The use of foundational ontologies in ontology development: an empirical assessment

C. Maria Keet

KRDB Research Centre, Free University of Bozen-Bolzano, Italy
School of Computer Science, University of KwaZulu-Natal, South Africa
keet@ukzn.ac.za

Abstract. There is an assumption that ontology developers will use a top-down approach by using a foundational ontology, because it purportedly speeds up ontology development and improves quality and interoperability of the domain ontology. Informal assessment of these assumptions reveals ambiguous results that are not only open to different interpretations but also such that foundational ontology usage is not foreseen in most methodologies. Therefore, we investigated these assumptions in a controlled experiment. After a lecture about DOLCE, BFO, and part-whole relations, one-third chose to start domain ontology development with an OWLized foundational ontology. On average, those who commenced with a foundational ontology added more new classes and class axioms, and significantly less object properties than those who started from scratch. No ontology contained errors regarding part-of vs. is-a. The comprehensive results show that the 'cost' incurred spending time getting acquainted with a foundational ontology compared to starting from scratch was more than made up for in size, understandability, and interoperability already within the limited time frame of the experiment.

1 Introduction

Ontologists tend to be outspoken about the usefulness of foundational (top-level) ontologies, such as BFO, DOLCE [1], GFO [2], and SUMO: either they are perceived to be essential or an impractical burden. Older ontology development methodologies that are still in use, such as METHONTOLOGY [3] and On-To-Knowledge [4], do not mention the use of a foundational ontology, but at the time of their development there were hardly any available, and the larger projects, such as GALEN and Cyc, developed their own. More recent methodologies, such as the NeON Methodology [5], mention it either in passing as part of ontology reuse in general, or explicitly, as in OntoSpec [6], and it is considered as an essential component in the OBO Foundry project [7, 8]. A foundational ontology is also used with, among others, top-level domain ontologies, such as BioTop [9], in the coordination of lightweight ontologies and thesauri in KOKO [10], and with domain ontologies, e.g., [11, 12]. The underlying assumptions of the proponents of the use of a foundational ontology are that ontology developers, once introduced to foundational ontologies, naturally will use it, be it right from the start or to align their ontology with one of them, because

i. it facilitates ontology development because one does not have to reinvent the wheel concerning basic categories and relations, and
ii. using a foundational ontology improves overall quality and interoperability.

On the other hand, such foundational ontologies are criticised as being too abstract, too expressive, too comprehensive for 'simple' or domain ontologies, and it takes too much time to understand them in sufficient detail. In addition, expressivity issues and the difference between a foundational ontology's take on how to represent attributes (e.g.,with qualities and qualia), and OWL's data properties with 'application ontologies' (*de facto*, OWLized formal conceptual data models) for ontology-driven information systems increases the perceived gap further [13]. Trying to answer whether the former or the latter stance holds cannot be carried out by simply collecting all ontologies on the web through, e.g., Swoogle or the TONES repository and counting the inclusion of a foundational ontology, because a substantial amount of them are experimental or tutorial ontologies or OWLized non-ontological resources (e.g., thesauri), one cannot always assess their quality regarding correct representation of the subject domain, the choice why a foundational ontology was used or not is unknown, and it is unknown how much resources went into developing the ontologies; hence, with this much uncertain parameters, one cannot draw any conclusions.

What is needed to commence ascertaining the validity of one or the other stance, are controlled experiments. This paper reports on one such experiment, which, to the best of our knowledge, is the first of its kind. After basic training in OWL and foundational ontologies (BFO, DOLCE, and part-whole relations), 52 participants developed 18 domain ontologies—all of them a "computer ontology"—in the timeframe of 24 hours. One third actually used a foundational ontology, and its developers had added, on average, more classes and class axioms than those who developed the ontology from scratch (albeit not statistically significant), and had added significantly less new object properties thanks to reusing those provided by the foundational ontology. Hence, the foundational ontology facilitated domain ontology development at least to the point that even with a very short timeframe, the investment required for using a foundational ontology was already more than evened out. In addition, they had slightly less errors and were more interoperable regarding the usage of part-whole relations in particular, thereby improving overall quality and interoperability. These results justify extending ontology development methodologies, or developing a new one, with a foundational ontology, both regarding when to choose one and how and where it aids the actual modelling.

The remainder of this paper is structured as follows. After describing the materials and methods of the experiment in Section 2, we present the results in Section 3 and discuss them in Section 4. We consider different directions of extensions of methodologies in Section 5, and conclude in Section 6.

2 Materials and methods

The methodology for the experiment was as follows.

1. Lecture on purpose and usefulness of using a foundational ontology and overview of its contents (3-4 hours);
2. Divide course participants into smaller groups of 1-4 participants;
3. Provide the participants with instructions, being:
   (a) Develop a domain ontology about computers (i.e., it should contain that what you expect to find in a 'computer ontology' if you would search for it online);
   (b) You have the following input options:
      i. *tabula rasa*, i.e., start from scratch with an empty OWL ontology and do not import anything;
      ii. Use an OWLized foundational ontology (options provided: DOLCE, BFO, GFO);
      iii. And/or use the OWLized taxonomy of part-whole relations;
   (c) Name your ontology with the names of the group participants;
   (d) Time to develop the computer ontology: 24h from start to handing it in;
   (e) The ontology will not be graded, but is part of an experiment that will be discussed after having handed in the ontology;
4. Evaluation:
   (a) Assessment of the OWL files on usage of foundational ontologies, ontology metrics (language used, classes and object properties added etc.), and errors made;
   (b) Open questions with the participants regarding motivations of (non-)usage and modelling issues.

The materials used for the experiment were OWLized foundational ontologies provided to the course participants, including the respective URIs, being `DLP3971.zip` (a set of DOLCE ontologies), `bfo-1.1.owl` (BFO), `gfo.owl` (GFO), and two versions of the taxonomy of part-whole relations with a simplified DOLCE taxonomy, which is based on [14] (`pwrelations.owl` and an extended version `mereotopoDOLCE.owl`). A highly simplified version of DOLCE (^{my}DOLCE-lite$_{mini}$, for short) is depicted in Fig. 1 and a summary of the OWLized taxonomy of part-whole relations is included in Fig. 2. The ontology development environment was Protégé 4.1beta with the integrated Hermit v1.2.4 automated reasoner. No restrictions were put on the participants to use, or not, non-ontological resources, such as textbooks, Wikipedia, product catalogs etc.

3 Results

In this section we describe the setting of the experiment, provide a characterisation of the participants, and describe quantitative and qualitative results of the ontologies that were developed. This will be discussed afterward in Section 4.

3.1 Setting

The experiment was carried out in three sessions during a course on a comprehensive introduction to ontology engineering at Universidad de la Habana

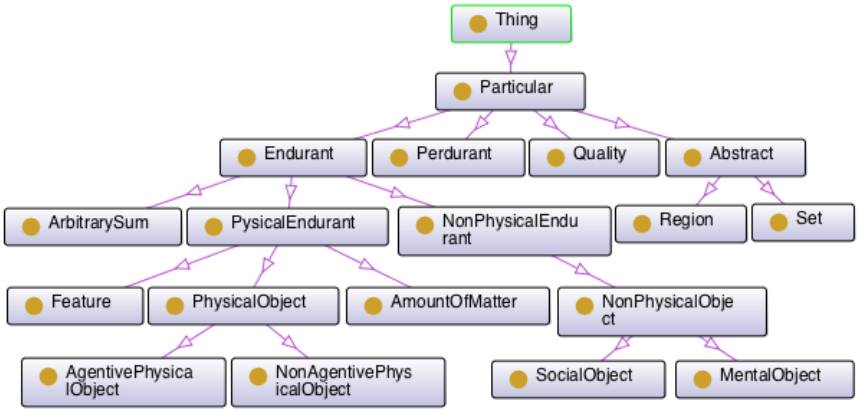

Fig. 1. Graphical rendering of ^{my}DOLCE-lite$_{mini}$ in the Protégé 4.1 OntoGraf tab.

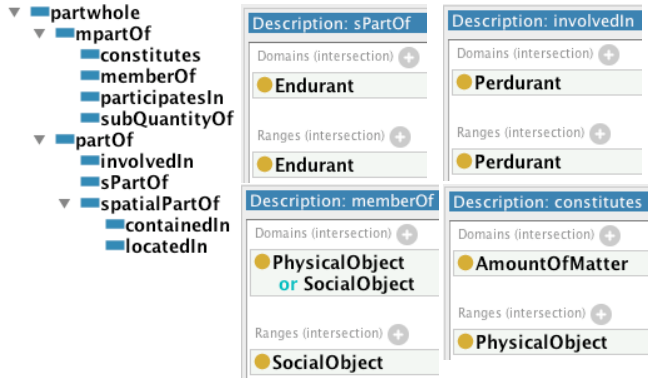

Fig. 2. The OWLized version of the taxonomy of part-whole relations [14] (left) with screenshots of domain and range restrictions of four object properties (right).

(UH) and Universidad de Ciencias Informáticas (UCI) in Cuba, and CSIR Meraka in South Africa in 2010. Regarding the course outline and content, for indicative purpose, the syllabus and slides of the latest installment are available at `http://www.meteck.org/teaching/SA/MOWS10OntoEngCouse.html`. After a session on OWL and OWL 2, they contained about a 1.5 hour introduction in foundational ontologies in general and DOLCE and BFO in particular, which was followed by 1-1.5 hours on part-whole relations. The experiment commenced afterward in the labs.

The reason for choosing the subject domain of computers for the experiment is that most participants can be considered domain experts in that area (see Section 3.2) and it lends itself well not only for a wide range of different entity types but also the need to use part-whole relations in one way or another. In addition, at the time of the experiment, it was asserted that there was no computer

ontology or similar available online and therefore no bother to search for it. (In fact, there is a serious ontology about software and programming languages [12], but it was inaccessible to the participants at the time of the experiment.)

3.2 Characterisation of the participants

The amount of participants in the courses who handed in ontologies is 16 (UH) + 27 (UCI) + 8 (Meraka) = 52. One of the participants was a biologist with an interest in ontologies, three were in interdisciplinary areas (juridical AI, IT & education, and computational linguistics) and the remaining 48 participants were computer scientist; hence, 51 participants did have at least some modelling experience, have had at least one logic course, and can be considered also domain experts. No participant did have any formal training on OWL before the course and only two participants in the last session had had a course on Description Logics. The participants of the last session did receive some training on Protégé prior to the ontology engineering course, whereas for the other two installments, the Pizza Ontology tutorial was advised as self-study and several exercises were carried out in the preceding days. Thus, all participants were relatively novice ontology developers.

Most of the participants were studying either for a MSc or PhD (n=48), or a researcher, lecturer or professor (n=4); lecturers who are also MSc or PhD student—a considerable amount—are counted in the former group. The participants' age was predominantly between 23-33 years, with five in the 45-65 year age bracket. 19 participants were female and 33 male. The subgroups the participants formed themselves were mixed.

The participants were principally interested in ontology engineering in that they needed to develop a domain ontology for their research projects, i.e., as a component of an ontology-driven information system, whereas a small amount (about 5) were carrying out research for ontologies, such as debugging ontologies and user interfaces.

3.3 Assessment of the ontologies

Ontology data and statistics. The 52 participants developed 6 (UH) + 8 (UCI) + 4 (Meraka) = 18 ontologies in groups of 1-5 participants. Six groups, or 1/3, used a foundational ontology, of which one imported the full DOLCE (all ontologies in `DLP3971.zip`), one DOLCE-Lite, two used `pwrelations.owl` that has both a taxonomy of part-whole relations and the DOLCE taxonomy of categories, and two groups used its extended version `mereotopoDOLCE.owl`; hence, no group used BFO or GFO.

Table 1 contains basic data of the slightly anonymized ontologies. In the remainder of the analysis, we exclude the biologist outlier (`52`) to ensure homogeneity in the notion of type of participant (this participant did not know what to add and was unfamiliar with logic [pers. comm.]).

In addition to the strict division between starting from scratch and using a foundational ontology, and for the purpose of analysis, we also consider a group

Parameter ⇒		Found. onto.		Language	New entities				new
	From scratch	DOL-CE	pwrel/ mereo-topo	(DL fragment)	class	obj prop	data prop	indi-vi-duals	class axioms
Ontology ⇓									
010203.owl	+			ALCIQ	16	4	0	9	34
0405.owl	+			SROQ	12	3	0	26	22
06070809.owl	+			ALCIQ	17	3	0	28	30
10.owl	+			ALCHI	17	6	0	1	18
111213.owl			+	SRIQ(D)	16	5	2	9	18
141516.owl			+	SRIQ	39	3	0	6	50
171819.owl	+			ALCHIF	20	6	0	2	20
202122.owl		+		SHIQ	44	2	0	0	52
232425.owl	+			ALCHIQ(D)	24	5	0	0	54
26272829.owl	+			ALCHIQ(D)	25	6	4	1	40
30.owl	+			ALCQ	36	2	0	0	44
3132333435.owl	+			ALCHF	36	4	0	0	55
36373839.owl			+	ALCROIQ(D)	24	0	7	3	34
40414243.owl			+	ALCROIQ(D)	22	0	4	3	32
444546.owl	+			AL	27	1	0	0	24
4748.owl	+			SHI	13	4	0	0	16
495051.owl	+			ALCI	10	5	1	11	13
52.owl		+		SHOIN(D)	1	0	0	0	1

Table 1. Basic characteristics of the computer ontologies developed by the participants; for each participant in a subgroup, a two-digit number was used in order of handing in the ontology. The language (DL) fragment was obtained from the Protégé 4.1beta 'active ontology' tab and is for indicative purpose only. 'Found. onto.' = usage of a foundational ontology.

of ontologies where its developers did inspect a foundational ontology, but did not use one in the submitted OWL file, being, at least, `0405`, `10`, `30`, `3132333435`, and `4748`. Upon inquiry, the main reason for not using one after all was time constraints and `4748` used the so-called Componency Ontology Design Pattern because either one of the foundational ontologies was "too much to handle" yet there was still the desire to reuse some existing material. Assessing averages, median, and standard deviation (Table 2), they are very similar to those who started from scratch, in particular compared to the substantial differences with those who started with a foundational ontology, and therefore this 'would have but did not do'-subgroup is not considered further as a specific subgroup.

While the differences in average and median are particularly favourable for those ontologies who started with a foundational ontology—i.e., having more classes and class axioms despite losing time in editing due to getting acquainted with the foundational ontologies—one also can observe quite some variation among the individual ontologies in Table 1. To this end, a Student t-test was performed on the two groups regarding new classes, new class axioms, and new object properties. p=0.145 for new classes, hence, barely not significant to claim

Parameter ⇒		New entities				New
		class	obj. prop.	data prop.	individuals	class axioms
Group ⇓						
All	Average	23.4	3.5	1.1	5.8	32.7
	Median	22	4	0	2	32
	StDev	10.1	2.0	2.0	8.8	14.3
Found. onto. reuse	Average	29	2	2,6	4.2	37.2
	Median	24	2	2	3	34
	StDev	11.9	2.1	3.0	3.4	14.0
From scratch	Average	21.1	4.1	0.4	6.5	30.8
	Median	18.5	4	0	1	27
	StDev	8.7	1.6	1.2	10.3	14.6
Inspect found. onto.	Average	22.8	3.8	0	5.4	31
	Median	17	4	0	0	22
	StDev	12.2	1.5	0	11.5	17.5

Table 2. Basic analysis of the new additions to the submitted ontologies; numbers are rounded off.

starting with a foundational ontology significantly speeds up ontology development. For new class axioms, p=0.420, hence, one cannot conclude anything either way. For new object properties, however, p=0.043, or: those who started with a foundational ontology added significantly less properties than those who started from scratch. Of the groups who started from scratch, 10 out of 12 invented a part-whole object property of their own, having names such as `hasPart`, `esParteDe` ('is part of' in Spanish), `compuestaPor` ('composed of' in Spanish), `hasComponent`, and so forth. Conversely, the groups who reused a foundational ontology availed of those part-whole object properties already present in the imported ontology—which, consequently, have a clear meaning compared to those in the other 10 ontologies.

A noteworthy observation is that, in analogy with software development, ontology development is not a factory line—more people in a group did not result in larger ontologies in the same amount of time. Other observations are that about 2/3 of the groups used qualified number restrictions and therewith uses OWL 2 DL compared to one of its profiles, and the ontologies in the most expressive OWL 2 DL fragment are typically those that reuse a foundational ontology. The data also shows that more emphasis had been put on adding a class hierarchy than class axioms involving object properties.

Qualitative aspects. In addition to the basic characteristics, one has to consider the quality of the ontologies concerning both the contents and the modelling errors. Considering the errors, the following can be observed. Unlike the well-known common error of confusing part-of with is-a among novice modellers, *none* of the 18 ontologies had this error. There were multiple cases of is-a vs instance-of confusion where types of processors and motherboards were modelled

as instances, and a few 'unexpected' results were encountered with the reasoner during development due to domain and range restrictions that were too restrictive in hindsight or had an axiom instead of the intended atomic class. There were several ontologies where the "NonSimpleRoleInNumberRestriction" was encountered either already during the development or after handing in the ontology, which was due to the use of `Min-`, `Max-`, or `ExactCardinality` with a non-simple object property. This was due to the interaction with the characteristics of the part-whole property that was not a simple object property anymore (see [15], of which the technical details about constraints on roles are described in [16]). A relatively minor issue concerns the difference between the naming of the ontology, i.e., changing the URI from a default like `.../Ontology123456789.owl` into a meaningful name, versus naming the OWL file: the participants did the latter, but not the former. Thus, while the sensitization of part-whole relations prevented one type of common errors, these errors observed are general mistakes that are not attributable to not using a foundational ontology.

The six ontologies that used a foundational ontology were analysed further. `141516` has `PC` as a subclass of DOLCE's `AgentivePhysicalObject` (APO), `40414243` has `Ordenador` (computer) as a subclass of `ArbitrarySum` (AS), and the other four have it as a subclass of `NonAgentivePhysicalObject` (NAPO). `40414243` motivated that each computer is a "varying collection of things", and "therefore" an arbitrary sum, whereas `141516` deemed APO appropriate because a computer "is a physical object that does things". DOLCE's motivation to distinguish between APO and NAPO, however, is that the former is assumed to have beliefs, desires, and intentions and are typically attributed only to persons, and the 'arbitrary' in AS is to be taken more arbitrarily than those collections that make up a computer (for which one can identify constraints) [1]; hence, computer as an (in-)direct subclass of NAPO is the appropriate category. A straightforward explanation of the details of these DOLCE categories immediately resolved the difference, converging to NAPO. Eleven of the 12 other ontologies had added `Computer` (or similar) to the ontology, but none was the same or even alike and a resolution was not attempted due to time constraints.

4 Discussion

We discuss the test results and limitations of the set up in order with the claims regarding the benefits and problems of using a foundational ontology, and subsequently consider other factors that did or might affect the results.

4.1 Reuse of entities vs. too comprehensive and too complicated

The main explanation for why the groups who used a foundational ontology did not lag behind those who started from scratch—in fact, quite to the contrary—is that they availed of the imported classes and object properties, so there was no time lost with, among others, discussing how the part-whole relation should be named, thereby avoiding having to reinvent the wheel concerning basic classes

and object properties. From the other perspective, one might assume that choosing the right category from a foundational ontology is time-consuming. If the latter were the case, and provided one wants to import a foundational ontology, then it certainly would have been easier for the former group to import BFO instead of DOLCE or the taxonomy of part-whole relations, because BFO is a bare taxonomy of 39 classes, 0 object properties, and 107 class axioms (in the DL language $\mathcal{ALC}$)[1] compared to 37 classes, 70 object properties, and 94 class axioms in DOLCE-Lite (in $\mathcal{SHI}$; the much larger OLWized DOLCE is in $\mathcal{SHOIN}(D)$) and the part-whole relations with the ^{my}DOLCE-lite$_{mini}$ has 18 classes, 13 object properties, and 31 class axioms (in $\mathcal{SRI}$). Yet, this did not occur and therefore this informal criticism cannot be substantiated with the data obtained in the experiment, whereas the former claim of speeding up ontology development by using a foundational ontology, can. Nevertheless, in this context it is worthwhile to observe that the current OWLized DOLCE versions are too expressive for the OWL 2 EL and OWL 2 QL profiles [17], which may affect its use and reuse. In addition, the developers of `36373839` and `40414243` *manually deleted* classes and object properties because, according to its developers, they were perceived to be unnecessary and cluttering the ontology. This indicates a possible use for partial imports that is currently still not implemented (works in ontology modules is in progress [18]), or a 'hide' feature either in the graphical interface or also at the logical level, as implemented in, e.g., CASL [19]. A follow-up experiment may want to include a scenario with more different—expressive and slimmed—versions of DOLCE to figure out what is, or are, the 'optimal' DOLCE version(s) for practical ontology engineering.

It is unclear why the participants chose DOLCE over BFO; that is, despite asking for it, the answers were not of sufficient detail to warrant drawing any conclusions. If we assume some foundational ontology FO_A is more suitable for tasks T_A or subject domain D_A and FO_B for T_B or D_B, then this should be known and have objective arguments why this is the case so that it can be taken into account in ontology development. We are not aware of the existence of such an assessment, so that it is more likely that developers—be it in this experiment or for real ontology development—choose a particular foundational ontology for compatibility with other existing or envisioned ontologies and/or infrastructure, or philosophical motivations, or subjective preferences.

4.2 Quality and interoperability

A general, and well-known, problem in assessing the second claim mentioned in the introduction—a better quality ontology—is to determine what are the unambiguous objective parameters by which one can say which ontology is really a *better* ontology. We took only a minimal approach to it in the assessment, such as the actual errors made (is-a vs instance-of) and avoided (is-a vs part-of),

[1] It is claimed recently [8] that "BFO" now has to be understood as BFO + the Relation Ontology; the RO passed the revue in the lectures, but because of the ambiguous status of the combination at the time, it was not included in the experiment.

the ambiguity of names/labels that especially in the 'from scratch' ontologies bore little semantics, if any (compuestaPor), logically correct but unintended mistakes with respect to the subject domain, and language errors (non-simple role in number restriction). Based on such basic metrics, the subgroups who used a foundational ontology fared a little better, but it is not entirely free of debate.

In addition, it may neither prevent nor fully solve certain differences in representation of knowledge. For instance, the ontologies that reused a foundational ontology had `Computer` at three different places in the DOLCE taxonomy, and, overall, 17 ontologies had `Computer` (or similar or a synonym) added as a class to the ontology, with 5 of them as a defined concept, and none was the same. There were three different principle directions taken by the subgroups: either computer was some collection of macro-components, such as tower, keyboard, and monitor, or it was some collection of its parts, such as RAM, CPU, motherboard etc., or something that has both hardware and software, where two ontologies had two of the three perspectives combined (`232425` and `26272829`). This surprised the participants, especially because intuitively it is obvious what a computer is and they were of the opinion that they were describing the characteristics of the same physical objects. How this has to be resolved in the realist-BFO way [8] or some other, perhaps more practical, way [20], is a different topic.

Compared to `Computer`, this was easier for `Software`, partially thanks to its underspecification in the ontologies and partially thanks to the fact that there is a domain ontology about software and programs that extends DOLCE [12]. As it turns out, it is easy to align at least the participants' DOLCE-based computer ontologies with this one. More precisely, four of the six ontologies had `Software` in their ontology, of which one as a subclass of AgentivePhysicalObject (like it did with `Computer` and which can simply be resolved in the same manner), and three had it as a subclass of NonPhysicalObject (`202122`, `36373839`, and `40414243`); that is, in the same branch as the more refined ontology by Lando and co-authors [12] and therewith relatively easy to merge.

4.3 Other factors

A complicating factor in ontology development in the first two sessions of the experiment was the natural language barrier in conjunction with the new design of Protégé 4. Whereas Protégé 3 uses icons for the familiar Description Logic symbols ($\forall$, $\sqcap$ etc), they have been changed into keywords in Protégé 4; more precisely, *English* keywords. Such an 'anglification' is not helpful in an international setting, which is a setting that ought to have been assumed for tools for the Semantic Web as part of its internationalization objective [21]. It was not intuitive to figure out what the keywords were (compared to immediate understanding of the symbols) so that modellers lost time finding the appropriate ones and it resulted in ugly spanglish that hampered understandability, such as `Ordenador subClassOf utiliza exactly 2 Perifericos_Principales` (vs. the more intelligible `Ordenador` $\sqsubseteq = 2$ `utiliza.Perifericos_Principales` that would have been obtained with Protégé 3). For proper internationalization and supporting ontology development environments tailored to subject domain experts,

the keywords should be provided in various languages (or to provide the option to add them for one's preferred language), and in order to cater for different types of modellers, an option to switch from keywords back to the natural language-independent symbols will be welcome.

Concerning the preparatory training of the participants, it may seem possible to argue in three directions regarding foundational ontology reuse: either 1/3 is a low reuse percentage because the lecturer had not taught the matter sufficiently well, or that even with good teaching there is just 1/3 of the groups who reused a foundational ontology voluntarily, or that thanks to good teaching it is an impressive 1/3 of the ontologies where people voluntarily reused a foundational ontology. Aside from the formal and informal student evaluations (unanimously positive for the first and third installment), this can be settled partially by carrying out the experiment with other lecturers, but this is beyond the scope of the current experiment. In addition, the purpose of the experiment was not communicated to the participants, because one of the parameters was to examine how many groups would *voluntarily* choose to use a foundational ontology without suggestive interference. The downside of this was that groups followed three distinct strategies in ontology development: either they were focussed on adding as much as possible ('adding more entities is better') or they were more concerned with discussions how to model the various entities as good as possible (e.g., "what constitutes a computer?", "is software is a physical object?"), or experimenting with the reasoner (e.g., "will `wrong_computer3` make the ontology inconsistent?"). Follow-up experiments may want to focus solely on the ontology quality dimension, informing the participants about this beforehand, and, by dividing the subgroups into two: one where the people are forced to use a foundational ontology, one where they should not.

Last, although the time allotted to domain ontology development may seem short, one has to bear in mind that the developers were also wearing their hat as domain experts and did receive logic training beforehand, thereby mitigating the short timeframe. While the participants in each installment of the course were highly motivated, this might be different for other experiments so that it may be beneficial to build in more precise timing with compulsory lab sessions.

5 On enriching methodologies

Given the cautiously positive outcome in favour of reuse of a foundational ontology, one has to look ahead at where, how, and in which methodology this can be incorporated, which requires inclusion of at least two main components: choosing which foundational ontology to reuse and how to use it in the modelling process.

5.1 Extending high-level methodologies to include a foundational ontology usage step

Including a decision point to choose a foundational ontology somewhere in the procedure is fairly straightforward, be it by extending existing methodologies

that do not address foundational ontologies explicitly yet or a new methodology based on a set of criteria the methodology has to meet (such as outlined in [22, 5]). For instance, for the relatively well-known METHONTOLOGY, the addition would be both in the "conceptualization" stage that has intermediate representations made by the domain experts and in the "formalization" stage where the domain-expert understandable model is transformed into a formal or semi-computable model. For the former case, the ontology can be offered in any format because it is used for modelling guidance only; for the latter case, and assuming a Semantic Web setting, then the ontology should be available in one of the OWL species. For the NeON methodology [5], this means extending its "Scenario 3" with an explicit section on 'foundational ontology' (in addition to the current "general or common ontology") and creating a new so-called "Filling card" for foundational ontologies, which can have the following contents according to the standard filling card headings:

- *Definition*: Foundational Ontology Reuse refers to the process of using a foundational ontology to solve different problems, such as non-interoperability of ontologies and losing time reinventing known modelling solutions.
- *Goal*: The goal of this process is to find and select a foundational ontology that is either to be integrated in the stand-alone ontology or ontology network being developed or to be imported at the start of ontology development.
- *Input*: Competency questions included in the ontology requirements specification document of the ontology to be developed (see [5]), at least one file in an implementation language for each such ontology, and, when available, a (set of) table(s) comparing the candidate foundational ontologies to be reused across the same criteria.
- *Output*: A foundational ontology integrated in the ontology being developed.
- *Who*: Ontology developers involved in the ontology development, such as domain experts, knowledge engineers, and practice-oriented philosophers.
- *When*: The foundational ontology reuse process should be carried out after the "ontology specification activity" and before other "ontological resource reuse".

Such a high-level filling card, however, does not yet aid the modeller in updating the contents of the ontology, for which we have to look at the second type of methodologies in the next section.

5.2 Augmenting the modelling exercise

Neither one of the high-level extensions says anything about how to choose between one or the other foundational ontology or how the chosen ontology is to be integrated in the modelling exercise. Concerning the former, the lofty goal of an "ontology library" of interchangeable foundational ontologies that was envisioned in [1]—thereby avoiding the need to choose between one or the other foundational ontology—is yet to be realised. Concerning the latter, we have to

look at another 'type' of methodology[2]. Noy and McGuinness' Ontology Development 101 (OD101) [23] and Kassel's OntoSpec [6] focus specifically on how to represent entities in an ontology, such as conducting a property analysis, assessing cardinalities, distinguishing between is-a or instance-of and so forth, whereby OntoSpec relies on the OntoClean and DOLCE foundations to stimulate good modelling practices and OD101 is based on the authors' own experiences in ontology development. To the best of our knowledge, there are no tutorial ontologies and exercises in conjunction with DOLCE, BFO or GFO[3] and practical examples may need to be added at least to the lectures but even more so in a structured fashion for all leaf categories of the foundational ontologies in each methodology. For OD101, this means rewriting steps 4-7 and sections 4-6 to reflect the guidance from foundational ontologies. For OntoSpec, it means extending the few elaborate examples to cover all DOLCE leaf types and, assuming it is to be used within a Semantic Web setting, converting its current representation from DOLCE-OS (a "semi-informal OntoSpec language" [6]) into a suitable OWL species. For both cases, however, a new solution to the issue of choosing the appropriate part-whole relation is required. This may be achieved by automating the decision diagram in [24] extended with a 'cheat sheet' with examples and informal definitions for each of the foundational ontology's main categories (alike Table 1 in [1]).

A brief illustration is described in the following example.

Example. Let us take the African Wildlife tutorial Ontology, and the wish to represent that the elephant's tusks (ivory) are made of apatite. There are three classes: `Elephant`, `Tusk`, and `Apatite`. Although reading the text in [1] would be better, in this case its examples in Table 1 are already helpful: examples for Non-Agentive Physical Object (NAPO) are "a hammer, a house, a computer, a human body" and for Amount of Matter (M) they are "some air, some gold, some cement"; hence, `Elephant` and `Tusk` are subclasses of NAPO, and `Apatite` is a subclass of M. Then, considering the relations between them, we can avail of the OWLized part-whole relations property hierarchy, which has its properties typed with the DOLCE categories (recollect Fig. 2). Thanks to knowing the DOLCE category of each of the three classes, it straightforwardly follows that each tusk is a structural part of, `sPartOf`, elephant in our wildlife ontology and that tusk is `constitutedOf` apatite. ♢

These suggestions for enhancing extant ontology development methodologies are incomplete. However, note that the aim of this work was first to examine whether it makes sense practically to use foundational ontologies, if they are used voluntarily, and how, in a mode that is based not only on theoretical motivations,

[2] That is, at present there are two strands of methodologies, but they may well become integrated into one larger methodology at a later stage.

[3] except for informal notes with very few examples at `http://keet.wordpress.com/2010/08/20/african-wildlife-ontology-tutorial-ontologies/` by this author and at `http://www.itee.uq.edu.au/~infs3101/` by Robert M. Colomb.

but, moreover, whether this can be motivated from the perspective of hard data in ontology development. As was demonstrated with the results obtained in the experiment, a comprehensive extension of the methodologies indeed will be of practical use.

6 Conclusions

We have investigated assumptions surrounding foundational ontology reuse in a controlled experiment with 52 developers who designed 18 domain ontologies. One-third of the ontologies were developed using a foundational ontology. Concerning the contents, on average, those who commenced with a foundational ontology added more classes, more class axioms, and significantly less object properties. The comprehensive results showed that the 'cost' incurred in spending time getting acquainted with a foundational ontology compared to starting from scratch was more than made up for in better quality and interoperability already with the limited duration of the experiment. Because of the positive results, we considered possible extensions to extant methodologies, which have to be at two levels: choosing which foundational ontology to reuse and how to use it in the modelling.

We are working on a tool to help choosing the appropriate part-whole relations and consider future extensions to help choosing when one foundational ontology would be better to reuse than another. It requires further investigation why the participants preferred DOLCE over BFO, and what the outcome will be if also much larger ontologies such as Cyc or SUMO were to be added to the options in a controlled experiment. It may be interesting to see similar experiments with other types of participants, such as with non-computing domain experts with experience in modelling.

Acknowledgements The visit to CSIR Meraka, South Africa, was partially funded by the EU FP7 Net2 project "Enabling a network of Networked Knowledge" and partially by "Technologies for Conceptual Modelling and Intelligent Query Formulation" within the Executive Programme of Scientific and Technological Co-Operation between the Italian Republic and Republic of South Africa.

References

1. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Ontology library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003). (2003) http://wonderweb.semanticweb.org.
2. Herre, H., Heller, B.: Semantic foundations of medical information systems based on top-level ontologies. Knowledge-Based Systems **19** (2006) 107–115
3. Fernandez, M., Gomez-Perez, A., Pazos, A., Pazos, J.: Building a chemical ontology using METHONTOLOGY and the ontology design environment. IEEE Expert: Special Issue on Uses of Ontologies **January/February** (1999) 37–46

4. Staab, S., Schnurr, H., Studer, R., Sure, Y.: Knowledge processes and ontologies. IEEE Intelligent Systems **16**(1) (2001) 26–34
5. Suarez-Figueroa, M.C., et al.: NeOn methodology for building contextualized ontology networks. NeOn Deliverable D5.4.1, NeOn Project (2008)
6. Kassel, G.: Integration of the DOLCE top-level ontology into the OntoSpec methodology. Technical Report hal-00012203, Laboratoire de Recherche en Informatique d'Amiens (Oct 2005) http://hal.archives-ouvertes.fr/ccsd-00012203.
7. Smith, B., et al.: The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology **25**(11) (2007) 1251–1255
8. Smith, B., Ceusters, W.: Ontological realism: A methodology for coordinated evolution of scientific ontologies. Applied Ontology **5** (2010) 79–108
9. Beisswanger, E., Schulz, S., Stenzhorn, H., Hahn, U.: BioTop: An upper domain ontology for the life sciences. Applied Ontology **3**(4) (2008) 205–212
10. Hyvönen, E.: Preventing ontology interoperability problems instead of solving them. Semantic Web **1** (2010) 33–37
11. Keet, C.M.: Factors affecting ontology development in ecology. In Ludäscher, B., Raschid, L., eds.: Data Integration in the Life Sciences 2005 (DILS2005). Volume 3615 of LNBI., Springer Verlag (2005) 46–62 San Diego, USA, 20-22 July 2005.
12. Lando, P., Lapujade, A., Kassel, G., Fürst, F.: Towards a general ontology of computer programs. In: Proceedings of the 2nd International Conference on Software and data Technologies: ICSOFT 2007. (2007) 25-27 July 2007, Barcelona, Spain.
13. Keet, C.M.: Dependencies between ontology design parameters. International Journal of Metadata, Semantics and Ontologies **5**(4) (2010) 265–284
14. Keet, C.M., Artale, A.: Representing and reasoning over a taxonomy of part-whole relations. Applied Ontology **3**(1-2) (2008) 91–110
15. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 web ontology language structural specification and functional-style syntax. W3c recommendation, W3C (27 Oct. 2009) http://www.w3.org/TR/owl2-syntax/.
16. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. Proceedings of KR-2006 (2006) 452–457
17. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles. W3c recommendation, W3C (27 Oct. 2009) http://www.w3.org/TR/owl2-profiles/.
18. Stuckenschmidt, H., Parent, C., Spaccapietra, S., eds.: Modular Ontologies—Concepts, Theories and Techniques for Knowledge Modularization. Springer (2009)
19. Kutz, O., Lücke, D., Mossakowski, T., Normann, I.: The OWL in the CASL – designing ontologies across logics. In: Proc. of OWLED'08. (2008) Karlsruhe, Germany, October 26-27, 2008.
20. Lord, P., Stevens, R.: Adding a little reality to building ontologies for biology. PLoS ONE **5**(9) (2010) e12258
21. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. J. Web Semantics **1**(1) (2003) 7
22. Dahlem, N., Guo, J., Hahn, A., Reinel, M.: Towards an user-friendly ontology design methodology. In: International Conference on Interoperability for Enterprise Software and Applications (I-ESA'09). (2009) 180–186 April 21-22, Beijing, China.
23. Noy, N., McGuinness, D.: Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory (March 2001)
24. Keet, C.M.: Part-whole relations in object-role models. In: 2nd International Workshop on Object-Role Modelling (ORM'06). Volume 4278 of LNCS., Berlin: Springer-Verlag (2006) 1116–1127 Montpellier, France, Nov 2-3, 2006.