenario concerning discussions of RDF vs. OWL or, at its core, taxonomy vs. ontology. This cannot be resolved by drawing a line and labeling one side “no explicit semantics required” and the other side “explicit semantics required”. Meaning is also conveyed by structure, and any reasoning requires structural semantics for identifying additional and missing elements.

1.2. Taxonomy versus ontology

This section in brief compares taxonomies with ontologies as representative example for (not) applying the appropriate representation model for a specific use case.

- A taxonomy is a hierarchical classification mechanism, with generalization/specialization relationships and subtype inheritance based on the very general meaning of “is-a” (one class “is-a” subclass of another, for example).
- An ontology is broader in scope than this as it contains a much richer set of valid relationships, such as “composition”, “if-then-else”, “and”, “or”, “not”, and the clear distinction between schema and instances.
- Taxonomy represents a small set of valid relationships, whereas ontology supports reasoning to deduce new classifications. Nevertheless, the backbone of an ontology consists of a generalization/specialization hierarchy of concepts, i.e., a taxonomy [23].

It is much harder to establish a correct ontology than it is to build up a correct taxonomy, because the relationships have to be explicitly specified in a way that supports reasoning. While taxonomic structures may be able to be derived almost wholly, and existentially, from the exemplars available, ontology development is iterative and requires many different levels of testing to ensure that the inferential span is correct.

Consequently, building up the complex structural and relational model of an ontology, including the long and arduous iterative development process, when only a taxonomy is needed, will provide a perception the ontology production was a waste of the time and resources invested into the project.

In order to further illustrate this point, in the next section three case studies are introduced that show important use cases for both ontological technology and the perceived requirement for ontologies. Case Study 1 describes a situation where users actively do not want, or think that they need, an ontology despite clear evidence to the contrary. Case Study 2 provides an example where users have established a set of ontologies but do not either really need them or use the (many) ontologies that are created to support interoperation. Case Study 3 shows a situation where, due to confusing terminology, users believe that they are using an ontology, when they are not, and potentially also do not need one.

3. Case studies

3.1 Case study 1: manufacturing industry

Why are ontologies not (yet) of concern to the manufacturing industry? Some reasons are discussed in the following case study that identifies the requirements for computer aided manufacturing aiming at product quality optimization and workflow efficiency.

In the last two decades, more and more company divisions gained a benefit through a wide support of various information systems. As a result, the potential for further optimization in recent years has decreased. Existing information systems ideally represent exactly the process of specific business sectors (e.g. construction, or manufacturing). But in many cases the support of the entire process chain in a manufacturing enterprise is still missing.

A specific information system normally stores its data in a purpose-built database, separated from the other, existing, databases already in use. Moving the data to a central database would prevent redundancies, inconsistencies and in addition would provide homogenous data structures. But due to a strong coupling of the software and the corresponding data structures, data is heavily tied to data schema; a central database scenario is not realistic. The resulting software adaptation would be too time- and cost-intensive.

Alternatively, data interchange mechanisms can be established between several information systems. In a manufacturing company the entire product life cycle with its involved information systems finally has to establish such data interchange mechanisms to support the entire process chain. Moreover, the business logic that overlaps the information system has to be implemented on the software side. These challenges grow with the size of the company and the number of active and installed information systems and are a significant burden for large, complex, companies.

The automotive industry is a representative example. If the final product, a car, causes a problem, the customer will visit a car repair company where the fault is detected and stored in a specific database. The 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. Meanwhile, possible error-prone parts (e.g. parts of the engine) may still be produced. Consequently, in order to save costs, the time between fault detection at the car repair company and construction improvements at the manufacturer side should be as short as possible.

In order to optimize this process, firstly, quality management has to identify the relevant attributes of the detected fault. In the next step affected bills of material have to be delimited and in combination with the recorded time of technical changes within the bills of material the cause of a production problem can be detected.

The described analysis procedure requires the interaction and cooperation of several experts across different divisions – a manual knowledge alignment problem with temporal and geographic constraints, using humans as middleware. Furthermore, data is not available on time, because export and import procedures are potentially not sufficiently efficient. At this time, data transformation, data assignment and combination and data analysis are performed manually in the majority of companies, because an integrated process view is still not implemented. Even once all of the data manipulation has been carried out, the interconnection of the identifying keys in different information systems is not possible in each case because, e.g., the identifier of the final product has no connection to the bill of material in the production. Bills of material are managed on a class level, whereas final products are managed on the instance level.

Shifting from the class level to the instance level is also caused by new manufacturing requirements, e.g., when using carbon-fiber-reinforced plastic parts. As well as considering the bills of material, it is necessary to also obtain process parameters (temperature, pressure, etc.), which have to be stored during the manufacturing process in order to ensure the reproducibility and to improve the co-operation between the development and the manufacturing division. Now a requirement for tracking is introducing more and more load inside the production process in terms on monitoring and early alignment of data to workflow.

For quality optimization purposes manufacturing companies have to bridge the gaps between different information systems in order to be able to establish analytical data processing over historical data that supports reactive adaptations and decisions. For a far-reaching optimization, which reflects also the increasing number of parameters during the development and manufacturing processes, more active data analysis is necessary to be able to create cause and effect forecasts.

The discussed aspects are reflected in the following derived optimization requirements, where requirement 1-3 are collaboration challenges and requirement 4-5 are common data engineering challenges:

1. Global consideration of workflows
2. Defragmentation of workflows
3. Integrated information flows
4. Data preparation, data cleansing
5. Establishment of a global meta data model
    a. Mapping of incompatible identifiers
    b. Harmonization (syntax, scale, …)
    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 not a real requirement for sharing, as we have already identified.

Using ontologies is efficient for formal models aiming at compiling and classifying information and resources in several knowledge domains like the (semantic) web. In such an open-world assumption 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 it is necessary for business-to-business integration or in life sciences [28, 42, 44].

Data quality improvement, as it is required for optimization purposes in manufacturing companies, demands homogenous, consistent views on distributed information that focus on syntactic and structural heterogeneities. Such a homogenous view is a global schema resulting from an integration of individual schemas each representing one of the distributed information systems. It is not efficient to represent a database schema as ontology. The reasons are that a database schema:

- is already an abstract model of the real world,
- is explicitly available,
- is machine readable,
- does not require reasoning,
- is normally developed for a limited number of applications whereas ontologies are representing a consensus of a larger number of partners modeled by a set of experts in a specific domain.

Businesses such as the automotive industry have no interest in sharing their hard-won business processes, as this may, in their opinion, dilute their business advantage for no real benefit. Thus, while these businesses would benefit from the use of an ontology for the internal, intra-business communication, the closed-world model will dominate any attempts to produce a more generic, or open, ontology for wider-scope information exchange.

3.2 Case study 2: semantic web in tourism

Travel and tourism are 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 is more user-centered, users can express opinions about their preferences, rate different places (like hotels, bars, visiting places etc.), carry out social networking and contribute to the formation of a user-centric data corpus that may be seen alongside business-provided web elements.

There are countless key players in the tourism sector, some concerned with transport, accommodation, gastronomy, the offer of tourism services (such as leisure facilities), and the management of tourism destinations (and their cultural offers). Each player has different information needs, and each key player has a specific perspective of the overall tourism domain. Hoteliers are concerned with issues that will affect their room occupancy and return rate, where restaurateurs worry about food fads, gastro tourism trends and table occupancy.

Consequently, a vast amount of information is generated, processed and applied, so that the most effective technology has to be used to manage it, in order to provide decision- and action-making.

Today's information management solutions for the complex tasks of tourism intermediaries are still at an early stage from a semantic point of view. Furthermore, information technology starts to play a challenging role in the domain of tourism, such as semantic web and web 2.0. During the last couple of years, the "emergence of ontologies" has led to a fundamental enhancement of web-based travelling and information systems, which allows the use of semantic technologies. Researchers and key players of the tourism domain determine the use of semantic description technologies to cope with a number of challenging requirements related to the tourism sector. The main motivation of establishing ontologies in tourism domain is threefold: Firstly, ontologies are used to compensate for the interoperability problem that is associated with the alignment and integration of heterogeneous data sources. The domain is characterized by a large set of different information systems, with different scopes, basic technologies and architectures as well as information structures [45]. Due to this, ontologies are, by definition, able to integrate the different information sub domains of several key players. Secondly, ontologies provide a formal basis for providing recommendations, inferential analysis, and creating new knowledge from the provided information (cf. data mining, opinion mining or sentiment analysis). Thirdly, the key players of tourism domain want to use ontologies for a more qualitative information search, for automatic discovery, negotiation and adaption/personalization of tourism services [14]. Summarized, the most relevant characteristics of ontologies in tourism are that a shared conceptualization is provided and that this conceptualization is a formal one, which means information is further processable in recommendation, inference and knowledge management systems.

After these initial motivations for establishing ontologies (or other semantic technologies) in the tourism domain, other application scenarios for tourism ontologies arise. The latest trends are recommender systems, person-computer interaction, ubiquitous computing, mobile technologies, search systems, location-based services, social media and system integration. A lot of research groups analyze users' behavior in searching for leisure information when planning a trip. Furthermore, decision-making systems are implemented based on contextual information to enable personalized trip planning. Ontology-based trip planning, user profiling, and modeling contextual information are increasing in popularity.

3.2.1 Ontologies, Taxonomies and Related Technologies in the Tourism Domain

Recently, industry, academia and several collaborative projects have designed different standards, catalogues, taxonomies and ontologies that should help to manage the heterogeneous tourism concepts and their data. But there is a missing semantic unification, to share information among different participants (i.e., how can the knowledge requirements of the restaurant manager within a hotel with the hotelier-specific concerns of the general manager be aligned?).

Standards. A study of the existing standards (e.g., accommodation or hotel classifications) is a prerequisite when developing an ontology in the area of tourism. There are standards for terms and classification, which are summarized as follows:

- Accommodation Facility Classification (Deutscher Tourismusverband e.V.) is a classification system for accommodation facilities that has the aim to enable more precise product positioning and therefore better sales opportunities. It comprises almost all of the terms of room setup and service [11].
- German Hotel Classification (Deutsche Hotel- und Gaststättenverband, DEHOGA) [9] is used for defining the criteria an accommodation facility must fulfill. The more criteria an accommodation fulfills, the more stars it will receive.
- ISO 18513:2003 Tourism services - Hotel and other types of tourism accommodation [27] – This terminology (later adopted from a standard by the European Committee for Standardization) defines terms used in the tourism industry in relation to the various types of tourism accommodation and other related services.
- Thesaurus on Tourism and Leisure Activities (World Tourism Organization, WTO) [46] is a guide to tourism terminology for the standardization and normalization of a common indexation and research language, at an international level.
- XML Schema Documents (Open Travel Alliance, OTA) [32] provide typical concepts for describing events and activities in the travel sector.
- Extension of GoodRelations (a standard vocabulary for the commercial aspects of offers, cf. [26]) termed as ACCO provides an accommodation-ontology for hotels, vacation homes, camping sites and other accommodation offers for e-commerce [1].

Application ontologies, domain ontologies and taxonomies. Aside from the existing (classification) standards there exist many application ontologies (or taxonomies), which try to cover the (whole) tourism domain or other relevant subareas. Some of these ontologies are:

QALL-ME ontology [33], Hi-Touch [31], DERI e-Toruism ontology [10], TAGA [41], GETESS [38], EON-Travelling [13], OnTour [6], ebSemantics [12], AUSTO [35]. Most of them have similar concepts and hierarchies, describing a bundle of typical tourism objects, such as popular attractions, food and service, accommodation, transportation and infrastructure, tourism events (e.g., music festival) as well as tourism destinations (e.g., national park or lake region) or catch only a sub-area of tourism. An example of a sub-area includes ebSemantics, which provide a separate ontology for accommodation, event and gastronomy. Each of the listed ontologies is used for a different application scenario.

However, no single player has enough power to impose one single accepted standard ontology/taxonomy [45]. Standardizing the main vocabulary, taxonomies and ontologies is needed but the complexity is too high. Within the EU project HarmoNET (Harmonisation Network for the Exchange of Travel and Tourism Information) the participants have built an ontology, termed Harmonise ontology [8, 16], that address the interoperability problems in the area of tourism, and focuses on data exchange. The Harmonise ontology (IMHO) provides concepts for events and accommodation in order to allow modeling and saving concepts of transaction data. Additionally, the tourism harmonization network provides mapping rules for transformation. In the end, Harmonise did not achieve the required acceptance, so it did not get a standard in the tourism domain.

Design issues of tourism ontologies and taxonomies. Aside from the well-designed and well-applied standards, the ontologies in the listed applications and the attempt to provide a unification of tourism concepts (cf. Harmonise ontology) there are still several issues that address the problems of using ontologies in tourism.

In the tourism domain the frequent points of criticism are not that a taxonomy, or a vocabulary is pushed as an ontology, as it appears in other domains (cf. see case study 3: large scale data curation). While it is possible to find taxonomies/ontologies, these would not pass the validation and verification phase, because often there are missing statements, like allowed domain and ranges or often ontologies are not designed in a way that supports reasoning. Most of the mentioned ontologies are application ontologies that are hardwired with the source code of the application, a common usage in the semantic web that reduces the need to develop a truly shareable model of the underlying processing semantics. Furthermore, the last point hinders any further developments and re-engineering processes in the ontology engineering life cycle.

The critical points of weakness can be traced to the planning and specification phase of ontology engineering. Often the use of an ontology is already determined, while the applicability of an ontology is still not proven. In addition, a number of primary tasks in ontology engineering are often skipped, furthermore, competency questions that the ontology should answer, are missing. Consequently, many state-of-the-art ontologies in the tourism domain base their applications on ontologies and enable functionalities, which do not necessarily need an ontology. Thus, following questions arises concerning the tourism domain ontology's application scenarios:

Do we need an ontology to prevent the interoperability problems, as Harmonise does? Yes, but there are too many challenges that make a unification of vocabulary difficult. Standards, ontologies and taxonomies are of value to those developing tourism and travel standards. Due to the cultural and linguistic differences formulating standard tourism definitions is a difficult task. There are too many differences, i.e. in categorizing hotels. A four star hotel in Greece has additional required room setup (or in general, has a different set of general criteria) compared to a four star hotel in Austria. In addition, in many cases there are nationally-specific ratings, such as those found in Austria. Tiscover uses, in the similar way of using stars for rating, flowers (one to five) to rate Austrians homesteads, and gentians (one to five) for private rental holiday flats. Due to the heterogeneity of the tourism sector, the process of developing and maintaining a single tourism ontology that covers the whole tourism market, including geographical-, temporal-, and user-related information would be very tedious and would require an agreement on a shared vocabulary between the different tourism organizations and its key players. Hence, in order to cover the semantic space of the tourism domain and to facilitate interoperability between the different tourism services, a bundle of ontologies may be required. However, these ontologies should not be disconnected, but integrated around a core domain ontology, as proposed by the methodology in [39]. In detail, the core ontology should contain the common vocabulary of the tourism sector and can be extended by other ontologies in a modular way, such as ontologies for modeling time, location or user context. For further detailed information on the approach of modularized ontologies in tourism domain see [14].

Do we need an ontology for personalization (trip planning, recommendation) as TAGA, GETESS, Hi-Touch do? This question brings forth another one: why are machine learning algorithms such as k-nearest neighbor approach or collaborative filtering methods that are based on collecting and analyzing a large amount of information of users' behaviors not considered? Tourism project aims often can be reached more efficiently, if they are using other approaches, such as data mining methods (i.e., classification and association rules, which are also well-established on textual data), or information retrieval methods that provide a flexible access to information and provide a lot of well-tried mathematical models (like model 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 newly, unknown information in documents).

Do we need an ontology for complex search queries, as ebSemantics, DERI, e-Tourism and QALL-ME do? Possibly, as ontologies can help to produce more meaningful and accurate web site content. But this not low-cost option as enabling the formulation of complex search strategies requires a lot of content preparation. Firstly, the information provided on different web sites must be annotated with ontology concepts. Because each resource is structured differently, wrappers must be manually crafted for each individual data source. This task is a labor-intensive task, requiring many test-and-debug-cycles. Secondly, a mapping mechanism to ontology concepts must be available, and thirdly, a standard vocabulary is needed for general use in the tourism domain. Search engines can use ontologies for term matching (cf. preparing data for the normal web) and use them to handle more explicit queries accurately. But good-old indexing mechanism, and common mathematical information retrieval models cannot be replaced by ontologies. Indexing models do not, however, guarantee any depth to the search unless the deep search terms are visible at time of indexing. Also, where the searchers are unsure of precisely what they are searching for, their intention is effectively unclear, non-semantically-based search is far more likely to lead to false positives and negatives.

3.2.2 What have we learnt from this?

While ontologies are very useful in this application domain, there are many business concerns that subvert the attempts to form a common ontology. A fixed and formal understanding that, for example, a three-star hotel in Austria was equivalent to a four-star hotel in Greece would have a significant impact on businesses and consumers. It would be difficult to convince an entire country's tourist industry to place itself in a "second-best" situation, or upgrade their hotel ratings, over a short period. Such a systematic correction would, most likely, require centralized legislative oversight to achieve, and only then after months or years of debate.

This, it is not surprising that unification has not occurred, but there is still a place for ontologies, with mechanistic assistance for alignment and machine-learning mechanisms, to achieve a higher degree of alignment.

3.3 Case study 3: large-scale data curation

What happens when, as part of the need to curate large volumes of data, we need to choose between the production of a taxonomy or an ontology? We have already addressed the fact that incorrect selection will lead to either reduced expressiveness or excessive complexity of production, thus, matching the semantic requirement to the appropriate level of effort. Case Study 3 addresses a project that, while it was considered by the stakeholders to be ontological, was not based on an ontology. Despite being a large-scale knowledge representation project, the so-called ontology was a simple taxonomy that had been mislabeled and was not performing any ontological function.

Long-lived information systems accumulate large volumes of data over their lifetime. Such systems must also provide efficient indexing and curation mechanisms to ensure that all data in the system remains available to answer queries. As the development of semantically grounded storage systems is relatively recent, any long-lived information system is likely to have been built within a database management system (DBMS) and a relational DMBS (RDBMS) in particular.

While RDMBS have a number of advantages, the storage format is rigidly structured, depending upon fixed columnar sizes and positions of fields in records to place a valid interpretation over the digital information stored in those positions. An obvious solution to make this information available in a more accessible form is to export the records, and their relationships, in a form such as RDF/XML. However, this exported data is useless without a key to interpret the RDF in a way that allows the RDF to convey the same syntactic and semantic relationships that were, potentially implicitly, codified in the RDBMS. An overarching taxonomy or ontology, as previously discussed, provides the required key for a querying agent to make well-defined queries across the exported data corpus.

In this case study, a national database storing gigabytes of complex, multi-faceted data with a high degree of cross-class relations is discussed. The existing database is managed centrally but is curated in a distributed fashion, with stakeholders making data additions and modifications across their areas of authority. Browsing access is available to any users capable of opening a web browser.

Recently, the decision was made to make the database available in a shareable and extensible format to facilitate research by other groups who did not have the same level of access to the RDMBS software as the managing group. There are obvious issues with allowing more than one group simultaneous supervisor access to a shared database at the fundamental level. As it transpired, an existing, and parallel, project was already exporting some of the data from the RDMBS into an RDF/XML format to allow the analysis of the data for network interaction, and this work was nominally built around an existing ontology describing the key relationships between the essential classes (tables in the RDBMS).

However, when the original ontology was requested, to provide the basis for development in the new project, it became apparent that no formal ontology had been produced. The "ontology" was a list of headings and sub-groupings, with some relationships defined, but it had no range, domain, functional relationship, or inclusion/exclusion criteria. Synonym properties had been asserted with the FOAF ontology, but no formal statements on these properties had been made. While this is a meta model, and a representative data schema, this does not meet any of the requirements that constitute an ontology.

At this stage, it was obvious that the business logic that was employing the ontology could not have been using it in any mechanistic or machine-interpretable manner. On inspecting the code, the implicit semantics that were partially described in the "ontology", and that had been derived from the RDMBS master, had been hard-coded into the RDF analysis code that was being used to drive the export. Similar experiences were identified in Case Studies 1 and 2, where relationships and semantics were hard-coded and non-extensible or interpretable.

The software owner was unaware that an actual ontology was not in place, as the relevant code was working and produced output that was consistent with the concepts that should have been contained within the ontology. However, given that ontology production is a slow and time-consuming activity, the lack of an ontology allowed the project to be completed relatively quickly, giving a false indication of how long such a project should take. This is the balancing concern to the "wasted effort" scenario posited in Case Study 1. Rather than abandoning projects because they are seen as too hard, a project may be abandoned because it is seen as taking too long compared to a previous project that never delivered the objective.

To clarify the key problems identified in this project:

1. *Taxonomic confusion*: the term ontology was being used when a simple classification hierarchy was being employed.
2. *Loose definition*: the taxonomy itself was incomplete.
3. *Implicit semantics*: the implicit semantics of the RDMBS had not been formally quantified.
4. *Lack of abstraction*: an additional layer of implicit interpretation had been hard-coded into the business logic of the executing code.
5. *Planning skew*: the project as described was not what was delivered, and had had the most time-consuming component removed.

The taken remediation steps included the formalization of the taxonomy in order to provide range and domain information, as well as formal class hierarchies and the separation of object and data properties. What was identified was that a full ontology was unnecessary for this application, but this was identified from both user requirement and pragmatic system limits.

The user requirement was for a strong classification of the terms in the RDMBS in terms of their structure and their relationships. The RDBMS system was not designed to be extensible in terms of missing classes, or necessary relationships that had not been asserted, hence the use of an ontology for inference was unnecessary. This was also supported by reflection on the system's user base and typical use pattern. Given the size of the RDMBS, the equivalent generated triple store would be very large and any reasoning taking place over it would be exceedingly slow, despite the available power of the underlying server. Users were expecting to receive query results in very short time, and to be able to add data in a short time, with immediate available of added data. The addition of a reasoning step for this live data, given that no new class inference was required, was seen as being unnecessary. This removed the taxonomic confusion and allowed the delivery of a taxonomy as a valid solution. Also completing the taxonomy and encoding the data correctly addressed the loose definition and implicit semantics problem. Modifying the existing code to work with the new information model addressed the lack of abstraction requirement.

The production of a taxonomy is far more straightforward than that of a correspondingly sized ontology but does take longer than a hard-coding based on implicit semantics. However, time spent in implementing this is returned when future development is undertaken and the lifespan of this project was measured in decades, hence more effort was warranted. This addressed the planning skew concern by injecting real planning data back into the management model.

In this case, while an ontology was not necessary, the perceived natural extension of preparing data and schema for sharing was to develop an ontology. However, this was never achieved and, to understand why, we must remember that the sharing point had not been reached. While RDF conversion was in use for another project, only the products of the RDF conversion were being shared, not the RDF itself. Thus, the ontology never had to be produced until the actual sharing point started to loom in the near future.

3.4 Summary of the case studies

We present these case studies to clarify the reasons behind ontology use and the influence of business model and user perception on the resources devoted to these projects, and their ultimate success.

Some businesses wish to share no data externally, but need intra-business alignment for efficiency. Without the additional requirement to make their representations work with a greater community, these representations risk being pale shadows of existing business practices. These are neither visionary nor agents of change and, unless mandated at a higher level, may not be used extensively, maintained or developed. This is not going to provide the strong motivating case for widespread adoption of ontologies.

Some businesses strongly want to share data but cannot easily agree on alignment, or are so focused on their "unique" approach that their sharing is limited by the lack of alignment and multiple semi-clones of the representation are produced with little re-use or efficiency. This is a waste of effort and further frustrates efforts to promote ontological adoption.

The third example demonstrated a business where sharing was paramount, across a wide sector, and in a collaborative manner, yet lack of understanding of what was involved prevented the mature project from developing smoothly as some, quite understandable, shortcuts had been taken in the early development processes.

A final business model, that we did not discuss, is where global sharing, with an open view to adopting other processes, has been used to drive an efficient and effective ontology production, with the produced ontologies forming the basis of a large volume of interactions as part of an active and involved knowledge community. While this ideal is our desired goal, we must ask: "Where is this business, or family of businesses, outside of 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 any idealized view of global information sharing, without borders.

While biological sciences, and especially bioinformatics, have been identified as strong users of shared ontologies based on genomes and taxonomic classification, this is not surprising as these disciplines depend entirely upon these fundamental scientific concepts. A biological scientist needs taxonomies to classify life forms on a day-to-day basis, and the ontologies built for genome-based classification are a new toolset for the same problem – analogues and affordances drive adoption. Adoption in this community is, sadly, not a sign of wider spread acceptance as there are two key considerations in this sphere:

1. Scientists must share their knowledge in the form of publication to validate their claims and expose the results of their research, especially in drug and germ classification.
2. Without sound taxonomic basis, the publications will be immediately disputed, especially where strong claims are made.

For biological scientists we finally have a powerful application for ontologies, but it is not for transferable reasons. Other reasons to bring other communities to ontologies must be identified. Thus, there must be some perceived benefits of ontologies in association with certain data operations, or sharing activities, but it must be ensured that designers understand what these are, in detail, before proceeding.

The next section discusses the benefits of ontologies in detail.

4. Ontologies – the benefits

If ontologies are going to be used, then it should be for the right reasons. The word "ontology" should be more than a buzzword and it should certainly be more than an incorrect synonym for taxonomy. To assist in making the correct decision as to whether ontologies should be used or not the following key points need to be considered:

- What are the benefits of ontologies as concepts?
- Is reasoning the only advantage?
- Where is the profit in applying ontologies?

Benefits cannot be considered solely from a technical point of view, as purely technical arguments are not sufficient to convince business to adopt a certain technology. Business must also be provided with arguments that will 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 for this technology. By comparing technically superiority-based benefits with what users consider to be important, there is the possibility of uncovering the missing decision weight that would lead people to choose to correctly use ontological or taxonomic technology.

From a technical point of view the usage of an ontology can be of benefit for the following reasons:

- *Communication*: an ontology enables communication between systems, between humans and between humans and systems. Reducing machine translation steps is crucial. The dual readability is the greatest justification given for the use of formats such as OWL (which of course introduces problems of its own).
- *Identification*: unique Identifier, the concept of URIs, which uniquely identifies the meaning of concepts in a given domain of interest. Furthermore, URIs enable the reuse of an ontology. A fundamental problem is the reification of a concept, tying it to a physical item or real-world concept. Without reification, we have no real grounding and we cannot apply our reasoning to business. The existence of URIs allows us to tie our systems to the real world.
- *Domain Analysis*: to make domain assumption explicit, to share a common understanding of the structure of information and the intensive analysis of the domain. Ontologies are also the means to structure and organize knowledge, not only data.
- *Knowledge Transfer:* to facilitate knowledge transfer.
- *Reusability of Domain Knowledge*: to enable the reuse of domain knowledge and also to build upon existing knowledge to integrate it into a new knowledge caucus. The OWL mechanisms are very powerful for this and allow us a form of inheritance that is a bit fragile but still very usable.
- *Inference*: an ontology enables computational inference, which is in turn useful for deriving implicit facts.
- *T-Box/A-Box Separation*: clear separation of the ontological schema and its instances. OWL DL requires separation of classes, instances, properties, and data values, in opposite to OWL Full, in which classes can be instances or properties at the same time. In addition it is possible to identify those aspects of ontologies that fall within DL boundaries and those that do not.
- *Standardization*: The usage of a uniform language that enables protocols. There is an implicit bootstrapping problem, because everyone has to agree to an initial lingua franca in order to be able to standardize around it.

However, different user surveys (2006-2009) [5, 7, 25, 34, 36, 37] of academic and industrial research institutes and business concerns have identified benefits from a user-centered view using an ontology:

- different user provide shared knowledge (cf. knowledge transfer),
- enable knowledge reuse (cf. reusability),
- make domain assumptions explicit (cf. domain analysis),
- provide intensive analysis of a domain (cf. domain analysis),
- provide a clear separation of operational and domain knowledge.

4.1 Discussion of the different perceived benefits

The differences between the benefits of the technical point of view and the user-centric one reveal some interesting and explanatory aspects, why ontologies are in many cases misused. Figure 1 subsumes both views, the technical and the user-centric one, and shows the intersection of them.

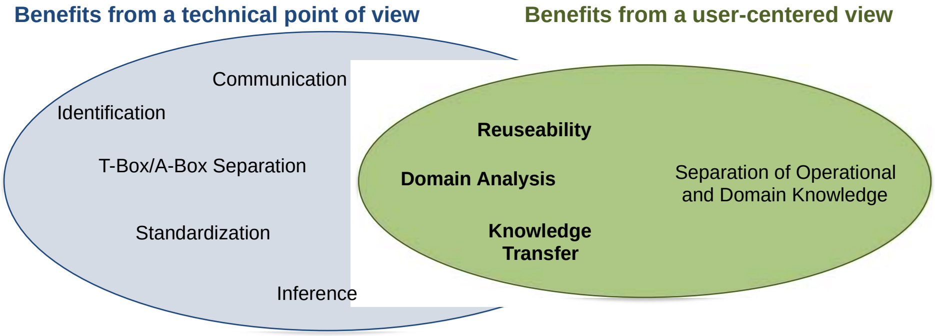

Fig. 1. Intersection of both views, the technical and user-centered one.

It is particularly noticeable that "inference", a key technical 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 in considering the complexity of a technology that would reduce cognitive load is a barrier to entry. Users would have to be aware of the complexity of the reasoner, its technology and the ability for a reasoner to substitute for missing experts in this area. Due to the misuse of ontologies as a taxonomic tool, their inference capabilities are rarely used, as there is insufficient reasoning and conceptual framework in the ontology.

The direct comparison of perceived benefits raises further questions.

Are the perceived benefits based on the correct technology? A survey of existing papers dealing with ontologies in the literature reveals that a great number of these papers are more accurately papers that deal with taxonomies. Many of the so-called ontologies are not validated, fully expressed or utilized in the appropriate manner. When 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 never used for reasoning or inference. Thus, if the necessity for expressiveness or decidability is understood in the context of the level of reasoning support required, then it is possible to successfully navigate the requirements and arrive at the sufficiently rich and expressive taxonomic language necessary for a given project.

However, if decidability is truly required, the expressiveness is heavily limited and it is very difficult to move ontologically limited systems into truly ontological frameworks. Given that the goal is the explicit capture of the domain, its entities and their relationships, any movement towards a partial solution that is less than truly expressive risks reducing the accuracy of the model. A similar analogue exists in graphical representation, such as those used for cartographic maps. Consider the path taken to move from one city to another, for example Linz to Vienna. There are many maps that could be used to see the position of these cities and the paths between them. The highest-level representation is, of course, the globe itself (cf. Figure 2).

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

To see this as a two-dimensional representation, suitable for assessing distances, a Mercator projection may be employed. This simple transformation to a two-dimensional model has already introduced well-known distortions into the actual surface of the globe. This is still not sufficiently accurate if the goal is to drive from Linz to Vienna, a road map, overlaying structures onto the 2D projection, is now required. If walking is the activity, then further detail is required, including the provision of trail maps for those areas where footpaths were not identified as part of roads. At each stage of this process, detail is increasing, as is the complexity of the model, until the desired level of complexity is reached. The "ground-truth" of this model is the reality itself, the atoms and atomic interactions that provide the paths upon which we walk or drive, but this is far beyond the level of complexity that is needed to achieve the goal.

A desire for accuracy must be tempered by a willingness to invest the time required producing an accurate model. Using a too complex model, such as trail maps for a driving trip, is inefficient as the wrong level of focus may be selected, leading 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.

Once a model has been created from ground truth, the limitations of the model must be clearly identified. It is as accurate as it has been constructed to be, and no more. If a more accurate model is required, the original must be consulted again. What impact does this 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 is far too complex, and wastes resources for no real benefit. Worse, in computational terms, a complex model is far more daunting and may not be tractable on the available technology. Again, when modeling a system into an ontology or taxonomy, if the full expressiveness is not captured on the first pass, a less-expressive system cannot be used to reconstruct the master domain but, instead, we must resurvey the original. A domain analysis and detailed conceptual design phase will reduce the possibility of a misleading design, but only if the level of expressiveness is chosen appropriately, otherwise the risk increases that important details will be ignored or design time wasted on trivialities.

Data modeling meets the same challenge; if in the conceptual design phase the modeling requirements are insufficiently defined, then it is nearly impossible to build a model in an appropriate abstraction level. The correct use of the abstraction concept demands for long-time experience and expert knowledge. The challenge can be seen in the following example, which describes the real word concept hotel room from two different views: a more abstract view is the view of reservation (expected date of arrival/departure, number of persons, extra bed for kids y/n, pets y/n) and a more specific one the inventory view, where each individual piece of furniture and interior equipment (like mini bar, TV, hair dryer) gets relevant for the data model. 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 for ontology design crucial, that means, that there is no reason to reinvent the wheel, but rather to use data modeling as best practice for ontology modeling.

What are users missing? Users are not always aware of the difference between these technologies or, at least, are not able to adequately express or understand the difference between these technologies as they are currently presented. To clarify user understanding and requirement, the following questions address a user's perception of satisfaction with taxonomies and ontologies. Answers to these questions may assist in identifying ways to increase the ubiquity of ontologies.

- These surveys have many questions but inference and reasoning are not. Inference mechanisms and reasoning are the two most important benefits of applying an ontology. To not ask this question means that no information can be collected on the ontological efficacy of the systems. Thus, the community must develop both understanding and practice in the application of an ontology so that it is obvious that these two aspects are always recognized as a benefit.
- Is it too complex? Data modeling and software engineering provide 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 project failure risk increases. Similarly, ontology engineering is potentially prone to attempting to produce an overly complex solution because an early focus on what is actually required has not been applied.
- Is it invisible? When the underlying technology is visible, the system is perceived differently and users must be trained to work with the visible framework. A successful ontological basis to a project provides all of the advantages of semantic classification, but without the burden of all users having to interact with the ontological underpinnings. Consider the Internet Protocol v4 address range and the Domain Name System. Correct and widespread use of names instead of addresses provides a useful abstraction that al-

lows a change of underlying technology (IPv6, in this case) without having to relearn a new set of names.
– Is the technology missing from most implementations? Are ontologies being used correctly or is the implementation a name and some fragments of a true approach?
– Are there missing experts? Consultation of experts is an essential step in mapping the domain and, if a key expert is missed, then only a partial model can be formed.
– Is there a missing of natural integration of structured ontology engineering procedures into modeling process?

Moreover, these are the points addressing a possible solution and finally leading to better acceptance of ontologies. Application of an ontology engineering methodology/method is – as it is standard in data modeling and software engineering – essential. Moreover, without a suitable ontology engineering methodology/method a project will likely fail, if compared to one, which follows a specific methodology/method. But before addressing the ontology engineering aspects it is important to be able to characterize the technologies in a way that encourages understanding of users in which technologies to use and knowledge sharing. The next section contains a discussion of the fundamental differences between the technologies and provides a clear illustration of where each technology can be classified.

5. A model for discussing ontologies and taxonomies

Several approaches of classifying types of ontologies are introduced. One research work that has to be pointed out is the classification of Hepp [24]. Hepp classified not ontologies but rather ontology projects by following six characteristics: (i) expressiveness, (ii) size of 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. This classification indeed supports in setting-out if an ontology is an appropriate solution for a specific project but does not support in selecting an appropriate technology. Thus, a more abstract classification that characterizes the differences of technologies is required. By any definition, taxonomies deal with structural relationships and the ontologies focus on conceptual relationships. Drawing a diamond-shaped diagram, with these two relationship criteria shown increasing as they moved up, provides us with something along the lines of Figure 4. Ontologies are clearly shown at the peak of diagram, being the most explicit representation of the classification and structural relationships known. At the other extreme, textual specifications, which have no machine-interpretable form or explicit structure or classification requirements, are found close to the origin. Structure without a great deal of conceptualization is shown with taxonomies, with the explicit grouping of concepts into an unstructured bag of concepts is identified with object-oriented classes from object-oriented programming.

While this initially appears to be a good classification, one extremely important concept is missing: that of the explicit conceptualization that is seen in strong data models. It is not enough to say that concept c_1 is like concept c_2 and that c_2 may be classified as a c_1, but rather it must be possible to express the conceptual relationships, which build-up the base for developing an expressive ontology.

The interaction between explicit conceptual representation, explicit structural representation and explicit classification representation can be shown in terms of a Venn Diagram (cf. Figure 3).

While this clearly shows, where each technology may be classified, it is not sufficient for assessing the effort required to successfully add each layer of explicit representation.

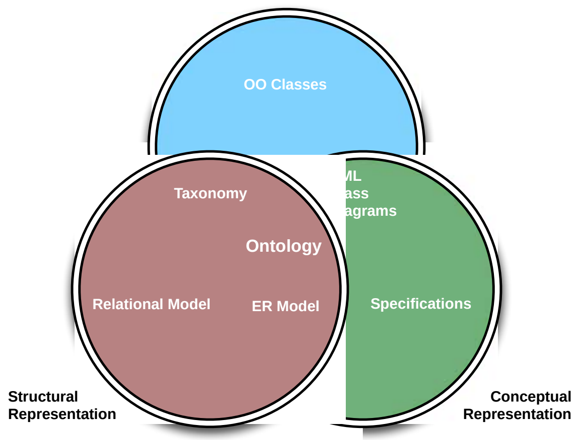

Fig. 3. Venn diagram that represents the interaction between explicit conceptual representation, explicit structural representation and explicit classification representation.

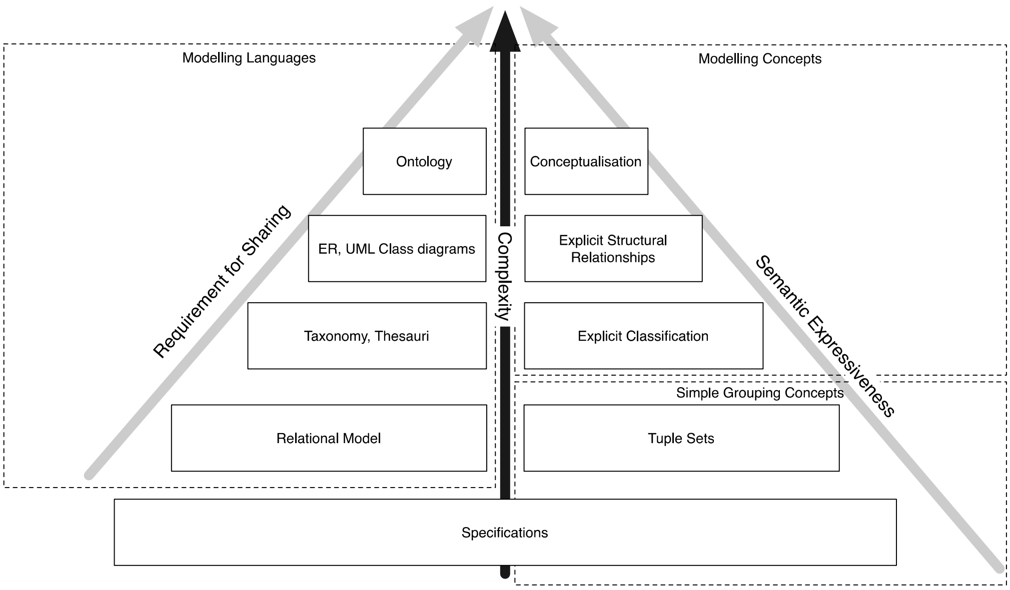

Fig. 4. Relationships between existing technologies.

The final diagram is recognition of the discrete nature that must, implicitly, be accepted in order to choose one technology over another. Figure 3 shows the relationships between the existing technologies. The expressiveness requirements have been included, as well as the requirement for sharing, as clearly defined arrows showing the direction of travel required to achieve these aims. Running from the bottom to the top is an arrow indicating the increasing complexity of the models to show the progression from specification to ontology. Finally, the diagram clearly separates the tools, on the left hand side, from the concepts, on the right hand side. The right hand side is further broken down into simple grouping concepts, at the bottom, and true modeling concepts that extend these simple groups to show relationships and more mature conceptual arrangements.

As would be expected from such a diagram, this also gives an indication of the level of effort required to achieve higher goals. Many specifications may exist, but it requires more and more effort to produce a correctly formatted ontology. The benefit however, is that the knowledge is now in a form that is conceptually explicit, capable of being shared and semantically expressiveness. This subsumes the purely structural requirement that was originally defined as an axis, as consistent and rigorous structure is a component of all of these characteristics.

From both a structural point of view, given the additional overheads required for semantic expressiveness and rigor in contemporary ontology languages, and also for the intellectual complexity required to truly capture the system knowledge, it requires far more effort to produce an ontology than it does to capture the same structure taxonomically.

While the magnitude of effort required to achieve a given level of modeling cannot explicitly be quantified, without further research, a list of considerations can be provided that should be addressed to assist in selecting the least complex and most expressive technology required for a given application. The list is:

– *Degree of sharing*: ontologies are at their most useful when assisting in the sharing and re-use of representations between a number of organizations. Outside of a company framework, there is no easy way to communicate disparate requirements and align across different organizations without an ontology. Thus, as the need for sharing increases, the likely requirement for an ontology increases.
– *Number of objects*: with a very small number of objects it is very unlikely that effort spent in a complex modeling process will be rewarded with any cost or resource savings. However, a large number of objects is complex to manage, has the likelihood to have of a large number of relationships and to demonstrate a variety of conceptual modeling problems. The larger the number of objects, the more likely it is to be rewarded by a greater investment of effort in a more complex modeling solution. Of course, if

the objects are all identical instances, and there are a small number of classes, complex modeling may not be as important.
– *Degree of re-use*: the effort invested in modeling is non-trivial and becomes greater as progress is made up through the complexity layers. If effort can be re-used a large number of times, costs are defrayed, lowering the cost of producing a more complex system.

The ideal candidate for an ontology is a highly shared, large and re-usable system. This, again, clearly illustrates why bioinformatics is such a heavy user of ontologies as the genome ontologies meet all three of these requirements.

Now a framework has been provided for measurement and comparison, and the beginnings of a mechanism to allow defensible selection of one technology over another. Furthermore, critical aspects in modeling ontologies must be reflected, thus in the following two sections ontologies are compared with other conceptual modeling techniques, in order to determine on the one hand the relative benefits and on the other hand to pick and choose aspects from which ontology engineering may benefits and therewith the user acceptance. We still have to identify the critical aspects that will comprise our final engineering process and this provides another parallel with conventional data modeling and software engineering, to see how we can adapt the lessons from these areas for ontology engineering.

6. Comparison of ontology engineering with conventional conceptual modeling

Other modeling approaches, such as entity-relationship diagrams or UML class models also enable the analysis, structuring and the organization of a domain of interest [17, 30]. In ontologies, the focus is upon a knowledge framework that classifies new entities of known taxonomic components, and extends, in an internally consistent manner, to handle new entities that were not defined at the original time of writing.

The classic OWL Pizza tutorial[5] is an excellent example of this as it is possible to invent new pizza types and add them into the model while remaining consistent with the old model. Significantly, the existence of classes can be inferred that were not explicitly created initially, identifying predictive aspects of the model. However, UML and ER also have the advantage to extend a model in a simple manner, by the use of the generalization concept, as may also be found in any other conventional data modeling approach. Therefore, by elimination, reasoning and prediction must be the key benefits of ontologies.

The previously mentioned surveys also specify barriers in modeling and using ontologies, but these challenges also must be tackled if using a specific modeling approach, such as modeling a data mining application. In this case an in-depth analysis of the domain is necessary, which requires an amount of time and human resources, including domain experts, knowledge engineers, and data mining experts. This is an important point: the production of ontologies is not simple and the process of modeling an ontology is both time- and resource-intensive. Points that result from the surveys [5, 7, 24, 25, 36] concerns regarding the comparative resource investment required to implement an ontological solutions over a conventional data modeling solution, such as ER or UML. One main advantage of ER is that the generated models are easy to understand, and therefore they can conduce as communication medium between several technical and non-technical stakeholders in a company. Quality criteria as expressiveness, easy applicability, minimality, and a theoretical fundament are the reasons why the ER model survived from 1976 until now. For ontology design there is no similar conceptual model similar to the ER model available. Because of the lack of understanding of the benefits that can be gained, ontologies risk being seen as too abstract to implement, and too far from useful modeling.

The definition of ontologies must extend beyond "good taxonomies", the role and importance of ontologies in practical applications are to be clarified. However, whichever definition is used must be easy to understand and couched in terms that technical and non-technical users alike can understand. Reviewing the important facts regarding ontologies reveals that many other modeling approaches, like ER models or UML, enable the analysis, structuring and organization of a domain of interest. However, ontologies provide a knowledge framework that not only classifies new entities of taxonomic components, but also can extend to handle new entities undefined at time of production. They provide in comparison to conventional modeling techniques a (i) consistency check, (ii) inference of new classes (reasoning) and (iii) predictive analysis (prediction).

Consequential, following demands on the conceptual modeling of ontologies are made:

[5] http://www.co-ode.org/ontologies/pizza/2005/10/18/pizza.owl, last visited: July 01, 2014.

- detailed domain analysis,
- more time, especially for considering reasoning and prediction aspects in design,
- greater demand of personal resources (domain experts, knowledge engineers, ontology experts),
- support (tool or experts) to guide the design process of knowledge framework.

These demands are nearly equivalent to those of conventional data models. The reasons why ontologies are not yet ubiquitous, thus, not only underlies an unsuitably selection of modeling technology, but rather is caused by (i) unknown (or unavailable knowledge about handling of) ontology modeling process, (ii) lacking support in modeling process because of missing state-of-the-art modeling tools, and (iii) missing longtime experience (cf. ER modeling is more or less state-of-the-Art since Chen proposes this popular kind of conceptual data modeling).

A simple notation such as the Chen Notation for ER modeling would likely increases the use in business scenarios. Therefore, the authors of the research group at the FAW institute of the Johannes Kepler University have started to develop a graphical notation for ontologies. The notation provides 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 [6] for transforming the ontology model into a formal ontology language (such as OWL DL, RDF, Manchester syntax) are proposed.

Furthermore, modeling an ontology would gain in popularity if its process model bases on well-established procedures, like data modeling or software engineering ones. In the next section ontology engineering methods and their comparison to software engineering ones are discussed. In addition, aspects are discussed why several software engineering models are state-of-art and yet ontology engineering does not have an accepted state-of-the-art model. [29]

7. Ontology engineering: a new development or just a new name?

As for software development software engineering models are indispensible, so ontology development necessitates also an appropriate model that leads – providing that suitable model is selected and is correctly employed in engineering process – to a formal, correct and satisfying ontological model. Gómez-Pérez et al. [19] prepared a comprehensive work about ontological methods and methodologies. Software engineering is a well-established and practically adopted discipline that has changed the way that software is produced world-wide. There are three areas where an ontological engineering discipline can be of assistance:

1. The recommendation and selection of the most effective model.
2. Bridging the gap between the model and its application.
3. Re-use of ontologies, with their resource-intensive production process.

To place this in context, software engineering is discussed to provide a comparison for those techniques that have emerged in ontological engineering.

7.1 Software engineering

Software engineering is now in its maturity and has progressed well beyond initial discussions of the waterfall model in 1970. Software engineering is well understood and can be taught as approaches, selection and specific methodologies to undergraduate students. Most, if not all, computer science and Engineering Schools will discuss software engineering as a fundamental discipline and expectation of practitioners. It has a set of key principles, one of which is *re-use*. All of the effort expended to build a component should be re-used as often as possible to achieve the maximum benefit from the effort. While this is simple in principle, many companies will have extremely similar code libraries and can and will share APIs but is certainly neither seamless nor easy to manage. Offsetting this, the majority of commonly used modules are likely to be sufficiently small that the reconstruction cost (reinventing the wheel) may be absorbed within a company without excessive visible cost.

7.1.1 Re-use: the bane of ontological engineering

The main question here is why there is not more reuse? Given the importance of saving money and effort in both business and research, reuse is an obvious way to make the best use of limited resources. Yet, ontologies are not being reused. Given that this is the single most likely factor to reduce cost in de-

[6] Similar to transformation rules required for transforming an ER-model into a relational model used in the field of relational databases.

velopment, why is reuse not a dominant driver in this area?

- Existing ontologies are often built for a single purpose from a single business focus. Ontology development is carried out in isolation with each business reinventing the wheel (as seen in case study 1).
- While tools exist, these favor the small-scale, application-oriented ontologies.
- Tools and concerns over Return On Investment (ROI) tend to get in the way of building large, comprehensive ontologies. People build what they need now, and to the lowest expressiveness level acceptable, rather than building a representational model that can be extended. A similar behavior can be seen in software engineering approaches, especially in agile development methodologies.

If an ontology is the right solution, then the effort is worthwhile and ontological work carried out at the start of a project may save a great deal of effort later, as conforming with an existing schema is considerably easier than carrying out a post-hoc alignment across multiple disparate sources. However, the expressiveness to be captured, or the level of sharing desired, may be limited by the time available and the processes employed. The case studies have already shown how desire for an ontology, even with well-defined need, does not easily translate into a successful implementation.

A key problem facing ontology development, at the start of any development process, is determining which concepts may be reused from previous attempts and which are unique to a given domain. Many businesses consider their processes and focus to be unique, despite strong evidence to the contrary. At the same time, naïve ontologists may look 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 high-reuse ontologies. However, even within the same business, many concepts are used in different ways in different points of the business. This requires the establishment of manual or, at best, semi-automatic alignment processes that are labor intensive – once again, this increases both cognitive and kinesthetic load. Manual alignment for reuse is, once again, an ROI issue.

People who wish to capture and represent their data and its relationships often do so, because they are convinced of the worth of their particular representation. Many businesses want to innovate because "their way is better" and thus their ontologies must be built, from the ground up, using concepts that make sense inside their conceptual context, without regard to pre-existing solutions. This was discussed in case study 2, where tens to hundreds of nearly-identical solutions exist to exactly the same problem.

When thinking about reuse, this naturally leads to the question "How are ontologies built?" Up to 80 % of survey respondents were not using an ontology engineering methodology [7, 34, 36]. This is not surprising as "ontology engineering" is in its infancy, barely registers as a search term on Google. Even the SemanticWeb.org page on ontology engineering, the theoretical champions of this discipline, has been idle for almost two years, with a brief update in June of 2012.

Discussing ontological engineering assumes that it is, in fact, possible to engineer ontologies: ontologies can be constructed by clearly defining what is to be achieved, measuring inputs and providing a predictable and well-instrumented process for delivering outputs. In order to justify an ontological engineering discipline, it must be demonstrated that it is possible to, decisively, separate all of the technologies discussed in a predictable, deterministic and rigorous manner.

Ontology re-use has already been identified as a problem, often because there is no real mechanism for detecting relevant ontologies that are useful for reuse. Ontological repositories exist, such as the Watson semantic web explorer[7], which is also available as a plug-in to the NeON ontology development environment[8]. However, the Watson search can only search across known, and shared ontologies, and uses keyword-based search. Even with format-aware extensions to this search, to allow searching for the keyword in certain syntactic constructs, there are still two major demands upon this search tool: that the correct word is chosen to match what the ontology's creators have used, and that they have chosen to make this ontology available. Hence, while Watson addresses some of the concerns, another bootstrapping problem emerges when re-using existing ontologies.

There will be a significant return on investment, if ontologies can be reused. Despite this, less than 50 % of new ontologies are reused within the sets of existing ontologies [37]. Ontologies built for highly spe-

[7] http://watson.kmi.open.ac.uk/WatsonWUI/, last visited: July 01, 2014.

[8] http://www.neon-project.org/nw/Ontologies, last visited: July 01, 2014.

cific purposes may not be suitable for re-use but many ontologies, including those described in the case studies, are suitable for re-use but, clearly, this is not happening. Missing collaborative tools for ontology-development can lead to the building of smaller and more application-oriented ontologies, at the expense of standardized (and comprehensive) ontologies.

Again, that there is no World Ontology, in the same way that there is no one store of ”true“ algorithms that can be composed to solve every programming problem. Of course, where there is a shared domain of interest, there is a shared requirement to establish vocabulary.

7.1.2 Making re-use work: aligning re-used software with new software

In both software and ontological engineering, there is overhead in re-using components, even if they can be located easily. Either the new software must be written in a way that conforms to the old or it must be possible to re-engineer the re-used components easily to interface with the new modules. This burden, of manual alignment, is, once again, a return on investment issue. The re-use burden is, therefore:

1. The time it takes to find the component in the re-use store.
2. The time it takes to integrate the component with the new software.
3. The time it takes to convince all interested parties that this new problem is not so special or unique that all software must be developed from scratch.

Consider, once more, the Watson Ontology finder. If the ontology, or the component that week, is there to be found in the first place, the correct search term must be used to locate it. Thus, the time spent in step 1 may be quite large and, if the component is not there to be found, effectively boundless. A common business and software development view is that the processes in use for a given organization or program are somehow superior, or different, to the competitors. Thus, item 3 may have surprisingly high weight and, if the re-use burden is high enough, the motivation for widespread reuse fades away. The final re-use burden is strongly affected by the motivation of other parties to share, software engineering practices to facilitate re-use, and political issues.

7.1.3 Adopting robust engineering practices

Whether the term is truly understood by all concerned, there is no doubt that any business wishing to develop software commercially will be able to clearly describe what its software engineering practices are and, in the vast majority of cases, be able to display the relevant software engineering methodologies. Adoption is almost complete within the software producing profession.

By comparison, the surveys clearly show that the majority of those surveyed (60-80 %) do not use an ontology engineering methodology for modeling their ontologies [7, 34, 36]. Some of the development environments do support a design methodology for ontologies, such as NeON, but this is not universal. The benefits of using a methodology are obviously not clear to all. Much as a lack of software engineering discipline can lead to poorly-defined production processes and hard-to-maintain software, failing to observe the correct discipline when developing ontologies has a similarly negative impact. The incorrect evaluation of the effort required to correctly produce ontologies might result in complex and confused activity, the incorrect handling of complex production methods, and the production of second-rate or inefficient support tools.

7.2 Ontology engineering

Ontology engineering is not a well-defined discipline, despite the best efforts of those producing carefully constructed tutorials to lead by example, as insufficient effort has been made to communicate with non-technical users. Similarly, within the emerging discipline of networking engineering, engineers try to bring the strong practices of software engineering to the large-scale ISP networking arena. This includes formal objects representing networks, network routing algebras and graph representations. However, before considering formal methods as a possible solution, it is necessary to be realistic in the assessment of network management, which is, to a great extent, self-documenting, with well-defined and standards-based components interacting in well-understood ways.

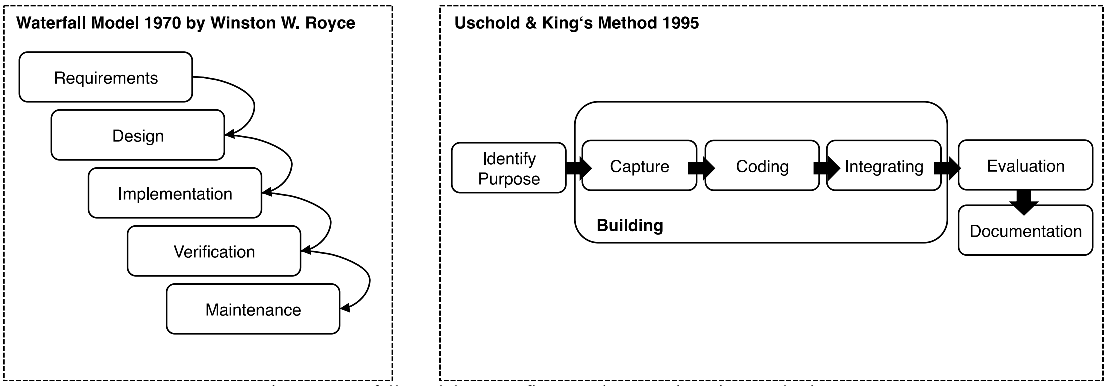

Fig. 5. Waterfall model versus first ontology engineering method.

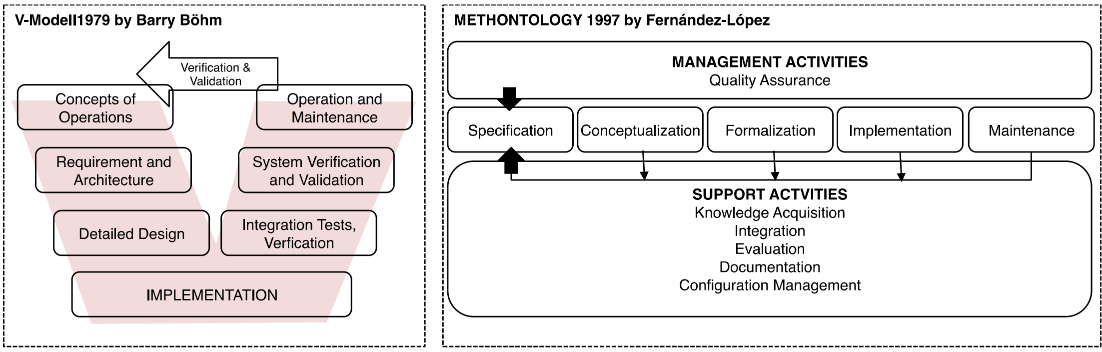

Fig. 6. Extended software engineering model versus extensive, novel methodology for ontology engineering.

The knowledge of the network is relatively simple; it is the instantiations of this knowledge base that are being manipulated. In ontology engineering the goal is to build an entire universe in shared discourse, rather than just a network built of well-understood components.

So the question is: "What can be learnt directly from software engineering?"

Despite the well-established software engineering techniques, ontology engineering methods are not applied. The following two figures (Figure 5 and 6) show the similarity between the software and ontology engineering methods/methodologies. One of the first software engineering approaches was the waterfall model, a sequential software development process. By examination of the literature, we can see that, some 25 years after the waterfall model, Uschold & King [43] proposed the first method for building ontologies. Figure 5 shows the strong similarities between both models.

The V-model, introduced by Barry Böhm 1979 is an extension of the waterfall model. The right side of the V represents integration of parts and their verification, leading to implementation. In the Ontological Engineering area, METHONTOLOGY [15], a methodology for ontological development and all-encompassing representation of a development lifecycle was developed within the Ontology group Universidad Politécnica de Madrid, 1997. In METHONTOLOGY there are also verification and validation activities, in form of management and support activities. This is similar to the V-model in many ways, but was developed 18 years later (cf. Figure 6).

While some might argue that the early software engineering methodologies are so deeply ingrained that it is natural to apply these in this way, the emergence of the more recently developed agile methodologies from software engineering is now being seen in ontological engineering. One example is eXtreme ontology design (XD). The main principles of XD are

- to understand the task and express it by means of competency questions,
- to reuse solutions, such as ontology design patterns[9], to evaluate the result against the task.

[9] The term design patterns is "borrowed" from the area of software development, Design Patterns by Christopher Alexander in the year 1964 vs. Ontology Design Patterns 2008 by Valentina Presutti and Aldo Gangemi as part of the NEON project [18].

What is most obvious from all of these observations is that, while software engineering is widely adopted and it is possible to work around poor distribution of components that leads to redevelopment energies being spent, this is not as true in ontology engineering. In ontology engineering, the components are more complex, and more effort has to be spent in rebuilding them. The most pressing need, however, is for the equivalent of the software engineering API, which is discussed further in the following section.

Where, in software engineering, there are well-defined models and known use cases, these do not exist to the same extent, or inspire the same confidence or discipline, in Ontological Engineering. The tourism case study clearly shows areas where ontologies are used, inappropriately and ineffectively, due to this lack of application knowledge and a well-formed discipline.

The languages used for software engineering have extensive community or commercial support, have in-built support for modular development and have rich and extensive Application Programming Interfaces (APIs). These APIs provide well-controlled access to language-specific libraries and, in turn, the libraries provide the core of a strong re-use platform. As an example, very few Java practitioners will write their own linked list implementation, unless needed for performance reasons, as one is defined in the API. Similarly, there is no need to rewrite core mathematical functions for most modern languages as these are also provided in the libraries and accessed through the APIs. Software engineering has a significant advantage: re-use components are automatically available to any person who uses a given language. This is evidence of a mature community.

There is a limited analogue of this in ontological engineering, as the well-defined ontologies and namespaces that define the core standards of OWL, RDF-S, RDF, XMLS and XML provide a common core of understanding. But, there is no single API that travels with all ontologies. The expression of a given concept requires us to locate the correct ontology, mandate its use and share it correctly.

Ontology engineering does not have a notion of the simple API signatures that facilitate shared library use in software engineering. Concepts are subtle, hard to capture and difficult to represent precisely. However, the methodology of software engineering is of little use unless the tools are of the same strength and applicability as those that have been developed for software engineering.

Having completed the review of software engineering techniques, with comparisons to some derived ontological engineering methodologies, it is possible to discuss the three principle aspects that should be brought to ontological engineering, namely:

1. The recommendation and selection of the most effective model.
2. Bridging the gap between the model and its application.
3. Re-use of ontologies, with their resource-intensive production process.

8. Choosing a model and appropriate tools

Choosing the correct model whether ER, UML or ontology, is a key decision that has a significant impact on the resources that are expended on the project and the success of the final outcome. Ontological engineering must provide us with a clear decision process to select the correct model, methodology and toolset to meet user requirements with the most efficient use of resources.

Industry has many concerns regarding the use and deployment of ontologies. These issues include:

– *Missing powerful applications*: the absence of a powerful application driving widespread adoption in the semantic web area, as previously discussed. Some powerful applications use similar technology, such as Google search. The community needs a powerful application that works over the top of existing systems and, where possible, does so with minimal additional work on the part of the user. Powerful applications need to be attractive to the widest set of possible users, not a subset. The semantic web, and ontologies, should not be reduced to the equivalent of gopher to the WWW.

– *Need of commercial applications*: ontologies are more applied in research prototypes than in commercial applications [24, 25, 34].

– *Missing cost-benefits analysis*: there must be some mechanisms to persuade business to adopt this approach.

– *Quality of ontologies*: the correctness of ontologies must be able to be measured and discussed. As stated, there is a balance between precision and recall, and precision is important. Ontologies must be reliable in their classification and in the consistency of the relationships that they provide. To assess the quality of an ontology, it is proposed that a directed search is carried out for a concept known to be in the ontology and confirm that the correct results are returned. For example, using the taxonomic relation “Linz is-a city”, it should be possible to formulate a query such as “city such as Linz”, and search for this phrase. If the resulting number of pages is less than a specified threshold, the taxonomic relation “Linz is-a city” cannot be validated.

In the area of ontology-based information extraction the situation is similar. Ontologies are used, but not in their full extent. Most of the time, ontologies (or taxonomies) are used for simple keyword matching - RDF would be sufficient here. There is no need for inference. Furthermore, there are no evaluations to support the hypothesis that ontology-based information extraction is better than rule-based or machine-learned information extraction without ontologies. RDF is more than sufficient for the majority of uses [25] and this is very disappointing in many ways - ontologies can give us so much more but at a much higher production cost.

8.1 Bridging the gap between model and application

The conceptual model, by itself, cannot be implemented. This is often overlooked in the development process: the model is defined and populated using the appropriate tools; this model must then be built into a system that will employ it. The choice of tools for model creation and model implementation will have a significant impact on the success, or otherwise, of the project. Ideally, ontological engineering should have an associated set of tools, or guidelines for tool production that lead to the semi-automated translation of models into applications.

Almost every ontology engineering method or methodology comes with a specification of its proposed conceptual modeling step. However, in comparison to conceptual modeling phase in relational database engineering (using the ER model), state-of-the-art ontology engineering tools do not support this conceptual phase or provide some quasi-standard notation for visualization as ER modeling does.

The machine-interpretable nature of a number of modeling solutions naturally leads to automated application-development tools, or integrated development environments, that can greatly assist the developer.

8.2 Reuse in ontological engineering

Reuse is crucial when dealing with ontologies, which are complex and expensive to construct correctly. Maximizing reuse defrays the cost of any ontologies that are produced locally for two reasons. Firstly, if a local ontology is used again locally, the development time is divided by the number of projects it is used for. Secondly, an active reuse strategy makes use of externally-defined ontologies, much as APIs and libraries allow the reuse of external code blocks at no development cost.

Rather than seeking to mimic software engineering libraries, which are functional blocks once conceptual alignment has taken place, pre-defined conceptual blocks can be used that can be built upon by systems that use ontologies.

To address this shortfall, a set of API-equivalents in ontological engineering is proposed, referred to as Reference Ontologies. A number of well-known software companies provide financial management and administrative organization software to businesses across the globe by taking a very straightforward approach. The process is:

1. Assume that all businesses follow a similar model and build a standard software system that uses these assumptions to provide the tools that this standard business will need.
2. Prepare for the eventuality that a specific business will need some re-engineering of the pre-prepared modules to provide an exact match to their business processes.
3. Develop a programming staff that will perform the required customization on demand.
4. Customize the software to the business, reusing most of your existing code base.
5. Finish implementation, scan the new code for anything useful that can be re-used elsewhere and integrate this back into the core product.

This model is an excellent representation of profitable software engineering. Maximize reuse of the correct implementation, employing the minimum effort required to meet the new specification.

From this, an equivalent in the sphere of ontologies is proposed, where reference ontologies are established for a number of domains. From a top-down view, these ontologies can be regarded as a common agreement on sets of statements that can be asserted about the nature of the knowledge that they describe, similar to the software engineering concept of patterns. From a bottom-up view, small reference ontologies may be considered for accepted concepts, from which larger reference ontologies can be built. Multiple viewpoints may result in the production of several ontologies from the same knowledge source.

Thus, comparing the ontological approach to that listed for software engineering, the process is:

1. Assume that certain key areas can be represented in a similar way and build a set of reference ontologies that describe key concepts or families of concepts.
2. Prepare for the eventuality that some rearrangement of concepts may be required based on a perception of the concepts as they are used in business or an application area. Because of this, any high-level concepts must be able to be decomposed to smaller, and effectively, indisputable concepts.
3. Develop a process for customization that allows the rearrangement of reference ontologies to form a new composed ontology, as well as providing a mechanism to clearly advertise the existence of this new composition, which is now a new reference ontology. This, in effect, constructs a *view* (citation from DB) of the ontology, similar to the view that one can encode across a database, and for similar reasons.
4. Customize the reference ontologies in ways that encode process-specific information, by using extensibility and mutability support inside the existing ontology language. Note: the final result will probably not be a reference ontology as some of these additions will be too specific to re-integrate.
5. Finish the implementation and integrate any useful modifications back into the parent reference ontology.

This process depends upon sharing and, given those issues already raised on businesses and proprietary information, it must be accepted that while some businesses will share their ontologies and their modifications, step 5 may not be a universally accepted step

8.3 Where to from here?

In this paper, two approaches are discussed as comparisons for ontological engineering; ER modeling for conceptual database design and software engineering for software design. Both of these approaches are widely accepted by a large community and are recognized as valid and standard practice. The goal is to remove the identified obstacles that are currently preventing ontologies from being applied by a broad community, in the same way that the community applies ER modeling and software engineering. In summary, the major concerns are:

– Missing powerful application and ubiquity.
– Ontologies are seen more as research interest than commercially viable.
– No good case for return on investment (cost/benefit analysis).
– Quality and the measurement of quality are very hard.
– The ongoing issue of reuse and manual alignment.
– Where ontologies are used, they are often underused or misused.

But, fundamentally, there are issues that relate to users choosing an ontology, even when all of the above questions have been addressed. But case studies show that application of ontologies leads to following unapt design issues:

– Taxonomic confusion exists because users cannot correctly distinguish ontology use cases from taxonomy use cases.
– Loose definitions (incompleteness) exist where the representation chosen is not checked or is used without all statements correctly constructed.
– Some of the semantics of the system are not expressed in the representation, leading to ambiguity and loose semantics.
– The solution is too concrete and has a corresponding lack of abstraction, which requires an additional layer of implicit interpretation in the business logic available. In the worst case, the business logic may be in active conflict with the concrete underlying representation.
– Users have no real idea of how much time it takes to construct a correctly-formed ontology or taxonomy, leading to planning skew.

Some of the most useful techniques where the term ontology is actually used do not need ontologies. For instance, simple keyword matching can be done in RDF, there is no need for an ontology.

While the modularity and granularity of software engineering can be naturally limited by language, there are no clear limits in place for the concepts, structure or expressiveness in an ontology. Chasing expressiveness can prevent the use of an ontology for inference (decidability limitations) and, even where decidability is possible, pragmatic constraints may prevent from waiting for an answer.

Software engineering is a rich source of potential improvements in the development of ontological engineering, because it is a highly technical discipline approach that has worked and is widely used today. Mastering method and approach selection and presenting it in a way that is easy to understand and apply may achieve substantial improvement. More and better tutorials and use cases are required and the technology needs to be made ubiquitous through necessity powerful applications that must use ontologies and to their full extent are the greatest requirement. Business interests must be convinced that the investment is worth it.

9. Conclusion

This research paper focuses on a critical reflection on ontologies and their applications in business. Therefore specific characteristics of ontologies and their benefits by reflecting business demands are discussed. In the course of this research work different definitions of ontologies are revealed, different aspects why ontologies are mainly applied in research and research related businesses and rarely in businesses are discussed, and the general misuse of ontologies is illustrated. For the general misuse and non-use of ontologies causes are identified and especially for the application of ontologies and its modeling procedures suggestions of improvements are proposed. Hence, the main research question and main motivation for this work are to find an answer for the question: *Are ontologies a buzzword or something really useful for business applications?*

The critical reflection of ontologies reveals three different explanatory statements that finally result in misuse of ontologies or non-use in business applications. These statements indicate (i) that both misuse and rare use of ontologies is due to the many existing, sometimes conflicting, definitions; (ii) that misuse can be traced back to imprecise specification of technologies; and (iii) that rare use derives from too abstract or too complex modeling processes.

The fundamental source of misusing and non-using ontologies aside from research projects and/or research related businesses already lie in the definition of the term itself. The many scientific publications on the definition of the term ontology make it indisputable that there are also many different understandings of the term itself; and de facto there is a great number of definitions for ontologies, as they are used within computer science. A definition that changes over time may be indicative of a lack of understanding in the field, or an inability to effectively communicate and share understanding. Whatever the reason is, introducing numerous definitions for one concept, especially a complex concept, leads to confusion and, consequently, people from various research communities can (and do) use the term ontology with different, partly incompatible meanings in mind. On the basis of the discussed definitions and the incompatible interpretation of ontologies the question may arise why ontologies are that popular in information technology and its research fields. And consequently, someone can ask why ontologies are not yet widely spread in the business world. In general, ontologies have to fulfill a central function if they meet the expectation to be effective and successful. This core function is the facilitation of communication between human and machine, which enables (semi-) automatic communication. In many cases a hyped buzzword raises expectations that are never fulfilled. Hence, it got difficult to discover who was or was not truly using ontologies. For this purpose three different case studies were presented to clarify the reasons behind ontology use and the influence of business model and user perception on the resources devoted to these projects, and their ultimate success.

In order to integrate ontologies into business applications, and thus, to provide a benefit from their advantages, three fundamental requirements were identified: (i) it has to be clearly defined what an ontology is; (ii) guidelines are necessary to answer the question as *when* ontologies should be used, *how* they can be used and *when they should not be used*; (iii) and finally an adequate representation languages must be applied.

The use cases once again make clear that a truly common understanding of what an ontology is must be established and the most appropriate technology must suggested for data modeling based on the kind of application and complexity considering the requirement of sharing, the semantic expressiveness, the complexity of the universe of discourse, and the size of the sharing community. These recommendations should finally result in an efficiently use of ontologies in a (business) project. Generally, ontologies can be specified using only informal means, such as UML class diagrams, entity-relationship models, or semantic nets, whereas conceptual entities in ontologies can also be defined mainly by formal means, e.g., by using axioms to specify the intended meaning of domain elements. This uncertainty leads to a broad spectrum of models, or concepts, or specifications, which are interpreted and published as ontology. In order to overcome the multitude of misuse a model for discussing ontologies and taxonomies that shows the relationships between existing technologies is proposed. The proposed model opposes modeling languages to modeling concepts and enables to select an appropriate method due to the consideration of the general requirement of sharing and its target semantic expressivity. The higher the demands for sharing and for semantic expressivity are the more complex the technology gets. According to this model ontologies are clearly shown at the peak of diagram (see Figure 4), being the most explicit representation of the classification and structural relationships known. Consequently, this model should help to answer the question as *when* ontologies should be used, and *how* they can be used.

The reasons why ontologies are not yet ubiquitous, thus, not only arises from an unsuitable selection of modeling technologies, but rather is caused by (i) an unknown (or unavailable knowledge about handling of an) ontology modeling process, (ii) lacking support of the modeling process because of missing state-of-the-art modeling tools, and (iii) missing longtime experience. The question that arises here: is ontology engineering – the modeling process of an ontology – too complex? Data modeling and software engineering come with a specification and design methodology that is suited to provide solutions that are sufficiently intricate without being overly complex. Both, a clear specification of the engineering process and a design methodology address a possible solution and finally lead to better acceptance of ontologies. Application of an ontology engineering methodology is – as it is standard in data modeling and software engineering – essential. Moreover, without a suitable ontology engineering methodology a project will likely fail, if compared to one, which follows a specific methodology. The most decisive reason for non-using ontology engineering methodologies for ontology design is that there is no conceptual model similar to the entity-relationship model available. Because of the lack of understanding of the benefits that can be gained, ontologies risk being seen as too abstract to implement, and too far from useful modeling. However, ontology design must be easy to understand and couched in terms that technical and non-technical users alike can understand. Reviewing the important facts regarding ontologies reveals that many other modeling approaches, like ER models or UML, enable the analysis, structuring and organization of a domain of interest. ER modeling is more or less state-of-the-art since Chen proposes this popular kind of conceptual data modeling. The best solution for making ontology engineering more amenable to businesses is to provide a simple notation such as the Chen notation for ER modeling. Furthermore, modeling an ontology would gain in popularity if its process model bases on well-established procedures, like data modeling or software engineering ones. The demands of ontology design are nearly equivalent to those of conventional data models. Such demands on the conceptual modeling of ontologies are (i) time and space for detailed domain analysis; (ii) more time, especially for considering reasoning and prediction aspects in design; (iii) greater demand of personal resources (domain experts, knowledge engineers, ontology experts); and (iv) support (tool or experts) to guide the design process of knowledge framework.

Finally, choosing the correct model whether ERM, UML or ontology, is a key decision that has a significant impact on the resources that are expended on a project and the success of the final outcome. Furthermore, ontology engineering has to provide a clear decision process to select the correct model, methodology and toolset to meet user requirements with the most efficient use of resources.

References

[1] Accomodation Ontology Language Reference and Accomodation Ontology Meta data, http://ontologies.sti-innsbruck.at/acco/ns.html, last visited: July 01, 2014.

[2] F. Bauer, M. Kaltenböck, Linked Open Data: The Essentials – A Quick Start Guide for Decision Makers, public available under: http://www.semantic-web.at/LOD-TheEssentials.pdf, last visited: July 01, 2014.

[3] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web. Scientific American Magazine. Retrieved March 26, 2008.

[4] W. Borst, Construction of Engineering Ontologies, PhD thesis, Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands, 1997.

[5] T. Bürger, E. Simperl, Measuring the Benefits of Ontologies, in Proceeding of OTM '08 Proceedings of the OTM Confederated International Workshops and Posters on the Move to Meaningful Internet Systems, pp.584 – 594, 2008.

[6] J. Cardoso, Developing an OWL Ontology for E-Tourism. In: Cardoso, J., Sheth, A. (Eds.), Semantic Web Services, Processes and Application, Springer, New York, 2006.

[7] J. Cardoso, The Semantic Web Vision: Where are We?, IEEE Intelligent Systems, pp.22-26, 2007.

[8] CEN Workshop Agreement, CEN Workshop on Harmonization of Data Interchange in Tourism - WS/eTOUR, Draft for Public Review, 2009.

[9] Deutsche Hotelklassifizierung, http://www.hotelsterne.de/userfiles/Files/downloads/de/Deutsche_Hotelklassifizierung_2010-2014.pdf, last visited: July 01, 2014.

[10] DERI, OnTour Ontology, http://e-tourism.deri.at/ont/index.html, last visited: July 01, 2014.
[11] Deutscher Tourismusverband: Classification of Tourism Offers, http://www.deutschertourismusverband.de/klassifizierung.html, last visited: July 01, 2014.
[12] ebSemantics Ontology, http://www.ebsemantics.net, last visited: July 01, 2014.
[13] EON Traveling Ontology, http://homepages.cwi.nl/~troncy/DOE/ontologies/EON- TravellingOntology-v0.2.daml, last visited: July 01, 2014.
[14] C. Feilmayr, C. Grün, R. Barta, B. Pröll, H. Werthner, Covering the Semantic Space of Tourism: An Approach based on Modularized Ontologies, in Proceeding of CIAO '09 Proceedings of the 1st Workshop on Context, Information and Ontologies, Article No. 1, ACM New York, 2009.
[15] M. Fernández-López, A. Gómez-Pérez, N. Juristo, Methontology: from Ontological Art Towards Ontological Engineering, in Procedings of Symposium on Ontological Engineering of AAAI, 1997.
[16] O. Fodor, H. Werthner, H., Harmonise — a Step towards an Interoperable e-Tourism Marketplace. International Journal of Electronic Commerce, 9/2, 2005.
[17] J. Fong, Mapping extended entity relationship model to object modeling techniques, ACM Sigmod Record, 24, 3, Sep, ACM, NY, 1995.
[18] A. Gangemi, V. Presutti, Ontology Design Patterns, in Handbook on Ontologies, International Handbooks on Information Systems 2009, pp.221-243, 2009.
[19] A. Gómez-Pérez, M. Fernandez-Lopez, O. Corcho, Ontological Engineering, with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web, First Edition Series: Advanced Information and Knowledge Processing, 2004.
[20] T. Gruber, A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), pp. 199-220, 1993.
[21] M. Gruninger, J. Lee, Introduction: Ontology Applications and Design, Communications of the ACM, Vol. 45, No. 2., 2002
[22] N. Guarino, P. Giaretta, Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars, editor, Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, IOS Press, Amsterdam, pp.25–32, 1995.
[23] N. Guarino, D. Oberle, S. Staab, What Is an Ontology?, Handbook on Ontologies, Steffen Staab, Rudi Studer (eds.), Springer, 2009.
[24] M. Hepp, Ontologies: State of the Art, Business Potential, and Grand Challenges, in: Martin Hepp, Pieter De Leenheer, Aldo de Moor, York Sure (eds.): Ontology Management: Semantic Web, Semantic Web Services, and Business Applications, ISBN 978-0-387-69899-1, Springer, pp. 3-22, 2007.
[25] M. Hepp, Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies, in Internet Computing, IEEE, vol. 11(1), pp.90-96, 2007.
[26] 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 (EKAW2008), Springer LNCS, vol 5268, pp. 332-347, 2008.
[27] 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, last visited: July 01, 2014.
[28] C.J. Janse, H. Kroeze, A. van Wigcheren et al., A genotype and phenotype database of genetically modified malaria-parasites. Trends in parasitology, 2010.
[29] M. Jarrar, J. Demey, R. Meersman, On Using Conceptual Data Modeling for Ontology Engineering, Journal on Data Semantics I, Lecture Notes in Computer Science Volume 2800, pp.185-207, 2003.
[30] R. Juric, The UML Rules, SIGSOFT software engineering Notes, 23, 1, ACM, NY, 1998.
[31] Mondeca, Semantic Web Methodologies and Tools for Intra-European Sustainable Tourism, White Paper, September 2004.
[32] Open Travel, http://www.opentravel.org/Specifications/OnlineXmlSchema.aspx, last visited: July 01, 2014.
[33] 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), Marrakech, Morocco, May 2008.
[34] T. Pellegrini, Semantic Web Awareness 2009 - A Comparative Study on Approaches to Social Software and the Semantic Web, Research Report, http://www.semantic-web.at/file_upload/1_tmpphpvuVU1T.pdf, last visited: July 01, 2014.
[35] N. Sharda, R. Jakkilinki, M. Georgievski, M. Ponnadal, Intelligent Visual Travel Recommender Systems Model for E-Tourism Websites, technical report, publisher: Gold Coast, Qld.: Sustainable Tourism CRC, 2008.
[36] E. Simperl, C. Tempich, Ontology Engineering: A Reality Check, in Proceeding of OTM Conferences 1, Vol. 4275, Springer, p. 836-854, 2006.
[37] E. Simperl, Reusing ontologies on the Semantic Web: A feasibility study, Data & Knowledge Engineering 68, pp.905–925, 2009.
[38] 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, In CIA '99 - Proceedings of the 3rd international Workshop on Cooperating Information Agents. Springer, Berlin, 1999.
[39] H. Stuckenschmidt, F. van Harmelen, Information Sharing on the Semantic Web, Springer: Berlin 2005.
[40] R. Studer, R. Benjamins, D. Fensel, Knowledge engineering: Principles and methods, Data & Knowledge Engineering, 25(1–2), pp.161–198, 1998.
[41] TAGA Ontology, http://taga.sourceforge.net/owl/index.html, last visited: July 01, 2014.
[42] M. Uschold, M. Gruninger, Ontologies: Principles, Methods and Applications. Cambridge University Press, 1996.
[43] M. Uschold, M. King, Towards a Methodology for Building Ontologies, in Skuce D (eds) IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montral, Canada, pp.6.1-6.10, 1995.
[44] M. Vujasinovic, N. Ivezic, E. Barkmeyer, Z. Marjanovic, Semantic B2B-integration Using an Ontological Message Metamodel. Concurrent Engineering, 2010.
[45] H. Werthner, F. Ricci, E-Commerce and Tourism, in Magazine Communications of the ACM - The Blogosphere CACM, vol.47(12), ACM New York, NY, USA, pp.101-105, 2004.
[46] World Tourism Organization: Thesaurus on Tourism & Leisure Activities. Madrid, Spain: World Tourism Organization, 2001.