Ontology design parameters for aligning agri-informatics with the Semantic Web

C. Maria Keet

Faculty of Computer Science, Free University of Bozen-Bolzano, Italy
keet@inf.unibz.it

Abstract. In recent years there have been many efforts in the development of bio-ontologies, where the applied life sciences can see the benefits reaped from, and hurdles observed with, such early-adopter efforts. With the plethora of resources, where should one start developing one's own domain ontology, what resources are available for reuse to speed up its development, for which purposes can the ontology be developed? We group inputs that determine effectiveness of ontology development and use into four types of parameters: purpose, ontology reuse, ways of ontology learning, and the language and reasoning services. We illustrate this for the agriculture domain by building upon experiences gained in previous and current projects.

1 Introduction

Only six years ago, multiple modelling issues for the applied life sciences were documented [1], which are currently being addressed, such as with the W3C's incubator group on modelling uncertainty in the Semantic Web (SW), or even surpass the required solution up to a point that is has generated new ones. The most notable advances are the mushrooming of freely available bio-ontologies, the notion of ontology design patterns [2] to save oneself of re-inventing the wheel, and the W3C standard for OWL as common ontology language in the SW. However, solving one problem moves the goal-posts. For instance, which ontologies are reusable for one's own ontology, what are the consequences choosing one over the other? The successor of OWL, draft OWL 2 [3], actually has 4 languages tailored for different purposes: which one should be used for what and when? We structure the main ontology design parameters to provide a brief and clear overview of the principal development options. Ontology development, in particular for highly specialised subject domains in the applied biosciences, is a challenging task and any reuse of information in some way can alleviate this bottleneck. One can both reuse ontologies and ontology-like artifacts, and carry out bottom-up development of ontologies through ontology learning. There are, however, interfering design choices due to the purposes of the ontology and the representation language and reasoning services. We illustrate these parameters with examples taken from the agriculture domain, based on prior and current experimentation with bacteriocins for food processing, semi-automated ontology development in ecology, and ontology-based data access in molecular ecology with horizontal gene transfer (e.g., [1, 4, 5]), and related literature.

2 Design parameters

2.1 Purposes of the ontologies

Arguably, one could take into account the possible aims for which the ontology will be developed. For the ontology purist, however, this is anathema, because an ontology is supposed to be implementation independent—even irrespective if an application will be linked to it or have any computational use at all—and as such has the sole purpose of representing reality. In the practice of ontology engineering, it does have an impact and, based on a literature review and survey [5], the different types of purposes can be summarised as follows:

A. Ontology-based data access through linking data to ontologies [6, 5];
B. Data(base) integration, most notably the strand of applications initiated by the Gene Ontology Consortium and a successor, the OBO Foundry [7, 8];
C. Structured controlled vocabulary to link database records and navigate across databases on the Internet, also known as 'linked data';
D. Using it as part of scientific discourse and advancing research at a faster pace [4, 9], including experimental ontologies in a scientific discipline and usage in computing and engineering to build prototype software;
E. As full-fledged discipline "Ontology (Science)" [10], where an ontology is a formal, logic-based, representation of a scientific theory;
F. Coordination and integration of Web Services;
G. Tutorial ontologies to learn modelling in the ontology development environment (e.g., the wine and pizza ontologies).

A real caveat with choosing explicitly for a specific goal is that a few years after initial development of the ontology, it may get its own life and be used for other purposes than the original scope. This, then, can require a re-engineering of the ontology, as is currently being done with the GO and FMA.

2.2 Reusing ontologies and ontology-like artefacts

With the mushrooming of ontology development, ontology repositories and semantic search systems, such as Swoogle [http://swoogle.umbc.edu/] and the TONES Ontology Repository [http://owl.cs.manchester.ac.uk/repository/] can be helpful. However, not all ontologies are just more of the same. The principal types of ontologies and ontology-like artifacts that can have a good potential for reuse in part or whole are:

1. Foundational ontologies that provide generic top-level categorisations;
2. 'Reference ontologies' that contain the main concepts of a subject domain;
3. Domain ontologies that have a (partial) overlap with the new ontology;
4. Legacy representations of information systems: conceptual data models of database and application software (sometimes called 'application ontologies'), terminologies, and thesauri;
5. For each of items 1-4, resource usage considerations, such as
   (a) The availability of the resource, such as openly available, copyright, and usage restrictions;

(b) If the source is being maintained or an abandoned one-off effort;

(c) The ontology is a result of a community effort, research group, or if it has already some adoption or usage;

(d) If it is subject to standardization policies or has stable releases.

The foundational ontologies can give a head-start by providing a basic structure, such as endurants being disjoint from perdurants, types of processes, attributes (qualities), and a set of basic relations; e.g., GFO, DOLCE, BFO, RO [11, 12]. Reference ontologies, on the other hand, are more restricted in scope of the content, but also intended for reuse, such as an ontology of measurements, of time units and 'top-level' ontologies for a domain, such as BioTop [http://www.imbi.uni-freiburg.de/biotop/] and an ontology of biological investigations (OBI, under development). Domain ontologies, in turn, can build upon such foundations and expand on it for the particular subject domain at hand, such as for traits of rice in Gramene extending GO and marine microbial loops reusing DOLCE [13, 4]. The applied life sciences domains have many terminologies and thesauri legacy material, of which a few are being adapted for the SW, such as the reengineering of AGROVOC [14] and reconfiguring and linking for the fisheries domain by using OneFish, AGROVOC, ASFA, and FIGIS with a DOLCE foundation [2]. Other candidates are AOS, and thesauri such as CAB International and CAT. In addition, one can 'ontologise' a conceptual data model and extend the contents. An example for bacteriocins, which are non-therapeutical antibiotics used for food preservation and food safety, is shown in Fig.1. The icons hide the OWL 2 DL in the Protégé ontology development tool, such as `Bacteriocin` $\sqsubseteq \exists$ `inhibits.MicroOrganism`, whereas the grey arrows denote a few of the myriad of possible extensions.

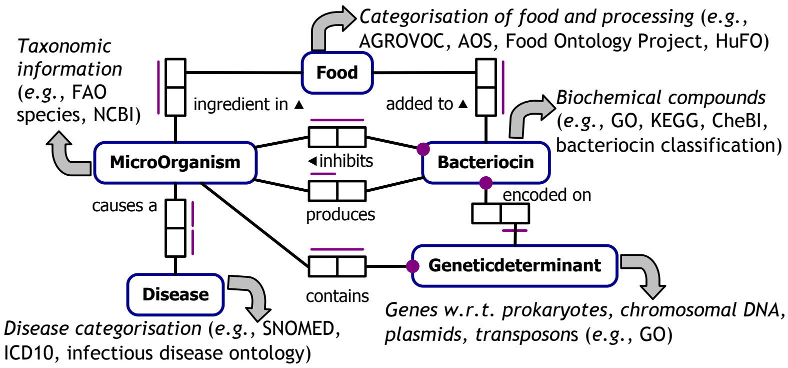
Categorisation of food and processing (e.g., AGROVOC, AOS, Food Ontology Project, HuFO)
Taxonomic information (e.g., FAO species, NCBI)
Food
ingredient in ▲
added to ▲
Biochemical compounds (e.g., GO, KEGG, CheBI, bacteriocin classification)
MicroOrganism
◄inhibits
Bacteriocin
causes a
produces
encoded on
Geneticdeterminant
Disease
contains
Genes w.r.t. prokaryotes, chromosomal DNA, plasmids, transposons (e.g., GO)
Disease categorisation (e.g., SNOMED, ICD10, infectious disease ontology)

Fig. 1. Section of the conceptual model of the bacteriocins database [1], with reuse of names for relations (e.g., *contains*) and where ontologies, terminologies, and thesauri can be added. This central part about bacteriocins is a candidate for an ontology content design pattern to structure and simplify adding new contents to the ontology. AOS: Agricultural Ontology Service; ChEBI: Chemical Entities of Biological Interest; GO: Gene Ontology; HuFO: Human Food and Nutrition ontology; KEGG: Kyoto Encyclopedia of Genes and Genomes; NCBI: National Center of Biotechnology Information.

2.3 Bottom-up development of ontologies through ontology learning

Although one will find something of use in the currently available ontologies, people often will have to develop at least part of the ontology themselves. There are several strategies to speed up this labour-intensive task, which focus on extracting in a semi-automatic way the subject domain semantics present in other legacy sources. The principal techniques are:

I. Extraction of types from data in database and object-oriented software applications, including database reverse engineering, least common subsumer, and clustering;
II. Abstractions from models in textbooks and diagram-based software;
III. Text mining of documents, including scientific articles and other Digital Libraries, to find candidate terms for concepts and relations;
IV. Wisdom of the crowds and usage of those tagging techniques;
V. Other (semi-)structured data, such as excel sheets and company product catalogs.

Reverse engineering is well-known in software development, which is being augmented with a logic-based approach to facilitate the step toward domain and application ontologies [15]. A similar approach in spirit is text mining that seeks to learn the candidate concepts and relations from documents [16]. This is, however, a highly iteratively process [17] that still requires considerable domain expert input (see [16] for a discussion). A different option is to extract knowledge from biological models, such as STELLA models for ecology and environmental sciences made with ISEE software, where, e.g., a STELLA "flow" is a perdurant (the *grazing* process by mesozooplankton) and "stock" corresponds to endurant (e.g., *Plankton*) [4]. One also can try to squeeze out the little semantics available in, say, excel sheets (but see also [9]). If also this fails to extract useful terms and relations, one could resort to the 'wisdom of the crowds'; however agriculture is highly specialised and perhaps not close to the hearts of many online users so that a controlled tagging game with agronomy students may yield better results.

2.4 Representation languages and reasoning services

Depending on the purpose(s)—and, in practice, available resources, such as time, money, domain experts, and available baseline material—one tends to end up with either (a) a large but simple ontology, i.e., mostly just a taxonomy without, or very few, properties (relations) linked to the concepts, where 'large' is, roughly, > 10000 concepts, so that a simple representation language suffices; (b) a large and elaborate ontology, which includes rich usage of properties, defined concepts, and, roughly, requiring OWL-DL; or (c) a small and very complex ontology, where 'small' is, roughly, < 250 concepts, and requiring at least OWL 2 DL. That is, a separate dimension that interferes with the previous parameters, is the choice for a representation language. Moreover, certain choices for reusing ontologies or legacy material, or goal, may lock one into the language that will be used to represent the ontology.

Different from OWL that divided itself between two Description Logics-based versions, OWL-DL and OWL-Lite, and an more liberal RDFS version, the final W3C draft of its successor, OWL 2, has one 'DL' version and three 'lighter-DL' versions [3]. The main motivation for including four DL languages in the standard is to allow tailoring the choice of ontology language to fit best with the usage scope in the context of a *scalable* and *multi-purpose* SW. At the time of writing, no applications exist yet that lets one seamlessly and transparently change one ontology language for another for a given OWL 2-formalised ontology. OWL 2 DL is most expressive and based on the DL language $\mathcal{SROIQ}$ [18], whereas OWL 2 EL and OWL 2 QL are smaller, 'computationally well-behaved', fragments to achieve better performance with larger ontologies and ontologies linked to large amounts of data in secondary storage (databases), respectively; OWL 2 RL has special features to handle rules. Differences between expressiveness of the ontology languages and their trade-offs are discussed in [19]. For instance, OWL 2 DL has the following features that OWL 2 QL does not have: role concatenation, qualified number restrictions, enumerated classes, covering constraint over concepts, and reflexivity, irreflexivity, and transitivity on simple roles. On the other hand, with the leaner OWL 2 QL one can obtain similar performance as with relational databases, whereas for OWL 2 DL one never can achieve that. In addition, not all reasoning services are possible with all languages, either due to theoretical or practical limitations. The current main reasoning services fall into three categories:

i. The 'standard' reasoning services for ontology usage: satisfiability and consistency checking, taxonomic classification, instance classification, and querying functionalities including epistemic and (unions of) conjunctive queries;
ii. Additional 'non-standard' reasoning services to facilitate ontology development: explanation/justification, glass-box reasoning, pin-pointing errors;
iii. Further requirements for reasoning services identified by users (e.g. [20]), such as hypothesis testing, reasoning over role hierarchies, and discovering type-level relations from ABox instance data.

Then, in a software-supported selection procedure, one should be able to select the desired purpose and reasoning services to find the appropriate language, or decide on purpose of usage of the ontology and one's language, and obtain which reasoning services are available. For instance, purpose A or B goes well together with OWL 2 QL and query functionalities, whereas for purposes D and E, OWL 2 DL and the non-standard reasoning services will be more useful.

3 Conclusions

To enhance the efficiency and effectiveness of the recent commencement of developing agri-ontologies, we described the four influential factors. These are (i) seven types of purpose(s) of the ontology, (ii) what and how to reuse existing ontologies and ontology-like artefacts, (iii) five different types of approaches for bottom-up ontology development from other legacy sources, and (iv) the interaction with choice of representation language and reasoning services. Future works pertain to setting up a software-mediated guidance system that can make suggestions how to proceed with ontology development given particular requirements; hence, to structure and make accessible more easily the 'soft' knowledge about ontology development, which then could feed into design methodologies such as methontology.

References

1. Keet, C.M.: Biological data and conceptual modelling methods. J. of Conceptual Modeling **29** (October 2003) http://www.inconcept.com/jcm.
2. Gangemi, A.: Applying ontology design patterns to practical expertise: roles, tasks and techniques in the agricultural domain. In: CSBio Reader. Volume 1., Free University of Bozen-Bolzano (Dec. 2005) 47–57
3. : OWL 2. Working draft, W3C (Dec. '08) http://www.w3.org/TR/owl2-syntax/.
4. Keet, C.M.: Factors affecting ontology development in ecology. In: Data Integration in the Life Sciences 2005 (DILS'05). Volume 3615 of LNBI., Springer (2005) 46–62
5. Alberts, R., Calvanese, D., DeGiacomo, G., et al.: Analysis of test results on usage scenarios. Deliverable TONES-D27 v1.0, TONES Project (Oct. 10 2008)
6. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. on Data Semantics **X** (2008) 133–173
7. Gene Ontology Consortium, .: The Gene Ontology GO database and informatics resource. Nucleic Acids Research **32**(1) (2004) D258–D261
8. Smith, B., et al.: The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. Nature Biotech. **25**(11) (2007) 1251–1255
9. Madin, J.S., Bowers, S., Schildhauer, M.P., Jones, M.B.: Advancing ecological research with ontologies. Trends in Ecology & Evolution **23**(3) (2008) 159–168
10. Smith, B.: Ontology (science). In: Proc. of FOIS'08, IOS Press (2008)
11. Masolo, C., Borgo, S., Gangemi, A., et al.: Ontology library. WonderWeb Deliverable D18 (v1.0, 31-12-2003). (2003) http://wonderweb.semanticweb.org.
12. Smith, B., et al.: Relations in biomedical ontologies. Genome Biol. **6** (2005) R46
13. Jaiswal, P., et al.: Gramene: development and integration of trait and gene ontologies for rice. Comparative and Functional Genomics **3** (2002) 132–136
14. Soergel, D., Lauser, B., Liang, A., et al.: Reengineering thesauri for new applications: the AGROVOC example. J. of Digital Information **4**(4) (2004)
15. Lubyte, L., Tessaris, S.: Extracting ontologies from relational databases. In: Proc. of the 20th Int'l Workshop on Description Logics (DL 2007). (2007) 387–395
16. Gliozzo, A.M., et al.: Results from experiments in ontology learning including evaluation and recommendation. Deliverable 7.3.1, NeoN Project (Dec. 15 2007)
17. Alexopoulou, D., et al.: Terminologies for text-mining; an experiment in the lipoprotein metabolism domain. BMC Bioinformatics **9**(Suppl 4) (2008) S2
18. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. Proceedings of KR-2006 (2006) 452–457
19. Keet, C.M., Rodríguez, M.: Toward using biomedical ontologies: trade-offs between ontology languages. In: Proc. of Semantic eScience. Volume WS-07-11 of AAAI. (2007) 65–68
20. Keet, C.M., Roos, M., Marshall, M.S.: A survey of requirements for automated reasoning services for bio-ontologies in OWL. In: Proc. of OWLED'07. Volume 258 of CEUR-WS. (2007)