Towards ontology composition from cognitive libraries

Stefano Borgo
Laboratory for Applied Ontology
ISTC CNR, Trento, Italy
stefano.borgo@cnr.it

Abstract

Sometimes knowledge engineers have to come up with new ontologies for some specific domain or application even though there are already ontological works in that area. Although there are techniques for ontology de/composition, a common problem is that the existing systems often characterise some notion in a way that is not quite what the knowledge engineer needs. The change of a few notions in a given ontology can be challenging: it is not easy to understand the impact of these changes.

In this paper we investigate another route. Assuming that the knowledge engineer has to deal with notions at the mesoscopic level that are cognitively clear, we propose to independently characterised them (in a sense to be discussed), and look at how they can be used to build an ontology such that it comprises only the needed notions and with the right meaning. Here we just explore some important steps of this approach, and list other problems that need to be faced.

Introduction

Ontology artefacts (Guarino 1998; Guarino, Oberle, and Staab 2009) are complex formal systems used to ensure readability and transparency of information models and to enhance interoperability across information systems. However, the development and tuning of ontologies, no matter whether focussed on foundational, reference or domain concepts, is still largely left to the personal skills and preferences of the ontologist (developing an ontology is comparable to a craft) and is one of the major bottlenecks towards their wide application.

Among the problems reported by ontology users three are particularly interesting for this paper:

- the construction of ontology artefacts is time-consuming even when trying to reuse existing systems (Fernández-López, Gómez-Pérez, and Suárez-Figueroa 2013; Simperl and Tempich 2006);
- the proper characterisation of the notions to include is complex (Guarino 1999) and
- any change in the formalisation of a notion can lead to logical problems (Ghilardi, Lutz, and Wolter 2006) or can force unexpected changes in other notions.

These problems arise because ontology artefacts tend to be large logical theories about general as well as specific notions[1] and this makes it impossible to modify the system relying on just intuition. Logical tools help to verify the consistency of a modified system and to build specific models, especially when we limit ourselves to decidable languages, but do not help to implement changes in a system that are aimed to preserve the modeller's intuitions, nor to model intertwined notions like those of action and agent. Furthermore, to be fully exploited ontology artefacts must cover (most of) the domain of application leading to build large systems that have to do justice of possible competing perspectives and ad hoc distinctions.

This paper investigates the construction of ontologies from cognitively motivated modules each focussing on a single notion. In particular, we rely on intuition, understood as a mix of cognition, domain expertise and common sense, to identify the needed notions and to select their characterisation. Assuming a library of such modules is available, we show how a Knowledge Engineer (KEer) can build her/his own ontology to meet intuition and needs. The paper is exploratory in nature and aims to only sketch the approach. This is done by looking at how the three notions of agent, action and artefact are understood in literature; how these should be formalised to form a library of modules; and how these modules can be merged into a single ontology. It also discusses several problematic issues that need to be resolved to turn this approach into a methodology for ontology construction.

Structure of the paper. After a section with preliminary observations, we propose to divide notions in two (contextualised) types which form the basis of our approach. Section "Cognitive categories" exemplifies key steps of the approach by discussing the notions of agent, action and artefact. The next section explains how to understand the definitions just introduced and the section after it, "The role of underspecified definitions", clarifies how they are used and why that

---

[1] In this paper we use the generic term 'notion' as a rough synonym of concept or category. A notion in this sense is formally modeled as a class.

form of underspecification is important. Next we explain how to deal with the other class of categories (called "structure categories") by relaying on a top-level framework. The final section highlights the remaining issues and some critical aspects.

Preliminary Observations

There can be a variety of motivations to build an ontology and these determine the key categories one needs to include in the system. When one has practical reasons like a specific application, as opposed to a research plan for systematising a topic or for a foundational ontology inspired by a particular worldview, he/she usually starts by listing a set of key terms, their intended meaning and related requirements. An ontology construction methodology aims to ensure, among other things, that the categories for these entities are correctly modelled and find a natural position in the ontology framework. Further categories are introduced for different reasons, e.g., (a) some categories are added for completeness (in an ontology for transportation one could include unicycles although they are not generally considered transportation means) and (b) some categories are added to structure the system's taxonomy (in the previous example, the category of physical objects and the subcategory of physical devices with the latter including that of transportation devices).

In practice, in many cases the categories one starts with are comparable in the sense that they carve up reality at some homogeneous level within the so called mesoscopic view, i.e., they characterise entities human beings typically perceive, manipulate and find relevant in their everyday life. Topics related to everyday life (services to the person, production of mechanical and mechatronic devices, organisational and institutional structure, logistics, transportation etc.) are almost exclusively at these perceptual/cognitive levels: people, animals, buildings, public places, appliances, social roles etc.

Since most of the ontologies that are built today naturally deal with categories at these perceptual/cognitive levels, it makes sense to leverage on this regularity to improve existing techniques for ontology construction. This paper makes the first steps to turn this observation into a suitable (and possibly promising) methodology.

Approach Sketched

Let us assume a KEer wants to construct an ontology for an application or for a specific company. She starts by embracing that particular viewpoint of a fragment of reality and then focuses on the key notions as understood in that application. For the sake of the presentation, let us assume that all available ontologies she could reuse misrepresent one or more of these key notions or make unsuitable choices relatively to some constraints. Since to build a robust and reliable ontology is complicated and time consuming, the KEer may try to modify one of these existing ontologies (Simperl 2009) by adding/deleting properties and rewriting relations; by tangling (or untangling) notions, e.g., remodelling an object category into a role category or vice versa (like distinguishing product and device, and moving the latter category into a role taxonomy); or by adding/removing single categories or articulated branches. In alternative, the KEer can use ontology modularity and matching approaches roughly consisting in isolating relevant parts of possibly different ontologies and merging them into a new system, see (Euzenat and Shvaiko 2007). For whatever reason one may use modularity (Borgo 2011), the reuse of existing ontological work is non trivial and, when specific constraints must be satisfied, it can be demanding and error-prone.

Another possibility is presented in the following sections where we concentrate on the modelling of predicates that identify classes (see our use of the term 'notion' in footnote 1) but the method, mutatis mutandis, should be applicable to properties and relations as well.

Starting from the application/domain-driven mandatory notions, at first the KEer defines (both in natural language and formally) each notion she needs to cover. This is done assuming that the other notions used in any of these definitions, are already correctly modelled (see the next Section for examples). This initial work leads to isolate a list of formally characterised categories and a list of uncharacterised categories. The latter set collects the notions that were used in some formal definition but were not themselves defined. In other words, the KEer associates a term in the formal language to each notion she considers but formally axiomatises in the language only some of these terms. There are different ways to arrive at these formal definitions but typically one follows the understanding of the notion in the given domain (from which the request of a description in natural language) and, with the help of competency questions (Grüninger and Fox 1995), fixes some constraints to capture that informal meaning. Some accompanying axioms may be added especially to bind the arguments of the occurring relations. When this step is completed, the KEer has produced a fairly complete list of categories deemed relevant to occur in the ontology at stake. Among these categories, some are associated with a formal (and typically partial) axiomatisation, these categories are called *domain categories*, the other are so far uncharacterised categories.

Recall that we target an ontology focusing on notions at some homogeneous mesoscopic/cognitive level. General categories like entity, object, happening and role, should not occur among the domain categories. It is however very well possible that they occur in the list of uncharacterised categories. The basic idea is that since these more general notions are hard to characterise correctly, it is better to borrow them from some well studied and characterised top-level ontology. In the language just obtained we divided the categories in two groups: domain and uncharacterised. We split the latter group in two classes: the *structural categories* and the *subsidiary categories*. By construction, the domain categories correspond to the key notions in which the KEer is interested. The structural categories (typically categories like entity, object, event, space, time etc.) are categories that organise the high-level structure of the ontology. They are not the focus of the ontology to be built but are necessary to obtain an ontological system. The subsidiary categories are the remaining categories: the categories that are neither at the centre of the KEer's concerns, nor serve to structure the ontology and still come up naturally when describing the domain categories. We will see some example in the next section.

The distinction between structural, subsidiary and domain categories is clearly fuzzy and contextual. Nonetheless, when an application scenario is fixed, the distinction becomes intuitively clearer and even inspiring to the point that we decided to leverage on it.

Assume now that a suitable general ontology comprising the needed high-level (structural) categories and their ontological relationships (parthood, participation, etc.) has been identified. The KEer isolates a coherent fragment of the top-level that includes all the structural categories (renaming them if needed), their formalisation, their characterising relationships, and that covers all the domain categories. In short, the KEer delegates the formalisation of the (typically complex) high-level notions to the top-level ontology or a suitable fragment of it. Note that, by construction, structural categories already come as a single theory, i.e., as a single module.

As we see, the distinction between domain and structural categories brings some intrinsic advantage since the KEer limits her work to the formalisation of those notions that are more accessible to common sense and well understood in practical use, while she harvests existing ontological work to organise and formalise the high-level categories which structure the overall system.

One crucial problem in any ontology construction approach via modularisation is the merging of the different modules. We anticipate that this remains an open problem even in the approach here sketched. Yet, there is some advantage since we need to merge only two modules: the one of domain categories (introduce next) and the one just discussed, that of structural categories.

Cognitive categories

Here we collect a few ways to understand three common notions, namely, agent, action and artefact. These notions are quite general compared to the standard domain notions that are the target of our modelling approach. However, these have the advantage of being simple, quick to introduce, largely discussed in the literature and intuitively clear without discussing an application scenario. Their informal descriptions have been collected from the literature and are well-known and accepted within their communities. However, we do not argue for or against their value, i.e., we are neutral about these informal characterisations since these are used only to exemplify our approach.

Since our approach requires to translate the notions in a logical language, for the sake of the presentation let us assume we have chosen some first-order language $\mathcal{L}$ whose interpretation is based in the usual logical and semantic machinery. Below we will gradually add information on the non-logical vocabulary of $\mathcal{L}$.

How to be Agent

The following definitions of (intelligent) agent are fairly simple and accepted in the literature. We formalise them just minimally and without considering further specialisations (e.g., introducing on a pair physical agents, software agents and so on). Our level of characterisation agrees with what is often done in today's ontologies and has the advantage to be simple to understand. Simplicity is crucial since our main goal is to present an approach, not a specific ontology.

Definition 1 (Agent).
An animate entity that is capable of doing something on purpose.[2]

This notion is meant to be quite inclusive comprising human and animal agents as well as robots and softbots. A quick analysis of the description identifies four key elements that together characterise this perspective: animate entity, capable, purpose, doing. We take these four elements plus agent itself as primitives, i.e., add them to the non-logical vocabulary with the indicated arity:[3] Animate(x), DoFor(x, z, p) [read: x does y for p], Purpose(p), and HasCapability(x, z) [read: x has capability z]. We also add Action(x) as auxiliary predicate.

Let us formalise this notion in $\mathcal{L}$ as follows:

$$Agent(x) \overset{\text{def}}{\leftrightarrow} Animate(x) \land \exists z, p\ (Purpose(p) \land HasCapability(x, z) \land DoFor(x, z, p)) \quad (1)$$

The auxiliary predicate is needed to constrain all the arguments of the new relations:[4]

$$HasCapability(x, z) \to Agent(x) \land Action(z) \quad (2)$$

$$DoFor(x, z, p) \to Agent(x) \land Action(z) \land Purpose(p) \quad (3)$$

<u>Observations</u>: (a) one could also formalise the definition using the following formula $Agent(x) \overset{\text{def}}{\leftrightarrow} Animate(x) \land \exists z, p\ (Purpose(p) \land Action(z) \land DoFor(x, z, p))$, or something more complex, e.g., by introducing a separate $Capability$ predicate. Our discussion is independent of the choice of the formula and of the non-logical vocabulary provided the resulting formulas capture fairly well the KEer's understanding of the notion; (b) as said, in this paper we do not discuss specific characterisations of relations and limit our interest to their domain and range (mainly to give a basic understanding of what they are binding). □

Definition 2 (Agent).
Something that acts in an environment.[5]

---

[2] This is the informal definition adopted by John Sowa, see http://www.jfsowa.com/ontology/agents.htm

[3] These are fresh elements in $\mathcal{L}$. Renaming or indexes can be used where needed.

[4] Note that we use the same numbering for axioms and definitions since the latter are technically seen as "if and only if" axioms, from which our choice of the $\overset{\text{def}}{\leftrightarrow}$ symbol.

[5] Informal definition proposed in (Poole and Mackworth 2010), see http://artint.info/html/ArtInt_3.html

Here there are just two characterising terms: to act and environment. As before, we take these two as new non-logical terms in the language $\mathcal{L}$. Our formalisation of this definition in first-order logic uses: Agent(x), Environment(y), ActIn(x, y) [read: x acts in y] and the auxiliary predicate Object(x). This implies that in the reading we formalise, environment is seen as an object.

We formalise this notion in $\mathcal{L}$ with a definition and an axiom as follows:

$$Agent(x) \overset{\text{def}}{\leftrightarrow} \exists y\ (Environment(y) \wedge ActIn(x, y)) \quad (4)$$

$$ActIn(x, y) \rightarrow Agent(x) \wedge Object(y) \quad (5)$$

Definition 3 (Agent)**.**

Anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.[6]

One can characterise this perspective via the following elements: Agent(x), Environment(x), Sensor(x), Actuator(x), PerceiveWith(x, y, z) [read: x perceives y with z], ActOnWith(x, y, z) [read: x acts on y with z] and the auxiliary predicate Object(x).

We now formalise this notion in $\mathcal{L}$ via one definition and two axioms:

$$\begin{aligned} Agent(x) \overset{\text{def}}{\leftrightarrow} \exists y, s, z(&Environment(y) \wedge \\ &PerceiveWith(x, y, s) \wedge ActOnWith(x, y, z)) \quad (6) \end{aligned}$$

$$Sensor(x) \rightarrow Object(x) \quad (7)$$

$$Actuator(x) \rightarrow Object(x) \quad (8)$$

$$PerceiveWith(x, y, s) \rightarrow Agent(x) \wedge Object(y) \wedge Sensor(s) \quad (9)$$

$$ActOnWith(x, y, z) \rightarrow Agent(x) \wedge Object(y) \wedge Actuator(z) \quad (10)$$

How to be Action

In this part we report a few definitions of (intentional) action which have been collected in (Trypuz 2008). Again, we use simple and generally well received informal definitions and, again, one could formalise them more deeply or precisely, but recall that our goal is to present an approach, not a specific ontology.

Definition 4 (Action)**.**

The event that is carried out by an agent.

This is a very general notion of action, it characterises an action as an event with a certain relation with an agent. The key terms for the formalisation are: Action(x), Event(x), Agent(x), and Does(x, y) [read: x does y]. We then define action, in the language of first-order logic $\mathcal{L}$ expanded with these non-logical predicates, as follows:

$$Action(x) \overset{\text{def}}{\leftrightarrow} Event(x) \wedge \exists y\ (Agent(y) \wedge Does(y, x)) \quad (11)$$

$$Does(x, y) \rightarrow Agent(x) \wedge Action(y) \quad (12)$$

Definition 5 (Action)**.**

An event done by an agent for a reason.

We take the following non-logical vocabulary to formalise this definition: Event(x), DoForReason(x, y, z) [read: x does y for z], Reason(x) and Agent(x). A formalisation of the definition in $\mathcal{L}$ with the new elements is:

$$Action(x) \overset{\text{def}}{\leftrightarrow} Event(x) \wedge \exists y, z\ (Agent(y) \wedge DoForReason(y, x, z)) \quad (13)$$

$$DoForReason(y, x, z) \rightarrow Agent(y) \wedge Action(x) \wedge Reason(z) \quad (14)$$

Definition 6 (Action)**.**

A bodily movement by an agent.

The definition does not refer to generic events but to the specific subclass of bodily movements. We take as our vocabulary: Movement(x), Does(x, y) [read: x does y] and Agent(x). The formalisation of this definition is:

$$Action(x) \overset{\text{def}}{\leftrightarrow} Movement(x) \wedge \exists z\ (Agent(z) \wedge Does(z, x)) \quad (15)$$

$$Does(x, y) \rightarrow Agent(y) \wedge Action(x) \quad (16)$$

How to be (Technical) artefact

Finally, we present two notions of artefact among the three presented in (Borgo et al. 2014). We report them in a simplified version to work with small formulas.

Definition 7 (Artefact)**.**

A (physical) object obtained by an agent by selecting a physical entity and attributing to it a (technical) quality.

This definition is quite general. According to this view, when one chooses a pebble to use it as a paperweight, she creates an artefact. The paperweight has a new distinct property with respect to the pebble, namely, the attributed capacity to perform as a paperweight. (For the sake of the presentation, we have ignored that the selected entity is ontologically constituent of the artefact.) The non-logical vocabulary that we will use to characterise this notion is: Object(x), Agent(x), Quality(x), Select(x, y) [read: x selects y], Attribute(x, y, z) [read: x attributes y to z]. Our rough formalisation is:

$$\begin{aligned} Artefact(x) \overset{\text{def}}{\leftrightarrow} Object(x) \wedge \exists y, q(&Agent(y) \wedge \\ &Quality(q) \wedge Select(y, x) \wedge Attribute(y, q, x)) \quad (17) \end{aligned}$$

$$Select(y, x) \rightarrow Agent(y) \wedge Object(x) \quad (18)$$

$$Attribute(y, q, x) \rightarrow Agent(y) \wedge Quality(q) \wedge \wedge Object(x) \quad (19)$$

The next definition is engineering oriented:

Definition 8 (Artefact)**.**

A (physical) object that is made by an agent and has some given behaviour.

---

[6] Definition in (Russell and Norvig 1995).

Here we use the following non-logical vocabulary for the formalisation: Agent(x), Behavior(x), Make(x, y) [read: x makes y], HasBehavior(x, y) [read: x has behavior y], IntendBehavioFor(x, y, z) [read: x intends y to be a behavior of z] and Object(x).

Then, here is a possible formalisation:

$$Artefact(x) \overset{\text{def}}{\leftrightarrow} Object(x) \land \exists y, b(Agent(y) \land Make(y, x) \land Behavior(b) \land Behave(x, b) \land IntendBehaviorFor(y, b, x)) \tag{20}$$

$$Make(y, x) \rightarrow Agent(y) \land Object(x) \tag{21}$$

$$Behave(x, b) \rightarrow Object(x) \land Behavior(b) \tag{22}$$

$$IntendBehaviorFor(y, b, x) \rightarrow Agent(y) \land Behavior(b) \land artefact(x) \tag{23}$$

We collected a variety of syntactic definitions about three cognitively relevant notions: agent, action and artefact. We assume that every formalisation of each single domain category has been checked to be satisfactory with respect to the intended meaning and logically consistent. The characterisation of domain notions is generally limited so that consistency is fairly easy to verify but this really depends on the structure of the axioms and the interactions between them. Note that notions can very well contain negative conditions. Different is the case of self-referential or recursive definitions like “an artefact is an object made by an agent which is not itself an artefact”. The possibility to include these depends on the chosen formal language.

Underspecified definitions

So far we have listed some informal understandings of common sense notions like agent, action and artefact. As we have seen, to model the intended constraints in logic one has to choose a suitable non-logical vocabulary. This is usually a delicate step that requires a general view of the system one aims to reach. Here, however, we do not assume there is such a general system. We want only to ‘state’ the constraints in a logical form. For this reason, we introduced as non-logical vocabulary the terms used in those informal definitions and considered them as unary or n-ary predicates depending on how we read the descriptions and how they are usually understood in natural language. That is, we wanted this step to be direct and unconstrained, even at the cost to be naïve.

Let us see what we have achieved. The interpretation of the term ‘agent’ in axiom (1) is constrained by a logical formula, namely, a conjunct in which one of the subformulas is an existential. This is indeed the structure of the Definition 1: an agent is a thing such that there exist other things in such and such relationship with it.

Clearly, the interpretation of the term ‘agent’ is captured by Definition 1 only if the interpretation of ‘animate entity’, ‘capable’, ‘doing’, and ‘purpose’ are as intended. This, however, is not something the informal definition of agent deals with. Indeed, the definition *assumes* that one knows what ‘animate entity’, ‘capable’, ‘doing’, and ‘purpose’ stand for. Furthermore, it does not even require that we all understand them the same way. This is exactly what we are doing in the formal language as well. We claim that, assuming we have an ontology that tells us (i.e. constrains) the interpretation of Animate, Purpose, HasCapability and DoFor, then axiom (1) can be used in that ontology to define the category of agents. This category will contain the entities that are seen as agents by our informal definition. Note that, although we are neutral on the interpretation of the relations HasCapability and DoFor, we constrain their arguments to be of a certain type: agents and qualities for the first, agents, qualities and purposes for the latter. This is not strictly necessary but since here we do not discuss relation characterisation in general, for the time being we make this minimal commitment.

Note that there is nothing special about Definition 1. If one prefers the view proposed by Definition 2, then she assumes that to be an agent it suffices to ensure that there is an entity of a certain kind and a special relationship between the agent and that thing. As it happens, this entity should be an environment and the relationship should constrain the agent to act in that environment. Assuming that $Environment$ and $ActIn$ are correctly constrained in the ontology, axiom (4) does the job as needed.

The formal definitions given by axioms (1) and (4) use different non-logical vocabulary except for the predicate $Agent$ that they both aim to define. This is not true for Definitions 2 and 3, corresponding to axioms (4) and (6), since these share also the predicate $Environment$. Generally speaking, two distinct definitions may very well use the same non-logical vocabulary. Note however that this is only a syntactic correspondence. One can interpret $Environment$ in axiom (4) differently from the interpretation of that predicate in axiom (6). The occurrences of $Environment$ in these two axioms are unrelated as this predicate is here not (yet) associated to a formal definition or characterisation. This holds for all the non-logical vocabulary, relations included.

Finally, note that the KRer should use just one definition for each term. If one needs to use a notion in more than one sense, clashes can be avoided by renaming. For example, one can include a notion of $Capacity$-$Agent$ (from Definition 1) and a notion of $Acting$-$Agent$ (from Definition 2), provided the non-logical vocabulary are distinguished where needed.

The role of underspecified definitions

We are looking for a methodology that makes it easy for the KEer to build an ontology which:

- is fairly well built in the sense that it characterises the categories via rich cross-categorical relations, thus beyond their positioning in the taxonomy or the list of their simple properties, and
- contains all the relevant notions and these are characterised as the KEer desires.

Assume a KEer needs to build an ontology for an application/scenario A about agents and tools (e.g. a reference ontology for manufacturing). At the moment, from our discussion on cognitive categories, all she can do is to choose a notion of agent suitable for scenario A, say Definition

1. Similarly, she chooses a notion of artefact matching A, say Definition 7. These choices isolate two sets of formulas which the KEer wants to put together but first, since some non-logical vocabulary is shared by the two characterisations, namely $Object$, she has to decide whether all $Object$'s occurrences have the same interpretation or not. If not, she needs to relabel $Object$ in one of the definitions to distinguish the two predicates. In our case, the notion of artefact is restricted to physical objects, thus $Object$ in (17) has a different meaning. She thus renames $Object$ in (17), (18) and (19) as $PhysicalObj$ and adds the formulas:

$$PhysicalObj(x) \rightarrow Object(x) \tag{24}$$

$$\exists x(Object(x) \land \neg PhysicalObj(x)) \tag{25}$$

At this point, the KEer puts together the formulas characterising the notion of agent and the notion of artefact to obtain a formal theory in the language $\mathcal{L}$ extended as needed. Let us call T_0^{voc} the set of axioms that constrains the non-logical vocabulary of the theory, i.e., $T_0^{voc} = \{(1), (2), (3), (17^*), (18^*), (19^*), (24), (25)\}$, where (n^*) indicates that in that formula the $Object$ predicate has been renamed $PhysicalObj$. We will assume that T_0^{voc} has been checked for consistency.[7]

There are several predicates in the non-logical vocabulary of T_0^{voc} that are not characterised, for instance, $Action$, $Purpose$ and $Object$. Depending on the application/scenario A, we can decide that constraining the notion of $Action$ is important and that of $Purpose$ is not. We then can choose a suitable definition of action according to the scenario A, say Definition 5. Note that $Agent$ and $Artefact$, if occurring in the chosen characterisation of $Action$, are as characterised earlier by the KEer.[8] Only the still undefined terms may need to be discussed, whether to unify or to distinguish them, in order to avoid conceptual mismatches. After all, our initial labelling was completely naïve and unconstrained. For instance, in our example the KEer should consider whether the notions of $Purpose$ and $Reason$ are to be unified (by renaming one or adding an equivalence axiom). Assuming this is not the choice of the KEer, her set of interest is $T_1^{voc} = \{(1), (2), (3), (17^*), (18^*), (19^*), (24), (25), (13), (14)\}$. Again, this theory has to be checked for consistency before moving on.

At this point we can go on and add characterisations of other notions that are deemed important in the scenario A. If these are not already formalised, as done for agent, action and artefact in Section *Cognitive categories*, the KEer can provide a suitable characterisation without much trouble: as before, the formalisation of a notion is done independently of other notions. As this process evolves, a library of (cognitive) category formalisations will start forming increasing reuse, reducing the time to build sets T_n^{voc} and reducing the KEers' future efforts in ontology construction. But if these notions are not deemed relevant in A, they will not be formally characterised. These are the subsidiary categories of the ontology for scenario A.

In this section we have discussed axioms and theories, not ontologies. We have shown how to build a logical theory that contains predicates with a formalisation driven by informal definitions. Such a theory is not an ontology: it does not give a view of reality, it does not commit to a vision of its constituents. To turn a theory, like the one based on the set T_n^{voc}, into an ontology a further step is necessary: the introduction of structural categories.

The need for a top-level structure

We have seen how to combine underspecified definitions (and accompanying axioms) into a logical theory. Let us assume that we have completed this step reaching a theory T_*^{voc} that satisfies the modelling view of the KEer for the scenario A at stake. In this theory, some predicates are axiomatised to take into account their informal definitions (the domain categories), others are not (the subsidiary categories). Our next goal is to develop an ontology from it. Fortunately, most of the work has already been done.

As a preliminary step, let us list all the predicates occurring in T_*^{voc}, call this L_*. Thus, L_* includes $Object$, $Purpose$, $Quality$, $Event$ and others.

An ontology encodes a view of the world by stating what is assumed to (possibly) exist and how to subdivide what exists in types depending on their essential properties. This subdivision is typically developed into a hierarchy of categories (formally these are classes), i.e., a taxonomy. Such a taxonomy is missing in our formalisation T_*^{voc} and the goal of this section is to provide one taxonomy coherent with the KEer's modeling choices.

Taxonomies are important but also complex to build correctly (Guarino and Welty 2009). For this reason, we suggest to reuse one of the existing foundational or upper ontologies. There are many one can choose from, e.g., BFO, BORO, DOLCE, GFO, UFO, YAMATO to name a few.[9] They are not equivalent. For instance, BORO rejects the existence of objects in the standard sense (the so called 3D entities) and if this restriction is in contrast with the scenario at stake, it should not be used. Some techniques to select among upper ontologies are being developed, e.g., (Khan and Keet 2012) although much work still needs to be done. The formal language in which the ontology is available is also an important factor in the decision. Fortunately, many ontologies are available in several languages and some tools to translate across languages or to merge them are also being developed (Lange et al. 2012).

The KEer includes the selected foundational ontology or a fragment of it in the theory of T_*^{voc}, provided the ontology (fragment) includes the structural categories (perhaps after renaming) and covers the domain and subsidiary categories in L_*. Here are three cases to guide the matching between

---

[7] If it is not consistent, then the chosen definitions capture incompatible views on the primitives and the KEer has to verify her choices or rethink her understanding of the domain.

[8] More precisely, they can be a specialisation or generalisation of these but this relationship has to be formally captured.

[9] http://en.wikipedia.org/wiki/Upper_ontology

the taxonomy and L_* under the assumption that the predicates in L_* are independent from each other:[10]

- If a predicate P of L_* identifies a category C in the taxonomy and the extension of C does not correspond to the KEer's informal interpretation of P, then rename P in L_* and in T_*^{voc} with a fresh predicate;
- If a predicate P of L_* identifies a category C in the taxonomy and the extension of C corresponds to the KEer's informal interpretation of the predicate, then leave L_* and T_*^{voc} unchanged;
- If the extension of a category C in the taxonomy includes more entities than the KEer's informal interpretation of a targeted predicate P in L_*, then introduce in the taxonomy a subcategory corresponding to P as a child of C and add an axiom to characterise the extension of the new subcategory with respect to C.

If these cases do not suffice to associate each predicate in L_* to a (subcategory of) a category in the ontology, then the KEer should reconsider the top-level ontology (or the fragment) she started with.

Are there specific characteristics that help to identify a suitable top-level taxonomy? This really depends on the set T_*^{voc} and the scenario A. Generally speaking, the taxonomy provides only a few hierarchy levels since once it arrives at notions in the mesoscopic "cognitive" level, T_*^{voc} itself will supply the domain categories as needed. An important structural aspect that the top-level ontology should provide is relative to the notions of space and time. Usually, it is expected that these come from the top-level ontology with just relatively few constraints, e.g., mereotopological relations for space and partial linear order for time. Usually cognitive concepts do not provide detailed information about space and time but their interactions may be sensitive to special assumptions on space and/or time. The problems that can arise are subtle and need to be studied more deeply.

Finally, although we concentrated on unary predicates that identify categories, T_*^{voc} constrains n-ary relations as well. For this reason, the ontology should provide information on basic ontological relations like structural relations (subclass, parthood and instance_of). Other ontological cross-categorical relations like participation, constitution and dependence, should be minimally constrained in the ontology.

To complete the example, Fig. 1 shows a taxonomy extracted from DOLCE-CORE (Borgo and Masolo 2009) which is suitable for our $Tvoc_1$. Of course, the KEer has to make some important choices to link her categories with the top-level ontology, e.g., whether to interpret $Purpose$ as a desired state, a subcategory of DOLCE-CORE $Object$, or as an information entity, a subcategory of DOLCE-CORE $Concept$. Note that often a foundational or upper ontology cannot be pruned of some branches due to cross-categorial relationships.

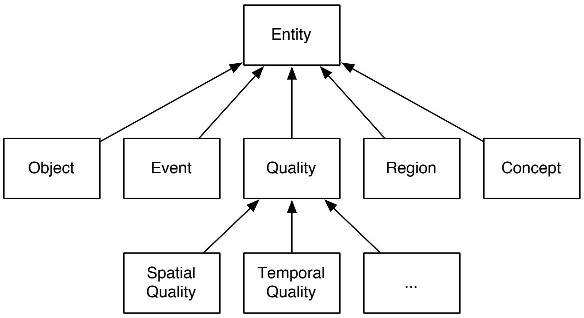

Figure 1: A top-level ontology suitable to structure our example.

Finally, the ontology obtained by merging T_*^{voc} and the chosen top-level ontology (possibly pruned) must be checked for consistency. This step, when not using a decidable language, is often a challenge for today's software.

Discussion

Our approach attempts to reduce the effort to produce a formal and application-driven ontology and to increase ontology reuse in applications. The difficulty to build robust application-driven ontologies has led to introduce many *light* axiomatised systems that are satisfactory neither for modelling nor for reasoning purposes. Furthermore, one needs well formalised and ontologically sound systems to ensure interoperability and robustness over time.

We aim to harvest as much as possible the KEer's intuitions at the level where these intuitions are mostly reliable and tested, i.e., at the level of the application. Where intuition may fail and everyday experience is not of help, like in the formalisation of general categories, we rely on well-known top-level ontologies since these are prepared and tested by expert ontologists. Our approach sees these two as phases of a single methodology for ontology construction and identifies their specific role in the construction process.

As all known methodologies, our approach has also some drawbacks. First, structure categories and the domain categories are typically axiomatised at different levels of precision. When a more homogeneous level is needed, one should deepen the formalisation of the domain categories, i.e., the most familiar to the KEer, since the others are already well characterised by construction.[11]

Second, the lack of an overall view for the characterisation of the domain categories leads to miss some interesting, where not important, connections. For instance, relations $DoesFor(x, y, z)$ and $Select(x, y)$, used to model Definition 1 and Definition 7, respectively, are one a qualifier of action (it says that the second argument is an intentional action) and the other the marker of an intentional action (a se-

---

[10] This is often not true, see our example with $PhysicalObj$ and $Object$. In these cases, first organise these predicates in hierarchies, then use the examples on the most general categories.

[11] The subsidiary categories are not problematic. If one of them is at some point considered relevant, it will be formalised and added to the list of domain categories. This can be done at any moment since the domain categories are all modeled independenty of each other.

lection) but this is not detected by the direct axiomatisation. These links require an analysis much deeper than what the standard KEer may be willing to do. The fact that even other existing methodologies, based on modularity or else, cannot cope with this issue indicates that some new idea is needed.
Third, from our limited experience the categories tend to be better characterised by cross-categorical relations. The addition of basic properties (attributes) was not discussed in this presentation and can be seen mostly as an independent task, better if guided by the adopted upper ontology.

Fourth, the notions of space and time are tricky. They should be introduced by the top-level ontology because of their generality and yet their characterisation should be flexible: some applications are based on a qualitative characterisation while others on a quantitative characterisation. It might be better to develop dedicated modules for these notions. This needs to be evaluated carefully. A similar observation applies to relations like parthood, constitution and dependence.

Fifth, the introduction of a library of cognitive notions as suggested in "Cognitive categories" section, raises the problem of how to identify the needed definition among all those available. Indeed, it is easy to generate many similar, yet not equivalent, informal definitions for the same notion. Of course, the KEer could generate a new one every time, due to the limited effort they require, giving up on reuse and making harder to evaluate the quality of the modules.

Finally, sixth, it remains unclear whether this methodology has actual (implementation) advantages in real applications and/or reduces the impact of known problems in other approaches (e.g. the merging of distinct modules). This point is something we cannot properly address at this stage of our investigation.

Aknowledgements

The approach was sketched in the author's keynote at WoMO 2011 and, under his supervision, Maxwell Henneh developed a few examples in Protégé as part of his bachelor thesis (Henneh 2014). The author thanks the reviewers for their comments and suggestions.

References

[Borgo and Masolo 2009] Borgo, S., and Masolo, C. 2009. Foundational Choices in DOLCE. In Staab, S., and Studer, R., eds., *Handbook on Ontologies*. Springer Verlag, 2nd edition. 361–381.

[Borgo et al. 2014] Borgo, S.; Franssen, M.; Garbacz, P.; Kitamura, Y.; Mizoguchi, R.; and Vermaas, P. E. 2014. Technical artifacts: An integrated perspective. *Applied Ontology Journal* 9(3-4):217–235.

[Borgo 2011] Borgo, S. 2011. Goals of modularity: a voice from the foundational viewpoint. In *Modular Ontologies (WOMO 11)*. Amsterdam: IOS Press. 1–6.

[Euzenat and Shvaiko 2007] Euzenat, J., and Shvaiko, P. 2007. *Ontology matching*. Heidelberg (DE): Springer-Verlag.

[Fernández-López, Gómez-Pérez, and Suárez-Figueroa 2013] Fernández-López, M.; Gómez-Pérez, A.; and Suárez-Figueroa, M. C. 2013. Methodological guidelines for reusing general ontologies. *Data & Knowledge Engineering* 86(0):242 – 275.

[Ghilardi, Lutz, and Wolter 2006] Ghilardi, S.; Lutz, C.; and Wolter, F. 2006. Did I damage my ontology? A case for conservative extensions in description logics. *In Proc. of KR2006.*

[Grüninger and Fox 1995] Grüninger, M., and Fox, M. S. 1995. The role of competency questions in enterprise engineering. In *Benchmarking—Theory and Practice*. Springer. 22–31.

[Guarino and Welty 2009] Guarino, N., and Welty, C. 2009. An overview on ontoclean. In Staab, S., and Studer, R., eds., *Handbook on Ontologies*. Springer Verlag, 2nd edition. 201–220.

[Guarino, Oberle, and Staab 2009] Guarino, N.; Oberle, D.; and Staab, S. 2009. What is an ontology? In *Handbook on ontologies*. Springer. 1–17.

[Guarino 1998] Guarino, N. 1998. Formal ontology in information systems. In Guarino, N., ed., *Proceedings of the Second International Conference on Formal Ontology in Information Systems*, 3–15. IOS Press.

[Guarino 1999] Guarino, N. 1999. The role of identity conditions in ontology design. In Freksa, C., and Mark, D. M., eds., *Spatial Information Theory - Cognitive and Computational Foundations of Geographic Information Science. Proceedings of International Conference COSIT '99.* Berlin: Springer Verlag. 221–234.

[Henneh 2014] Henneh, M. 2014. Study and development of a methodology to build ontologies in a modular way. Bachelor thesis, Free University of Bozen-Bolzano.

[Khan and Keet 2012] Khan, Z., and Keet, C. M. 2012. Onset: Automated foundational ontology selection and explanation. In *Knowledge Engineering and Knowledge Management (EKAW 2012)*. Springer. 237–251.

[Lange et al. 2012] Lange, C.; Kutz, O.; Mossakowski, T.; and Grüninger, M. 2012. The distributed ontology language (dol): ontology integration and interoperability applied to mathematical formalization. In *Intelligent Computer Mathematics*. Springer. 463–467.

[Poole and Mackworth 2010] Poole, D. L., and Mackworth, A. K. 2010. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press.

[Russell and Norvig 1995] Russell, S., and Norvig, P. 1995. *Artificial Intelligence. A modern approach.* Prentice-Hall, Egnlewood Cliffs.

[Simperl and Tempich 2006] Simperl, E. P. B., and Tempich, C. 2006. Ontology engineering: a reality check. In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*. Springer. 836–854.

[Simperl 2009] Simperl, E. 2009. Reusing ontologies on the semantic web: A feasibility study. *Data & Knowledge Engineering* 68(10):905–925.

[Trypuz 2008] Trypuz, R. 2008. *Formal Ontology of Action. A Unifying Approach.* Lublin (Poland): Wydawnictwo KUL.