A Core Ontology of Macroscopic Stuff

C. Maria Keet

Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za

Abstract. Domain ontologies contain representations of types of stuff (matter, mass, or substance), such as milk, alcohol, and mud, which are represented in a myriad of ways that are neither compatible with each other nor do they follow a structured approach within the domain ontology. Foundational ontologies and Ontology distinguish between pure stuff and mixtures only, if it contains stuff. We aim to fill this gap between foundational and domain ontologies by applying the notion of a 'bridging' core ontology, being an ontology of categories of stuff that is formalised in OWL. This core ontology both refines the DOLCE and BFO foundational ontologies and resolves the main type of interoperability issues with stuffs in domain ontologies, thereby also contributing to better ontology quality. Modelling guidelines are provided to facilitate the Stuff Ontology's use.

1 Introduction

Stuffs, or, depending on one's background, also called matter, mass, or substance, and normally denoted with mass nouns in natural language, are generally considered to deserve their own ontological category distinct from countable objects. They are entities such as wood, milk, water, honey, and whipped cream, that can be counted only in specific quantities. They are important in many subject domains, such as the environment (e.g., soil, air in the environment ontology EnvO [7]), medicine with the various body substances (blood and fat in SNOMED CT [29], and in manufacturing (an amount of oil greasing the machine, steel [18]). A more elaborate ongoing use case is that of traceability of food and its ingredients [11], concerning handling bulk goods [31], such as soybean, flour, and breakfast cereals: the flow of the ingredients have to be monitored in the food production chain for food safety and swift and adequate response in case of problems. This requires a way to track parts and portions of the bulk, and mixing them with other portions of stuff. From a modelling viewpoint, this requires representing the stuffs adequately, how the stuffs relate, and their temporal dimension; we focus on the first topic in this paper.

Related works about stuff can be divided roughly into philosophy, foundational ontologies, and domain ontologies. Discourses in Ontology (philosophy) concern portions, quantities, constitution, and distinguishing stuff from object [2, 5, 9, 12] and the nature of stuff and it representation [9, 24, 32], which, if it accepts stuff, remains at a very basic distinction between 'pure' stuff that has one kind of basis or granule (e.g., gold, water) and 'mixed' stuff (e.g., lemonade, milk), and the philosophy of chemistry [6, 25, 24], that concerns itself with, among others, deuterium in water and historical interpretations of the nature of stuff. The former is too generic and the latter too detailed for most domain ontologies. Foundational ontologies, such as DOLCE [23], BFO [4], and GFO [17], have no or one stuff category, hence do not provide any modelling guidance in that regard. Contrast this with domain ontologies, which have from several (BioTop [3]) to very many (SNOMED CT) kinds of stuff. For instance, SNOMED CT's Body substance has as direct subclasses, among others, Nervous system hormone-like substance (pure stuff), and Regurgitated food and Breath (mixtures), and EnvO's Environmental material has as direct subclasses Anthropogenic environmental material, Clay, and Foam. The main issue for handling stuff consistently is due to the gap between Ontology and foundational ontologies on the one hand, and domain ontologies on the other, and in particular the absence of methodological guidance where the former may be applicable to, and beneficial for, the latter. A recent overview of methodologies [15] notes the absence of foundational ontology use in all surveyed methodologies.

Thus, there is a disconnect between the foundational ontologies on the one hand, and domain ontologies on the other, yet there is a pressing need for adequate representation of stuffs in domain ontologies. To address this issue, we apply the notion of a bridging *core ontology* that connects the very general contents of a foundational ontology to the entities represented in domain ontologies. Core ontologies have been developed elsewhere for, among others, events and multimedia entities, and has some general guidelines for its development [27]. Here we describe the Stuff Ontology, which functions as such a core ontology. To address this comprehensively, one has to consider the essential features of stuff compared to other categories, which stuff categories have to be added to suffice for its application in domain ontology development, and find a way to formalise it adequately. We focus on the latter two, and assume that stuff exists and deserves its own category. We describe the modelling decisions, including the refinement of the notion of 'pure' stuff and specialisation of mixtures into homogeneous and heterogeneous mixtures, and several subcategories, and several roles and basic relationships have to be introduced, such as the 'basis' type (granule) a stuff is made up of and what sub (part) stuffs it has. A most precise formalisation requires either second order logic or a sorted logic, but to keep usability in mind, we formalise this ontology in OWL so that it can be imported easily into extant ontologies and actually be used in comparatively widely used software infrastructure for ontologies. We evaluate the core ontology against the set of desirable features specified in [27] and add (potential) usability to it, including a decision diagram, linking the stuff ontology to DOLCE and BFO, and illustrating how stuff can be represented more accurately in domain ontologies for interoperability by availing of this core ontology, using the nature of the stuff as modelling criterion rather than the various roles it may play.

In the remainder of the paper, we discuss other ontologies (Section 2) and present the stuff ontology in Section 3. This is followed by implementation aspects and evaluation of the core ontology in Section 4, and we close in Section 5.

2 Related works: Ontology and ontologies

Davis [9] identified five options how to represent stuff—particles, fields, two for continuous matter, and a hybrid model—where a hybrid one with portions and particles is the only one used in both the literature and ontologies. Among the hybrids, distinctions are drawn regarding shareability of parts, with sums for 'pure' kinds of stuff and rigid embodiments for mixtures [2], therewith distinguishing stuff with instances of a single element or molecule as part from those that have instance of more than one different type of molecule with a stuff-forming relation between them [2, 12], inclusion of time and modality [12, 25], and interaction of quantities of stuff and spatial character of mereological relations [25]. First, we grant stuff its own ontological category distinct from objects, and we agree with a hybrid model. Second, we restrict it here to the categories and kinds of stuff and an a-temporal view, because most ontologies and related computational infrastructure are only for atemporal logical theories.

Let us assess several foundational ontologies on their inclusion of stuff. DOLCE's Amount of matter has examples such as gold, wood, sand, and meat with as commonality "that they are endurants with no unity ... [and] are mereologically invariant" [23]; it has no subcategories. GFO has Mass_entity ≡ Amount of substrate, which is a "special persistant whose instances are distinct amounts at certain timepoints" [17]; it has no subcategories either. BFO v1.1 does not have stuff, though object aggregate leans in that direction (a mereological sum of its members, alike a fleet); pedantically, it means that according to the philosophical commitments of BFO, stuff does not exist. The most recent SUMO ([30] cf. [26]) has Substance, which is something "in which every part is *similar* to every other in every relevant respect...any *parts* have *properties* which are *similar* to those of the *whole*" (emphasis added). Its subclass PureSubstance covers both the elements and compounds and has subclasses such as GreenhouseGas (indeed a stuff) and HydrophilicLipidHead (part of a molecule, i.e., and object), whereas Mixture is "two or more pure substances, combined in varying proportions - each retaining its own specific properties.", with direct subclasses, e.g., Glass ($\geq$1 pure stuff) and Sewage ($>$1 pure and mixed stuffs). It has a partition into natural and synthetic substances, and with direct subclasses including Beverage, Electricity, and Juice. Thus, SUMO has categories and kinds of stuff, and, as we shall see, useful informal definitions, but the modelling criteria for the domain stuffs are too limited and not uniform. Cyc [8] has an undefined StuffType with subclasses TemporalStuffType and ExistingStuffType, and a relationship granuleOfStuff that has StuffType as domain and ObjType as range. Its hierarchy contains EdibleStuff as a direct instance of ExistingStuffType, which is a direct generalisation of FoodOrDrink, Nutrient, CerealFood, FruitAndVegetableFood, and so on. Aside from the instantiation issue, they are *roles* that both objects (e.g., apple) and stuffs (e.g., vitamin A) play from the anthropocentric viewpoint, not that they rigidly are those things.

Regarding domain ontologies, we could find only one that addresses stuff explicitly, being the ontology for maintenance actions of industrial objects [18]. It has at least seven subcategories of stuff, distinguishing between pure and mixed stuff (with further division in solution and emulsion, but not other colloids) and state-based divisions (solid, liquid, and gas); its KIF3.0 version is not available, however. BioPortal lists many ontologies that include stuff and 4 were selected randomly. The loggerhead nesting ontology [22] has Stuff with four subclasses: air, mucus, sand, water. EnvO's environmental material has direct subclasses such as clay and water (particular kinds of stuff) and foam (a type/category of stuff). Its food product—potentially overlapping with SUMO's food classes—contains stuffs, such as sugar (the sucrose, 'table sugar'), meat, milk, and so on organised mostly by source (animal, fish, plant) but mixed with other criteria (e.g., nonfat dry milk). In [2, 12]'s viewpoint, however, sugar is an unstructured or discrete stuff, and milk and meat are structured/nondiscrete, and of which it is already known that they require different constraints to handle portions in time [12]. SNOMED CT has many kinds of stuff. Considering even just its direct subclasses, they include Allergen class (with subclasses), oil, materials (with subclasses surgical material, adhesive agent, culture medium, body material (with subclass, among others, fat and crystal)), substance of abuse, chemical, biological substance, and body substance. However, crystal is a solid state, adhesive agent is function-oriented, and substance of abuse is contextual.

Thus, there is no set of modelling criteria within and throughout the ontologies, and a gap exists on guidance from Ontology and foundational ontologies for modelling stuffs, and its consequent diverse use in domain ontologies.

3 Overview of the Stuff Ontology

Typical examples of stuffs in the Ontology literature are Water and Gold versus Lemonade, Oil, Milk, and Wood, where the first two are called unstructured, discrete, or pure kinds of stuff and the latter structured, non-discrete, or mixed stuffs [2, 5, 12] without further refinements, which is insufficient to categorise stuffs in domain ontologies. The philosophy of chemistry concerning stuff, notably [6, 25], goes into more detail than is necessary for the current scope of macroscopic stuff (such as element decay, heavy water, the structure of solids, categorisation of crystals, whether Water is the same as H_2O). We will use the very same principles, however, as considered in those works:

- a granule, grain, or basis type of the stuff, which is at one finer-grained level than the stuff itself (see also [20]);
- a so-called stuff-forming relation between the entities 'in' the stuff, i.e., that it is made of at that adjacent finer-grained level;
- homogeneous versus heterogeneous matter;
- we concur with Brakel's conclusion that the focus has to be on "macroscopic sameness" [6] and Barnett's notion of "least portion" [2]: the least portion is the smallest portion that still exhibits the macroscopic properties of that kind of stuff.

The chemistry discussed in this section can be found in any chemistry textbook; we used [19, 28] and IUPAC's Gold Book [http://goldbook.iupac.org/] to cross-check the chemistry information, and more details, explanations, and examples can be found there (the author's prior education in food science and its lecture notes aided in devising the food examples).

We introduce several categories of stuff, and define most of them in the next two sections; the taxonomy is depicted in Fig. 1, where each of the direct subclasses are disjoint, except Bulk. To make this practically useful for ontology development, a representation in OWL 2 DL is preferred thanks to its expressiveness, computational infrastructure, and relatively wide uptake, which can be slimmed to one of its less expressive profiles if needed. This extensively annotated stuff ontology is available online (see Section 4), and some key definitions are included in this paper in DL notation. It required an adjustment due to expressivity limitations, but it is outweighed by the usability argument: using a second order logic, a many-sorted logic, or variable n-ary predicate, relegates the ontology to a paper-based exercise only that would still not be practically usable in ontology development and ontology-driven information systems for, among others, food processing.

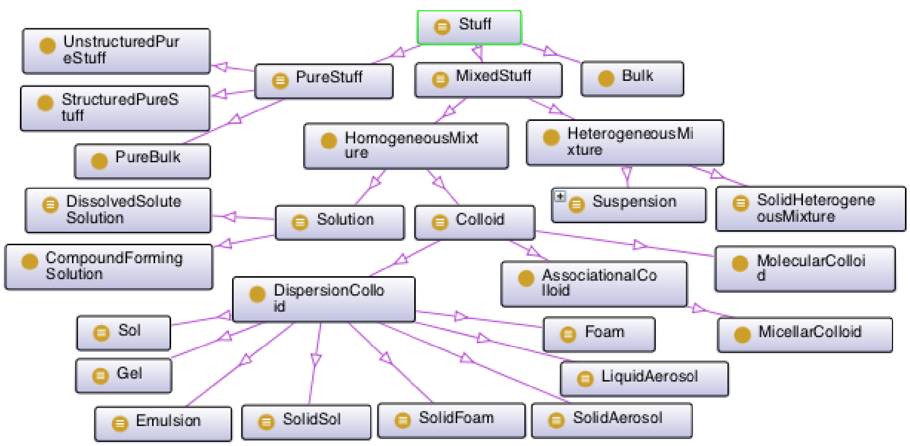

Fig. 1. Taxonomy of main categories of stuff (Ontograf rendering).

3.1 Pure stuff

Really pure stuff occurs very rarely in nature and is not interesting practically, depending on what one categorises under pure stuff, if anything, and if so, how. What is widely agreed upon is that there is pure stuff that consists of objects that instantiate only one most specific single type of granule: H_2O *molecules* in the case of water, Au *atoms* in the case of gold (one can go further down into isotopes, which is beyond the scope), but also the individual soybean, as *multi-molecular particle* (MMP) in the soybean storage silo handled as bulk. Molecules have as parts atoms and can be separated into the element or molecule form of the element and are created by putting together an amount of molecules such that each component portion loses its properties and become something else that has other properties than its constituent atoms or molecules. This is not the case for gold. There is a difference between an amount of stuff made up of instances of one of the elements as listed in the periodic table of elements, like Au, and pure stuffs that consist of molecules. They are both PureStuff, though, as the instances—some amount of that stuff—consist of instances of one type of 'basis', 'grain', or 'granule' only. Because 'grain' has specific connotations as individual entity and 'base' is a bit vague, we will use hasGranuleType to relate to the kind of entity at that next finer-grained level of analysis, which is a partial representation of the "constant basis principle" [12] and a generalisation of BioTop's and Cyc's granular parts, and is typed as in Eqs. 1-2, where MultiMolecularParticle is abbreviated as MMP. We will use hasSubStuff to relate a stuff playing the whole role with its part-stuff, in line with the taxonomy of part-whole relations [21] (Eqs. 3-4). This results in the definition for PureStuff in Eq. 5.

$$\exists\,\mathsf{hasGranuleType}.\top \sqsubseteq \mathsf{Stuff} \tag{1}$$

$$\top \sqsubseteq \forall \mathsf{hasGranuleType}.(\mathsf{Atom} \sqcup \mathsf{Molecule} \sqcup \mathsf{MMP}) \tag{2}$$

$$\exists\,\mathsf{hasSubStuff}.\top \sqsubseteq \mathsf{Stuff} \tag{3}$$

$$\top \sqsubseteq \forall \mathsf{hasSubStuff}.\mathsf{Stuff} \tag{4}$$

$$\mathsf{PureStuff} \equiv \mathsf{Stuff} \sqcap =1\,\mathsf{hasSubStuff}.(\mathsf{PureBulk} \sqcup \mathsf{StructuredPureStuff} \sqcup \mathsf{UnstructuredPureStuff}) \sqcap =1\,\mathsf{hasGranuleType}.(\mathsf{Atom} \sqcup \mathsf{Molecule} \sqcup \mathsf{MMP}) \tag{5}$$

For instance, TableSugar is a kind of PureStuff that has one type of granule, Molecule (more precisely: $C_{12}H_{22}O_{11}$), and its only sub-stuff is Sucrose. The only way to have this working in OWL, is to represent each kind of molecule as an instance in the ABox, instead of making them subclasses of Molecule. Ontologically, $C_{12}H_{22}O_{11}$ is a universal (or: OWL class) and there are very many instances of that molecule in the pot of sugar on your table. However, no-one is ever going to encode individual molecules in an ontology, and representing the molecules as classes would require second order logic regarding the knowledge about granule types.

PureStuff has three subtypes, being UnstructuredPureStuff, StructuredPureStuff, and PureBulk. The first one has as granule *one type of element*, such as a amount of gold consisting of instances of the element Au. The second one has as granule type *one type of molecule* (and the atoms are part of the molecules in a specific configuration), like water (H_2O), ethanol (C_2H_5OH), vitamin A, and so on; existing resources that contain such molecules are the Chemical Entities of Biological Interest (CHeBi) ontology [10] and the Chemical thesaurus [`http://www.chemthes.com`]. In chemistry, these entities are called *compounds*. The reasons for this distinction are the differences in granule and divisibility with compound formation and destruction in the right situation. One cannot simply put the right amount of C, H, and O atoms together and have as mereological sum ethanol, and ethanol can be destroyed by, among others, the enzyme alcohol dehydrogenase, such that the constituent atoms survive but not the molecule. Unlike the unstructured pure stuff, there is, in philosophy terminology, a so-called *stuff-forming relation* [2] for structured pure stuff, restricted to one that concerns covalent chemical bonds, which we denote with compoundFormingRelation and to which one can attach properties like the ratio and environmental parameters (not pursued here). This does not hold for unstructured pure stuffs. While it is the case, as with StructuredPureStuff, that dividing an amount of, say, gold to the smallest portion and down to an instance of the granule always remains gold, and no matter how we combine or sum different amounts of gold, it always remains gold (setting aside the ion formation to Au^{3+}), the compound formation with ratio of components to form the compound and right conditions is not applicable for its atoms are not bound covalently to atoms of another type, hence, it is not a compound. One could argue that elements are formed as well and therefore should be considered 'compounds' of protons, neutrons, and electrons, but this occurs at many orders of magnitude less than compound formation, and is one step more fine-grained than the next level of analysis, and therefore can be ignored. Third, and finally, PureBulk is pure stuff at the macroscopic level, where some bulk goods fit, such as an amount of tea biscuits or soybeans with as granule type the multi-molecular object TeaBiscuit (resp., Soybean). They are the macroscopic equivalent of the molecule and there is only one granule type; hence, with respect to the chosen granule type, an amount of tea biscuits with granules the multi-molecular particles as instances of TeaBiscuit is then just as much structured pure stuff. Because of the different single granule type, we shall categorise these under PureBulk.

In sum, PureStuff's subtypes differ in granule type they have, which is either one type of Atom (element), Molecule, or MultiMoleculeParticle (MMP), which are subtypes of Physical Object (represented in any of one's preferred foundational ontology); see Eqs. 6-8 for the simplified OWL version in DL rendering.

$$\begin{aligned}
&\mathsf{UnstructuredPureStuff} \equiv \mathsf{PureStuf} \sqcap \forall \mathsf{hasGranuleType.Atom} \sqcap \\
&\quad = 1\,\mathsf{hasGranuleType.Atom} && (6)\\
&\mathsf{StructuredPureStuff} \equiv \mathsf{PureStuf} \sqcap \forall \mathsf{hasGranuleType.Molecule} \sqcap \\
&\quad = 1\mathsf{hasGranuleType.Molecule} && (7)\\
&\mathsf{PureBulk} \equiv \mathsf{PureStuf} \sqcap \forall \mathsf{hasGranuleType.MMP} \sqcap = 1\,\mathsf{hasGranuleType.MMP} && (8)
\end{aligned}$$

Once it is determined the type of granule is of one kind, then there is a sameness in summation, i.e., two amounts of the same kind of pure stuff is still of that kind of pure stuff.

3.2 Mixed Stuff

Mixtures, also called "structured stuff" [12] or "nondiscrete stuff" [2], are illustrated in the philosophy literature with entities such as lemonade, oil, milk, and wood. While they are indeed all 'structured' in some way, what is meant is that they are mixed stuffs consisting of different stuffs related with a stuff-forming relation [2, 12] both when considered from the viewpoint of stuffs and as a collection of the chosen granule type. What holds for all individual mixtures is that they are composed of at least two amounts of stuff that instantiate distinct kinds of stuff (hence, have also distinct granules) that are mixed. To formalise this, we face the same issue as with pure stuff (second order logic, which is 'pushed down' into first order here as well), and that it is not possible to express that the two stuffs are distinct. For the latter reason, we use $\sqsubseteq$ instead of $\equiv$:

$$\mathsf{MixedStuff} \sqsubseteq \mathsf{Stuff} \sqcap \geq 2\, \mathsf{hasSubStuff.Stuff} \tag{9}$$

For instance, the mixture lemonade has water, sugar, and lemon juice as substuffs. Thus, mixtures are indeed different from pure stuff. However, they are not all of the same kind or not all in the same way to the extent that those differences have an effect on portions and parts. Informally, Sprite soda drink is a true *solution*, consisting of water, carbon dioxide, and sugar, whereas milk is an *emulsion* of protein and fat globules dispersed evenly in water, wood is yet again different, being a solid heterogeneous mixture, whereas mud is a *suspension* of unevenly distributed sand in water. The former two examples are kinds of homogeneously mixed stuffs and the latter is an example of heterogeneously mixed stuff, which we elaborate on in the next two subsections.

Heterogeneous mixed stuff HeterogeneousMixture is a combination of different stuffs, of which at least one has a fairly large particle size, that do not react chemically, and the stuffs that the mixed stuff is composed of can be separated by purely physical means (filtration, etc.). This can be sub-divided into a solid, liquid, and a gaseous version. The liquid version with solid particles (usually >1 μm) is called Suspension, which has as characteristics that the substance naturally separates, with separation due to sedimentation, creaming, flocculation, or coalescence unless they are 'stabilized' by some other stuff, as regularly happens in the production process; e.g., tomato juice, mud. A SolidHeterogeneousMixture is the solid version of a suspension; e.g., wood has a compartmentalisation of the components into cellulose, hemicellulose, lignin, and other stuffs like waxes. The interesting aspect here, is that the *state* of the stuffs become relevant for describing the mixtures cf. the pure stuffs, and the role they play in the mixture. Suspensions have a stuff in liquid state, called *continuous medium* or dispersion medium, and a stuff in the solid state, called *dispersed phase*, where their bearers, stuffs, can play each role depending on what is mixed. Take, e.g., the suspension Mud, where liquid water acts as continuous medium and sand as dispersed phase, whereas liquid water is the dispersed phase in Fog in the continuous medium air; hence, they are roles the stuffs play in the mixture, not that the stuffs are continuous medium (resp. dispersed phase).

Formalising this, we start with HeterogeneousMixture that has at least two types of stuff and granules (Eq. 10), inheriting from MixedStuff that it has at least two stuffkinds.

$$\begin{aligned} \mathsf{HeterogeneousMixture} \sqsubseteq\ & \mathsf{Mixture} \sqcap \geq 2\, \mathsf{hasSubStuff.(MixedStuff} \sqcup \mathsf{PureStuff)} \sqcap \\ & \geq 2\, \mathsf{hasGranuleType.(Molecule} \sqcup \mathsf{MultiMoleculeParticle)} \end{aligned} \tag{10}$$

For its two subtypes, we have to take into consideration the state that the stuffs are in, as that is an essential characteristic of the mixture. This is a property of a substance only, not as a reified solid, liquid, or gas (which have their own properties, like crystalline or malleable). A SolidHeterogeneousMixture is composed of stuffs that are in the solid state, and Suspension has the stuff that plays the continuous medium in the liquid state with stuff in the solid state dispersed in it, more than one kind of solid stuff may be suspended in the liquid continuous medium, and we use the common inheresIn relationship (from foundational ontologies) to relate roles with their bearers. Typing hasState with domain Stuff and range StuffState (i.e., $\exists\,\mathsf{hasState}.\top \sqsubseteq \mathsf{Stuff}$ and $\top \sqsubseteq \forall\mathsf{hasStuff}.\mathsf{StuffState}$), we define the two classes as follows:

$$\begin{aligned}
&\mathsf{SolidHeterogeneousMixture} \equiv \mathsf{HeterogeneousMixture} \sqcap \\
&\quad \geq 2\ \mathsf{hasSubStuff}.(\mathsf{Stuff} \sqcap \exists\mathsf{hasState}.\mathsf{Solid}) && (11)\\
&\mathsf{Suspension} \equiv \mathsf{HeterogeneousMixture} \sqcap \exists\mathsf{hasSubStuff}.(\mathsf{Stuff} \sqcap \\
&\quad \exists\mathsf{hasState}.\mathsf{Liquid} \sqcap \exists\mathsf{inheresIn}^{-}.\mathsf{ContinuousMedium}) \sqcap \\
&\quad \exists\mathsf{hasSubStuff}.(\mathsf{Stuff} \sqcap \exists\mathsf{hasState}.\mathsf{Solid} \sqcap \exists\mathsf{inheresIn}^{-}.\mathsf{DispersedPhase}) && (12)
\end{aligned}$$

Homogeneous mixed stuff In HomogeneousMixtures, the *mixed stuffs are distributed evenly* across the mixture[1]. There are two categories of stuff that can be created and remain a stable homogeneous mixture: solutions and colloids, where the former exists as one phase and the latter exists as two or more phases (which is sometimes also referred to as a homogeneous heterogeneous stuff); a *phase* is a physically distinct portion of the system, where the stuff occupying that region of space has uniform properties. A true Solution is a combination of at least two stuffs where the mixing occurs at the molecular level where some chemical reaction occurs, and the resultant is one phase. This can occur such that (*i*) one stuff that plays the role of Solute dissolves in the other stuff that plays the Solvent role such that no new compounds are formed, other than ignorable changes—like an -OH group to a $\text{-O}^{-} + \text{H}^{+}$—in the solution, e.g., some dissolved sugar in a cup of tea, or (*ii*) it can be compound-forming in a chemical reaction; e.g., to dissolve gold in aqua regia to obtain a solution with new $AuCl_4^-$ molecules. One can introduce a symmetric solutionFormingRelation, which is a stuff-forming relation between the solute and solvent, but also here one cannot assert that the stuffs that play the solute and solvent role must be distinct, so the solutionFormingRelation is only typed with Solvent as its domain and Solute as its range. Formally, a solution is made of at least two kinds of stuff where one plays the solvent and the other one(s) play(s) the solute role (Eq. 13), and each solvent must have at least one solute (Eq. 14).

$$\begin{aligned}
&\mathsf{Solution} \equiv \mathsf{HomogeneousMixture} \sqcap \exists\mathsf{hasSubStuff}.(\mathsf{Stuff} \sqcap \exists\mathsf{inheresIn}^{-}.\mathsf{Solute} \sqcap \\
&\quad \mathsf{hasGranuleType}.(\mathsf{Atom} \sqcup \mathsf{Ion} \sqcup \mathsf{Molecule})) \sqcap \exists\mathsf{hasSubStuff}.(\mathsf{Stuff} \sqcap \\
&\quad \mathsf{inheresIn}^{-}.\mathsf{Solvent} \sqcap \mathsf{hasGranuleType}.(\mathsf{Atom} \sqcup \mathsf{Ion} \sqcup \mathsf{Molecule})) \sqcap \\
&\quad \exists\mathsf{hasNrOfPhase}.\mathsf{int}_{=1} && (13)\\
&\mathsf{Solvent} \sqsubseteq \exists\mathsf{solutionFormingRelation}.\mathsf{Solute} && (14)
\end{aligned}$$

The other subclass of HomogeneousMixture is Colloid, which is a mixture with intermediate particle size, where one kind of stuff that plays the dispersed phase, is

[1] Note that 'homogeneous' can mean sameness of the parts, one phase, and/or macroscopic sameness, depending on the literature consulted [25].

microscopically dispersed *evenly* throughout another stuff, which acts as the continuous (or dispersion) medium; e.g., milk, mayonnaise, agar, and marshmallow. Particle size is the main distinguishing characteristic of colloids compared to solutions; practically, the liquid colloids can be distinguished from liquid solutions using the Tyndall effect[2] and among themselves based on either (*i*) the state of the continuous medium and dispersed phase (dispersion colloids), (*ii*) hydrophobicity (associational colloids), or (*iii*) it has molecules of colloidal dimension, i.e., very large molecules, dispersed in the medium (molecular colloids). For instance, whipped cream is a Foam that has as continuous medium stuff that is in its liquid state and as dispersed phase a stuff that is in its gaseous state; pigmented ink and blood are Sols that have as continuous medium stuff that is liquid and a dispersed phase with solid stuff; and milk and mayonnaise are Emulsions that have a liquid continuous medium and dispersed phase. Soap is a micellar (associational) colloid, with an aqueous liquid containing micelles formed by molecules with a hydrophilic head and hydrophobic tail. Latex and starchy stuffs with large polymers, such as wallpaper glue, are examples of molecular colloids.

These characteristics result in the second distinction with solutions: a colloid has at least two phases whereas solutions have only one. The phases of a dispersion colloid can be distinguished typically only under a microscope, and therefore they are perceived as homogeneous and possibly easily misrepresented as a solution. Third, the colloid has properties of its own, such as its freezing point, that is different from the separate stuffs making up the colloid.

Note that one cannot simply pour whipping cream into a bowl and it will become whipped cream by simple contact with the air, or put together oil and egg yolk arbitrarily and expect to obtain mayonnaise. They have to be mixed in a specific way such that the globules/particles are gradually added to the continuous medium and obtain the required surface tension to remain stable. As such, one can consider a specific *colloid-forming relation* between the distinct stuffs that compose the colloid, of which an instance comes into existence only when the stuff that plays the role of dispersed phase is mixed gradually with the continuous medium in order to become a stable colloid. Thus, the colloidFormingRelation holds between the continuous medium that inheres in some stuff and the dispersed phase that inheres in some stuff, not the stuffs per sé. One can philosophise whether they are roles the stuffs in the colloid play or the component-stuffs are occupiers of three-dimensional spaces that, in turn, fulfill a particular role in the space that the amount of colloid occupies. Here, we take the former, more compact, formalization. Formally, the general characterisation of Colloid is in Eq. 15. The dispersion colloid Gel is included as Eq. 16; the other seven follow the same pattern with the differences in states: Emulsion as a stuff in liquid state dispersed in another stuff that is also in liquid state and has the role of continuous medium; Foam as a gas dispersed in liquid; LiquidAerosol as a liquid dispersed in gas; Sol as a solid dispersed in liquid; SolidAerosol as a solid dispersed in gas; SolidFoam as a gas dispersed in solid; and SolidSol as a solid

[2] A beam of light passing through a true (liquid) solution is not visible, but light passing through a colloid will be reflected by the larger particles and therefore the light beam is visible.

dispersed in solid; they are included in the OWL file.

$$\begin{aligned}&\mathsf{Colloid} \equiv \mathsf{HomogeneousMixture} \sqcap \exists \mathsf{hasNrOfPhase.int}_{\geq 2} \sqcap \\ &\quad = 1 \mathsf{hasSubStuff.}(\mathsf{Stuff} \sqcap \exists \mathsf{hasState.StuffState} \sqcap \exists \mathsf{inheresIn}^{-}.\mathsf{ContinuousMedium}) \sqcap \\ &\quad = 1 \mathsf{hasSubStuff.}(\mathsf{Stuff} \sqcap (\exists \mathsf{hasState.StuffState}) \sqcap (\exists \mathsf{inheresIn}^{-}.\mathsf{DispersedPhase})) \quad (15)\end{aligned}$$

$$\begin{aligned}&\mathsf{Gel} \equiv \mathsf{DispersionColloid} \sqcap\ = 1\ \mathsf{hasSubStuff.}(\mathsf{Stuff} \sqcap \exists \mathsf{hasState.Solid} \sqcap \\ &\quad \exists \mathsf{inheresIn}^{-}.\mathsf{ContinuousMedium}) \sqcap = 1\ \mathsf{hasSubStuff.}(\mathsf{Stuff}\ \sqcap \exists \mathsf{hasState.Liquid} \sqcap \\ &\quad \mathsf{inheresIn}^{-}.\mathsf{DispersedPhase}) \sqcap \exists \mathsf{hasNrOfPhase.int}_{\geq 2} \quad (16)\end{aligned}$$

One can elaborate on surfactants, stabilizers, and enhancers; e.g., for whipping cream in homebaking, one can add solid sugar powder and cornstarch to keep the whipped cream stiff for longer, i.e., stabilise the colloid against unfavourable environmental conditions. Also, there are more types of gel, which are useful for a pure chemistry ontology, but have not been included in the stuff ontology due to its highly specialised usage; e.g., a ringing gel is a gel "with energy dissipation in the acoustic frequency range" [1].

This concludes the overview of the 25 categories of stuff, of which 16 are defined (see OWL file for the other definitions).

4 Implementation and evaluation of the core ontology

As mentioned in the introduction, the stuff ontology is aimed at bridging the foundational with the domain ontologies, keeping usability in mind. It meets all of Scherp et al.'s desirable core ontology features [27]—axiomatisation, extensibility, reusability, separation of concerns, and modularity—and we add usability. In the remainder of this section, we describe how these features are met.

4.1 OWL version of the stuff ontology

The introduction of Section 3 mentioned concessions were made to represent the knowledge in OWL, notably the design choice to stay within a decidable fragment of first order logic. The Stuff Ontology is highly axiomatised nevertheless, and has 52 classes, 18 object properties, 266 axioms and is in $\mathcal{SHIQ}(D)$ (OWL 2 DL). Because availing of automated classification services can be of use, especially for the colloids and solutions, it is more useful to not use just a generic untyped inheresIn relation, but one where the domain and range are declared in a way relevant for the ontology, being one where the range is Stuff and domain is StuffRole that subsumes ContinuousMedium, DispersedPhase, Solute, and Solvent. Therefore, srInheresIn is introduced as an object subproperty of inheresIn. Then, when an ontologist adds, e.g., Water, Sand, and Mud, with the water as the continuous medium and sand the dispersed in water, the reasoner classifies Mud as a subclass of Suspension, which it would not have done if we had used inheresIn. Also disjointness axioms have been added where relevant, and most entities have annotations with explanations and examples. The ontology and the extended versions (introduced below) are available online at `http://www.meteck.org/files/ontologies/`.

4.2 Modelling guidance for the main categories of stuff

Most types of stuff in the Stuff Ontology have a quite elaborate axiomatization, which potentially hampers its usability. Although it is not a highlighted feature of core ontologies, we deem efforts toward *usability* relevant, for it is needed to facilitate reusability in the field. In addition, options such as Ontology Design Patterns [14] and templates [13] can help and serve the modularity feature of core ontologies, and templates have been shown to be particularly useful with core ontologies, for being at the appropriate level of granularity according to the evaluation carried out by [13]. However, this will work for the Stuff Ontology only once one knows which stuffkind it is. To this end, we designed a decision diagram that in an informal way guides the domain ontology modeller to the appropriate type of stuff. This diagram is included in Fig. 2 and is intended as an informal aid and as a general overview of the Stuff Ontology's contents.

Let us step through it for Mayonnaise. It is not made up of only one type of atom, molecule or larger particles of the same type, hence, it is a MixedStuff. The stuffs it is made up of are evenly distributed throughout, hence it is a HomogeneousMixture. Let's say we do not know whether it consists of one phase, so we move to the alternative question asking whether the component stuffs of mayonnaise—egg yolk and oil—keep their phase, which is 'yes', hence, it is a Colloid. It has nothing to do with hydrophobicity or very large molecules, hence it is a Dispersion Colloid. As both the egg yolk and oil are liquids, option a) is chosen, and we arrive at the endpoint where mayonnaise is an Emulsion. A contextual template or stuffkind-dependent ODP can then be generated to facilitate adding the definition of the domain entity. A contextualised table on the left in Fig. 3 would suffice as input to generate the axiom on the right in the figure.

A separate `stuff-example.owl` contains several domain stuff entities to illustrate the automated classification of stuffkinds in the correct category, such as Mayonnaise and Gold. These entities are distinctly domain ontology entities and therefore they are not included in the Stuff Ontology itself; likewise, a line has been drawn not to include the fine-grained chemistry scientific knowledge (such as the aforementioned ringing gel). Thus, the feature of separation of concerns has been adhered to.

4.3 Ontology interoperability

To evaluate whether the Stuff Ontology can serve as a core ontology that bridges foundational ontologies and domain ontologies, we first link it to DOLCE and BFO, and subsequently turn to domain ontologies.

Linking to a foundational ontology. We tested linking the Stuff Ontology to DOLCE-Lite and BFO v1.1.1, which was successful and these combined ontologies are also online. One might argue against aligning it to BFO because it does not have stuff, but the EnvO is being aligned to BFO [7], entailing that stuff has to be handled in some way. The links in brief: Stuff is made equivalent to DOLCE's Amount of matter and a subclass of material_entity in BFO, both

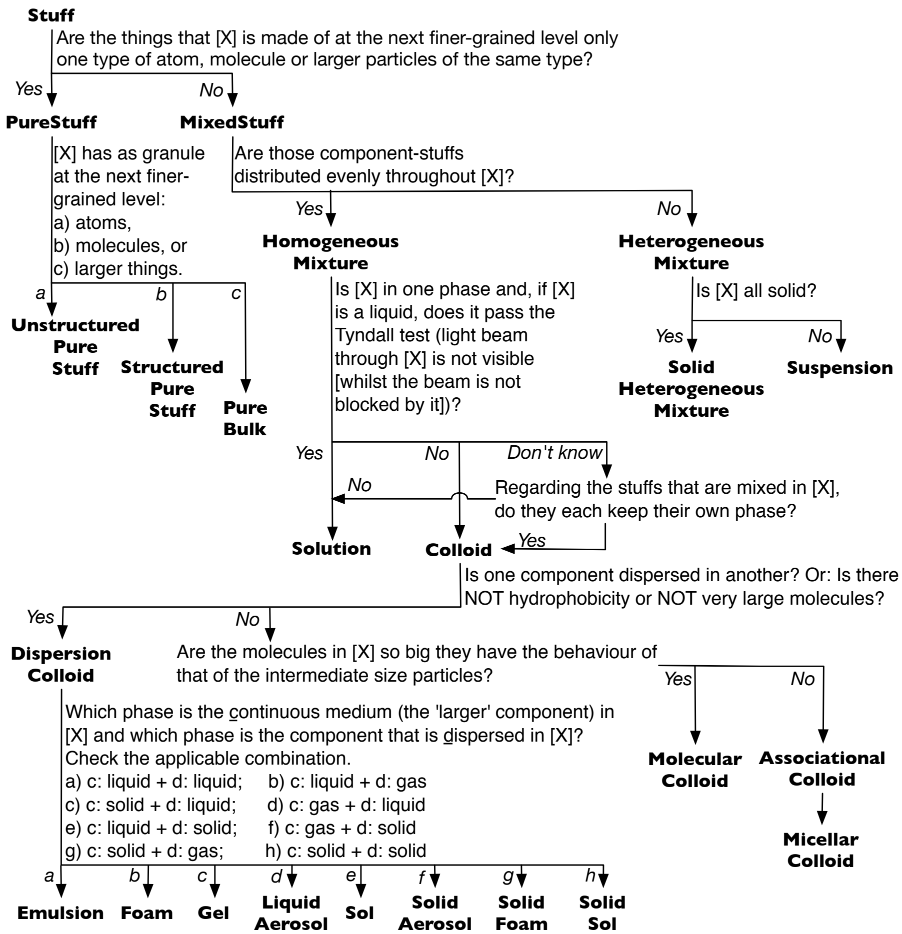

Fig. 2. Decision tree to find the principal kind of stuff of a domain stuff entity.

A. Extensible sample template

Feature	Answer
Mayonnaise has as sub-stuff:	Eggyolk
Mayonnaise has as sub-stuff:	Oil
The stuff that is the continuous medium is:	Egg yolk
The stuff that is dispersed in it is:	Oil
Click here to add more sub-stuffs	

B. Definition added to the ontology

$$\mathsf{Mayonnaise} \equiv \mathsf{DispersionColloid} \sqcap {=1}\ \mathsf{hasSubStuff}.(\mathsf{Eggyolk} \sqcap \exists \mathsf{hasState}.\mathsf{Liquid} \sqcap \exists \mathsf{srInheresIn}^{-}.\mathsf{ContinuousMedium}) \sqcap {=1}\ \mathsf{hasSubStuff}.(\mathsf{Oil} \sqcap \exists \mathsf{hasState}.\mathsf{Liquid} \sqcap \mathsf{srInheresIn}^{-}.\mathsf{DispersedPhase}) \sqcap \exists \mathsf{hasNrOfPhase}.\mathsf{int}_{=2}$$

Fig. 3. Template to easily ask for the required variables (A), and the definition it will then populate (B), shown for the example where we know Mayonnaise is an Emulsion.

also have the 'auxiliary' classes used in the Stuff Ontology, such as PhysicalObject linked to dolce:Physical object and bfo:object, and StuffRole is made a subclass of bfo:role and of dolce:Non-physical endurant. hasPart is made equivalent to dolce:part and inheresIn is made a subrelation of generic-dependent. There are no mappings for the object properties to BFO, because relations are only in the BFO-related Relation Ontology, not in BFO v.1.1.1. The resultant `stuff-dolcelite.owl` and `stuff-bfo11.owl` are consistent, and available at the aforementioned URL.

Stuff Ontology for domain ontologies: interoperability. Core ontologies should be useful for domain ontologies, notably to improve the quality, broaden interoperability, and to compute interesting derivations. Due to space limitations, we only provide use cases of ontology interoperability with a few classes (milk, mud, blood, and sugar) and a note on relations, so as to illustrate the underlying modelling issues and solution.

Consider again Milk, which is a colloid *irrespective of one's preferred context*, i.e., the 'nature' of that thing is being a type of stuff. Looking at some of the afore-mentioned ontologies, one notes that Cyc has it as a type of bodily secretion [8] and Galen[3] has it as a NAMEDBodySubstance, which therewith complicate adding soy milk, whereas SNOMED CT [29] has it as a type of Dairy foods and Envo [7] combines the two views by having it as subclass of both 'animal derived beverage' and 'milk product' (both are subclasses of 'food product'), whereas another environment ontology, tailored to microbes (MEO[4]), has it as a portion of secreted substance from the mammary gland, which is an organ, which is an animal-associated habitat for micro-organisms. Linking up the interpretations of milk across the various ontologies and that is least disruptive for the source ontologies, is that in all cases, the milk that is referred to is a stuff and, more precisely an emulsion. The Stuff Ontology can facilitate such interoperability and serve as a 'lingua franca' of the stuffs, either by having links from a stuff ontology extended with domain stuffs, containing, say, a domain-stuff:Milk $\sqsubseteq$ stuff:Emulsion, or where a domain ontology adds a subsumption link to emulsion, e.g., envo:'milk' $\sqsubseteq$ stuff:Emulsion. In a similar way, one can reconcile Mud, which is a stuff:Suspension of itself, whereas in the domain ontologies, it ihas as direct superclass any of the following: it is a meo:rock, sand and soil (which is a meo:geosphere), an envo:'environmental material', cyc:liquid ground, and a cyc:mixture. Likewise, Galen considers Blood as a subclass of Soft tissue, whereas Cyc and SNOMED CT have it as a body fluid (among other things), but, as a stuff, it is in any case a Sol (colloid with liquid continuous medium and solid dispersed phase). Sugar is somewhat trickier: MEO and Envo categorise it as food product (a role it plays), whereas Cyc and SNOMED have it as subclass of Carbohydrate and Organic compound, respectively. However, as a stuff, table sugar/sucrose Sugar is a StructuredPureStuff, and its granule, the molecule $C_{12}H_{22}O_{11}$, is the carbohydrate.

Besides aiding interoperability with respect to classes, it also does so for the relations (OWL object properties). cyc:granuleOfStuff, biotop:'has granular part' and [18]'s ingredients all match the notion of hasGranuleType, so one either can replace those relations or link each to the Stuff Ontology's hasGranuleType. Knowledge about relating component-stuffs is not widely represented in the ontologies, for

[3] `http://www.co-ode.org/ontologies/galen#Milk`; last accessed: 4-7-2014.
[4] `http://purl.jp/bio/11/meo/MEO_0000629`; last accessed: 4-7-2014.

it requires a careful analysis between portions, parts, and substuffs. hasSubStuff can already aid with some aspects; e.g., adding which beverages hasSubStuff some alcohol can be represented in SUMO, Cyc, or SNOMED CT, knowing that they then have the same representation of that knowledge and are interoperable on that aspect. Generally, though, the issues with portions, sub-quantities, contiguous parts, scattered parts, and summation principles are yet to be resolved fully (see, e.g., [12, 16, 21] for preliminary results), and we expect that the Stuff Ontology can assist in resolving that, for it already eliminates some philosophical debates on scattered portions (like sending oil molecules to Venus), as individual molecules are not portions of mixtures, and likewise it sets conditions on parts and portions especially regarding colloids and solid heterogeneous mixtures.

5 Conclusions

The gap between foundational and domain ontologies regarding stuff (matter/mass/substance) was filled by applying the idea of a 'bridging' core ontology: the Stuff Ontology. The ontology distinguishes between pure and mixed stuffs, and their sub-categories, such as solutions, colloids, and suspensions, and includes a few core relationships, such as a stuff's granule type and what stuffs a stuff is made of. The Stuff Ontology is highly axiomatised and in OWL DL for practical usability, it was successfully aligned to the DOLCE and BFO foundational ontologies. A decision diagram provides modeling guidelines applicable to the stuffs in domain ontologies, enhancing their quality and interoperability.

Current work concerns the interaction with portions and parts, and future work pertains to a use case on modelling food and bulk goods in food processing.

References

1. Alemán, J., et al.: Definitions of terms relating to the structure and processing of sols, gels, networks, and inorganic-organic hybrid materials. Pure and Applied Chemistry 79(10), 1801–1829 (2007)
2. Barnett, D.: Some stuffs are not sums of stuff. The Philosophical Review 113(1), 89–100 (2004)
3. Beisswanger, E., Schulz, S., Stenzhorn, H., Hahn, U.: BioTop: An upper domain ontology for the life sciences - a description of its current structure, contents, and interfaces to OBO ontologies. Applied Ontology 3(4), 205–212 (2008)
4. BFO: Basic formal ontology ((last accessed February 2012)), http://www.ifomis.org/bfo
5. Bittner, T., Donnelly, M.: A temporal mereology for distinguishing between integral objects and portions of stuff. In: Proceedings of AAAI'07. pp. 287–292 (2007), Vancouver, Canada
6. Brakel, J.v.: The chemistry of substances and the philosophy of mass terms. Synthese 69, 291–324 (1986)
7. Buttigieg, P.L., et al.: The environment ontology: contextualising biological and biomedical entities. Journal of Biomedical Semantics 4, 43 (2013)
8. Cyc: Opencyc. Online (version 2012-10-08 last accessed: 2-7-2014), http://www.cyc.com/platform/opencyc

9. Davis, E.: Ontologies and representations of matter. In: Proceedings of AAAI'10. AAAI Press (2010), Atlanta, Georgia, USA, July 11-15, 2010
10. Degtyarenko, K., et al.: ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res. 36, D344–D350 (2008)
11. Donnelly, K.A.M.: A short communication - meta data and semantics the industry interface: what does the food industry think are necessary elements for exchange? In: Proceedings of MTSR'10. CCIS, Springer (2010)
12. Donnelly, M., Bittner, T.: Summation relations and portions of stuff. Philosophical Studies 143, 167–185 (2009)
13. Francescomarino, C.D., Ghidini, C., Khan, M.T.: Grounding conceptual modelling templates on existing ontologies. In: Proceedings of KEOD'13. pp. 199–206. INSTICC, SCITEPRESS (2013), Vilamoura, Portugal, 19-22 September 2013
14. Gangemi, A., Presutti, V.: Ontology design patterns. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 221–243. Springer Verlag (2009)
15. Garcia, A., et al.: Developing ontologies within decentralized settings. In: Semantic e-Science. Annals of Information Systems 11, pp. 99–139. Springer (2010)
16. Guizzardi, G.: On the representation of quantities and their parts in conceptual modeling. In: Proceedings of FOIS'10. IOS Press (2010), Toronto, Canada
17. Herre, H.: General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In: Theory and Applications of Ontology: Computer Applications, chap. 14, pp. 297–345. Springer, Heidelberg (2010)
18. Höfling, B., Liebig, T., Rösner, D., Webel, L.: Towards an ontology for substances and related actions. In: Proceedings of EKAW'99. LNAI, vol. 1621, pp. 191–206. Springer (1999)
19. Jakubke, H.D., Jeschkeit, H.: Concise Encyclopedia Chemistry [translated from the German "ABC Chemie"]. Walter de Gruyter & Co (1993)
20. Keet, C.M.: A Formal Theory of Granularity. Phd thesis, KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy (April 2008)
21. Keet, C.M., Artale, A.: Representing and reasoning over a taxonomy of part-whole relations. Applied Ontology 3(1-2), 91–110 (2008)
22. Loggerhead Nesting Ontology: (last accessed: 27-1-2012), `http://bioportal.bioontology.org/ontologies/1024`
23. 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
24. Needham, P.: Compounds and mixtures. In: Handbook of the Philosophy of Science, Vol. 6: Philosophy of Chemistry, pp. 271–290. Elsevier (2011)
25. Needham, P.: Macroscopic mixtures. Journal of Philosophy 104, 26–52 (2007)
26. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of FOIS'01 (2001), Ogunquit, Maine, October 17-19, 2001
27. Scherp, A., Saathoff, C., Franz, T., Staab, S.: Designing core ontologies. Applied Ontology 6(3), 177–221 (2011)
28. Silberberg, M.S.: Chemistry: the molecular nature of matter and change. McGraw Hill Higher Education, 4th edn. (2006)
29. SNOMED CT: Online (version 31-1-2014 last accessed: 3-7-2014), `http://www.ihtsdo.org/snomed-ct/`
30. SUMO: (last accessed: 2-5-2012), `http://www.ontologyportal.org/`
31. Thakur, M., Donnelly, K.A.M.: Modeling traceability information in soybean value chains. Journal of Food Engineering 99, 98–105 (2010)
32. Zimmerman, D.W.: Coincident objects: Could a 'stuff ontology' help? Analysis 57(1), 19–27 (1997)