



HAL
open science

A knowledge base with modularized ontologies for eco-labeling: application for laundry detergents

Da Xu, Mohamed Hedi Karray, Bernard Archimède

► To cite this version:

Da Xu, Mohamed Hedi Karray, Bernard Archimède. A knowledge base with modularized ontologies for eco-labeling: application for laundry detergents. *Computers in Industry*, 2018, 98, pp.118-133. 10.1016/j.compind.2018.02.013 . hal-01945196

HAL Id: hal-01945196

<https://hal.science/hal-01945196>

Submitted on 5 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <http://oatao.univ-toulouse.fr/20083>

To cite this version:

Xu, Da  and Karray, Mohamed Hedi  and Archimède, Bernard 
A knowledge base with modularized ontologies for eco-labeling: application for laundry detergents. (2018) *Computers in Industry*, 98. 118-133. ISSN 0166-3615

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

A knowledge base with modularized ontologies for eco-labeling: Application for laundry detergents

Da Xu, Mohamed Hedi Karray*, Bernard Archimède

Laboratoire Génie de Production, LGP, Université de Toulouse, INP-ENIT, Tarbes, France

ABSTRACT

Along with the rising concern of environmental performance, eco-labeling is becoming more and more popular. However, the complex process of eco-labeling is demotivating manufacturers and service providers to be certificated. The knowledge contained in eco-labeling criteria documents is not semantically exploitable to computers. Traditional knowledge base in relational data model is not interoperable, lacks inference support and is difficult to be reused. In our research, we propose a comprehensive knowledge base composed of interconnected OWL (Ontology Web Language) ontologies. This ontology based knowledge base allows reasoning and semantic query. In this paper, a modularization scheme about ontology development is introduced and it has been applied to EU Eco-label (European Union Eco-label) laundry detergent product criteria. This scheme separates entity knowledge and rule knowledge so that the ontology modules can be reused easily in other domains. Reasoning and inference based on SWRL (Semantic Web Rule Language) rules in favor of eco-labeling process is also presented.

Keywords:

Ontology engineering
Ontology modularization
Knowledge base
OWL imports
SWRL
Eco-labeling

1. Introduction

Since the late 1980s, there has been a growing demand for products that do less harm to the environment. The public willingness to use buying power as a tool to protect the environment provides manufacturers with an opportunity to develop new products [1]. From a global point of view, promote of environment-friendly consumption and production will contribute not only to the life quality but also the economy itself. But how does a consumer judge and make good choices to reduce environmental impacts? How should we assess the validity of a statement about a product or service's environmental impacts? The need of evaluating a product's environmental performance has led to the establishment of eco-labels. Nowadays, most of the knowledge and criteria about eco-labeled products are published in official journals, web pages, and all kinds of documentation. Usually, this knowledge is presented in such complex regulation and specification documents that it is difficult to be understood even by humans. The integration of this knowledge into software requires that it must be exploitable to machines. However, until now, there is still a lack of computable format of that. Besides,

traditional knowledge base in relational data model is not interoperable, lacks inference support and is difficult to be reused. In order to better understand these criteria and rules, stakeholders need a common and machine accessible presentation of the knowledge. To address such problems, in our research, we propose an ontological knowledge base composed of modularized ontologies. This scheme has been applied to the creation of the ontology knowledge base of EU Eco-label's laundry detergent products.

Due to the fact that EU Eco-label is a large and complex labeling system covering dozens of products and service groups, it is difficult and unrealistic to cover all its products and services in the research stage. Thus, we decide to choose laundry detergent products group which has a middle size knowledge volume to be our study case. The rest of the paper will follow this Outline: The first section presents a state of the art of eco-labeling and modularized ontology; in Section 3, an overview of the criteria document and requirement analysis is presented; The third section talks about how the terminology of ontology is retrieved; Section 5 presents detailed design and construction of the ontology. In particular, an entity-rule separation pattern is introduced. Basic idea of this separation is to put descriptive entity knowledge and subjective rule knowledge into different modules. This pattern is proven to be in favor of modularity and extendability, especially for the rule module. It can also be applied to the other product groups' ontology building and even other similar criteria-like document's knowledge extraction; the fifth section is about how to utilize

* Corresponding author.

E-mail addresses: da.xu@enit.fr (D. Xu), mkarray@enit.fr (M.H. Karray), bernard.archimede@enit.fr (B. Archimède).

reasoner to do the reasoning upon the ontology knowledge base and the argumentation, which is very important to eco-labeling decision support process; in Section 7, we have a brief evaluation and analysis for the ontology; Section 8 is about some discussion of experience feedback, and learned lessons. Finally, in the last section, we have conclusion, discussion and future work.

2. State of art

2.1. Eco-label and EU Eco-label

According to Global Eco-labelling Network¹ (GEN), “eco-labelling” is a voluntary method of environmental performance certification and labelling that is practiced around the world. An “eco-label” is a label that identifies overall proven environmental preference of a product or service within a specific product/service category. They usually concern the whole life cycle of the product and are issued by a third party [2]. Eco-labeling has a number of benefits from various points of view. First, eco-labeling is a good way to inform consumers of the environmental impacts of selected products. In the practice of some existent eco-labeling, the fitness of use and human health aspects are also included. All this information will help a consumer make decision out of different willingness. Then, eco-labeling is generally cheaper than regulatory controls in terms of global economics. By empowering customers and manufacturers to make environmentally supportive decisions, the need for regulation is kept to a minimum. This is beneficial to both government and industry [3]. Eco-labeling will also stimulate market development and encourage continuous improvement on products and services.

EU Eco-label is a successful example among all the eco-labels. Created in 1992, EU Eco-label is the only official European ecological label authorized for use in every member country of the European Union [4]. Until 2011, there are over 1300 enterprises that have been issued EU Eco-label licenses. By September of 2014, there are already over 43,000 products or services being labelled [5]. France is always an important contributor to EU Eco-labeling. By March of 2016, 486 enterprises in France have obtained EU Eco-label licenses in various product groups and that makes France the first place as for the enterprises’ possession of EU Eco-label licenses. As illustrated in Fig. 1, the removal of certain product group (e.g. IPV: Indoor paints and varnishes, SSC: Soaps, shampoos, and hair conditioners, and OPV: Outdoor paints and varnishes.) which happened in 2016 indicates that the alteration of EU Eco-label criteria is continuous. It also implies that the change of knowledge and rules. Although the size of LD (Laundry detergents) group is not the largest, it keeps increasing in the recent 4 years.

EUEB (European Union Eco-labeling Board) is responsible to develop and regularly review eco-label criteria. EUEB will set up an advisory body including representatives on behalf of different stakeholders. Feasibility study will be carried out to draft the environmental criteria. At last, representatives from every member state will be summoned to vote to approve the criteria or the guideline [6]. The guideline developed by the advisory body, together with the possible amendment or annex will be the baselines for the knowledge base that we developed in this work.

2.2. Ontology and modularized ontology

Derived from philosophy, in computer science, we refer to an ontology as a special kind of information object or computational artifact [7]. Studer et al. [8] gave definition stating that: “An

ontology is a formal, explicit specification of a shared conceptualization”. Today, so many ontologies and knowledge repositories have been developed and adapted into applications, especially in biomedical domains [9]. Successful examples and platforms are BioPortal,² UniProt,³ LEO,⁴ etc.

Despite quite amount of ontologies of different domains are developed, a lot of problems are encountered when knowledge engineers as well as general users want to understand and reuse the ontologies into their own development. As for the application of ontology, there is definite need to gather knowledge from multiple remote ontological sources. It is known that, when knowledge is distributed, the idea to collect all knowledge and put them into a single repository (i.e. the integration approach) is very difficult to implement, because of semantic heterogeneity calling for human processing [10]. Another very important reason is the low reusable design of these ontologies. Good ontology design pattern has drawn the attention of many researchers. In [11] and [12], a method to describe ontology design pattern is presented. A Semantic Web portal called *OntologyDesignPatterns.org*⁵ is also available. However, most of the submitted patterns are cataloged in Content Ontology Design Patterns which means that the patterns themselves may contain certain semantics and domain knowledge, which may still set obstacles to ontology reuse. Also, most of these patterns’ structure is hard to be modularized and very few of them care about modularity in a specific way. Thus, better engineering principle and philosophy about ontology modularity is needed.

Generally speaking, there are two important aspects of ontology modularization: independently developing modules that can be integrated coherently and uniformly (ontology composition) or extracting such modules from an integrated ontology for supporting a particular use cases (ontology decomposition) [9]. Most of our research focus on the first aspect and we emphasize more on reusing, inference and change management of ontology knowledge base.

To achieve ontology modularity in a distributed scenario, different methods and schemes have been proposed. For example, E-Connection is proposed as a set of “connected” ontologies. An E-Connected ontology contains not only information about classes, properties and their individuals, but also a new kind of properties, called Link Properties, which establish the connection between the ontologies [13]. Another interesting approach is Distributed Description Logics (DDL) framework [14] and the distributed reasoner DRAGO (Distributed Reasoning Architecture for a Galaxy of Ontologies) [15] as formal and practical tools for composing modular ontologies. Also, there is Package-Based Description Logics as another formalism that supports contextual reuse of knowledge from multiple ontology modules [16]. While, these methods and formalism have more or less logic compatibility problems when we try to use them together. For example, the underlying logic formalism of E-Connection is OWL-DL (i.e. SHOIN); logic formalism for DDL is SHIQ; when it comes to Package-Based Description, it turns into SHOIQ. Very few of these methods have full compatibility and equal logic expressiveness as OWL standard. This could limit large scale reasoning and modification between heterogeneous and distributed modular ontologies. From practical perspective, these methods have not been applied in such a considerable scale. Most of the methods focus on low-level modularization of syntax and semantic level, a higher level consideration which

² <http://bioportal.bioontology.org/>.

³ <http://www.uniprot.org/>.

⁴ <http://leo.informea.org/>.

⁵ http://ontologydesignpatterns.org/wiki/Main_Page.

¹ <http://www.globalecolabelling.net/>.

TOTAL EU ECOLABEL PRODUCTS & SERVICES PER PRODUCT/SERVICE GROUP

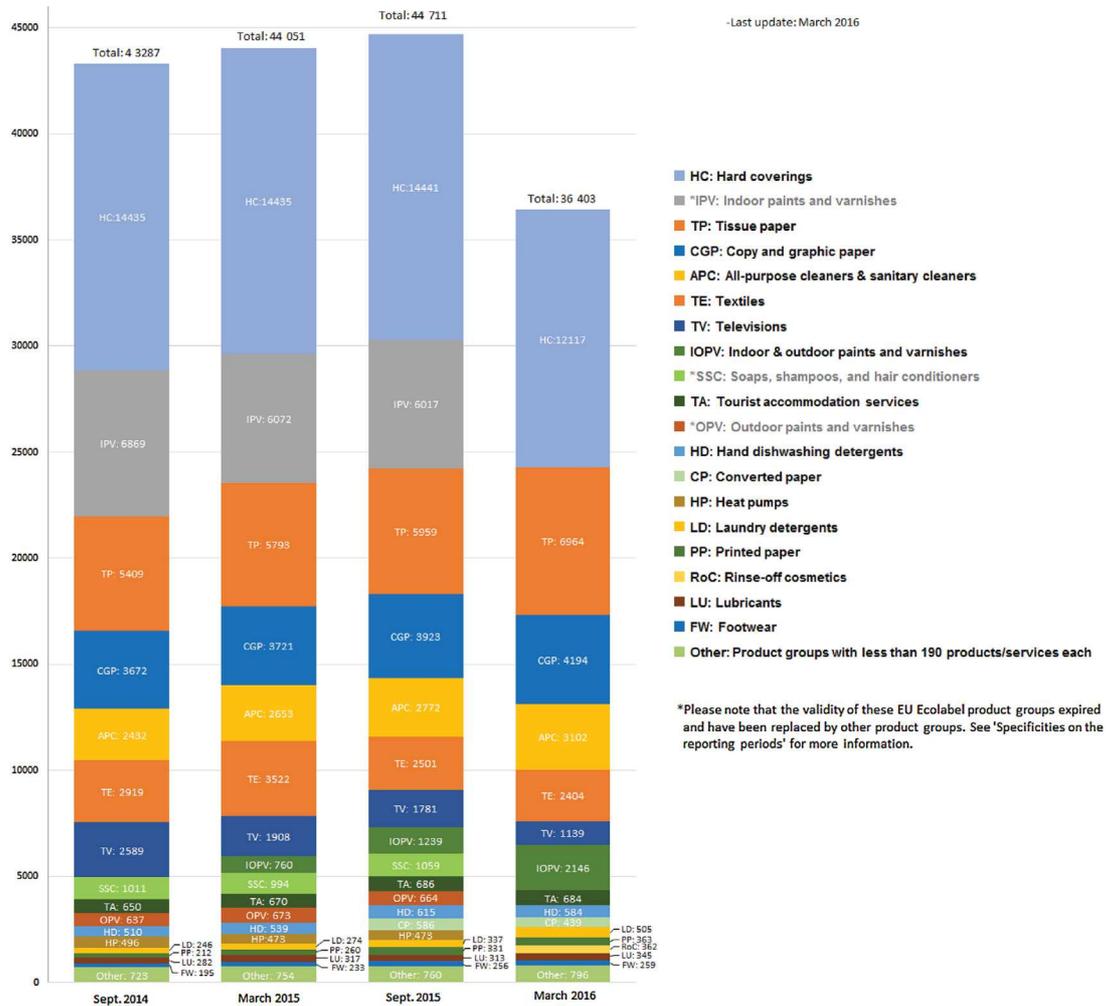


Fig. 1. Total EU Eco-label products & services per product/service group.

cares more about conceptualization itself and engineering efficiency is still lacking.

In [17], the authors have identified a high-level view of the framework for modularity. The dimensions of the framework are related as follows. A module's use-case results in modules of a certain type. A module of a certain type is created by a modularization technique. Modularization techniques result in modules with certain annotation features or properties. This work provides feasible engineering guidance for module partition or extraction. It seems to work well with descriptive conceptualization, but it does not address how to deal with rules.

As for OWL ontology, the current OWL imports syntax already provides the ability of modularization to a certain extent. It is very interesting to see that in [18] the authors present use cases for modular development of ontologies using the OWL imports mechanism. For cases (Ontology organization and factoring, interfaces between ontologies and between ontologies and software, ontology localisation, and ontology extension.) are presented to illustrate how to make use OWL syntax as well as imports constructors to build modularized ontologies. They have chosen to implement all modules as separate files. In our research, we have also taken the same approach. However, in their research, we have not seen how rules are addressed in a modular design.

In this work, we apply a method using imports syntax to build OWL ontology knowledge base with SWRL rules⁶ in which smaller ontology components can be maintained and reused more easily. We expect to explore and find out some useful design principles and engineering experience regarding to original OWL ontology scheme.

Like software engineering, engineering methodologies are also required in ontology development. Yet, in our opinion, ontology engineering is not as mature as software engineering because of its shorter history and limited relative scale of practice. In spite of that, quite several ontology development methods have been proposed, e.g. TOVE, METHONTOLOGY, DILIGENT, NeOn Methodology [19–22]. Most of these methods follow a “water fall” pattern. Common characteristics that can be generalized from these methods are iteration and refinement. In our ontology development, we don't rely on only one methodology exclusively, instead, we have adapted and customized those useful steps from all these methodologies to have a development method that best suit the

⁶ SWRL is an abbreviation for Semantic Web Rule Language, it extends the set of OWL axioms to include Horn-like rules. It thus enables Horn-like rules to be combined with an OWL knowledge base. More details can be found at <http://www.w3.org/Submission/SWRL/>.

task. The key steps in our development method are: requirement analysis, capture of motivating scenarios and competency questions, terminology collection, modeling, test reasoning and argumentation, evaluation and analysis. The rest part of this paper will describe these steps and present the modularized ontologies in detail.

3. Requirement analysis, motivating scenarios and competency questions

Firstly, let's have a brief overview of the current eco-labeling process for laundry detergent products. As EU Eco-label has been undergoing for more than twenty years in European Union, a well-defined coordination between the EU Commission and other member countries' competent bodies has been established. On the official web site of EU Eco-label,⁷ detailed documentation is provided to enterprises to facilitate the application process. On the same site, there is also a detailed product group catalog and corresponding criteria for each product or service group.

Usually, when a new product or service is about to be added into the product group catalog, various stakeholders and domain experts will be assembled. After a careful survey and discussion, a technical report will be drafted. According to this technical report, a feasible criteria will be made and then put into practice under the authorization of EU commission. From time to time, necessary revise or amendments to the criteria may be applied. As a result, the information implied in each product or service criteria becomes a complex knowledge system which involves multiple domains' expertise, standards and best practice. Take laundry detergent for example, criteria is set for each of the following aspects:

1. Dosage requirements.
2. Toxicity to aquatic organisms: Critical Dilution Volume (CDV).
3. Biodegradability of organics.
4. Excluded or limited substances and mixtures.
5. Packaging requirements.
6. Washing performance (fitness for use).
7. Points.
8. Consumer information.
9. Information appearing on the EU Eco-label.

These criteria have been published in *Commission decision of 28 April 2011 on establishing the ecological criteria for the award of the EU Eco-label for laundry detergent 2011/264/EU*.⁸ This commission decision is composed of regulation articles, annex where each item of the criteria is explained, and appendix. The regulation articles are not very interesting as it gives only administrative declarations and reference. Most of the knowledge about laundry detergent is elaborated in the annex and appendix. Criterion "Dosage requirements" specifies the reference product dosage recommended for each wash. Qualified detergent products should not exceed certain value. "Toxicity to aquatic organisms" specifies the maximum CDV value for qualified products. Similarly, in the next criterion "Biodegradability of organics", it indicates that the content of organic substances in the product that are aerobically non-biodegradable (not readily biodegradable) (aNBO) and/or anaerobically non-biodegradable (anNBO) shall not exceed certain limits. Criterion "Excluded or limited substances and mixtures" prohibits some sensitive or hazardous substances as ingredients. "Packaging requirements" points out acceptable threshold weight/utility ratio (WUR) of the product. "Washing performance" is more about the

product's performance test. The applicant shall provide a test report indicating that the product fulfills the minimum requirements specified in the test. Criterion "Points" provides an indicator matrix of points. Each option has 1 or 2 points. A minimum of 3 points shall be achieved for a qualified product. Criterion "Consumer information" examines if the dosage instruction, washing recommendations, or pretreatment information are properly printed on the product's package. The last criterion "Information appearing on the EU Eco-label" is about the optional text showing on the EU Eco-label.

After reading and analyzing the criteria document for laundry detergent products, we have identified two important motivating scenarios or basic requirements concerning our ontology knowledge base. The first one is saving candidate product's detailed description. For example, some applicant wants his product to be eco-labeled, a description of the product should be provided. Product's critical physical and chemical characteristics, parameters, textual information or other specification should be instantiated in the ontology and can be queried afterwards. More technically speaking, both TBox and ABox should be preserved in the ontologies. The other important motivating scenario is judging whether some candidate product is qualified to be labeled or not. This scenario requires inference support for ontology.

Based on these two scenarios, some competency questions have been defined. We expect that the ontology to be developed can answer questions like:

- CQ1:** If this product is qualified to be eco-labeled?
- CQ2:** What is the quantitative value of this product's certain physical or chemical characteristics? (Critical dilution volume, biodegradability, weight/utility ratio, etc.)
- CQ3:** Does this product contains excluded or limited substances and mixtures?
- CQ4:** In which countries is this product being sold?
- CQ5:** What is the reference dosage per wash for this product?
- CQ6:** What is the corresponding EU Risk Phrase for some GHS Hazard Statement?
- CQ7:** What physical or chemical characteristics does some ingredient have? What are their values?

.....

One thing that draws our attention is that, among those 9 criteria, some are not suitable to be modeled in ontologies. In our research, we had expected our knowledge base to cover all the criteria, but we found that some complex criterion is difficult to be translated in ontology. Because both the syntax and semantic complexity of this criterion exceed what is allowed by OWL language. For example, the specification of consumer information (Criterion 8) has almost no quantitative parameter's requirement, instead, whether the information showing on the package is good or not is mostly subject to the judgment of human experts. As for the washing performance (Criterion 6), a test report is needed. The production of this report must be carried out by a certificated laboratory and then reviewed by human experts too. Another example is the criterion of points (Criterion 7). In this criterion, it is required to calculate the points that a candidate product accumulates. With regard to OWL 2 and SWRL which are monotonous in terms of logic, it is hard to modify an already built model or do accumulative calculation by itself. If we translate such kind of criterion into OWL ontology forcefully, we may encounter very bulky ontology structure. Because for every single points item, we may have to use a property to save the points, then a set of corresponding rules has to be translated and established to calculate the points of this item. Such efforts will greatly increase modeling complexity and affect the reasoning performance. Thus, for the sake of a better inference performance of the decision

⁷ <http://ec.europa.eu/environment/ecolabel/how-to-apply-for-eu-ecolabel.html>.

⁸ <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32011D0264>.

support process, we decide to take a trade-off strategy that criterion 1, 2, 3, 4, 5 are chosen to be translated into ontology. The rest of the criteria will be implemented by external traditional program logic, but the verification result of these criteria will be stored in the ontology knowledge base as well.

4. Terminology collection

At first, we tried to utilize some Ontology Learning techniques. After some survey work, Text2Onto [23] was chosen to be the tool that extracts ontology from the criteria document. Unexpectedly, the result of Text2Onto⁹ was not satisfactory. After parsing the criteria document in text, only about a dozen classes were identified, two object properties were identified. For the other ontology learning tools, either no download links are provided, or the tool is not runnable. Since automatic extraction of ontology did not work very well. Even we agree that automatic extraction may help in some cases [24], we decided to do it manually.

The first critical task before modeling is to identify the terminology of the ontology. Because we have the experience that once a terminology is acquired, class definition and class hierarchy will be easily retrieved from the terminology. Then, the definition of object property and data property will correspondingly become easier. In this step, we have utilized card sorting and laddering techniques that are described in [25]. Useful terms were identified and recorded when we roughly browsed the document. In this first step, both nouns and verbs were recorded. Multiple iterations were carried out to make sure we don't miss important terms. Then, we tried to group these terms into different catalogs. For example, "preservative", "fragrance", "stabilizer", "coloring agent", "substance", and "solvent" describe things in the same field, so they should be cataloged into a same group. Next step, we put these grouped terms into "ladders". In other words, terms were organized by "is-a" relationship in hierarchy structure and this structure became the prototype modeling of our ontology. In the previous example, "substance" has a more generic meaning, then it was laddered in a higher level than the others in the hierarchy; the other terms associated it through "is-a" relation in the lower level. At last, a review to all the selected terms were conducted with domain expert making sure the modeling is complete.

5. A modularized modeling

Since we already have a prototype modeling of the ontology composed of the selected terminology. Here in this step, we should translate the modeling into specific ontology syntax.¹⁰ The axioms of class, properties, and individuals should be inserted. Put it more vividly, the output of terminology collection is more like building a skeleton of the ontology; the modeling in this step is closer to enrich the ontology with flesh and blood. As we have stated in the beginning of this paper, a very important issue of our research is "reuse". In pursuit of better re-usability, we propose a modularized methodology to separate the entity model (static conceptualization) and rule model (dynamic conceptualization). In other words, we should identify in which part the knowledge about laundry detergent is relatively constant, and in which part frequent changes may take place. As a result of this, in Fig. 2, we have two kinds of modules: one is the entity module with solid border line, which represents the relative static conceptualization; the

other is the rule module with dotted border line, which represents more dynamic criterion rules that relay on entity module.

In our design, still in Fig. 2, the main module named *laundry_detergent* contains generic concepts, roles and individuals of the domain. For the other more generic entities, module *laundry_detergent* reaches to them via dependencies. In OWL 2 scheme, we can implement this dependency by using *import* syntax, which means an ontology will use all those concepts and relationships from the imported ontology. For our laundry detergent product group, we have entity module *iso_standards*, which contains all the ISO standards references; *ghs_hazard_statement*, in which stores all the hazard statements and codes of GHS (Globally Harmonized System of Classification and Labeling of Chemicals); *regulation_european_commission*, where stores all the European Commission regulation reference; *european_risk_phrases*, where all relevant European risk phrases of chemicals are listed; *commission_decision*, which refers to all relevant European Commission decision documents; *didlist*, which is a database for detergent ingredients. As we have put them into independent modules, they are easier to be imported and reused by other domain ontologies. Please note that, although the main module *laundry_detergent* imports these sub-modules, it does not mean that *laundry_detergent* need all the content in them. Maybe only a part or even a very small part of content is useful for the upper-level modules.

5.1. Module Laundry_detergent

This module is the skeleton of the laundry detergent domain ontology. Almost all the important domain concepts and relationships are defined in this ontology module. Fig. 3, a class diagram in UML illustrates the main classes defined in this module. On the right side of the diagram, we can find a hierarchy of the candidate laundry detergent product and there are five kinds of laundry detergents that are concerned in this criterion: *color safe detergent*, *heavy duty detergent*, *low duty detergent*, *fabric softener*, and *stain remover*. The core candidate laundry detergent class is associated with several other parameter classes via object properties. These properties or relations are developed from the verbs that are identified in the terminology collection process. Object properties are important part for a complete laundry detergent product profile. These object properties link the other parameter class to the core candidate laundry detergent product class. As illustrated in Fig. 3, each instance of candidate laundry detergent must have at least one kind of chemical as ingredient. It is required to specify the manufacturer of the product, the countries where it will be sold, and the product type. Each candidate laundry detergent should also be associated with one and only one parameter instance for the critical dilution volume, reference dosage, weight utility ratio, aerobically non-biodegradability, and anaerobically non-biodegradability. For each parameter class, a data property *hasValue* has been defined in order to assign concrete value to different parameters. *hasFunctionalUnit* is defined to specify various kinds of functional units (e.g. g/kg wash, ml/kg wash) for this concrete value. If some parameter value of a candidate laundry detergent doesn't comply to the criteria, it will be cataloged into the rejected detergent class.

5.2. Module Didlist

This module is the conceptualization of the detergent ingredient database. In EU Eco-label laundry detergent product criteria, this database is recorded in an excel file, which is not very convenient to be used in applications or other software systems. This module is interesting because it will be reused in other product group criteria. We have developed an excel scanner to read

⁹ The version we used is here <http://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/text2onto/text2onto-071109.zip>.

¹⁰ The laundry detergent criteria ontology can be accessed on Github: <http://github.com/xudadd/EU-Ecolabel-laundry-detergent-product-criteria-ontology>.

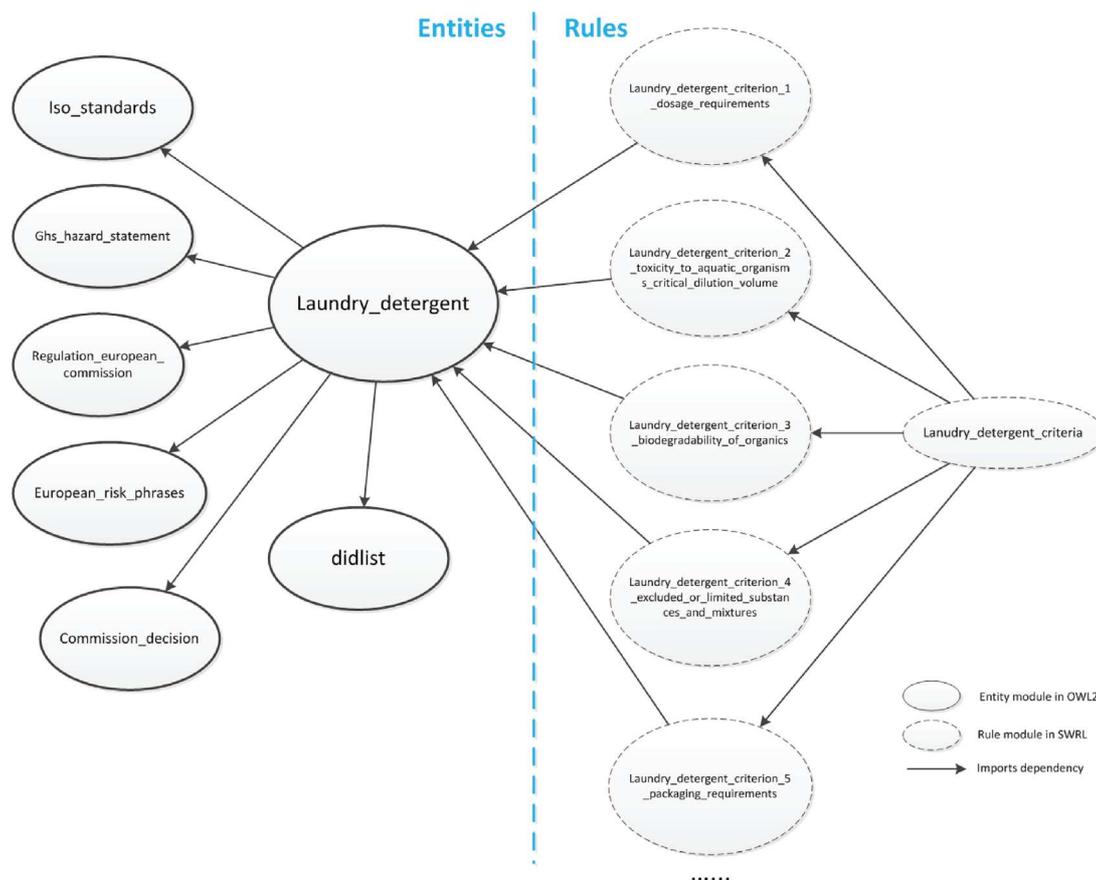


Fig. 2. Ontology modularization schema for EU Eco-label laundry detergent product group.

this excel file, then generated this module as OWL files. Fig. 4 is the representation of this module in UML class diagram. In this module, all the detergent ingredients are sub-classified into groups: amphoteric surfactants, anionic surfactant, cationic surfactant, non-ionic surfactant, preservative, and other ingredients. Various functional units are identified by scanning the whole excel file. Full name label and annotation are attached to each of them accordingly. Each ingredient has one and only one anaerobic degradation characteristic e.g. “N” means anaerobically not biodegradable; Each ingredient has one and only one kind of aerobic degradation characteristic e.g. “I” means aerobically inherently biodegradable, but not readily biodegradable.

5.3. Module *European_risk_phrases*

This module covers all the European Risk Phrases specification. Since European Risk Phrases is an external standardization reference that appears in criterion 4, it’s better to keep these specifications to be an independent module. Most of this module is the risk phrase individuals. Each risk phrase individual has two data property assertions, e.g. individual “R49” *hasRiskCode* “R49”; *hasPhraseStatement* “may cause cancer by inhalation”. This module is reusable in other EU Eco-label product group.

5.4. Module *Ghs_hazard_statement*

Similar to previous module *European_risk_phrases*, GHS (Globally Harmonized System of Classification and Labelling of Chemicals) is also an external reference in criterion 4. A mapping between GHS statement and European Risk Phrases is presented in this criterion. A module following almost the same pattern as

module *European_risk_phrases* is modularized. Most of this module is hazard statement individuals. Each hazard statement individual has two data property assertions, e.g. individual “H261” *hasHazardCode* “H261”; *hasHazardStatement* “In contact with water releases flammable gases”. This module can be reused in the other EU Eco-label product groups, like all-purpose cleaners, cosmetic products.

5.5. Module *Iso_standards*, *Regulation_european_commission*, and *Commission_decision*

These modules store the external documentation reference. They record relevant EU documents, standard, commission decision or regulations that are referred in this detergent laundry criteria. These dependency and references contribute to a better understanding of the criteria in a bigger picture. Fig. 5 presents the structure of these three modules. Each of these individuals is equipped with URLs that link to external resources. These three modules can also be reused and supplemented by other domains and other EU Eco-label product groups.

Several advantages exist in this modularized design. As more coherent concepts and relationships are gathered together to form modules, it’ll be easier to manage knowledge and data in large scale. Complex conceptualization can be achieved by integrating multiple small modules. Also, it’s easier to configure and replace modules rather than to make slight changes directly in a large structure. Take the same example in Fig. 2. We have a general conceptualization of laundry detergent product which is stored in domain module *laundry_detergent*. This major ontology module can be replaced by other modules describing other product groups while still making

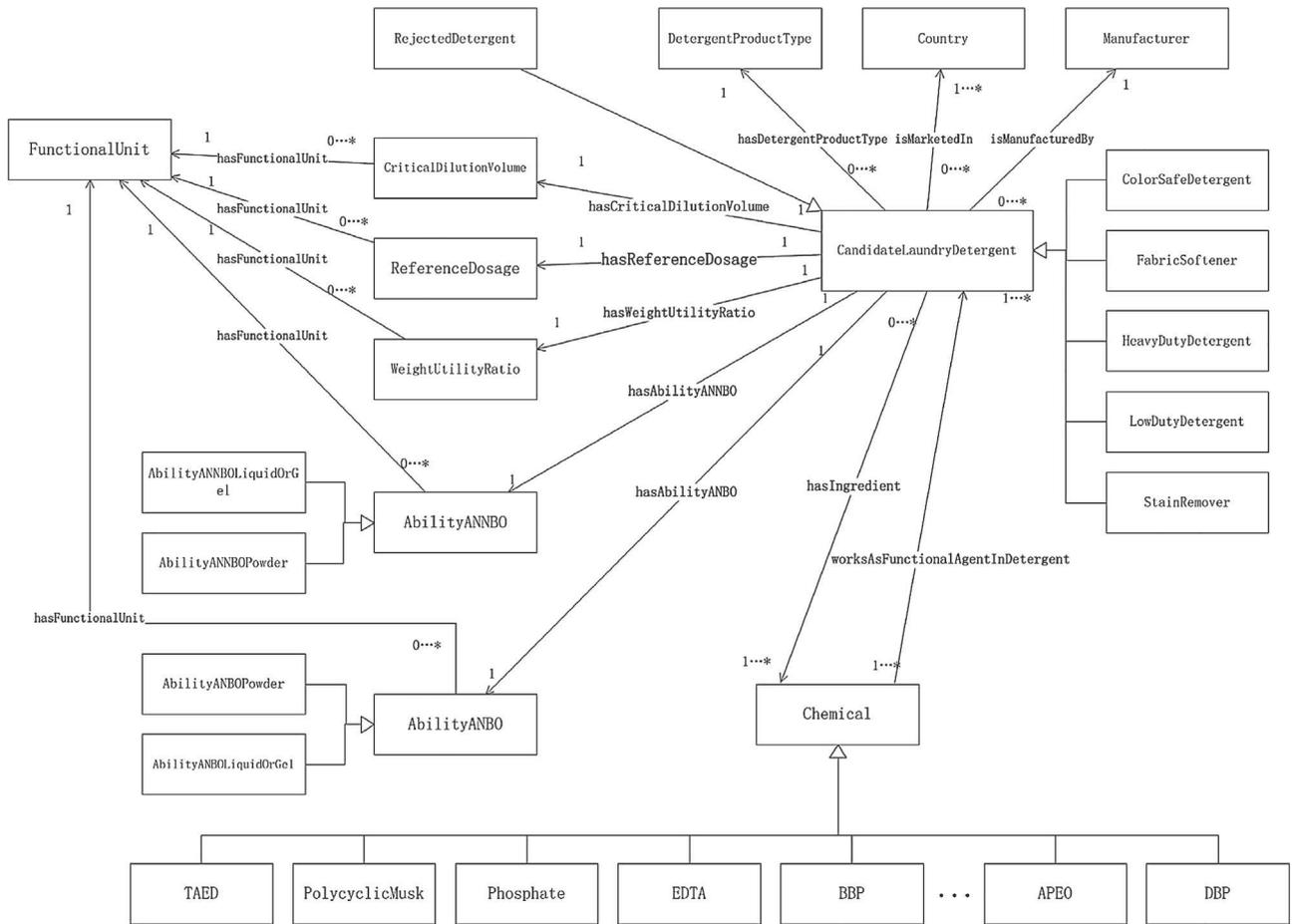


Fig. 3. Structure of Module *Laundry_detergent* presented in UML.

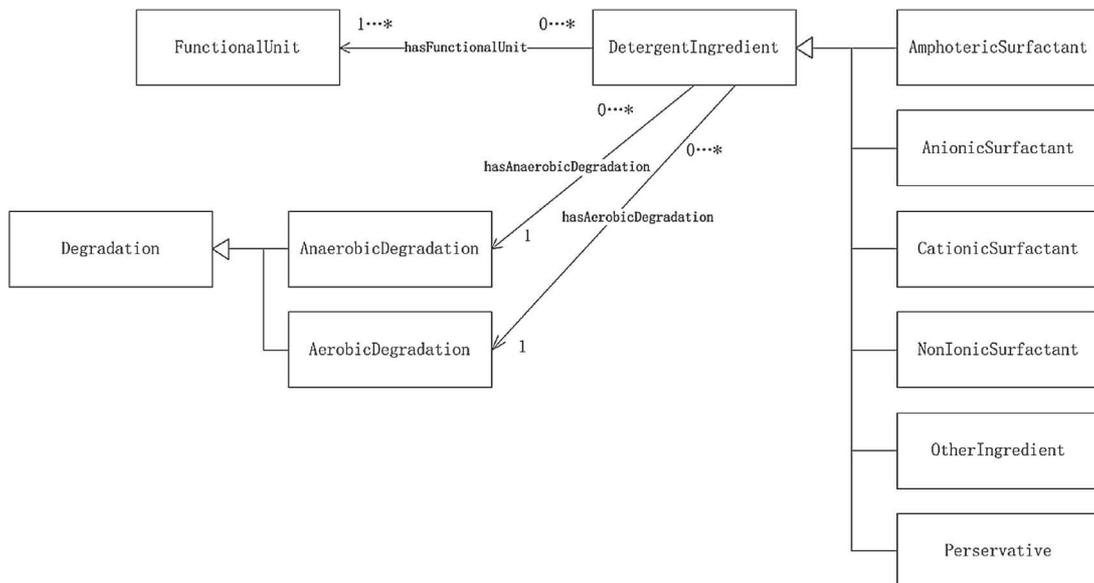


Fig. 4. Structure of Module *Didist* presented in UML.

use of sub-modules like *didlist*, *ghs_hazard_statement* and *european_risk_phrases*, etc. This actually happens in at least two other product groups “rinse-off cosmetic products” and “all-purpose cleaners and sanitary cleaners” which use the same detergent ingredient database (Fig. 6). Re-usability is achieved by extracting

the common knowledge module and have it shared between domain ontologies.

Modularization implies separation of conceptualization. In our case, we can see that it will be practical to extract rules from ontology modules. In other words, it's better to keep subjective

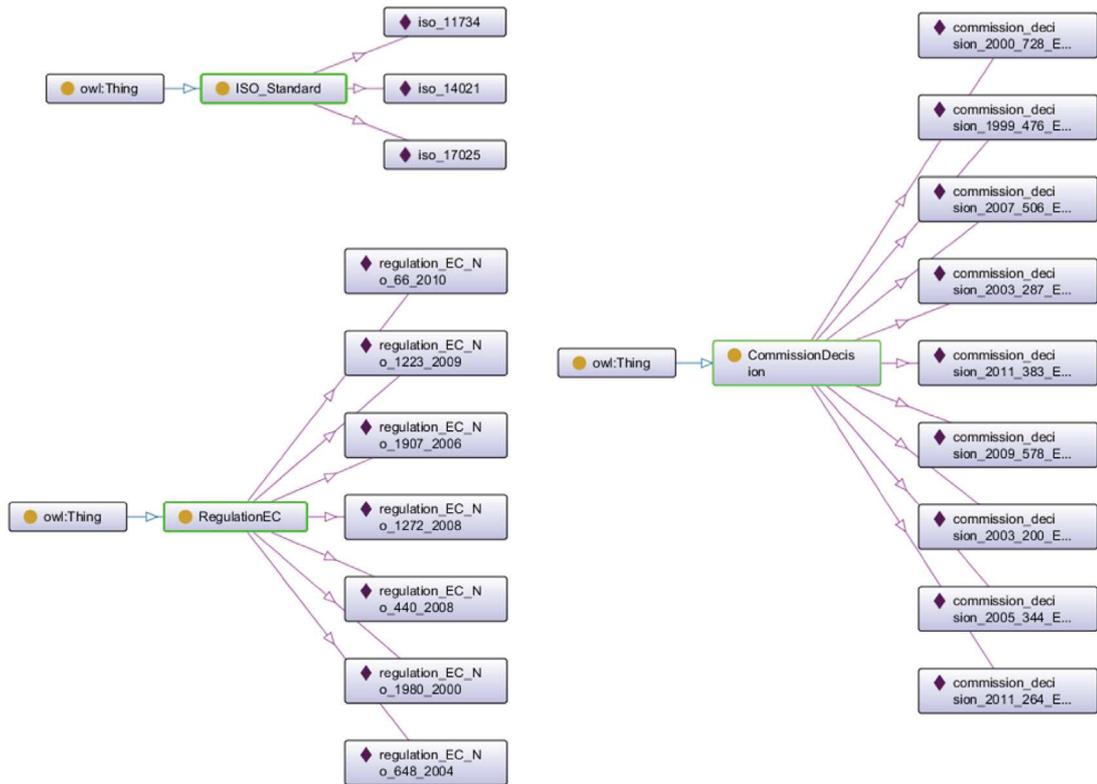


Fig. 5. Structure of module *Iso_standards*, *Regulation_european_commission* and *Commission_decision* illustrated in OntoGraf Protégé plug-in.

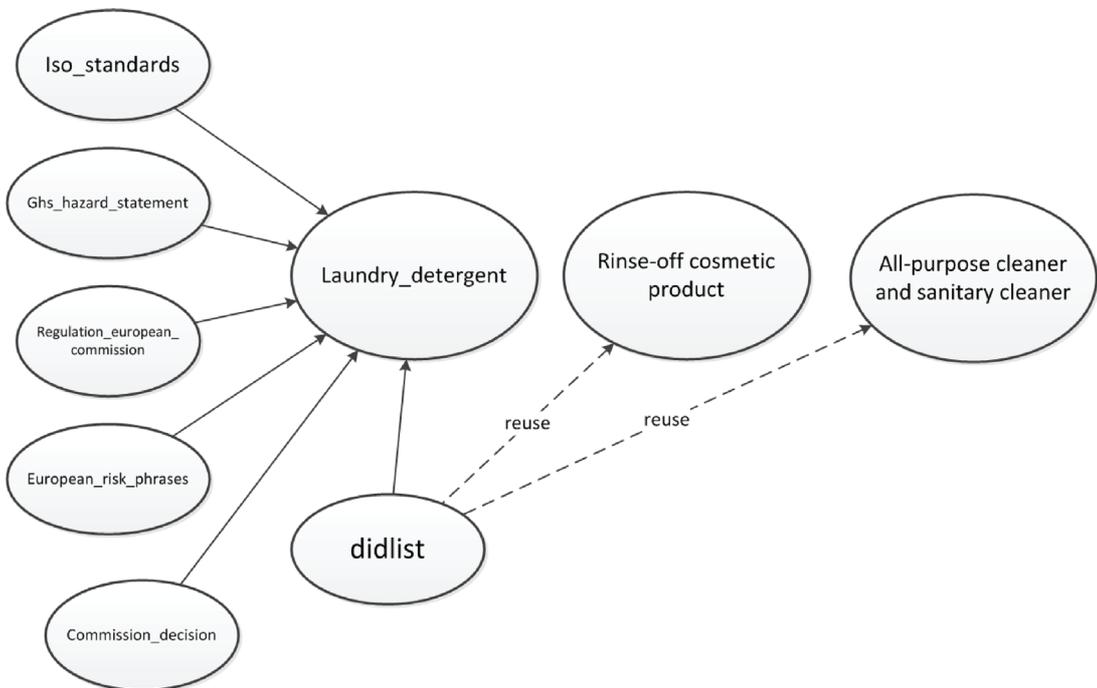


Fig. 6. Basic reuse pattern which happens between laundry detergents, rinse-off cosmetic products and all-purpose cleaners and sanitary cleaners.

constraints and world description separated. We call this the separation of rules and entities. For example, in the detergent ontology shown in Fig. 2, ontologies represented in ellipses with solid borders are concept-centered, which means the main function of these ontology is to describe the concrete world. These ontologies contains concepts and relationships that are

meant to describe or record the facts about the real world. On the other hand, as for a product group's guideline or criteria, quite much of this information is involved with human objectives. They are the rules and willingness that human beings impose to the world. Generally speaking, the description of the concrete world does not change as much as human's subjective willingness and

rules. In our research, we implement such separation between rules and entities in order to loose the coupling between these two aspects, and then realize a better reusability. This separation of rules and entities is a significant difference between our modularization method and previous ones.

For detergent products, the concentration of different chemical ingredients has to comply with certain limit and standard. We can hardly say that such goal-oriented specification is plain description of the world. Moreover, such rules may change time after time. This actually happens, because the product guideline keeps being updated as EU Commission keeps generating new amendments or revise. In our approach, we have each criterion item be an independent module (not completely independent actually, as these rule modules may also have dependencies to other external or internal ontology modules). For example, each of the 5 criterion of the laundry detergent product group is made into an independent OWL file. In the OWL file, firstly, the fundamental entity modules are imported (in Fig. 2, module hierarchy whose root is *laundry_detergent* is imported by all the five criterion), then SWRL rule axioms are inserted. As each criterion is distributed in its corresponding module alone, we can easily replace them with new rules and manage them in a configurable way without impacting the others.

At last but not the least, for the criteria ontology as a whole, an entry module is introduced to include all the criteria, e.g. the *laundry_detergent_criteria* module on the right side of Fig. 2. For applications, once the ontology entry is provided, the whole ontology composed of all the entity and rule modules will be retrieved. With this configurable design, expansion and alteration to the ontology will be easier. For example, when a new criterion is about to be approved by the commission, in Fig. 7, we can update the product criteria to a new version by adding a new rule module and new entry called *Laundry_detergent_criteria_2.0* without losing trace of the previous one. The newly added rule module could be

about another new criterion or just an update version of existent criteria. The removal of certain module is similar, all we need is to introduce another entry module. For example, if the new entry module imports criterion 2, 3, 4, 5, thus criterion 1 will be removed from this version of criteria ontology.

In this subsection, we've introduced a modularized modeling of EU Eco-label laundry detergent product criteria. The separation of entity modules and rules is one of the major contributions of our work. The main reference sources used in this work are the official criteria documentation for laundry detergent product group. This documentation consists of multiple PDF files (about forty pages in all). Two developers and an expert in eco-labeling (Certification engineer for Detergent products in a private company) are involved in the modeling and development process. For the requirement analysis process, we have conducted a careful reading of all the documentation which took about one week of time. The terminology collection process took about three weeks. The module partition and entity module modeling took us about four weeks. Then we almost spend double time i.e. 8 weeks for the translation and modeling of the SWRL rules. For each step we adopted an agile methodology in which we used DevOPS practice (Development and Operations) [26]. We initiate what we called DevExp (Development and expert) loop through feedback from expert to developers. The goal is to amplify the feedback loop so that the process is swift and seamless. The feedback loop led to an increase in efficiency of the ontology construction. The role of the expert was to check the output, identify problems if any and validate each step.

6. Reasoning and argumentation

A considerable advantage of using OWL ontology is that the underlying DL (Description Logic) formalism allows reasoning.

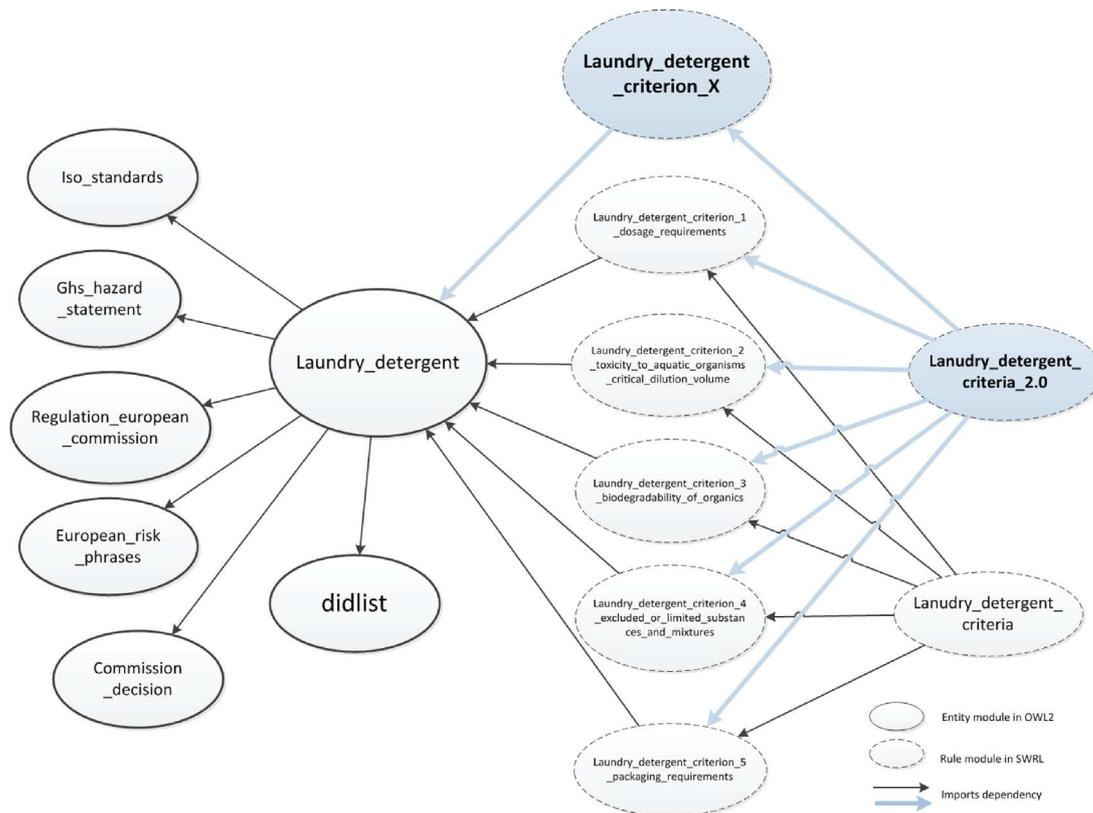


Fig. 7. Ontology expansion by adding new rule module of criteria for detergent product group.

Product type	Dosage, powder/tablet	Dosage, liquid/gel
Heavy-duty laundry detergent, Colour-safe detergent	17,0 g/kg wash	17,0 ml/kg wash
Low-duty laundry detergent	17,0 g/kg wash	17,0 ml/kg wash
Stain remover (pre-treatment only)	2,7 g/kg wash	2,7 ml/kg wash (*)

(*) Estimated average dose to be used in CDV calculations. Actual dosing will depend on number of stains in any given wash-load. The estimated dose is based on a dosage of 2 ml per application and 6 applications per wash-load of 4,5 kg (liquid stain remover).

Fig. 8. Criterion of dosage requirements. The recommended dosage for each wash shall not exceed the amounts above.

Actually, the DL computation complexity and the development of reasoners are very important research issues for ontology and ontology engineering. Investigating the trade-off between the expressivity of DLs and the complexity of their inference problems has been one of the most important issues in DL research [27]. As for the expressiveness of OWL, it's mostly related to the underlying DL expressiveness. Concerning the latest W3C discussion, there are three OWL schemes on different levels of expressiveness: OWL-Lite, OWL-DL, OWL-Full [28]. OWL-DL is best supported by reasoners because it is decidable which means a proper trade-off between reasoning performance and expressivity. A famous algorithm implemented by many modern DL reasoners is called Tableau Algorithm [27].

In Protégé editor, several third-party reasoners have been developed as plug-ins. In fact, today's reasoners can also stand along as APIs or even independent tools. Since Protégé is an open source project, for almost all its reasoner plug-ins, we can find APIs that can be integrated into programming language like Java or C++. Some common reasoners for Protégé (the version we used is Protégé 5.0.0 beta 24): Fact++ is a sound and complete reasoner for SHOIQ (the same description logic underlying OWL-DL) [29]. Pellet is also a sound and complete reasoner that is said to support E-Connections and that would be very interesting for our research [30]. Hermit [31] works best with our ontology knowledge base as for the SWRL rules, all the reasoning tasks involved in this paper is completed by Hermit (The version we used is 1.3.8.413).

Now, let's have a look at the SWRL rules. As stated in previous sections, after checking all the criteria in the laundry detergent product, we found that only the first five criteria are proper to be translated into SWRL rules. The main function or objective of SWRL rules is for determining whether a candidate should be rejected or accepted. They are manually translated by ontology developers. For the check and validation of these rules, reasoner will be used to see if there is syntax or variable errors. If errors or inconsistency exist in the rules, reasoning process will be blocked. For the final validation of the rules, we will apply a reasoning comparison. Besides the rules and reasoning process, we will conduct manual evaluation in which human beings read the criteria documentation and check the product's profile, then compare the reasoning result with the manual evaluation result. If they have the same results, it proves that the rules have been correctly translated and modeled.

More specifically, take the first criterion for example, it is about the recommended dosage of detergent for each wash. The details of this criterion is shown in Fig. 8. For each type of product, since the value for "powder/tablet" and "liquid/gel" is the same, we merge the two requirements into one. In Protégé, the SWRL rules are edited in a tab as in Fig. 9. It's written in the popular Manchester Syntax [32]. Please note that here we have another advantage of dividing rules into modules. For that reasoning is a pretty costly computation task, it will be interesting to make reasoning separate and distributed. In our modularization of ontologies, by putting SWRL rules in different modules, unnecessary interference

between rules is avoided. For example, some domain experts finish editing criterion No.1 and he wants some test, all he needs to do is to choose the rule module of criterion No.1 and start the reasoner. The reasoning will be based only on criterion No.1 because the other criteria rules are stored in the other rule modules and are exempted from current ontology composition and test.

We assume the readers have basic ideas about the syntax and semantics of SWRL. (A good reference of SWRL specification can be found on the W3C web site¹¹) The basic idea for the criteria rules is introducing two concepts called *RejectedDetergent* and *CandidateLaundryDetergent*. As long as the profile of some detergent product doesn't comply with the criteria rules, this product should be classified as an individual or instance of *RejectedDetergent*. In other words, this class can be treated as the "objective" of the reasoning task. In the beginning of the eco-labeling reasoning process, we input product profile as individual of the *CandidateLaundryDetergent* class, once the reasoning process is started, criteria rules are applied upon it. After the reasoning, if an product individual is classified under the class of *RejectedDetergent*, then we assert that this product doesn't comply with EU Eco-label criteria.

In practice, the criteria ontology will work somehow like a template for real laundry detergent product profile. A new product profile imports the criteria ontology entry (module *Laundry_detergent_criteria* in Fig. 2 for example), then a detergent product profile ontology is constructed according to the pre-defined specification in the criteria ontology. After the reasoning, all the reasoning and inference result of this profile ontology will be stored in our knowledge base as reference cases for further reuse or review. Thus, the knowledge base will be composed mainly of two parts: a criteria ontology repository that stores all kinds of EU Eco-labeling products' criteria in modularized ontologies; and a historical case repository that reserves all the product profiles' reasoning results.

In the rest part of this section, a simple product profile example will be presented to show how explanation is generated at the end of reasoning. Typically, we save a candidate detergent profile in Manchester Syntax¹² which is illustrated in Fig. 10. One product could be marketed across several different European countries at the same time and this fact is expressed in the axiom expression *isMarketedIn min 1 Country*. *isMarketedIn* is an object property that we've defined and *min* is the restriction type which means the cardinality of this property is at least one. Table 1 shows the product's parameters in detail.

Compared with the criteria value, the two known ingredients in Table 1 don't have any hazard code, neither are they in the list of excluded or limited substances. This means they are good to be added into laundry detergent products. However, some of this product's parameter value exceeds the criteria value, e.g.

¹¹ <http://www.w3.org/Submission/SWRL/>.

¹² <http://www.w3.org/TR/owl2-manchester-syntax/>.

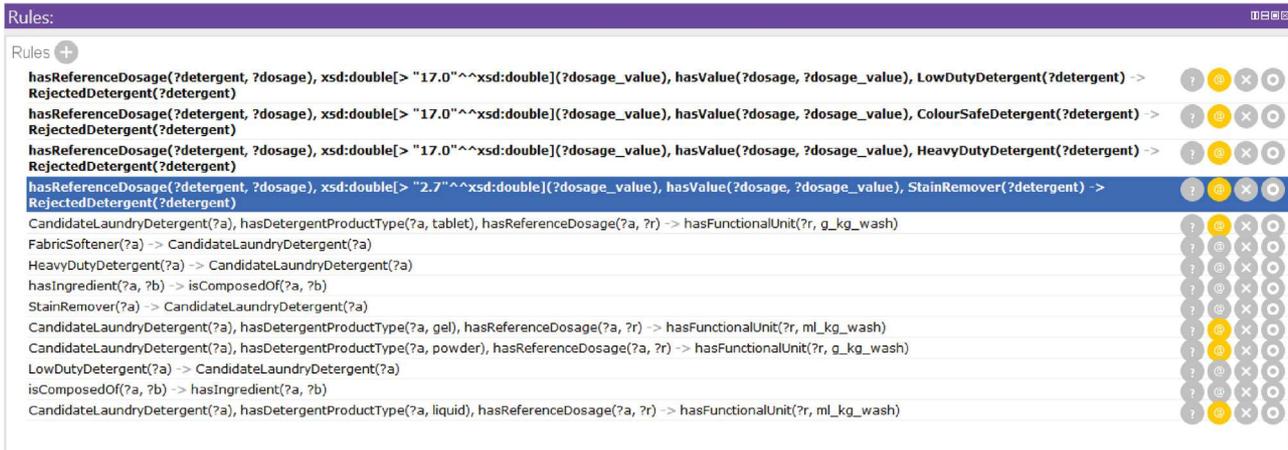


Fig. 9. SWRL rules edited in Protégé editor.

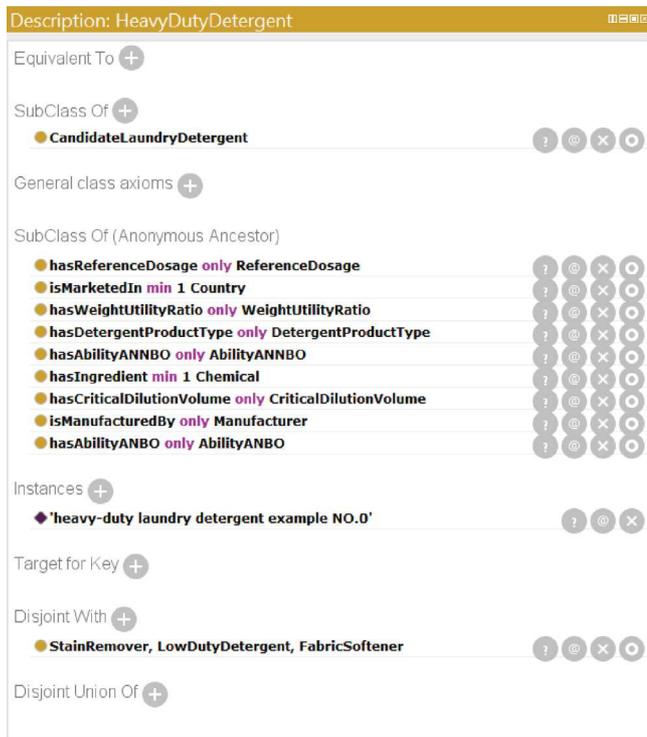


Fig. 10. Core concept *HeavyDutyDetergent* defined in Manchester Syntax axioms in Protégé editor.

recommended dosage and weight utility ratio, so it should be considered as a *RejectedDetergent*. After launching Hermit reasoner, we can get an inferred class hierarchy shown on the left side of Fig. 11. We find that our example individual *heavy-duty laundry detergent example NO.0* has been classified under the concept *RejectedDetergent*. All the axioms with yellow background color indicates that they are inferred by the reasoner. If we click on the small question mark suited on the right side of each newly inferred axiom, we can check the explanations of how the reasoner reaches to this new inference. Fig. 12 shows several explanation items for this new inference result and why our example product profile is classified into the *RejectedDetergent*. From explanation No.1, we can see that our example breaks the rule of recommended dosage and explanation No.2 is about the weight utility ratio rule. In our case, there are 15 explanation items found. If we scroll down in the window shown in Fig. 12, we can find all the others.

7. Evaluation and analysis

The laundry detergent ontology is the first criteria ontology that we have developed for EU Eco-labeling. Another two important criteria ontologies about rinse-off cosmetic product and all-purpose cleaner are under development. All these ontologies will be included in a knowledge base framework. Adjustment and improvement in favor of global performance are being taken into account. Evaluation of single ontology and the whole knowledge base is also undergoing. The advantage of the design of modularization and separation has been observed by researchers as module *Didlist*, module *European_risk_phrases*, etc. can be directly reused by newly developed ontologies.

As we have presented in requirement analysis section, a very important motivation of this ontology development is to judge

Table 1 Detailed parameters of product profile example: *heavy-duty laundry detergent example NO.0*.

Property parameter	Value	Criteria value
Product type	Liquid	
Recommended dosage (reference dosage)	20.0 ml/kg wash	17.0 ml/kg wash
Sales country	France	
Weight utility ratio (WUR)	2.0 g/kg wash	1.5 g/kg wash
Critical dilution volume (CDV)	30,000.0 l/kg wash	35,000.0 l/kg wash
Aerobically non-biodegradability (aNBO)	0.5 g/kg wash	0.55 g/kg wash
Anaerobically non-biodegradability (aNBO)	0.5 g/kg wash	0.7 g/kg wash
Known ingredient	Acetic acid; C8-18-Amphoacetates	

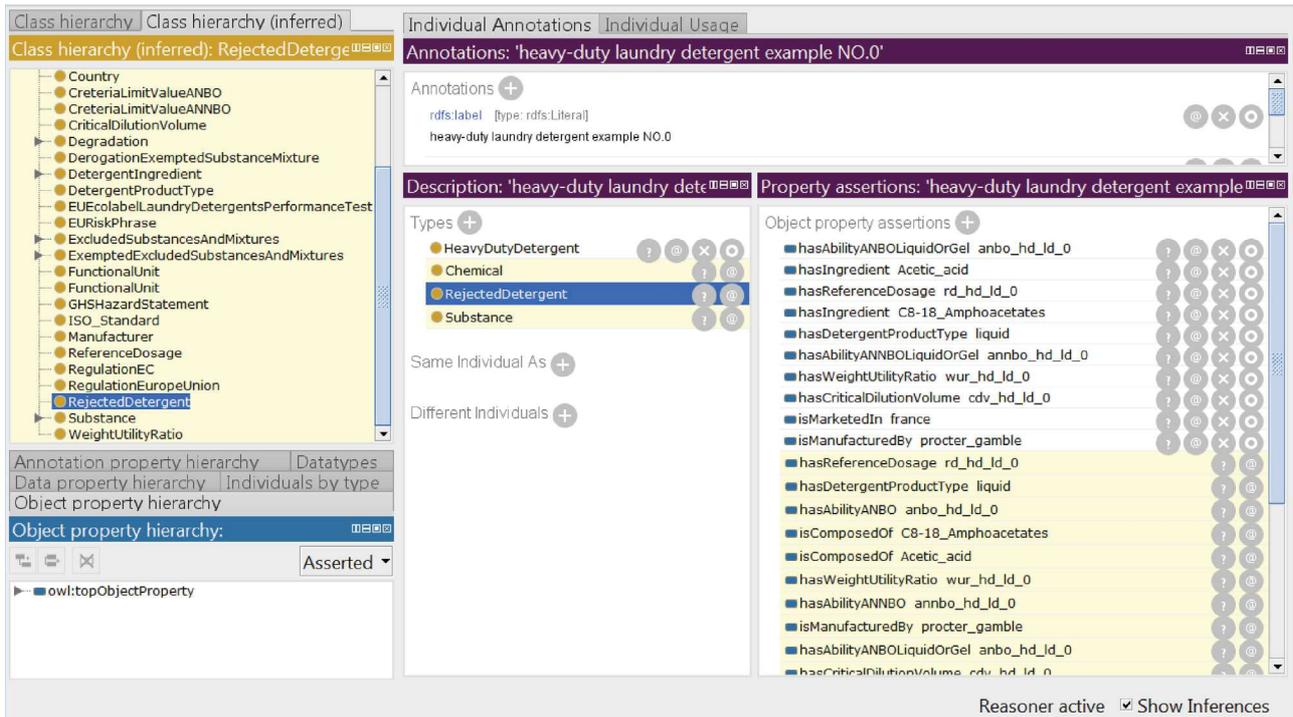


Fig. 11. Reasoning result shown in Protégé editor.

whether a candidate product is qualified to be labeled. According to a classification of ontology evaluation approaches in [33], our evaluation approach is closer to an application-based evaluation method, i.e. using the ontology in an application and evaluating the results. We have seen that this laundry detergent criteria ontology is successfully applied in a decision-support system in [34]. Synthetically, taking into account the criteria and aspects introduced in [35] and [36], we have evaluation result as following:

Syntax. The criteria ontology is described in standard OWL syntax.

Semantics. Since SWRL rules are defined in the ontology and inference support is a basic requirement of our ontology, multiple reasoners e.g. Fact++, Hermit, and Pellet have been applied to check and verify the semantic consistency. So, the ontology is always logically consistent.

Vocabulary. Almost all the classes, properties and individuals in the ontology have a meaningful identifier which follows Camel case naming pattern. For those entities that have abbreviation names and vague meaning names e.g. *CDV* and *H400*, a *rdfs:label* axiom is added as complement.

Structure. The structure of our ontology is relatively simple, the depth of both class hierarchy and property hierarchy is not more than two. The most out degree for an individual that reaches to other individuals via properties is 14. Taking all the modules into account, 68 classes, 46 object properties, 21 data properties and 460 individuals are defined and stored in our ontology. The number of total axioms is 5786. DL expressivity is *ALCHQ(D)*. Our ontologies can be easily understood and manipulated by other knowledge engineers.

Documentation. Each module of the ontology has a textual annotation. For those key terms that come from specific domain glossaries, textual annotation and external links are provided. For every SWRL rule, annotation as well as the corresponding anchor position in the document is indicated.

As regards to more specific validation, the competency questions that are defined in requirement analysis section have been translated into SPARQL queries. They work fine with our

ontology and correct result can be queried. Here are two examples as listed below (Figs. 13 and 14).

CQ1: If this product is qualified to be eco-labeled?

CQ2: What is the value of this product's certain physical or chemical characteristics? (critical dilution volume, biodegradability, weight/utility ratio, etc.)

Besides this intuitive evaluation, we have also applied a more systematic evaluation method that is presented in [37]. Three types of evaluation measures have been identified and considered: structural measures, functional measures, and usability-profiling measures. In practice, more principles and parameters are used to reflect the quality of ontology:

- a. Cognitive ergonomics:
 - Depth: Maximum 3.
 - Breadth: Maximum 12 for classes; maximum 115 for individuals. Tangledness: Low.
 - Class/property ratio: 1.01 (68/67).
 - Annotations: 49.
 - Anonymous classes: None.
- b. Transparency:
 - Modularity design: 12 modules (7 entity modules and 5 rule modules).
 - Axiom/class ratio: 85.09 (5786/68).
 - Patterns: No.
 - Specific differences: No.
 - Accuracy: Good.
 - Complexity: Medium.
- c. Computational integrity and efficiency:
 - Logical consistency: Good.
 - Disjointness ratio: 0.97 (66/68).
 - Restrictions: Well defined and annotated.
 - Cycles: None.
- d. Meta-level integrity:
 - Meta-level consistency: Good.
 - Tangledness: Low.
- e. Flexibility:

Show regular justifications All justifications
 Show laconic justifications Limit justifications to

Explanation 1 Display laconic explanation

Explanation for: 'heavy-duty laundry detergent example NO.0' Type RejectedDetergent

- 'heavy-duty laundry detergent example NO.0' Type HeavyDutyDetergent In 2 other justifications ?
- 'heavy-duty laundry detergent example NO.0' hasReferenceDosage rd_hd_ld_0 In 1 other justifications ?
- hasReferenceDosage(?detergent, ?dosage), xsd:double[> "17.0"^^xsd:double](?dosage_value), hasValue(?dosage_value, ?dosage_value), HeavyDutyDetergent(?detergent) -> RejectedDetergent(?detergent) In NO other justifications ?
- rd_hd_ld_0 hasValue "20.0"^^xsd:double In NO other justifications ?
- EquivalentProperties: hasValue, hasValue** In ALL other justifications ?

Explanation 2 Display laconic explanation

Explanation for: 'heavy-duty laundry detergent example NO.0' Type RejectedDetergent

- 'heavy-duty laundry detergent example NO.0' hasDetergentProductType liquid In 13 other justifications ?
- 'heavy-duty laundry detergent example NO.0' hasWeightUtilityRatio wur_hd_ld_0 In 13 other justifications ?
- wur_hd_ld_0 hasValue "2.0"^^xsd:double In 13 other justifications ?
- hasDetergentProductType **Domain** CandidateLaundryDetergent In NO other justifications ?
- CandidateLaundryDetergent(?a), xsd:double[> "1.5"^^xsd:double](?wur_value), hasDetergentProductType(?a, liquid), hasWeightUtilityRatio(?a, ?wur), hasValue(?wur, ?wur_value) -> RejectedDetergent(?a) In 13 other justifications ?
- EquivalentProperties: hasValue, hasValue** In ALL other justifications ?

Explanation 3 Display laconic explanation

Explanation for: 'heavy-duty laundry detergent example NO.0' Type RejectedDetergent

- 'heavy-duty laundry detergent example NO.0' hasDetergentProductType liquid In 13 other justifications ?
- 'heavy-duty laundry detergent example NO.0' hasWeightUtilityRatio wur_hd_ld_0 In 13 other justifications ?
- wur_hd_ld_0 hasValue "2.0"^^xsd:double In 13 other justifications ?
- hasWeightUtilityRatio **Domain** CandidateLaundryDetergent In NO other justifications ?
- CandidateLaundryDetergent(?a), xsd:double[> "1.5"^^xsd:double](?wur_value), hasDetergentProductType(?a, liquid), hasWeightUtilityRatio(?a, ?wur), hasValue(?wur, ?wur_value) -> RejectedDetergent(?a) In 13 other justifications ?
- EquivalentProperties: hasValue, hasValue** In ALL other justifications ?

Explanation 4 Display laconic explanation

Explanation for: 'heavy-duty laundry detergent example NO.0' Type RejectedDetergent

- 'heavy-duty laundry detergent example NO.0' hasDetergentProductType liquid In 13 other justifications ?
- 'heavy-duty laundry detergent example NO.0' hasWeightUtilityRatio wur_hd_ld_0 In 13 other justifications ?

OK

Fig. 12. Reasoner's explanation to why heavy-duty laundry detergent example NO.0 is not good.

Snap SPARQL Query:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ld: <http://www.enit.fr/xuda/ontologies/laundry_detergent#>

SELECT ?subject
WHERE {
  ?subject rdf:type ld:RejectedDetergent.
}
  
```

Execute

?subject
ld:hd_ld_example_0

1 results

Fig. 13. Query answer for CQ1, the result shows the ID of product.

- Modularity: Good.
 Partitioning: Functional partition and entity&rule separation.
 Context-boundedness: Unknown.
- f. Compliance to expertise:

- Precision: Medium.
 Recall: Good.
 Accuracy: Good.
- g. Compliance to procedures for mapping, extension, integration, adaptation:
 Accuracy: Good.
 Recognition annotations (esp. lexical): 16.
 Modularity: Excellent.
 Tangledness: Low.
- h. Organizational fitness:
 Organizational design annotations: 13.
 Commercial/legal annotations: None.
 User satisfaction: Good.

From the evaluation results, we know that our ontology is competent for the laundry detergent product evaluation task. In spite of that, there are still some aspects that need improvement, e.g. there is no existent patterns reused in our ontology design and more annotations are still needed. If reused in other contexts, how would our ontology and modules react is still unknown. We will keep work on these drawbacks in the future.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX Id: <http://www.enit.fr/xuda/ontologies/laundry_detergent#>
SELECT ?subject ?property ?value
WHERE {
    ?subject rdf:type ?subType.
    ?subType rdfs:subClassOf Id:CandidateLaundryDetergent.
    ?subject ?property ?object.
    ?object Id:hasValue ?value.
}

```

subject	property	value
'heavy-duty laundry detergent example NO.0'	hasCriticalDilutionVolume	"30000.0"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasReferenceDosage	"20.0"^^<http://www.w3.org/2001/XMLSchema#double>
test_2_detergent	hasCriticalDilutionVolume	"20000.0"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasAbilityANBOLiquidOrGel	"0.5"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasWeightUtilityRatio	"2.0"^^<http://www.w3.org/2001/XMLSchema#double>
'heavy-duty laundry detergent example NO.0'	hasAbilityANNBOLiquidOrGel	"0.5"^^<http://www.w3.org/2001/XMLSchema#double>

Fig. 14. Query answer for CQ2.

8. Experience and lessons learned

By developing this modularized ontology knowledge base, we have acquired some interesting experience and lessons about ontology design and application. As far as we can see, people have been trying to build more and more complex knowledge representation. If we take documents, which are written in whatever language, as a model or representation of knowledge. To some extent, developing ontology is like a translation process that translates models of human language to formal knowledge representation which can be accessible by machines. As the expressiveness of human language is very high, a computable modeling and translating scheme that has competent expressiveness is needed. The expressiveness and modeling complexity of ontology language has been increasing. We can see this from the evolution of OWL to OWL 2. It is also observed that, in the early days of ontology research, simple knowledge content e.g. medical terminology often used to be the object of study. Today, complex documents e.g. specifications, legal terms, executive orders are expected to be made into ontology. In order to handle more complex knowledge representation or modeling in human language, more comprehensive consideration should be taken into account. The entity-rule separation pattern as well as modularization is such kinds of consideration and exploration that try to handle such more and more sophisticated modeling tasks. As we have discussed in the beginning of Section 5, descriptive entity-related knowledge is relatively constant which

means they don't change very much. While, the subjective rule-related knowledge part could be altered frequently. We put them in separation in order to better manage and control the change. The philosophy generalized from this entity-rule separation and modularization pattern can be applied into other modeling or application domain. When dealing with criteria alike knowledge representation, we can apply this entity-rule separation pattern to model descriptive entity-related knowledge and subjective rule-related knowledge into different models, which will facilitate reuse and maintenance.

Fig. 15 is a more detailed mind map specification for the application of this entity-rule separation pattern. The point of our learned lesson is that before diving into the concrete modeling, higher level abstraction and conceptualization should take precedence. In our case, the target documentation is the Eco-labeling criteria. According to the characteristics of the document and the domain knowledge, modularization scheme based on the entity-rule separation pattern is proposed. Then, in each module, the concrete modeling and potential reuse proceed. However, in reality, the boundary of each task could not be very clear. For example, reusability is a very important factor when we decide to set up *Didlist*, *Ghs_hazard_statement* and *European_risk_phrases*. In even more generalized cases and other domains, other modularization schemes are also possible. It depends on the objective and application scenario of the modeling. However, in our research, we have seen that, instead of direct and premature modeling, extra work before that is in favor of a good ontology quality and

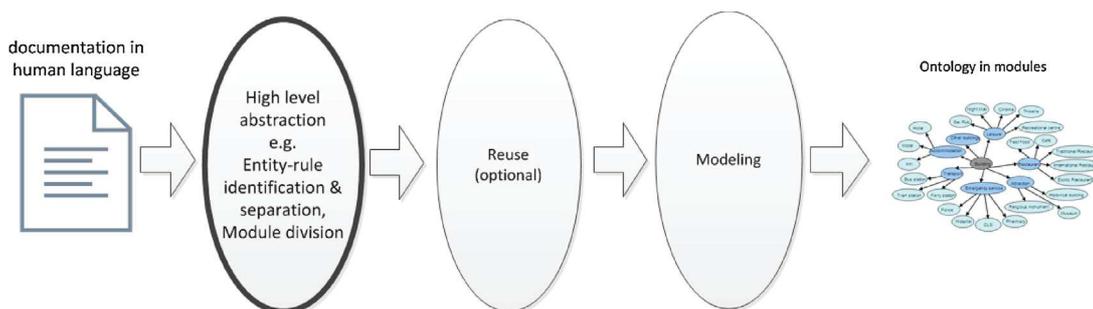


Fig. 15. Before modeling the ontology or even reusing, a high level abstraction e.g. entity-rule separation and module division is needed sometimes.

reusability, especially in a top-down ontology development approach.

From engineering point of view, Ontology construction is tedious and time-consuming process. The use of some NLP techniques to partially automate the terminology collection phase may accelerate the process by identifying classes and properties especially in the case of huge volume of sources.

We also learned that even though a knowledge base based on ontology is developed, it seems that we could not burden all the work upon ontology. Concerning the knowledge underlying in the EU Eco-labeling criteria document, our knowledge base for now only covers limited amount of knowledge. Part of this is due to the limit of expressiveness of OWL language. Another reason is because sometimes well informed and experienced human labor is more competent for aesthetics and usability assessment. For example, the criterion NO.8 of this laundry detergent criteria talks about the consumer information such as the dosage instructions, information on the packaging, and additional claims on the packaging. Instead of assigning these works to human experts, we can imagine how hard and costly it will be to implement and train an AI system to do that. But the research on AI and NLP is nevertheless worth of it as we still expect that computers should eventually accomplish such sophisticated job.

9. Conclusion and future work

In this paper we have seen what is eco-labeling and EU Eco-label. To popularize eco-labeled products and services in order to achieve a more competent and ecological economy, a better eco-labeling process is needed. Our approach is based on a knowledge base composed of identified domain knowledge by means of ontologies, which will be the foundation of further decision support process development. In this paper, an OWL ontology knowledge base for laundry detergent is established. A separation pattern between entity and rules as well as the modularization of each criterion is proposed to realize better modularity. The modules of the knowledge base can be browsed and reused by other systems in order to achieve a data interoperability and knowledge sharing. By using a modularized design, as presented in Section 5, entity modules and rule modules are managed separately, we have improved re-usability and maintainability of ontology in the face of change. Once certain rule module requires updating, the update operation will be exclusively restricted within this rule module without affecting the other rule modules. Such feature will be very useful for ontology knowledge base management in real-time cases. If the knowledge base is very big, it's not necessary to stop or shut down the service for some small and partial changes, instead, our modular design allows change or reconfigure ontologies on-the-fly.

However, consensus on an OWL-compatible syntax for a modular ontology language that can express both inter-module concept subsumptions and inter-module role relations is still lacking. It would be interesting to investigate whether OWL can be re-modeled with a new modular semantics or it has to be extended with a new set of constructors to replace *owl:imports* [38]. In our future work, we are planning to develop new syntax and semantic constructors for more convenient ontology modularization and integration. Our current research invests much effort on domain-centric ontology knowledge base and ontology modularization. Based on those ontology modules that are already developed and presented in this paper, we'd like to extend our knowledge base by developing ontologies of new product groups e.g. cosmetic products and all-purpose cleaner by reusing modules and integrating NLP techniques. Then, based on the local reuse and modularization, we are going to connect or merge multiple

ontology knowledge bases remotely via Semantic Web. Thus, how to simultaneously deal with ontology modularization, ontology integration and ontology mapping or alignment will be one of the important topics in our future research work.

Acknowledgment

Authors would like to thank Cedric POULENARD the Head of Household Product Division at IRFAQ (Research Institute for Formulation Analysis and Quality control) for his availability and advices.

References

- [1] Eco-labeling Markets, International Institute for Sustainable Development. http://www.iisd.org/business/markets/eco_labeling.aspx (accessed 05.03.16).
- [2] A. Lefébure, R.R. Muñoz, Communicating to Consumers in Sweden with Eco-labels: Is the Message Getting Through? (2011) (accessed 25.08.15).
- [3] S. van der Panne, V. Ihamäki, E. van Ooij, Green Public Procurement in the European Union and the Use of Eco-labels, (2014) (accessed 28.08.15).
- [4] The EU Eco-label Scheme - What is it? <http://www.ecolabels.fr/en/the-eu-eco-label-scheme-what-is-it> (accessed 05.03.16).
- [5] Facts and Figures - Ecolabel - European Commission. <http://ec.europa.eu/environment/ecolabel/facts-and-figures.html> (accessed 05.03.16).
- [6] How are Guidelines Drawn Up? - Ecolabel. <http://www.ecolabels.fr/en/the-eu-eco-label-scheme-what-is-it/how-are-guidelines-drawn-up> (accessed 05.03.16).
- [7] N. Guarino, D. Oberle, S. Staab, What is an ontology? Handbook on Ontologies, Springer, 2009, pp. 1-17.
- [8] V. Rudi Studer, R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, Data Knowl. Eng. 25 (1) (1998) 161-197.
- [9] J. Pathak, T.M. Johnson, C.G. Chute, Survey of modular ontology techniques and their applications in the biomedical domain, Integr. Comput.-Aided Eng. 16 (3) (2009) 225-242.
- [10] C. Parent, S. Spaccapietra, An overview of modularity, Modular Ontologies, Springer, 2009, pp. 5-23.
- [11] A. Gangemi, Ontology design patterns for semantic web content, International Semantic Web Conference, Springer, 2005, pp. 262-276.
- [12] A. Gangemi, A. Gómez-Pérez, V. Presutti, M.C. Suárez-Figueroa, Towards a Catalog of Owl-based Ontology Design Patterns, (2007) (accessed 25.08.15).
- [13] B.C. Grau, B. Parsia, E. Sirin, Ontology integration using ϵ -connections, Modular Ontologies, Springer, 2009, pp. 293-320.
- [14] L. Serafini, A. Taminin, Composing modular ontologies with distributed description logics, Modular Ontologies, Springer, 2009, pp. 321-347.
- [15] L. Serafini, A. Taminin, Drago: distributed reasoning architecture for the semantic web, The Semantic Web: Research and Applications, Springer, 2005, pp. 361-376.
- [16] J. Bao, G. Voutsadakis, G. Slutzki, V. Honavar, Package-based description logics, Modular Ontologies, Springer, 2009, pp. 349-371.
- [17] Z.C. Khan, C. Maria Keet, An empirically-based framework for ontology modularisation, Appl. Ontol. 10 (3-4) (2015) 171-195.
- [18] A. Rector, S. Brandt, N. Drummond, M. Horridge, C. Pulestin, R. Stevens, Engineering use cases for modular development of ontologies in owl, Appl. Ontol. 7 (2) (2012) 113-132.
- [19] M. Uschold, M. Gruninger, Ontologies: principles, methods and applications, Knowl. Eng. Rev. 11 (2) (1996) 93-136.
- [20] D. Jones, T. Bench-Capon, P. Visser, Methodologies for Ontology Development, (1998) (accessed 25.08.15).
- [21] Y. Sure, S. Staab, R. Studer, Ontology engineering methodology, Handbook on Ontologies, Springer, 2009, pp. 135-152.
- [22] M.C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The neon methodology for ontology engineering, Ontology Engineering in a Networked World, Springer, 2012, pp. 9-34.
- [23] P. Cimiano, J. Völker, A framework for ontology learning and data-driven change discovery, Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Lecture Notes in Computer Science, Springer, Springer, 2005, pp. 227-238 vol. 3513.
- [24] S. Mittal, N. Mittal, Tools for ontology building from texts: analysis and improvement of the results of text2onto, IOSR J. Comput. Eng. (IOSR-JCE), e-ISSN (2013) 2278-0661.
- [25] Y. Wang, Y. Sure, R. Stevens, A. Rector, Knowledge elicitation plug-in for protégé: card sorting and laddering, The Semantic Web-ASWC 2006 (2006) 552-565.
- [26] M. Loukides, What is Devops? Infrastructure as Code, O'Reilly Media, 2012, pp. 25.
- [27] F. Baader, U. Sattler, An overview of tableau algorithms for description logics, Stud. Log. 69 (1) (2001) 5-40.
- [28] L. Deborah, McGuinness, F. Van Harmelen, et al., Owl web ontology language overview, W3C Recomm. 10 (10) (2004) 2004.
- [29] D. Tsarkov, I. Horrocks, Fact++ description logic reasoner: system description, International Joint Conference on Automated Reasoning, Springer, 2006, pp. 292-297.

- [30] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, Y. Katz, Pellet: a practical OWL-DL reasoner, *Web Semant.: Sci. Serv. Agents World Wide Web* 5 (2) (2007) 51–53.
- [31] R. Shearer, B. Motik, I. Horrocks, Hermit: a highly-efficient owl reasoner, *OWLED*, volume 432 (2008) 91.
- [32] M. Horridge, N. Drummond, J. Goodwin, A.L. Rector, R. Stevens, H. Wang, The manchester owl syntax, *OWLed*, volume 216 (2006).
- [33] J. Brank, M. Grobelnik, D. Mladenić, A Survey of Ontology Evaluation Techniques, (2005) (accessed 25.08.15).
- [34] D. Xu, M.H. Karray, B. Archimède, A semantic-based decision support platform to assist products' eco-labeling process, *Ind. Manage. Data Syst.* 117 (7) (2017) 1340–1361.
- [35] D. Vrandečić, *Ontology evaluation*, *Handbook on Ontologies*, Springer, 2009, pp. 293–313.
- [36] M. Sabou, M. Fernandez, *Ontology (network) evaluation*, *Ontology Engineering in a Networked World*, Springer, 2012, pp. 193–212.
- [37] A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann, *Modelling ontology evaluation and validation*, *European Conference on the Semantic Web: Research and Applications* (2006) 140–154.
- [38] J. Bao, V.G. Honavar, *Divide and conquer semantic web with modular ontologies: a brief review of modular ontology language formalisms*, *Proceedings of the 1st International Conference on Modular Ontologies-Volume 232* (2006) 1–14 CEUR-WS.org.