Formal Ontology in Information Systems
M. Donnelly and G. Guizzardi (Eds.)
IOS Press, 2012

doi:10.3233/978-1-61499-084-0-205

Integrating OntoClean's Notion of Unity and Identity with a Theory of Classes and Types

Towards a Method for Evaluating Ontologies

A. Patrice Seyed [1]
Department of Computer Science and Engineering
Center for Cognitive Science
University at Buffalo, NY, USA

Abstract.
This paper provides a reformulation of OntoClean's notion of Unity and Identity within a formal theory of classes, and evaluates how the reformulations apply to BFO's theory of types, which was previously given within the same formal theory. For Unity, a definition schema and explication together express the underlying dependency between a unifying relation and some proper subrelation of the 'part of' relation, which together define how a particular of a class is a whole. For Identity, the notion of an identity criterion is ontologically grounded and formalized as an identity procedure. For both Unity and Identity the formulations are expressed within a sorted first-order logic, where staying within first-order expressivity proved difficult in past work. With our reformulations in hand we evaluate the primary type dichotomy for material entities of BFO, *Object* and *ObjectAggregate*. Together with the work that integrates OntoClean's notion of Rigidity with BFO's theory of types, this work augments ongoing efforts to build software designed to evaluate and standardize OBO Foundry candidate ontologies, of which BFO is the upper level ontology.

Keywords. ontology, OntoClean, BFO

Introduction

This paper provides a reformulation of OntoClean's notion of Identity and Unity [1] within a formal theory of classes [2] and evaluates how the reformulations apply to BFO's theory of types [3]. Together with our integration of OntoClean's notion of Rigidity with BFO's theory of types [2], this work augments ongoing efforts to build software designed to evaluate and standardize OBO Foundry candidate ontologies (e.g, the BFO-Rigidity Wizard Plugin for Protégé 4 [4]), of which BFO is the upper level ontology.

The current work builds upon [2], where the categorical unit *property* and *type* are unified under that of *class*, and a reformulation of Rigidity is given in service of a defi-

[1] Corresponding Author: A. Patrice Seyed, Department of Computer Science and Engineering, University at Buffalo, 201 Bell Hall, Buffalo, NY, 14260, USA; Email: apseyed@buffalo.edu.

nition of *type*. In that and the current work BFO's theory of particulars is also assumed, where particulars are entities confined to specific spatial, spatiotemporal, or temporal regions (e.g., a specific grasshopper in front of me, its life, or the time interval that its life spans, respectively). Formally, a particular x instantiates A at a time t iff x is a member of the class A at t and A satisfies the formal conditions for being a type [2].

1. Reformulating OntoClean's Notion of Unity for our Formal Theory of Classes

OntoClean's notion of Unity is heavily influenced by [5]. By Guarino and Welty's (G&W's) account, Unity is a metaproperty that a modeler assigns to properties to help distinguish, for each instance of the property, it's parts from the rest of the world. If a property has Unity, this means there is some *unifying relation* that binds together certain parts of each instance of that property such that the parts compose the whole object. When a modeler tries to identify a unifying relation that applies to each object of a property, she is trying to answer: *What are the parts such that they form a whole*? [1] cite Simons when offering an explanation of what it means for an object to be a whole, which is as follows:[2]

> Every member of some division of the object stands in a *certain relation* to every other member, and no member bears this relation to anything other than members of the division. [5, p. 327]

Simons emphasizes that this certain relation holds only among parts of *a certain division*. These parts form a whole system.

[1] apply the theory of [5] for a *closed system* to their theory of Unity. An object x is closed under a relation r, or simply *r-closed*, iff, if y is a part of x, and if y is in the r relation to z, then z is a part of x.[3] Although not cited, Guarino and Welty also apply Simons' theory of a *connected system* [5] to their theory of Unity. An object x is connected under a relation r, or *r-connected* iff, if y and z are a part of x, then y and z are related by r. G&W apply these notions to what it is to be an *integrated whole* (although they do not give it formally for Unity):

> An object x is a (contingent) **r**-integrated whole if there exists some division of x such that it is a closed system. **r** will be called a base unifying relation for x [1].

(The use of 'contingent' here means "at some time" and is with respect to one snapshot in time only.) In more recent work [6] say that x is "whole under r" and the definition above regarding "some division" is omitted altogether. In both [1] and [6], they provide a definition for what it means for an object to be whole under a unifying relation, ω.

If x is *whole under* ω at t, then if y is a part of x at t and z is a part of x at t then y and z are in the ω relation at t; furthermore, if y is a part of x at t, and y and z are in the ω relation at t, then z is a part of x at t.

[2]Note that 'member of' is used here to provide in set-theoretic terms the relationship between a part of an object and the object, which is different from our standard usage as the relationship between a particular and a class.

[3]Relations introduced from other works that are not or not yet included in our formal system are introduced in italics and their respective axioms are introduced in plain English.

Therefore a unifying relation ω is reflexive, symmetric, and transitive, i.e., an equivalence relation.[4] The notion of "some division" is omitted in this definition, however it is crucial to the theory because it implies that ω does not hold between just any parts.

They define that x is an *intrinsic whole* under ω if for all times x exists it is *whole under t*. Applying this notion to properties, property ϕ is *unified under* ω iff for each instance of ϕ it is an *intrinsic whole* under ω. [7] proves that if ϕ is *unified under* ω, then the instances of ϕ are non-overlapping wholes, i.e., they do not partially overlap with other entities with the same property. G&W define three categories of properties based on these notions: *Unity*, *Non-Unity*, and *Anti-Unity*. A property has *Unity* if there is a relation it is unified under, a property has *Non-Unity* if there is no one relation it is unified under, and finally, a property has *Anti-Unity* if there is no instance of the property that is an intrinsic whole under some relation.

G&W's theory of Unity [1] includes a purported non-triviality stipulation, that it is not the case that there is a universal unifying relation such that every object is an intrinsic whole under it. [7] shows that this axiom does not accomplish its intent; it rules out a universal unifying relation, but also allows for an infinite number of other unifying relations that are trivially true.

We reformulate the notion that a property has Unity, i.e., that it has a unifying relation, under our theory of classes. We provide a definition schema for introducing it under the meta-predicate **Unified_Under**:

Definition Schema 1.

$$
\begin{aligned}
&\mathbf{Unified\_Under}(A,\omega,\mathbf{p}) =_{\text{def}}\\
&\quad \forall x(\exists t(\mathbf{member\_of}(x,A,t)) \rightarrow\\
&\qquad \forall t(\mathbf{exists\_at}(x,t) \rightarrow\\
&\qquad\quad \forall y(\mathbf{p}(y,x,t) \rightarrow\\
&\qquad\qquad \forall z(\mathbf{p}(z,x,t) \leftrightarrow\\
&\qquad\qquad\quad \omega(z,y,t))))) \wedge\\
&\quad \forall wvt(\mathbf{p}(w,v,t) \rightarrow \mathbf{part\_of}(w,v,t)) \wedge\\
&\quad \neg\forall wvt(\mathbf{part\_of}(w,v,t) \rightarrow \mathbf{p}(w,v,t))
\end{aligned}
$$

As with any schemata, the constants which are applied, i.e., the constants that take the place of A, ω, and **p** replace occurrences of A, ω, and **p** of the wffs in the definiens (i.e., right hand side) of the definition schema. As a definition schema, A serves as a meta-variable that represents any particular class and ω and **p** serve as meta-variables that represent any particular relations.

As mentioned, a unifying relation only holds among parts of a *certain division* and not arbitrary parts. To capture this, within our formulation we apply a proper subrelation **p** of the **part_of** relation of BFO/RO, instead of the generalized parthood relation applied by G&W, which corresponds what we have provided as **part_of**. Here **p** is some relation that is based on a restricted notion of parthood; however, it is not always clear how best to formalize this relation as it applies to a class A, therefore specifying **Unified_Under**(A,ω) can serve as a shortcut. Clearly however, there is a dependency between the unifying relation and the proper subrelation of **part_of** within the definition schema of **Unified_Under**.

Unifying relations have a transitive, symmetric, and reflexive nature:[5]

[4] Clearly these are properties of binary relation; as discussed previously, we assume that the relation is between two entities at some time represented in the third argument.

[5] Each proof is given in [4].

Metatheorem 1.

$$\begin{aligned}
&\mathbf{Unified_Under}(A,\omega,\mathbf{p}) \rightarrow \\
&\quad \forall x(\exists t(\mathbf{member_of}(x,A,t)) \rightarrow \\
&\quad\quad \forall t(\mathbf{exists_at}(x,t) \rightarrow \\
&\quad\quad\quad \forall y(\mathbf{p}(y,x,t) \rightarrow \\
&\quad\quad\quad\quad \forall zw(\omega(y,z,t) \wedge \omega(z,w,t) \rightarrow \omega(y,w,t))))) \wedge \\
&\quad\quad\quad\quad \forall zw(\omega(y,z,t) \rightarrow \omega(z,y,t))))) \wedge \\
&\quad\quad\quad\quad \omega(y,y,t))
\end{aligned}$$

Therefore, a unifying relation has the properties of an equivalence relation. Furthermore, as shown by [7] for Unity, for our reinterpretation w.r.t. classes, for each member of a class, if all and only its parts are unified by some relation, then the member does not partially overlap with other members of that class:

Metatheorem 2.

$$\begin{aligned}
&\mathbf{Unified_Under}(A,\omega,\mathbf{p}) \rightarrow \\
&\quad \forall xyt(\mathbf{member_of}(x,A,t) \wedge \mathbf{member_of}(y,A,t) \rightarrow \\
&\quad\quad \mathbf{exists_at}(x,t) \wedge \mathbf{exists_at}(y,t) \rightarrow \\
&\quad\quad\quad \exists z(\mathbf{p}(z,x,t) \wedge \mathbf{p}(z,y,t)) \rightarrow \\
&\quad\quad\quad\quad \forall w(\mathbf{p}(w,x,t) \leftrightarrow \mathbf{p}(w,y,t))))
\end{aligned}$$

We introduce here a relation that corresponds to RCC8's *connected with*, **connected_with**(x,y,t), in service of introducing a candidate unifying relation **connected_with**$_{\mathbf{tr}}(x,y,t)$, the transitive closure of **connected_with**. For an example, this relation holds between a hand and a torso at some time.

We consider whether *Ball* is unified under the **connected_with**$_{\mathbf{tr}}$ relation—that there are certain parts of a ball that are connected via a chain of connections, and only those parts are connected in this manner. We do not suggest that all parts of any ball hold in this relation; in fact, this is disallowed by the definition schema for **Unified_Under**, where **p** is a proper subrelation of **part_of**. This is also clearly reflected in the domain— for the class *Ball*, the internals of some balls may contain loose, disconnected pieces of the ball that are parts and not connected. Even though all parts of a solid ball are connected in this manner (i.e., in the **connected_with**$_{\mathbf{tr}}$ with every other part), this does not hold for all balls, therefore *Ball* is not unified under the **connected_with**$_{\mathbf{tr}}$ relation.

If we also consider a class *Human Skeletal System*, it is not simply unified under the relation **connected_with**$_{\mathbf{tr}}$, either. For example, the Achilles' tendon (calcaneal tendon) connects the plantaris, gastrocnemius (calf) and soleus muscles to the calcaneus (heel) bone, but these three muscles are not part of the human skeletal system.[6] A more specific relation or **part_of** relation is required, here to unify the class *Human Skeletal System*. In a human's skeletal system the bone parts form a path that is connected by joints. Therefore the more specific relation that unifies the certain parts is *connected by joints*. We could equally define a proper subrelation of **part_of** that is restricted to bones. Clearly then, there is a dependency between the subrelation of **part_of** and the unifying relation a class is unified under. We discuss potential unifying relations with respect to BFO's type dichotomy *Object* and *ObjectAggregate* in more detail in **Section 3 and 4**.

[6]We acknowledge there is a system called the 'musculoskeletal system' which includes muscles, as well as bones, cartilage, tendons, ligaments, joints and other connective tissues. For our example, we only discuss the skeletal system to outline connected entities that are not part of that system.

2. Reformulating OntoClean's Notion of Identity for our Formal Theory of Classes

Identity is a relation that every object has to itself and to nothing else. A criterion of identity as a way to determine when the identity relation holds, or informally, to recognize an object as the same again [8]. It is difficult to discuss the identity of a class of objects without presupposing what the objects *are* based on an assumed class definition, therefore identity criteria are better expressed as "identifying" criteria. [9] advises that this is permissible, because it is unavoidable in so many cases, that an identity criterion make reference to the class of objects the criterion of identity is being given for. However, it must not presuppose the criterion of identity for the class of objects whose identity criterion is being given [9]. For example, a criterion for the identity of events cannot be *having the same causes and effects* if causes and effects are themselves events [9]. Similarly a criterion of identity for sets should not be *having the same subsets*.

Identity criteria that are both necessary and sufficient include *occupying the same spatio-temporal region* for material entities or processes, and *having the same members* for sets. Outside these examples, identity criteria that are both necessary and sufficient are rare. To address this issue, G&W define necessary and sufficient criteria of identity separately. Another issue they address is identity with respect to time; identity can be defined with respect to one time (synchronic) or defined across times (diachronic). Therefore G&W provide time arguments that allow for either kind of identity criterion.

In many cases, analysis of identity can be limited to detecting the features that are just necessary for keeping the identity of a given entity, based on what can be described as essential properties. It is on these properties that G&W base necessary criteria of identity. According to G&W, a necessary criterion of identity θ of a property ϕ is defined such that for x and y that are instances of ϕ at t and t_1, respectively, and exist at t and t_1, respectively, if x and y are the same object, then they are the same under θ (i.e., $\theta(x,y,t,t_1)$ holds).[7] Where θ stands for 'having the same genotype' $\theta(x,y,t,t_1)$ is read x at t and y at t_1 have the same genotype. For G&W, 'same under' captures the intuition that, based on the identity criterion θ, there is some characteristic feature that is unique to the entity to which the criterion is applied [6, p. 5].

In considering what an identity criterion is ontologically about, for such a criterion to be applied, there must be some procedure, some instance of BFO's *Process* type, during which the identities of x and y are evaluated. More formally, **confirms**(P,x,y,t,t_1) means that, for x at t and y at t_1, procedure type P confirms x and y are the same thing. We discuss why the predicate **confirms** applies to a type, P, rather than a particular instance of P, shortly. We define this proposed notion under our theory of classes, and define a predicate **Necessary-IP**(A,P), which means that a class A has a *necessary identity procedur*e P:

Definition 1. $$\textbf{Necessary-IP}(A,P) =_{\text{def}} \forall xytt_1((\textbf{member\_of}(x,A,t) \wedge \textbf{exists\_at}(x,t) \wedge \textbf{member\_of}(y,A,t_1) \wedge \textbf{exists\_at}(y,t_1)) \rightarrow (x{=}y \rightarrow \textbf{confirms}(P,x,y,t,t_1)))$$

The nature of **confirms**(P,x,y,t,t_1) is such that it is not necessarily the case that there is a specific instance of P that has confirmed that x and y are the same; more accurately,

[7] The sense of 'instance' and 'property' is based on G&W. The property labels 'Sugar' and 'Hydrophillic' are shortcuts for the more descriptive labels 'being sugar' and 'being hydrophillic'. 'instance' is used differently in BFO and 'property' is used differently in philosophy at large.

the predicate relies on past instances of P. If **confirms**(P,x,y,t,t_1) then there is at least one particular whose identity has been confirmed in the past by an instance of P.[8] More formally, if **Necessary-IP**(A,P) then there is some w that is a member of A and exists at a time t_2, some v that is a member of A and exists at a time t_3 and **confirmed**(P,p,w,v,t_3,t_4), which means that w and v were confirmed as the same by a procedure instance p of P:

Axiom 1. (**Necessary-IP**(A,P) $\rightarrow$
$\exists p w v t_2 t_3$(**member_of**(w,A,t_2) $\wedge$ **exists_at**(w,t_2) $\wedge$
member_of(v,A,t_3) $\wedge$ **exists_at**(v,t_3)) $\wedge$
confirmed(P,p,w,v,t_2,t_3))

Axiom 2. **confirmed**(P,p,w,v,t_2,t_3) $\rightarrow$ $\exists t$(**instance_of**(p,P,t))

Since when **Necessary-IP**(A,P) you have **confirmed**(P,p,w,v,t_2,t_3), for any identity procedure type P such that **confirms**(P,x,y,t,t_1) it is linked to some instance p of P that has been applied. For an instance p of the procedure class P, there is some part of the procedure where a result of w is derived, and some part of the procedure where a result of v is derived, and finally, there is an end part of the procedure where these results are compared to determine whether or not w and v are the same thing. The procedure p need not occupy contiguous spatio-temporal regions.

We provide a formalization of identity procedures that more concisely represent the aforementioned procedure parts and their results. This is best explained with the additional predicates **result_of_procedure** and **matches**. The former predicate, **result_of_procedure**(p_1,w,t_2), is a function that maps to some result of procedure p_1 that applies to the entity w at t_2 (but need not span t_2). For example, p_1 may be a process that has as a result the fingerprint pattern of a person. To evaluate identity for an entity, another result must be acquired; hence, a second procedure, p_2, is applied to an entity v at t_3, (**result_of_procedure**(p_2,v,t_3)). Therefore **matches**(**result_of_procedure**(p_1,w,t_2), **result_of_procedure**(p_2,v,t_3)) means the result of the first procedure "matches" the result of the second procedure (e.g., two fingerprint patterns match). What 'matches' means here depends entirely upon the identity procedure type, and it also depends upon the identity of other things, since, as we discuss shortly, there is a recursive nature to identity procedures. Given these formulations we have the following axiom:

Axiom 3. **confirmed**(P,p,w,v,t_2,t_3) $\rightarrow$
$\exists p_1 p_2$(**matches**(**result_of_procedure**(p_1,w,t_2), **result_of_procedure**(p_2,v,t_3)) $\wedge$
part_of(p_1,p) $\wedge$ **part_of**(p_2,p))

The practical use of **matches** is that when applied to two results, if p_1 and p_2 are parts of an instance of P that is a necessary identity procedure, if false, then x and y are not identical. That said, because **confirms**($P,x,y,t,t1$) captures the notion of an identity procedure categorization applicable to every member of a class, we take the **confirms** predicate to be our primary notion for formalizing the relation between identity procedure types and classes of particulars to which the identity procedures apply.

With respect to our example, one such procedure is *DNA_Profiling*. When **Necessary-IP**(*Person*,*DNA_Profiling*) holds, if x exists and is a member of *Person* at t and y exists

[8] We also observe that for P to be a legitimate identity procedure, there are many occurrences, i.e., members of P where the identity of an entity has been confirmed in the past.

and is a member of *Person* at t_1, if x and y are identical, then **confirms**(*DNA_Profiling*,x,y,t,t_1) holds. Therefore if **Necessary-IP**(*Person*,*DNA_Profiling*) holds, by our formulation of **confirmed** (**Axiom 2**), it is true that an instance of *DNA_Profiling* has in the past served to confirm necessary identity for an instance of the class *Person*.

DNA profiling requires the object being evaluated have a genotype, and since this is what we consider an essential property for people, it is accurate to presume that the existence of a person at some time entails the existence of their genotype at the same time. Note that even though *DNA profiling* is a procedure type that confirms the necessary identity of people, it is clearly not sufficient for confirming identity, due to the existence of genetically identical twins.

The notion of a necessary identity procedure is perhaps more intuitive to think of in terms of the contrapositive of the nested implication, $x{=}y \rightarrow$ **confirms**(P,x,y,t,t_1) of **Definition 1**. If two objects are not confirmed as identical by procedure P, they do not have the same essential properties, therefore they certainly cannot be identical. In the context of a modeler thinking about what a necessary identity procedure of a class is, it is helpful, in order to identify essential properties, for her to answer the question: *What feature must change or no longer exist for a member of the class, at some time t, to no longer be the same thing at a time after t?*

According to G&W a sufficient identity criterion θ of a property ϕ is defined such that for x and y that are instances of ϕ at t and t_1, respectively, and exist at t and t_1, respectively, if $\theta(x,y,t,t_1)$ holds, then x and y are identical. We reconsider criteria again, for sufficient identity, and put forth a notion of sufficient identity procedures under our theory of classes. **Sufficient-IP**(A,P) means that a class A has a *sufficient identity procedure* P:

Definition 2. $\textbf{Sufficient-IP}(A,P) =_{\text{def}} \forall xytt_1((\textbf{member\_of}(x,A,t) \wedge \textbf{exists\_at}(x,t) \wedge \textbf{member\_of}(y,A,t_1) \wedge \textbf{exists\_at}(y,t_1)) \rightarrow (\textbf{confirms}(P,x,y,t,t_1) \rightarrow x{=}y))$

Take for example, a sufficient identity procedure *FingerprintMatching* for the class *Person*.[9] When **Sufficient-IP**(*Person*,*FingerprintMatching*) holds, if x exists and is a member of *Person* at t and y exists and is a member of *Person* at t_1, if **confirms**(*FingerprintMatching*,x,y,t,t_1) holds, that is, if an instance of the type *FingerprintingMatching* confirms x at t and y at t_1 are the same person, then x and y are identical.

The procedure type *FingerprintingMatching* is defined under the following natural language parse: 'a procedure in which fingerprint patterns *that exist* are analyzed, the results of which are comparable to confirm identity'. The procedure requires a fingerprint pattern that represents an actual fingerprint's pattern. Because fingerprints can be removed, it is not possible to compute and compare fingerprint patterns between arbitrary people *at any time*; therefore, the assumption that the fingerprint pattern being evaluated during the procedure in question exists is needed as a basis for the class definition of *FingerprintMatching*, in order for it to be a legitimate sufficient identity procedure.

This clarification of sufficient identity procedures brings attention to an important point about what we consider necessary and sufficient procedures for identity. In each case, the procedure involves the identity of functions which map from the objects of the class in question (**Axiom 3**). [14, p. 20] noted that identity criteria often make use of the

[9] The exact precision for unique identification by fingerprinting is debated.

notion of identity itself, and can only do so informatively by alluding to the identity of things of another class. With respect to identity procedures, for the necessary identity procedure class *DNA profiling* of the class *Person*, it is dependent on the identity of genotypes, which must account for genetic variations over time that are due to mutations. By this token, the corresponding identity criterion is non-primitive and can be reduced to identity of functions mapped from individual people to their genotype, i.e., 'genotype of'.

Applying G&W's formulation, if $\theta(x,y,t,t_1)$ holds where θ is 'having the same genotype', this implies that the genotype of x at t is identical to the genotype of y at t_1. If a person x at t and a person y at t_1 are identical under the sufficient identity criterion *having the same fingerprint pattern*, there is some fingerprint pattern of x at t and fingerprint pattern of y at t_1 which are identical. Nevertheless, by defining identity procedures for necessary and sufficient identity, instead of criteria, these issues are dealt with more simply, and by a designated identity procedure type that is a subtype of *Process*.

We designate necessary and sufficient identity procedures to be two kinds of identity procedures (**IP**), and define a procedure that is both necessary and sufficient for identity (**N&S-IP**). The necessary and sufficient identity procedure for the duration of time of a process is *TimeMeasurementProcedure*. Here, the measurements, i.e., the values that results from measurement of time, of x and of y, are identical according to some specific scale.

[10] discuss how Non-Rigid properties seem to only "carry" (i.e., inherit) identity criteria, for example the property *being a student* inherits its identity from *being a person* which "supplies it". An identity criterion proposed to be "supplied" (i.e., not inherited) by *being a student*, for example *having the same registration number*, is only held within certain durations of the student's existence. Given this limitation, Guarino and Welty decide that identity criteria that are not held by Rigid properties are not of interest to their theory. Therefore they exclude these "local" identity criteria, like *having the same registration number*. We provide this informal part of their theory formally and with respect to classes and identity procedures:

Axiom 4. $\mathbf{IP}(A,P) \rightarrow \exists B(\mathbf{Rigid}(B) \wedge \mathbf{IP}(B,P) \wedge \mathbf{subclass\_of}(A,B))$

It follows trivially that, for an identity procedure type P of a Non-Rigid class A, there is some Rigid class B with that identity procedure type that is a superclass of A.

G&W define that a property ϕ "supplies" an identity criterion iff ϕ is Rigid, has the identity criterion, and does not have a parent with that identity criterion. We consider their definition for a notion of supplying an identity procedure, and provide it in terms of classes. If A supplies an identity procedure P, that means all other classes with identity procedure P are subclasses:

Definition 3. $\mathbf{supplies\text{-}IP}(A,P) =_{\mathrm{def}} \mathbf{IP}(A,P) \wedge (\forall B(\mathbf{IP}(B,P) \rightarrow \mathbf{subclass\_of}(B,A))$

It trivially follows that if a class A supplies an identity procedure P, and is a subclass of a class B that has that identity procedure, then A and B are identical.[10] By **Axiom 4**, **Definition 3**, the axiom on the identity of classes, it also follows that if a class supplies an identity procedure, it is Rigid:

[10] This assumes an axiom on the identity of classes; as given in [4]: A and B are identical iff a subclass of one another.

Theorem 1. $\exists P(\textbf{supplies-IP}(A,P)) \rightarrow \textbf{Rigid}(A)$

Furthermore, there is some class that supplies every identity procedure:

Axiom 5. $\exists A(\textbf{IP}(A,P)) \rightarrow \exists B(\textbf{supplies-IP}(B,P))$

From this and the definition of **supplies-IP** it follows that if a class has an identity procedure that it does not supply, there must be some superclass that supplies it:

Theorem 2. $\textbf{IP}(A,P) \wedge \neg\textbf{supplies-IP}(A,P) \rightarrow \exists B(\mathrm{A}\neq B \wedge \textbf{subclass\_of}(A,B) \wedge \textbf{supplies-IP}(B,P))$

For example, *Primate* supplies the identity procedure *FingerprintMatching*, which is inherited by classes *Human* and *Gorilla*. In OntoClean, if a Non-Rigid property has an identity criterion θ then it is subsumed by a Rigid property that supplies it. We also provide this in terms of classes and identity procedures, which follows trivially from **Axiom 5** and the definition of **supplies-IP**:

Theorem 3. $(\textbf{Non-Rigid}(A) \wedge \textbf{IP}(A,P)) \rightarrow \exists B(A\neq B \wedge \textbf{subclass\_of}(A,B) \wedge \textbf{supplies-IP}(B,P))$

It follows that classes inherit necessary and sufficient identity procedures:

Theorem 4. $(\textbf{Necessary-IP}(A,P) \wedge \textbf{subclass\_of}(B,A)) \rightarrow \textbf{Necessary-IP}(B,P)$

Theorem 5. $(\textbf{Sufficient-IP}(A,P) \wedge \textbf{subclass\_of}(B,A)) \rightarrow \textbf{Sufficient-IP}(B,P)$

Because every type is a class, identity procedures are also inherited by types.

G&W discuss the notion that if two identity criteria are incompatible, then a property cannot have both. Examples are given but the notion is not formalized. We define **Incompatible-IP**(P,Q) to mean no class has identity procedures P and Q. If P and Q are incompatible (exclusively necessary or sufficient identity procedures) of classes A and B, respectively, and A and B are types, then they are disjoint types.

The utility of the **Incompatible-IC**(P,Q) for modeling is that if a class is purported to have both P and Q identity procedures, then there is a mistake in classification or a mistake in assignments of identity procedures to classes. It is a decision a modeler must make to resolve the inconsistency.

3. Unity and Identity of BFO's Object

BFO 1.1 defines *Object* as a type where every instance is "a material entity that is spatially extended, maximally self-connected and self-contained, and possesses an internal unity. The identity of object entities is independent of that of other entities and can be maintained through time" [3, p. 48]. Examples include an organism, a heart, a chair, a lung, and an apple. In the BFO sense, objects are not merely the sum of their parts, and thus can survive the gain and loss of *some* parts.

Also, instances of *Object* are described as having "internal unity", which we suggest is a relation closed under boundary parts. *Object* is not unified under **connected_with$_{\mathbf{tr}}$**, for many reasons, including that it does not transitively hold for just the parts of each particular that is an instance of *Object*. For example, there are fused particulars, such as conjoined twins, where, even though **connected_with$_{\mathbf{tr}}$** holds between any two parts of the fused totality, still the fused totality is not considered to be a single unified particular, but a fusion of two particulars. Here the intuition of Kaplan's proof is easily applied to reject **connected_with$_{\mathbf{tr}}$**—by also lacking a clearly defined proper subrelation of **part_of** to delineate certain parts—as a unifying relation for *Object*, due to overlapping parts. Like unifying relations, practically useful identity procedures for BFO's *Object* types are those found for subtypes at the more specific domain ontology level.

4. Unity and Identity of BFO's Object Aggregate

BFO 1.1 defines *Object Aggregate* as a type where every instance is "a material entity that is a mereological sum of separate object entities and possesses non-connected boundaries" [3, p. 48]. Examples include a heap of stones, a group of commuters on the subway, a collection of random bacteria, a flock of geese, and the patients in a hospital. Every object aggregate is composed of at least two distinct objects:

Axiom 6.
$$\begin{aligned}&\mathbf{instance\_of}(x,\mathit{ObjectAggregate},t) \rightarrow\\ &\quad \exists yz(\mathbf{part\_of}(y,x,t) \wedge \mathbf{part\_of}(z,x,t) \wedge\\ &\qquad \mathbf{instance\_of}(y,\mathit{Object},t) \wedge\\ &\qquad \mathbf{instance\_of}(z,\mathit{Object},t) \wedge y \neq z)\end{aligned}$$

For *ObjectAggregate*, there is no one unifying relation that it is unified under (which under G&W's theory satisfies non-Unity) because there is some instance that shares parts with another instance. For example, Barack Obama is a part of the aggregate of people present for the State of the Union Address he gave on January 25, 2011, and at the same time a part of the aggregate of people intending to run for U.S. President in the 2012 election.

Equally, there is no one identity procedure that applies to *ObjectAggregate*, but we can distinguish primarily two types of aggregates based on whether or not a specific identity procedures holds. Specifically, there are aggregates (1) whose identity is strictly based on certain parts that sum the whole, and (2) whose identity is not identified in this manner. We refer to these types of aggregates using corresponding subscripts to maintain emphasis on how we define them here.

Each instance of *ObjectAggregate$_1$* is a mere grouping of spatially separated particulars, where these particulars serve as the parts, although not the only parts, of the aggregate. These specific parts compose the sum of the aggregate, and this kind of aggregate cannot survive the gain and loss of these parts. For example *utensils in your kitchen* is precisely every utensil in your kitchen, at some time *t*. If one utensil is chipped at t_1 it is the same as the aggregate at *t*, but if the same utensil is destroyed at t_2, then it is no longer the same as the aggregate at *t*.

Does this mean that an instance of ObjectAggregate$_1$ can survive some changes in its parts? The answer to this question depends on the part relationship under question. The member/aggregate relation is defined as that between an object and an object aggregate;

it is intransitive, irreflexive, and asymmetric. A fiat part of a stone (i.e., a part that does not have distinct boundaries)[11] in an aggregate of stones is not a member of the aggregate of stones. In some literature, including [12], the member/aggregate relation is defined separate from the 'part of' relation, but in other works it is a subrelation of 'part of'.

This sort of contextually defined 'part of' relation is not given for BFO/RO, and the **part_of** relation is used for the composition of both objects and object aggregates. The motivation behind BFO's position of strictly using **part_of** is for maintaining transitivity of parthood across levels of granularity. BFO/RO's **part_of** relation is true to classical mereology, therefore **part_of** is always transitive. Given the transitivity of **part_of**, a fiat part of a stone in an aggregate of stones is a part of the aggregate.

It is worth noting that some argue that the relationship between a stone and an aggregate of stones is not 'part of' at all. Under this view the relationship is more specialized and must address the nature of the whole, as covered by the member/aggregate relation. There is also a tendency to associate the 'part of' relation with physical connectedness, and under such an account the relation does not apply to the composition of aggregates. For those who argue this position, the reading of "a piece of a stone is part of a stone aggregate" goes against common-sense knowledge representation, and this much is true.

We propose a relation for BFO[12], **part_of_aggregate**, to represent the member/aggregate relation, where **part_of_aggregate** is a proper subrelation of **part_of**:

Axiom 7. $\textbf{part\_of\_aggregate}(x,y,t) \rightarrow \textbf{part\_of}(x,y,t)$

Axiom 8. $\neg\forall xyt(\textbf{part\_of}(x,y,t) \rightarrow \textbf{part\_of\_aggregate}(x,y,t))$

Under this formulation if a stone is a member of an aggregate, the stone is also more basically a part of the aggregate; therefore, the transitivity of **part_of** is still maintained across grains. For example, if you observe what is an object aggregate at the microscopic level, you might "zoom out" to what is an object at the human eye level. From either level of granularity what composes the particular is a part of it. Introduction of this grain-specific 'part of' relation, maintains this transitivity, via its super-relation, **part_of**.

Given this newly introduced relation, we can now formalize the (necessary and sufficient) identity procedure of a class based on its extensionality (*ME*): a member is the same over time iff it is inspected to have the same "member parts":

Axiom 9.
$$\begin{aligned}&\textbf{N\&S-IP}(A, ME) \leftrightarrow \\ &\quad \forall xyt((\textbf{member\_of}(x,A,t) \wedge \\ &\qquad \textbf{member\_of}(y,A,t)) \rightarrow \\ &\qquad\quad (x{=}y \leftrightarrow \forall z(\textbf{part\_of\_aggregate}(z,x,t) \leftrightarrow \\ &\qquad\qquad \textbf{part\_of\_aggregate}(z,y,t))))\end{aligned}$$

ME is a necessary and sufficient identity procedure in which the member parts are inspected to be the same. By this axiom and our description of the types $ObjectAggregate_1$ and $ObjectAggregate_2$, it is the case that $\textbf{N\&S-IP}(A, ME) \rightarrow A{=}ObjectAggregate_1 \wedge A{\neq}ObjectAggregate_2$. To answer the question posed, an instance *a* of $ObjectAggregate_1$ can survive changes in some parts, but not changes of parts in the **part_of_aggregate**

[11] A fiat part object part is part of a object but is not demarcated by any physical discontinuities, e.g., upper and lower lobes of the left lung [3].

[12] In this context reference to BFO assumes inclusion of the Relation Ontology (RO).

relation with *a*. Instances of *ObjectAggregate*$_2$ are such that they can survive both kinds of changes, therefore *ME* is not a necessary or sufficient identity procedure for this type. Defining a class as having the identity procedure type of *ME* imposes that all subclasses of it have this identity procedure; otherwise it is a modeling mistake. There are additional interesting dichotomies subtyped beneath *ObjectAggregate*$_2$, with respect to the roles the member parts play, that we leave for future work.

Next we consider unifying relations for instances of *ObjectAggregate*. Take for example the relation *has the same parents as* proposed as a relation the class *Aggregate of Siblings* is unified under [13, p. 6]. [13] There are many other similarly defined unifying relations for aggregates, for example, the unifying relation *has the same immediate boss as* and *being located in the same designated spatio-temporal region as*, the former holding for member parts of an aggregate of certain co-workers, and the latter holding for member parts of an aggregate of audience members of some event.

There are other purported unifying relations of classes of aggregates which suffer from a problem that they only hold at the particular level and do not hold where the **Unified_Under** relation applies, at the class level. For example, if the organization *PETA* is an intrinsic whole under the relation *pays dues to PETA*, there is not a more generalized relation like *pays due to an organization* that applies to all social organizations, because clearly, there is some person-part of one organization that is a person-part of a different organization at the same time (therefore it does not transitively hold). The same can be said, for an aggregate *a* and a purported unifying relation *being an aggregate part of a*. In both cases these relations are not a unifying relation at the class level. Furthermore, the latter relation and others of its kind are also self-referring to the aggregate in question, and are therefore trivial.

Ultimately then, as reinforced by our examples, the utility of our definition schema for **Unified_Under** is that we apply it for a specific class *A* and purported unifying relation ω, which may or may not hold as a unifying relation for purported subclasses of *A*. If it does not hold, then the modeler must identify that either the purported subclasses are not subclasses of *A*, or the ω is not a relation that *A* is unified under.

Unified_Under does not cover the stronger notion of Non-Unity introduced by G&W (which can be given informally), however it does have it has immediate utility in that it covers the notion of not having a specific kind of Unity, i.e., relative to an identified unifying relation. Anti-Unity primarily holds for classes whose members are considered amounts of matter, which we address in the following section.

5. Amounts of Matter

An amount of matter is the kind of entity that usually falls under terms that, in everyday language, take singular verbs, cannot occur with numerals (unless elliptical for some measurement), and takes determiners like 'some', 'little', and 'much', as opposed to 'every', 'few', and 'many'. These terms include 'gold', 'sugar', and 'water'. As mentioned, these terms refer to classes that satisfy G&W definition of Anti-Unity.

BFO holds the position that two distinct entities cannot occupy the same space at the same time; therefore, what is referred to as a piece of tofu and the tofu "stuff" it is

[13] Note that the description 'has the same parents as' is intended to be interpreted as "born of the same parents as" as opposed to "at some time, has the same parents as".

made of must be identical [15, p. 12]. However the tofu stuff is assumed to have different existence conditions, since, for example, if it is split into one hundred pieces it still exists while the portion of tofu no longer exists. Hence the notion of amounts of matter, in light of BFO's theory, leads to contradictions, in this case that some x exists at a time t and does not exist at t.

Further, the purported class *AmountOfMatter* is inherently cross-granular in nature, having features of both *Object* and *ObjectAggregate*, which causes problems for BFO. More specifically, the tofu stuff is a whole object, given that all the parts are physically connected by a chain of connections, while at the same time it is an aggregate, and one in which its parts need not be connected to be a part. For BFO, an ontology represents a certain, specific level of granularity [3], and by this approach, the types *Object* and *Object Aggregate* are disjoint. Clearly these assumptions, together with the conclusion that the portion of tofu and the tofu stuff are identical, result in an inconsistent ontology. Ultimately, then, something that demonstrates the existence conditions of an amount of matter is not a particular in BFO's domain.

6. Conclusions

In our reformulation of OntoClean's notion of Unity, we set forth a definition schema and explication which reflects that a unifying relation depends on a specific 'part of' relation, where the dependency is mutual, which together help define how a particular is a whole. In our reformulation of OntoClean's notion of Identity, we ontologically ground the notion of an identity criterion in what we consider an identifying procedure process, which is an instance of BFO's *Occurrent* type. In both cases we were able to express the reformulations within a sorted first-order formal system, which proved difficult in previous OntoClean work. Further, we formally provided, within our formal theory of classes and types, notions such as supplying an identity procedure, and incompatibility among identity procedures, which in previous literature were only given informally.

The notion of necessary identity procedures can help a modeler identify precisely what features of the members of Rigid classes make them members of those classes, which can be applied to compare any two members in a procedure that determines that they are distinct or the same. Therefore, necessary identity procedures provide a facility that supplements our formal theory of Rigidity, ergo types, and vice versa. The notion of a sufficient identity procedure is more clearly epistemic in nature and less relevant to an upper ontology with the ontological position such as BFO. Fingerprints can be removed and social security numbers can be changed, all while the particulars they correspond to can continue to exist. Nevertheless, sufficient identity procedures can be useful for modeling, as they are inherited, and for ontologies that are linked to databases they can be useful to the designation of primary keys.

Given these reformulations within a formal system that assumes our theory of classes [2], we consider how they apply to the main dichotomy of material entities in BFO's theory of types, *Object* and *ObjectAggregate*. For the former, we concluded that proposed unifying relations and identity procedures have the most utility at the domain ontology level. For the latter, we isolated a division between aggregates defined on member parts and those that are not. In the process we proposed a new relation for BFO, **part_of_aggregate**, which is applied to define a necessary and sufficient identity procedure for those aggregates defined on member parts.

In our analysis we also concluded that the particulars conceived as amounts of matter, which cross-cut the *Object* and *ObjectAggregate* types, do not fall within BFO's domain of particulars. An ontologist that is deciding upon an upper ontology must consider how this aspect of BFO's theory of types might affect the models of their domain.

We expect that our reformulation of the Unity and Identity components of OntoClean provide formal and intuitive value to the existing literature on OntoClean. In future work, we will incorporate the resulting axioms and definitions to our existing decision tree algorithm and a Protégé 4 Plugin [4], which is currently based on the integration of the Rigidity component of OntoClean and BFO's theory of types. Ultimately, this work will help novice and expert ontologists build ontologies for the OBO Foundry that are compliant with BFO.

References

[1] N. Guarino and C. Welty, Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis. Proceedings of ECAI-2000, IOS Press, Amsterdam, (2000), 219–223.

[2] A.P. Seyed and S. C. Shapiro, Applying Rigidity to Standardizing OBO Foundry Candidate Ontologies, Proceedings of the International Conference on Biomedical Ontologies (ICBO) (2011), 175–181.

[3] A. Spear, Ontology for the Twenty First Century: An Introduction with Recommendations. Technical report, University at Buffalo, (2007).

[4] A. P. Seyed, A Method for Evaluating and Standardizing Ontologies. Ph.D. Dissertation. Department of Computer Science and Engineering, University at Buffalo, (2012).

[5] P. Simons, Parts: A Study in Ontology. Clarendon Press, Oxford, (1987).

[6] N. Guarino and C. Welty, Identity and Subsumption. The Semantics of Relationships: an Interdisciplinary Perspective. Editors: Green, R. and Bean, C. A. and Hyon, Kluer, Dordrecht, (2001), 111–126.

[7] A. N. Kaplan, Towards a Consistent Logical Framework for Ontological Analysis. FOIS. Editors: Chris Welty and Barry Smith. (2001), 244–255.

[8] G. Frege, The Foundations of Arithmetic. Blackwell: Oxford, (1950).

[9] E. J. Lowe, What is a Criterion of Identity? The Philosophical Quarterly. 39, 154, 1–21, (1989).

[10] N. Guarino and C. Welty, A Formal Ontology of Properties. Proceedings of EKAW-2000. Editors: R. Dieng and O. Corby, Berlin, (2000).

[11] C. Welty and W. Andersen, Towards OntoClean 2.0: A framework for Rigidity. *Applied Ontology*, 1(1), IOS Press, Amsterdam (2005), 107–116.

[12] G. Guizzardi, Ontological Foundations for Structural Conceptual Models. Centre for Telematics and Information Technology, (2005).

[13] A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari, Understanding Top-Level Ontological Distinctions. Proceedings of IJCAI-01 Workshop on Ontologies and Information Sharing, (2001), 26–33.

[14] E. J. Lowe, Kinds of Being: A Study of Individuation, Identity, and the Logic of Sortal Terms. Blackwell Publishers, Oxford, 1989.

[15] P. Grenon. BFO in a Nutshell: A Bi-categorical Axiomatization of BFO and Comparison with OntoClean, Technical Report, (2003).

[16] B. Smith, The Logic of Biological Classification and the Foundations of Biomedical Ontology. Handbook on Ontologies: Invited Papers from the 10th International Conference on Logic, Methodology and Philosophy of Science. Elsevier-North-Holland, (2003).

[17] W. Andersen and C. Menzel, Modal Rigidity in the Ontoclean Methodology. *Formal Ontology in Information Systems*. Editors: A. C. Varzi and L. Vieu, IOS Press, Amsterdam, (2004), 119–127.

[18] M. Carrara. Identity and Modality in OntoClean. *Applied Ontology* 1(1), IOS Press, Amsterdam, (2004), 128–139.

[19] B. Smith and W. Ceusters, Ontological realism: A methodology for coordinated evolution of scientific ontologies. *Applied Ontology* 5(3-4), (2010), 139-188.

[20] N. Guarino and C. A. Welty, An Overview of OntoClean, *Handbook on Ontologies*, (Eds.) S. Staab and R. Studer, Springer Verlag, Berlin, (2004), 151–159.