How to *Reuse* a Faceted Classification and Put it on the *Semantic* Web

Bene Rodriguez-Castro, Hugh Glaser, and Les Carr

School of Electronics and Computer Science, University of Southampton,
Southampton SO17 1BJ, UK
{b.rodriguez,hg,lac}@ecs.soton.ac.uk
http://www.ecs.soton.ac.uk

Abstract. There are ontology domain concepts that can be represented according to multiple alternative classification criteria. Current ontology modeling guidelines do not explicitly consider this aspect in the representation of such concepts. To assist with this issue, we examined a domain-specific simplified model for facet analysis used in Library Science. This model produces a Faceted Classification Scheme (FCS) which accounts for the multiple alternative classification criteria of the domain concept under scrutiny. A comparative analysis between a FCS and the Normalisation Ontology Design Pattern (ODP) indicates the existence of key similarities between the elements in the generic structure of both knowledge representation models. As a result, a mapping is identified that allows to transform a FCS into an OWL DL ontology applying the Normalisation ODP. Our contribution is illustrated with an existing FCS example in the domain of "Dishwashing Detergent" that benefits from the outcome of this study.

Keywords: facet analysis, faceted classification, normalisation, ontology design pattern, ontology modeling

1 Introduction

Ontologies remain as one of the key components needed for the realization of the Semantic Web vision. They bring with them a broad range of development activities that can be grouped into what it is referred to as Ontology Engineering. Ontology Engineering for the Semantic Web is a very active research area and has experienced remarkable advancements in recent years, although it is still relatively new compared to other engineering practices within Computer Science or other fields. A constant ongoing effort in Ontology Engineering deals with harnessing the field with sound development methodologies analogous to those successfully employed in Software Engineering for decades. One of the objectives of these methodologies is to address areas of the ontology development process vulnerable to ad-hoc practices that could potentially lead to unexpected or undesirable results in ontology artifacts.

This paper describes a specific, very recurrent modeling scenario in ontology development, subject to such vulnerability. The scenario consists of domain-specific concepts that can be represented according to multiple alternative classification criteria. To the best of our knowledge, guidelines for the conceptualization and representation of domain-specific concepts prone to be described based on multiple (potentially alternative) classification criteria, has not been explicitly considered in the context of ontology modeling for the Semantic Web.

General examples of domain-specific concepts that exhibit the characteristics described abound, going from a “bibliographic reference”, (which could be classified according to several criteria such as “subject”, “author”, “publication venue”, etc.); to a “toy” (which could be classified based on “suitable age”, “brand”, “subject type”, etc.). The list of examples can go on. We have seen in our own experience that lack of specific design guidelines leaves ample room for conceptual errors when trying to develop a simple domain-specific ontology model for such concepts. For example, common mistakes when trying to represent these concepts and their classification criteria are to use subsumption relations between classes when in fact a *part-of* relation would be in order, or to use subsumption to model relationships that are outside OWL DL expressivity altogether.

Other examples of domain-specific concepts that can fit into the modeling scenario described are particularly interesting because they are used in well-known ontology development literature using OWL. They include: “Wine” [1], “Person” (in the context of family history relations) [2], or “Pizza” [3]. However, in none of them, they refer explicitly to the various classification criteria of the domain concept that are considered implicitly, nor attempt to represent these criteria explicitly in the respective ontology models developed.

To assist with these issues, we aim to put forward an initial set of basic design guidelines to mitigate the opportunity for ad-hoc modeling decisions in the development of ontologies for the problem scenario described. To obtain the conceptual model of a domain-specific concept and its multiple classification criteria we examined a simplified model for facet analysis in the field of Library and Information Science [4]. The outcome of this facet analysis is a Faceted Classification Scheme (FCS) for the domain concept in question where in most cases a *facet* would correspond to a *classification criterion*. To obtain an ontology representation of the FCS, we examined the Normalisation Ontology Design Pattern (ODP) [5] [6] [7]. A comparative analysis between a FCS and the Normalisation ODP revealed the existence of key similarities between both knowledge representation paradigms. The similarities allowed us to identify a series of mappings to transform a FCS into an OWL ontology applying the Normalisation pattern. Moreover, the ontology model obtained through this process contains a valid OWL DL representation of the classification criteria involved in the characterization of the domain concept.

To illustrate our contribution, we used throughout the document an existing FCS example in the domain of “Dishwashing Detergent” [8]. In fact, there are aspects of the work presented in this paper that could be viewed as a follow-up to [8] in the context of the Semantic Web and we attempted to acknowledge that in the title.

There is an important use case worth highlighting for motivating the need of this work as well. That is the modeling of the concept "Fault" in the domain of resilient and dependable computer systems. The representation of "Fault" is part of an ontology featured in a web portal knowledge base (RKBExplorer[1]) for the project ReSIST[2] (Resiliance for Survivability in Information Society Technologies) [9].

The rest of this paper is structure as follows: Section 2 provides a comparison to previous work closely related to our proposal; Section 3 describes the structure and elements of a generic FCS; Section 4 does likewise regarding the Normalisation ODP; Section 5 introduces the alignments identified between both knowledge representation paradigms to enable the transformation of a generic FCS into a normalised ontology; Section 6 provides promising opportunities for further development and finally, Section 7 concludes the paper with some final remarks.

2 Related Research

Previous work that defines mappings between different semantic models include [10]. The authors performs a rigorous and comprehensive comparative analysis between the primitive elements of three semantic models: the Semantic Web Ontology Language (OWL), the Relational Database Model (RDBM), and the Resource Space Model (RSM). Based on the identified mappings between every two models, a detailed set of criteria is provided to transform one of them to the other. The most relevant to us is the mapping between RSM and OWL because of its similarities with the conversion between a FCS and OWL that we propose here.

The RSM is defined as a semantic model for specifying, organizing and retrieving diverse multimedia resources by classifying their contents according to different partition methods and organizing them according to a multidimensional classification space. A FCS is also a multidimensional classification space and comparing the primitive elements of a FCS and a RSM the following mapping is instantly revealed:

- The domain of the FCS (the target domain concept) corresponds to the overall resource space, the RS element in the RSM.
- A facet in the FCS corresponds to an axis X_i in the RSM.
- A facet term in the FCS corresponds to a coordinate C_i in the RSM.
- A facet is covered and exhausted by the set of terms associated to it in a FCS. The same principle holds in a RSM for an axis and the set of coordinates associated to it, $X_i = \langle C_i1, C_i2, ..., C_in \rangle$.
- An item to be classified by the FCS corresponds to a point p in the RSM.

[1] `http://www.rkbexplorer.com/`

[2] `http://www.resist-noe.org/`

- The principle of homogeneity and mutual exclusivity for the facets of a FCS implies that the corresponding RSM satisfies the third normal form (3NF).

These mappings show that a generic FCS can be converted into a RSM, which in turn can be converted into an OWL model using the RSM to OWL mappings in [10]. Now there are two possible paths to convert a FCS into an OWL model.

- Path 1: FCS to RSM via mappings above and RSM to OWL via mappings in [10]. Let us refer to this OWL model as O1.
- Path 2: FCS to OWL via mappings presented in our paper using the Normalization ODP. Let us refer to this OWL model as O2.

There are important differences between the ontologies O1 and O2. An important difference is due to the RSM to OWL conversion in [10]. RSM describes mainly classification semantics and as the authors explain, this means that there is no semantic loss when converting from RSM to OWL but there might be semantic loss when transforming an OWL model that includes richer semantics into a RSM. This also means that, in terms of W3C standards, the expressivity level of the resultant OWL model O1, will be within the RDF Schema or OWL Lite boundary.

On the other hand, the ontology O2 is within OWL DL and presents richer OWL semantics than O1, provided by the Normalization ODP. These additional OWL DL semantics in O2 enable one of the main features of the normalization pattern such as the automatic classification and maintenance of complex subsumption relations by a reasoner. So while O1 is a valid OWL description of the FCS that it is based on, O2 using our proposed method provides additional semantics at the OWL DL level that support a richer description and additional features of the classification criteria considered in the initial FCS.

Additional research that made use of facet analysis in Library and Information Science to build computational ontologies includes [11]. Giunchiglia et al. introduces the concept of Faceted Lightweight Classification Ontology as "a lightweight (classification) ontology where each term and corresponding concept occurring in its node labels must correspond to a term and corresponding concept in the background knowledge, modeled as a faceted classification scheme".

Similarities to our approach include:

- The use of a FCS to model certain background knowledge and to derive and ontology based on it.
- Each concept in the ontology model obtained in our method also corresponds to a concept in the FCS.

There are important differences where our approach deviates from that in [11] probably due to the different type of problems that we are trying to address respectively. Giunchiglia et al. are trying to address the lack of interest and difficulties on the user side to build and reuse ontologies while we are concern about identifying explicit guidelines to represent the notion of multiple classification criteria in domain concepts. Additional differences include:

- The expressive level for the resultant ontology model in our method is OWL DL. In contrast, [11] focuses on lightweight classification ontologies which expressive level would loosely correspond to no more than RDF Schema in terms of W3C Standards. Many of the features found in our method can not be implemented using solely RDF Schema semantics.
- The type of FCS used in [11] is based on a Universal Faceted Classification System. On the other hand, we have focused on simpler custom domain-specific FCSs to serve as an starting point for our initial proof of concept. This helped us to limit the complexity of the classification criteria to consider and represent in the corresponding ontology.

3 Faceted Classification Scheme

This section remarks the main features of a FCS involved in the comparative analysis to the Normalisation ODP for a given domain of discourse, while a thorough overview of facet analysis and FCSs can be found in [4] [12]. The latter also explores how FCSs compare to other knowledge representation approaches in classification and provides an account of its strengths and limitations.

Denton characterized a FCS for a given domain as follows: "a set of mutually exclusive and jointly exhaustive categories, each made by isolating one perspective on the items (a facet), that combine to completely describe all the objects in question, and which users can use, by searching and browsing, to find what they need" [8](§ 0).

However, in order to develop a FCS it is required to go through the process of Facet Analysis. Vickery describes Facet Analysis as: "The essence of facet analysis is the sorting of terms in a given field of knowledge into homogeneous, mutually exclusive facets, each derived from the parent universe by a single characteristic of division" [8](§ 2.3).

The key to Facet Analysis and FCSs is the notion of *facet*. Spiteri simplified existing principles used in established Universal FCSs in Library Science and introduces: "The Principles of Homogeneity and Mutual Exclusivity state respectively that facets must be homogeneous and mutually exclusive, i.e., that the contents of any two facets cannot overlap, and that each facet must represent only one characteristic of division of the parent universe" [4].

In this sense, each facet can be designed separately and it models the domain of discourse from a distinct aspect. Each facet consists of a terminology, a finite set of terms that exhaust the facet. This set of terms is also referred to as *foci*.

There are numerous types of FCSs that vary in complexity. For example, FCSs that include several subject fields containing multiple facets and subfacets [13](§ 8, Fig. 1). However, the rest of this section characterizes the elements of a simple generic FCS that this paper will refer to hereafter.

3.1 Structure and Elements

Definition 1. *Elements of a simple generic Faceted Classification Scheme:*

- Target Domain Concept (TDC).
- Facets: Facet1, Facet2, ..., rest of facets.
- Terms or foci (organized by facets):
  - Facet1: F1Term1, F1Term2, ..., rest of terms in Facet1.
  - Facet2: F2Term1, F2Term2, ..., rest of terms in Facet2.
  - ... rest of terms by facet.
- Set of items (from the TDC) to classify: Item1, Item2, ..., rest of items.

The following notation is introduced to refer to the elements of a generic FCS in Def. 1:

- TDC denotes the domain of discourse. The domain-specific concept targeted by the FCS.
- $Facet_i$ denotes one of the facets of the FCS.
- F_iTerm_j denotes one of the terms of $Facet_i$.
- $Item_x$ denotes one the items from the domain of discourse to be classified.

Example 1. The structure below recaps the final FCS developed for the "Dishwashing Detergent" domain example in [8](§ 2.4). The elements of the schema fit into the generic structure presented in Def. 1.

- The TDC element is populated with the domain "Dishwashing Detergent".
- $Facet_i$ elements are populated with the facets: "Agent", "Form", "Brand Name", "Scent", "Effect On Agent", and "Special Property".
- F_iTerm_j elements are populated with the terms or foci listed below (grouped by facet):
  - Agent: dishwasher, person.
  - Form: gel, gelpac, liquid, powder, tablet.
  - Brand Name: Cascade, [...], Palmolive, President's Choice, Sunlight.
  - etc.
- $Item_x$ elements are populated in this case with two example items to classify:
  - "President's Choice Antibacterial Hand Soap and Dishwashing Liquid".
  - "Palmolive Aroma Therapy, Lavender and Ylang Ylang".

4 Normalisation Ontology Design Pattern

This section highlights the main characteristics of the Normalisation ODP relevant to the comparative analysis to a FCS.

The Normalisation pattern is classified as a "Good Practice" ODP in the catalog of ODPs introduced in [6] [7] (available online[3]). It can be applied to any OWL DL ontology that consists of a polyhierarchy where some *semantic axes* can be pointed. Each of those axes will be a *module*. One of their most powerful features, is the ability of logical reasoners to link these independent ontology modules to allow them to be separately maintained, extended, and re-used.

The pattern also establishes a series of requirements that a normalised ontology should meet, some of which are summarized below:

[3] `http://odps.sourceforge.net/`(§ Normalisation)

- The essence for the normalisation proposal is that the primitive skeleton of the domain ontology should consist of disjoint homogeneous trees (also referred to as *modules*) [5].
- Each primitive class that is part of the primitive skeleton should only have a primitive parent, and primitive sibling classes should be disjoint, creating the *modules* [6](§ 4.3.2.1).
- This implies that for any two primitive concepts either one subsumes the other or they are disjoint. Assertion of multiple inheritance relations among primitive concepts are not allowed [5].
- Normalisation allows exactly one unlabelled flavour of *is-kind-of* link corresponding to the links declared in the primitive skeleton. All others are inferred by the reasoner [5].

4.1 Structure and Elements

OWL Classes. There are several examples of the generic structure of the Normalisation ODP in the literature [6](§ 4.3.2.1), [7](§ 6.5.1, § A.13) and online[3]. Figure 1 presents the specific version of the generic structure that this paper will refer to hereafter, which preserves the required characteristics of the pattern. Every node of the tree in Fig. 1, denotes an owl:Class. The symbol "(≡)" indicates that the corresponding node is a *defined* class. Otherwise, the node is a *primitive* class.

Figure 2 depicts a further generalization of the structure in Fig. 1 and introduces the following notation:

- :TDC denotes a primitive class representing the domain concept being normalised.
- :$Module_i$ denotes a primitive class that represents one of the modules.
- :M_iClass_j denotes a primitive class that represents a subset of the module class :$Module_i$.
- :M_iClass_jTDC denotes a defined class that represents a subset of the target domain concept class :TDC. Every class :M_iClass_jTDC is defined based on its relationship to the single corresponding class :M_iClass_j that it is derived from.
- :$SpecificTDC_x$ denotes a primitive class that represents a subset of the target domain concept class :TDC and an entity from the domain to be classified. Every class :$SpecificTDC_x$ is described based on its relationship to various classes :M_iClass_j from potentially different modules. As a consequence of this relationship, the classes :$SpecificTDC_x$ could introduce the polyhierarchy scenarios in the ontology model that the Normalisation ODP aims to manage.

OWL Properties. In addition to the structure of classes presented in Fig. 2 and 1, the pattern requires the use of object properties to link the various modules :$Module_i$ to the different subclasses of the target domain concept :M_iClass_jTDC and :$SpecificTDC_x$. In general terms, an object property of the type :$hasModule_i$ is declared for every module :$Module_i$ in the pattern.

```
owl:Thing
  |-- :Module1
     |-- :M1Class1
     |-- :M1Class2
     |-- (... rest of subclasses of Module1)
  |-- Module2
     |-- :M2Class1
     |-- :M2Class2
     |-- (... rest of subclasses of Module2)
  |-- (... rest of modules and subclasses)
  |-- :TargetDomainConcept (or :TDC)
     |-- (≡) :M1Class1TDC
     |-- (≡) :M1Class2TDC
     |-- (≡) (... rest of defined classes based on Module1)
     |-- (≡) :M2Class1TDC
     |-- (≡) :M2Class2TDC
     |-- (≡) (... rest of defined classes based on Module2)
     |-- (≡) (... rest of defined classes based on
                  subclasses of the rest of modules)
     |-- :SpecificTDC1
     |-- :SpecificTDC2
     |-- (... rest of specific items from the TDC
              to be represented and classified)
```

Fig. 1: Generic structure of the Normalisation ODP.

4.2 Implementation

One of the main features of the Normalisation ODP is to enable a reasoner to mantain the subsumption relations between a class :$SpecificTDC_x$ and the various classes :M_iClass_jTDC involved in its description. This feature is accomplished encoding the conditions of the subsumption relation as restrictions in the implementation of the classes :M_iClass_jTDC and :$SpecificTDC_x$.

Definition 2. *The implementation of a generic defined class :M_iClass_jTDC is given as follows:*

```
:M_iClass_jTDC
        rdf:type owl:Class ;
        rdfs:subClassOf :TDC ;
        owl:equivalentClass [ rdf:type owl:Restriction ;
                              owl:onProperty :hasModule_i ;
                              owl:someValuesFrom :M_iClass_j
                            ] .
```

This implementation indicates that:

- Every :M_iClass_jTDC class is equivalent to an anonymous class described by an existential property restriction.

```
owl:Thing
  |-- :Module_i
     |-- :M_iClass_j
  |-- :TargetDomainConcept (or :TDC)
     |-- (≡) :M_iClass_jTDC
     |-- :SpecificTDC_x
```

Fig. 2: Generic structure of the Normalisation ODP.

- The restriction is on the object property $:hasModule_i$ associated to the module $:Module_i$ given that $:M_iClass_jTDC$ is derived from $:M_iClass_j$ (a subclass of $:Module_i$).
- The filler of the restriction is the class $:M_iClass_j$, which $:M_iClass_jTDC$ is derived from.

Definition 3. *The implementation of a generic class* $:SpecificTDC_x$ *is given as follows:*

```
:SpecificTDC_x
        rdf:type owl:Class ;
        rdfs:subClassOf :TDC ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :hasModule_i ;
                          owl:someValuesFrom :M_iClass_j
                        ] ,
                        [ ... rest of existential restrictions for
                              every class :M_iClass_j that participates
                              in the description of :SpecificTDC_x
                              via the property :hasModule_i
                        ] .
```

This representation indicates the following:

- Every class $:SpecificTDC_x$ is subsumed by a variable number of anonymous classes. More specifically, one anonymous class for every class $:M_iClass_j$ of every module $:Module_i$ that is linked to the description of $:SpecificTDC_x$. Every anonymous class is represented by an existential property restriction such as:
  - The restriction is on the object property $:hasModule_i$ associated to the module $:Module_i$ given that the description of $:SpecificTDC_x$ is linked to $:M_iClass_j$ (a subclass of $:Module_i$).
  - The filler of the restriction is the class $:M_iClass_j$, that is linked to the description of $:SpecificTDC_x$.

This implementation of the classes $:M_iClass_jTDC$ and $:SpecificTDC_x$ respectively, enable a reasoner to infer and maintain the subsumption relations between a given class :$SpecificTDC_x$ and the various classes :M_iClass_jTDC that it is related to.

Specific examples of the Normalisation ODP in the literature [6](§ 4.3.2.1), [7](§ 6.5.1, § A.13) and online[3] demonstrate the features of the pattern in specific use case scenarios.

5 Alignment of a FCS to the Normalisation ODP

A comparative analysis between the main characteristics of a FCS and the Normalisation ODP presented in previous sections, indicates the existence of key similarities between the elements in the generic structures of both conceptual models.

One such key similarity lies in the notion of *facet* in FCSs and the notion of *module* (or *semantic axis*) in the Normalisation ODP. Both elements represent one perspective of the domain being modelled, a single characteristic of division, a single criterion of classification in their respective paradigm.

Another key similarity is linked to the requirement for facets in a FCS to be homogeneous and mutually exclusive and likewise the requirement of modules in the Normalisation ODP to be comprised of primitive classes arranged in a structure of disjoint homogeneous class trees.

These key similarities prompt us to identify a mapping between the elements of both conceptual models that allows to transform a FCS into a normalised ontology model. In this first approach, the mapping aims to keep the design choices of the resultant normalised ontology as simple and straight-forward as possible, without compromising any of the requirements and features of both FCSs and the normalisation mechanism. This approach might not be suitable for converting all possible schemas into a normalised ontology but it is an attempt to provide an initial set of basic design guidelines. These guidelines can be extended hereafter to support more complex cases of FCSs.

The main principle is to represent each *facet* as a independent *module* or *semantic axis*. Following this principle makes the application of the Normalisation ODP *almost* straight-forward. Moreover, the resultant ontology includes the representation of the multiple alternative classification criteria that were considered in the original FCS for the target domain concept.

Table 1 summarizes the alignment of the elements in the generic structure of both conceptual models. This alignment enables the conversion from a FCS to an OWL DL ontology by applying the Normalisation ODP.

- The first column (leftmost), contains the elements of a generic FCS as introduced in Sect. 3, Def. 1.
- The second column contains the elements of the Normalisation ODP generic structure as introduced in Sect. 4, Fig. 2.
- The third column represents the selected OWL notation for the elements of a generic FCS in the context of the Normalisation ODP generic structure.

– The forth column (rightmost), indicates the OWL implementation chosen for every element. The selection complies with the requirements of the normalisation mechanism.

Library Science	**Ontology Modeling**		
FCS	**Norm. ODP**	**FCS in Norm. ODP**	**OWL Implementation**
TDC	$:TDC$		owl:Class (primitive)
$Facet_i$	$:Module_i$	$:Facet_i$	owl:Class (primitive)
	$:hasModule_i$	$:hasFacet_i$	owl:ObjectProperty
F_iTerm_j	$:M_iClass_j$	$:F_iTerm_j$	owl:Class (primitive)
	$:M_iClass_jTDC$	$:F_iTerm_jTDC$	owl:Class (defined) (≡)
$Item_x$	$:SpecificTDC_x$		owl:Class (primitive)

Table 1. Alignment of a Faceted Classification Scheme to the Normalisation ODP

Based on the principle of representating each facet as a module, the underlying ideas behind the mappings in Table 1 can be outlined as follows:

– The target domain concept TDC represents the domain of discourse of both a FCS and the Normalisation ODP. The primitive class $:TDC$ fulfills that role in the normalised ontology.
– A facet $Facet_i$ from a generic FCS corresponds to a module $:Module_i$ in the Normalisation ODP, therefore it becomes a primitive class $:Facet_i$ in the normalised ontology model.
– A facet $Facet_i$ from a FCS also becomes an object property $:hasFacet_i$ in the normalised ontology, given that for every module $:Module_i$ in the Normalisation ODP, there is an object property $:hasModule_i$.
– From the relationship between facet and module, it follows that a facet term F_iTerm_j from a FCS maps to a module subclass $:M_iClass_j$ from the Normalisation ODP. Both elements represents the same notion in their respective conceptual models. A subvidision, a refinement of the facet or module that they complement respectively. Therefore, a facet term F_iTerm_j from a FCS becomes a primitive class $:F_iTerm_j$ in the normalised ontology.
– A facet term F_iTerm_j from a FCS also produces a defined class $:F_iTerm_jTDC$ in the normalised ontology, given that for every primitive class $:M_iClass_j$ in the Normalisation ODP, there is a corresponding defined class $:M_iClass_jTDC$.
– Every item $Item_x$ to be classified in the FCS aligns to a class $:Specific_x$ that is automatically classified by a reasoner in the Normalization ODP. Therefore, every element $Item_x$ is represented as a primitive class $:SpecificTDC_x$ in the normalised ontology.

The rest of this section details the characteristics of the resultant normalised ontology model that is obtained by applying the Normalisation ODP to a generic FCS. The application of the pattern is driven by the alignments summarized in Table 1. The process is illustrated using the example of the "Dishwashing Detergent" FCS presented in Sect. 3.1, Ex. 1.

5.1 Structure and Elements

OWL Classes. Figure 3 depicts the placement of the elements of a generic FCS into the generic structure of the Normalisation ODP based on the structure of the pattern in Fig. 2 and the corresponding mappings from Table 1.

```
owl:Thing
  |-- :Facet_i
     |-- :F_iTerm_j
  |-- :TargetDomainConcept (or :TDC)
     |-- (≡) :F_iTerm_jTDC
     |-- :SpecificTDC_x
```

Fig. 3: Placement of FCS elements into the Normalisation ODP generic structure.

Example 2. Now let us populate the generic ontology structure in Fig. 3 with the specific elements of the "Dishwashing Detergent" FCS example. Figure 4 presents the overall normalised ontology class diagram obtained.

It is important to note that the structure in Fig. 4 includes axioms to comply with the requirement already stated of the Normalization ODP. That is, the skeleton of primitive classes consists of disjoint homogeneous tress where each primitive class only has a primitive parent, and primitive sibling classes are disjoint, creating the modules. This normalization requirement complies as well with the FCS requirement of facets being homogeneous and mutually exclusive based on the alignments in Table 1.

OWL Properties. Based on the mapping between the elements $Facet_i$ and $:hasFacet_i$, the object properties created for the "Dishwashing Detergent" FCS example are: `:hasAgent`, `:hasForm`, `:hasBrand`, `:hasScent`, `:hasEffectOnAgent` and `:hasSpecialProperty`.

5.2 Implementation

Defined Classes. The generic implementation of a defined class $:F_iTerm_jTDC$ in terms of FCS elements is straight-forward based on the definition of $:M_iClass_jTDC$ (Sect. 4, Def. 2) and the corresponding mappings from Table 1.

Example 3. Let us illustrate the implementation of a defined class in the "Dishwashing Detergent" FCS example. Consider the facet "Agent" which contains the terms "Person" and "Dishwasher". Based on Table 1, these FCS elements fit into the normalised ontology as follows:

```
owl:Thing
  |-- :Agent
     |-- :Person
     |-- :Dishwasher
  |-- :Form
     |-- :Gel
     |-- :Gelpac
     |-- (... rest of terms in the facet "Form")
  |-- :BrandName
     |-- :Cascade
     |-- :Electrasol
     |-- (... rest of terms in the facet "Brand Name")
  |-- :Scent
     |-- :GreenApple
     |-- :GreenTea
     |-- (... rest of terms in the facet "Scent")
  |-- :EffectOnAgent
     |-- :AromaTherapy
        |-- :Invigorating
        |-- :Relaxing
  |-- :SpecialProperty
     |-- :Antibacterial
  |-- :DishwashingDetergent (:TDC)
     |-- (≡) :ManualDishDetergent
     |-- (≡) :DishwasherDishDetergent
     |-- (≡) :GelDishDetergent
     |-- (≡) :GelpacDishDetergent
     |-- (≡) (... rest of subclasses for each term in the facet "Form")
     |-- (≡) :CascaseDishDetergent
     |-- (≡) :ElectrasolDishDetergent
     |-- (≡) (... rest of subclasses for each
                                          term in the facet "Brand Name")
     |-- (≡) :GreenAppleDishDetergent
     |-- (≡) :GreenTeaDishDetergent
     |-- (≡) (... rest of subclasses for each term in the facet "Scent")
     |-- (≡) :AromaTherapyDishDetergent
        |-- (≡) :InvigoratingDishDetergent
        |-- (≡) :RelaxingDishDetergent
     |-- (≡) :AntibacterialDishDetergent
     |-- :PresidentsPersonLiquidAntibacterial
     |-- :PalmoliveAromaTherapyLavenderYlangYlang
     |-- :SpecificDishDetergent3
     |-- (... rest of specific dish detergent classes
              :SpecificDishDetergent_x to classify)
```

Fig. 4: Normalised ontology structure of the FCS for the "Dishwashing Detergent" domain concept. The symbol "(≡)" denotes a defined class.

- :$Facet_i$ is populated with :Agent.
- :$hasFacet_i$ is populated with :hasAgent.
- :F_iTerm_j is populated with :Person and :Dishwasher.
- :F_iTerm_jTDC is populated with :ManualDishDetergent and :DishwasherDishDetergent.

As an example, let us focus on the class :DishwasherDishDetergent. Based on the correspondence above, the implementation of :DishwasherDishDetergent in the normalised ontology can be stated as follows:

```
:DishwasherDishDetergent
          rdf:type owl:Class ;
          rdfs:subClassOf :DishDetergent .
          owl:equivalentClass [ rdf:type owl:Restriction ;
                                owl:onProperty :hasAgent ;
                                owl:someValuesFrom :Dishwasher
                              ] ;
```

The implementation of the rest defined classes in the "Dishwashing Detergent" FCS shown in Fig. 4 follows the same rationale.

Classification Classes. The generic implementation of a class :$SpecificTDC_x$ in terms of FCS elements is straight-forward following the implementation of :$SpecificTDC_x$ given in Sect. 4, Def. 3 and the applicable mappings from Table 1.

Example 4. To illustrate the representation of a specific dishwashing detergent, let us reuse one of the classification examples presented in [8](§ 2.4). Applying the "Dishwashing Detergent" FCS developed in the cited reference, the item "President's Choice Antibacterial Hand Soap and Dishwashing Liquid" is classified as follows:

- Agent: person
- Form: liquid
- Brand Name: President's Choice
- Scent: (none)
- Effect on Agent: (none)
- Special Property: antibacterial

According to the guidelines derived from Table 1 and the implementation of a class :$SpecificTDC_x$ given in Sect. 4, Def. 3, the description of the example detergent reveals the following mappings:

- :TDC is populated by :DishDetergent.
- :$SpecificTDC_x$ is populated by :PresidentsPersonLiquidAntibacterial.
- There are 4 existential restrictions. One per facet term involved in the description of the specific detergent at hand ("person", "liquid", "President's Choice", and "antibacterial"). Therefore, for each restriction:

- $:hasFacet_x$ is populated with :hasAgent, :hasForm, :hasBrandName and :hasSpecialProperty respectively.
- $:F_iTerm_j$ is populated with :Person, :Liquid, :PresidentsChoice and :Antibacterial respectively.

Therefore, the implementation of this particular detergent in the normalised ontology can be stated as follows:

```
:PresidentsPersonLiquidAntibacterial
         rdf:type owl:Class ;
         rdfs:subClassOf :DishDetergent ,
                         [ rdf:type owl:Restriction ;
                           owl:onProperty :hasAgent ;
                           owl:someValuesFrom :Person
                         ] ,
                         [ rdf:type owl:Restriction ;
                           owl:onProperty :hasForm ;
                           owl:someValuesFrom :Liquid
                         ] ,
                         [ rdf:type owl:Restriction ;
                           owl:onProperty :hasBrandName ;
                           owl:someValuesFrom :PresidentsChoice
                         ] ,
                         [ rdf:type owl:Restriction ;
                           owl:onProperty :hasSpecialProperty ;
                           owl:someValuesFrom :Antibacterial
                         ] .
```

This description makes explicit the relationship between the specific detergent class and every term of every facet that participate in the facet classification of the item. Moreover, it enables a reasoner to infer that :PresidentsPersonLiquidAntibacterial is a subclass of the defined classes :ManualDishDetergent, :LiquidDishDetergent, :PresidentsChoiceDishDetergent and :AntibacterialDishDetergent.

A version of the complete normalised ontology model for the "Dishwashing Detergent" FCS example is available online in RDF/XML format[4].

5.3 Discussion

There are certain design choices that need to be taking into account in the transformation process of a FCS into a normalised ontology.

[4] http://dl.dropbox.com/u/1666716/Attachments/detergent_fcs_normalisation_prot3x_owl.owl

Self-standing versus Partitioning concepts. The original normalisation mechanism recommends to differentiate classes in the ontology model based on whether they represent a *domain* entity (also known as self-standing or indenpendent entity) or a *modifier* entity (also known as refining or dependent entity) [5]. The Normalisation ODP derived from the original mechanism does not explicitly request this distinction [6](§ 4.3.2.1) [7](§ 6.5.1, § A.13). For simplicity, the proposed transformation guidelines considered all entities to be *domain* entities.

Domain and Range of Object Properties. The Normalisation ODP does not prescribe the domain and range of object properties in the pattern. It only requires that domain and range property restrictions do not introduce overlap between primitive concepts that are intended to be disjoint. This scenario can take place when the domain or range of a property is set to more than one class which results in the intersection of the classes involved. Based on the definition of the object property :$hasFacet_i$, the natural choice of domain and range would be the target domain concept class :TDC and the corresponding facet class :$Facet_i$ respectively. That is:

```
:hasFacet_i rdf:type owl:ObjectProperty ;
            rdfs:domain :TDC ;
            rdfs:range :Facet_i .
```

Mutual Exclusion of Facets. There is a characteristic of a FCS system that could lead to some confusion. A FCS requires its facets and facet terms to be mutually exclusive in the conceptual model of the scheme. However, it allows to use multiple terms from the same facet to classify an item from its domain of discourse.

This characteristic is illustrated in the classification example of the detergent "Palmolive Aroma Therapy, Lavender and Ylang Ylang" given in [8](§ 2.4) using two terms of the same facet to classify the item (Scent: lavender, ylang ylang).

The feature of *functional* property provided by OWL allows to capture this behaviour in an ontology model given that existential property restrictions lead to *unsatisfiability* if a functional property is inferred to have two or more disjoint values.

In terms of our ontology model: (a) the primitive classes :Lavender and :YlangYlang, subclasses of :Scent, are disjoint to comply with FCS and Normalisation requirements; and (b) as per Def. 3, the representation of the class :PalmoliveAromaTherapyLavenderYlangYlang includes two existential restrictions on property :hasScent over the classes :Lavender and :YlangYlang respectively. Under these conditions, if :hasScent is a functional property, the class :PalmoliveAromaTherapyLavenderYlangYlang would be infered to be unsatisfiable.

6 Future Work

There are interesting opportunities in the short term for further development and enhancement of the design guidelines presented here to transfrom a FCS into an OWL DL ontology.

Multiple FCSs. One such opportunity lies in the ontology modeling front. We can identify additional design scenarios that present in our opinion, attractive incremental challenges. Consider the situation where two (or more) different domain-specific FCSs are to be transformed into a single normalised ontology model (i.e. a FCS for "Dishwashing Detergent" and a different FCS for "Tooth Cleaning Products"). A situation with two FCSs can lead to the following design scenarios:

1. FCS1 and FCS2 do not have any element in common (facet or facet term).
2. FCS1 and FCS2 do have some element in common (facet or facet term).
3. The domain of discourse (TDC) of FCS1 appears as a facet or as a facet term in FCS2 (or vice versa).

Case 1 would be the simplest. The Normalisation ODP could be applied separately to FCS1 and FCS2 following our tranformation guidelines and combine the outcome into a single ontology model. The only difference between Case 1 and having only one FCS, is that the ontology model obtained will include two $:TDC$ classes, (provided by FCS1 and FCS2 respectively) and the rest of the ontology elements ($: Facet_i$, $: hasFacet_i$, $: F_iTerm_j$, etc.) will be populated with the elements of both FCS1 and FCS2.

Case 2 and 3 on the other hand, could potentially lead to a myriad of different modeling issues that have not been yet explored. Going forward, we would like to extend our transformation guidelines to support scenarios such as Case 2 and 3.

Automation. An additional opportunity being considered is the automation of the normalised ontology creation process. Provided a source FCS and the mappings identified here, an application could automatically or semi-automatically generate the corresponding normalised ontology artifact. Depending on the complexity of the source FCS, user intervention might be required to disambiguate among several valid design choices available and assist the application to select the preferred option. To materialize this application, we are evaluating the development of a plug-in or extension for some of the most popular open ontology development frameworks.

7 Conclusions

This paper has presented an initial set of basic design guidelines to develop an ontology model within OWL DL that supports the representation of multiple (potentially alternative) classification criteria of a specific domain concept.

At the moment, there seems to be a lack of explicit guidance in the ontology development literature on how to represent multiple classification criteria of ontology domain concepts, leaving ample room for ad-hoc practices that can lead to unexpected or undesired results in ontology artifacts. In our quest to address this void, we examined the procedure to develop a Faceted Classification Scheme (FCS) which contains the conceptualization of various classification criteria (facets) of a specific target domain concept. A series of mappings between the elements of a generic FCS and the Normalization Ontology Design Pattern (ODP) have been identified that allow us to convert a generic FCS into an OWL DL ontology model following a consistent and systematic approach. The resultant ontology model includes the representation of the various classification criteria of the domain concept that were considered in the original FCS. An existing example of a FCS for the domain concept Dishwashing Detergent is used to illustrate the main steps of our conversion rationale.

The guidelines presented in this first effort do not solve all possible unexpected or undesired results in ontology artifacts when representing multiple classification criteria. Not all ad-hoc situations are eliminated and not all types of generic structures of FCSs have been considered either, (which can be material for future forthcoming research). However, they make explicit the conceptualization of existing classification criteria and we believe they already provide a valuable and novel contribution to the tool-set available to ontology engineers to address this particular and recurrent modelling scenario. By using a consistent and systematic approach, the opportunity for ad-hoc decisions in the development process that could lead to modeling errors is significantly reduced.

Acknowledgments. Many people have contributed directly and indirectly to this work and we thank them all. In particular, everyone that participated in the online discussion "The notion of a 'classification criterion' as a class"[5][6] via the mailing lists: "ontolog-forum@ontolog.cim3.net" and "public-owl-dev@w3c.org"; and the British Chapter of the International Society for Knowledge Organization (ISKO).

References

1. Welty, C., McGuinness, D.L., Smith, M.K.: OWL web ontology language guide. W3C recommendation, W3C (February 2004) http://www.w3.org/TR/2004/REC-owl-guide-20040210/.
2. Krötzsch, M., Patel-Schneider, P.F., Rudolph, S., Hitzler, P., Parsia, B.: OWL 2 web ontology language primer. Technical report, W3C (October 2009) http://www.w3.org/TR/2009/REC-owl2-primer-20091027/.
3. Horridge, M., Drummond, N., Jupp, S., Moulton, G., Stevens, R.: A practical guide to building owl ontologies using the protege-owl plugin and co-ode tools edition 1.2. Technical report, The University Of Manchester (March 2009)
4. Spiteri, L.: A simplified model for facet analysis: Ranganathan 101. Canadian Journal of Information and Library Science **23**(1/2) (1998) 1–30
5. Rector, A.L.: Modularisation of domain ontologies implemented in description logics and related formalisms including owl. In: K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture, New York, NY, USA, ACM (2003) 121–128

[5] `http://lists.w3.org/Archives/Public/public-owl-dev/2010AprJun/0009.html`
[6] `http://ontolog.cim3.net/forum/ontolog-forum/2010-04/msg00051.html`

6. Egana-Aranguren, M.: Ontology Design Patterns for the Formalisation of Biological Ontologies. MPhil Dissertation, Bio-Health Informatics Group, School of Computer Science, University of Manchester (2005)
7. Egana-Aranguren, M.: Role and Application of Ontology Design Patterns in Bio-ontologies. PhD thesis, School of Computer Science, University of Manchester (2009)
8. Denton, W.: How to make a faceted classification and put it on the web. Online (November 2003) http://www.miskatonic.org/library/facet-web-howto.html.
9. Rodriguez-Castro, B., Glaser, H.: Whose "fault" is this? untangling domain concepts in ontology design patterns. In: Workshop on Knowledge Reuse and Reengineering over the Semantic Web in the 5th European Semantic Web Conference. (June 2008)
10. Zhuge, H., Xing, Y., Shi, P.: Resource space model, owl and database: Mapping and integration. ACM Trans. Internet Technol. **8**(4) (2008) 1–31
11. Giunchiglia, F., Dutta, B., Maltese, V.: Faceted lightweight ontologies. In Borgida, A., Chaudhri, V.K., Giorgini, P., Yu, E.S.K., eds.: Conceptual Modeling: Foundations and Applications. Volume 5600 of Lecture Notes in Computer Science., Springer (2009) 36–51
12. Kwasnik, B.H.: The role of classification in knowledge representation and discovery. Library Trends **48**(1) (1999)
13. Vickery, B.: Faceted classification for the web. Axiomathes **18**(2) (June 2008) 145–160