

# 1. Personalization of ontologies visualization: use case of diabetes

Laia Subirats<sup>1,2</sup>, Rosa Gil<sup>3</sup>, Roberto García<sup>3\*</sup>

<sup>1</sup> Eurecat, Centre Tecnològic de Catalunya, Unitat de eHealth, C/Bilbao 72, 08005 Barcelona, Spain.

<sup>2</sup> Universitat Oberta de Catalunya, eHealth Center, Rambla del Poblenou 156, 08018 Barcelona, Spain.

<sup>3</sup> Universitat de Lleida, Jaume II 69, 25001 Lleida, Spain

\* Corresponding author: [roberto.garcia@udl.cat](mailto:roberto.garcia@udl.cat)

## Abstract

In P4 medicine, which faces the challenge of building personalized medicine, a semantic-based personalized visualization is key to enhance both patient and other stakeholders' experience. Ontologies are a formal way to represent knowledge, and their visualization depends considerably on the role or user who is visualizing them. In the same way, like databases have virtual tables or views to tailor the data to application needs, ontologies should facilitate different perspectives on semantic data, customized to the needs of all stakeholders. This is especially true in the case of medicine, where the data consumers have quite varied roles, like patient, professional or policymaker. This study presents the current state of the art of personalization in ontology visualization initiatives, a brief summary of the diabetes mellitus domain, and existing ontologies in the diabetes domain. It also presents an approach for the personalization of ontologies visualization based on the implementation of the overview, zoom/filter and details interaction patterns. This is done by adapting the Rhizomer tool so different views can be generated in the context of personalized medicine. All this is validated through a use-case of a new ontology to model the diabetes domain from an existing open dataset of around 70000 diabetic patients extracted from American hospitals. The conclusion is that the application of this approach has the potential to enhance personalization of medicine ontologies and their visualization.

## 1.1 Introduction

P4 medicine [1] is a term widely used to indicate that P4 medicine is Predictive, Preventive, Personalized and Participatory. P4 medicine is the challenge of systems medicine, where hospitals try to maximize patients' experience (see patients

experience measurement of Mayo Clinic [2]). Taking into account that medicine should be personalized, the visualization of information should be personalized as well. Given that ontologies are the formal definition of concepts and domains, ontologies should be personalized too. Therefore, the visualization of information of all stakeholders is customized according to their needs and expertise. In this study, an approach of personalization of ontologies is applied to a diabetes use case.

This chapter is structured as follows: In Sec. 1.2 a state of the art is presented, and it contains a description of current initiatives of personalization of ontologies visualization, a background of diabetes mellitus and existing diabetes ontologies. In Sec. 1.3 the proposed approach is presented and in Sec. 1.4 a case study in the diabetes domain is presented. Finally, in Sec. 1.5 conclusions and future work are outlined.

## **1.2 State of the art**

The state of the art is divided into a brief summary of existing initiatives in personalization of ontologies visualization, a background of diabetes mellitus and a description of existing diabetes ontologies.

### ***1.2.1 Personalization of ontologies visualization***

There are some surveys of visualization ontology methods such as [3], where they group these methods in the following categories: indented list, node-link and tree, zoomable, space-filling, focus + context or distortion and 3D Information landscapes. In the following paragraphs several methods to visualize ontologies will be presented describing its main advantages and disadvantages.

*Ontology views of Bioportal* (indented list). Bioportal [4] is the biggest medical ontology repository. There, more than 500 ontologies can be found, some of them receiving a high number of views. One of the most visited ontologies is SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms) [5]. Bioportal offers the possibility to add ontology views to the ontology. According to [6], ontology views are subsets of one or more ontologies which are useful to work with. The views of SNOMED CT comprise topics such as a nursing ontology, Spanish edition, anatomy, organism or test findings. When clicking a view of an ontology, a new navigation through this view is enabled in Bioportal. Its main advantage is that it is modulable and it enables to focus on one part of one or more ontologies. The disadvantage is that problems of merging ontologies can appear when the number of ontologies involved in the view is high.



that it supports different roles in visualizing the same ontology, while the main disadvantage is that it is very domain-focused and should be adapted to each domain.

The gap this research paper wants to fill is to generate more usable and customizable interactive presentations of ontologies and associated data for lay-users, which goes beyond simple hierarchical representations or the drawbacks of graph representations [14].

To do so, we leverage the widely used Shneiderman's interaction patterns for information visualization [15] to use them with web ontologies and semantic data. This approach is presented in Sec. 1.3 together with validation results that show it improves the User Experience. The results are further validated in the context of personalized medicine through a scenario detailed in Sec. 1.4.

The advantage is that not only the hierarchy of the ontology is available but also enables to navigate through the instances and its properties, enables advanced interaction features like pivoting, and provides different graphical representations.

### ***1.2.2 Diabetes mellitus***

Diabetes Mellitus is a disease marked by high levels of sugar in the blood. WHO foresees that diabetes will be the 7th leading cause of death in 2030 [16]. There are three types of diabetes: Type 1, Type 2 and gestational diabetes mellitus. Complications of diabetes can be divided into hypoglycaemia; hyperglycaemia, micro-vascular damage and macro-vascular damage.

In the last years, the interest in reducing hospital readmissions has increased due to its potential to reduce healthcare costs and improve care. Since 2012, Centers for Medicare and Medicaid Services decrease their funding to hospitals with too high readmissions rates [17]. Studies usually differentiate between readmissions of less than 30 days and readmissions over longer time periods. Particularly, the interest in reducing diabetes hospital readmissions has increased because of the growth of the burden of diabetes [18]. In order to reduce diabetes readmissions with interventions, risk factors of readmissions should be better understood. Owing to the fact that it would be very costly to apply intervention measures to all diabetic patients, studies usually focus on high-risk patients.

There are currently several risk factors which predispose to diabetes such as genetics, race, physiological measures and habits. In Catalonia, there are some guidelines for the harmonization of the pharmacological treatment of type 2 diabetes mellitus [19]. There are several key points for the treatment of hypoglycaemia of type 2 diabetes mellitus: glycemic control (maintain HbA1c level under 7%), non-

pharmacological treatment (based on education on the management of the disease and a healthy lifestyle), initial pharmacological treatment, double therapy, therapy combined with insulin, triple therapy and reevaluation of the treatment.

Type 2 diabetes usually is associated with other comorbidities that can have an impact in the treatment of this disease which are: renal insufficiency; cardiac insufficiency; hepatic insufficiency; cardiovascular disease; obesity; and elderly, pluripathological and disable patient.

This research paper applies as a use case the diabetes domain.

### ***1.2.3 Diabetes ontologies***

There are several existing initiatives to model diabetes with ontologies. One of them is [20], which combines fuzzy ontology modeling with case-based reasoning (CBR) [21]. Moreover, in [22] also a fuzzy ontology is created to provide diet recommendations to diabetic patients.

Another initiative is [23], which provides an ontology to identify barriers in Diabetes of type 2. This study could be linked not only with the environmental factors such as climate and financial assets appear, but also with body functions such as emotional functions and the ICD where items such as hypoglycemia and obesity appear. [24, 25] are initiatives which automate diagnosis of diabetes in Electronic Health Records (EHR). Furthermore, incorrect and incomplete data can be addressed by ontologies.

If we look for diabetes ontologies in Bioportal, we found three ontologies. The first one is BioMedBridges Diabetes Ontology (DIAB) [26], an ontology of 375 classes developed by Helmholtz Zentrum München with more than 3000 accumulated views.

The second one is Diabetes Mellitus Diagnosis Ontology (DDO) [27, 28], an ontology with more than 6400 classes that contains symptoms, drugs, tests, complications, etc. which has also more than 3000 accumulated views. More specifically, the DDO ontology has 6444 classes, the root class is an entity, which has continuous, occurrent and symptom subclasses, which divide the concepts into 3 groups. There are few object properties (42), which focus on the relationships between the patient, disease and drug classes. The few data properties that exist are usually demographic classes. There are also few individuals, almost all of the class curation status specification. DDO does not integrate other ontologies.

And finally, the third ontology is Diabetes Mellitus Treatment Ontology (DMTO) [29] which models the creation of customized treatment plans for diabetic

patients. The ontology is based on Basic Formal Ontology (BFO) [30] and the Ontology for General Medical Science (OGMS), has over 10000 classes, more than 3400 accumulated views and is used in four Bioportal projects. More specifically, the DMTO ontology has 10700 classes, the Thing root class, and it has 14 subclasses that belong to the domain of time, entity, food, nutrients and meals. There are a considerable amount of object properties (191) and data properties (124), and a considerable number of rules. However, there are few individuals (63), many of them related to time. The ontology is based on Basic Formal Ontology (BFO) and Ontology for General Medical Science (OGMS). In addition, it extends the Diabetes Mellitus Diagnosis Ontology (DDO) ontology by adding treatment classes and axioms to the existing diagnostic part. The advantages of reusing ontologies are that when you make changes to DDO they are updated to DMTO in such a way that it is easier to maintain the ontologies, that is to say the ontologies are modular with all the advantages that this entails: easier to reuse, and better detail and maintenance.

The created ontology reuses concepts of ontologies focused on diabetes such as DMTO, more general health ontologies such as SNOMED CT, and also international classifications such as ATC and ICD.

### 1.3 Approach

Our starting point to deal with ontology visualization is the fundamental set of tasks for visual data analysis proposed by Shneiderman [15]. In the following, we present each task associated with the chosen interaction pattern and Information Architecture component [31] to implement the pattern:

- **Overview:** to obtain a full picture of the data set at hand. At this stage, we propose to apply the Global Navigation interaction pattern or the Directory Navigation pattern. In Information Architecture terms, the former can be implemented using navigation menus users are used to see at the top or on the left of websites. The later, as sitemaps that can be represented using text or graphically, for instance using [32].
- **Zoom and Filter:** to zoom in on items of interest and filter out uninteresting items. Here the proposal is to employ some form of Faceted Navigation. Once we have zoomed in by selecting the kind of things we are interested in from the navigation bar, facets are the Information Architecture components that help users to filter out those data items that are not interesting to them.
- **Details:** after zooming and filtering the user arrives at concrete resources of interest. At this point, the user can obtain details for those resources, which in the case of Semantic Web data means to retrieve properties for the resources plus those properties pointing to them. This step is related to the Details on Demand interaction pattern and can be implemented via a simple list of properties and

values of the resource of interest or through a specific visualization tailored to the kind of resource at hand, e.g., a map for geo-located resources.

We elaborate these interaction patterns in the context of semantic data. We have chosen these patterns because they are simple and, so users are very comfortable using them. They are part of the "culture" about how information is presented in the Web so they can be quickly learned. However, though they look like the common ones, the semantic versions of them should be capable of giving access to the richer semantic data they are built on top of.

The aim is to make it possible that lay users, not just Semantic Web technologies experts, can benefit from the Semantic Web. The following subsections detail how these user tasks are implemented through user interface components in the context of a Semantic Web application.

### ***1.3.1 Overview***

"Overview" is the first user task when dealing with a dataset. The objective is that the user is capable of getting an idea about the overall structure of the dataset. In the case of Semantic Web and a Linked Data dataset, this overview usually helps to identify the main types of entities in the dataset, the most instantiated classes, and how they are structured, e.g., their hierarchical structure. In addition to an overview from a class instantiation point of view, it is also possible to build an overview of how data is classified into topics coming from a thesaurus. In this case, we consider the "broader than" and "narrower than" relations among topics. To provide this overview of the data, we propose to employ the Global Navigation or the Directory Navigation interaction patterns. The former is provided by navigation menus, detailed in Section 1.3.1.1, the later through graphical representations like treemaps, described in Section 1.3.1.2.

#### **1.3.1.1 Navigation Menus**

Navigation menus, in the case of websites, let users navigate through different sections and pages of the site. They tend to be the only consistent navigation element, being present on every page of the site.

Traditionally, user-centered design techniques like Card Sorting are used to develop the navigation menus of websites. This technique requires a lot of time and effort from developers, while most of this effort is wasted as soon as the structure of the dataset changes and the statically defined menu gets obsolete. For instance, if new content items are introduced, or an existing part of the content becomes more relevant, the Card Sorting should be repeated at least in part.

In the case of websites built on top of semantic data, we have the opportunity to automate part of the process of generation and maintenance of the navigation menus. This is possible because semantic data is structured by thesauri and ontologies, which hierarchically organize the kinds of things described in the dataset. They not only specify all the classes or concepts but also which entities belong to a particular class or are related to a specific concept.

Consequently, if there are fewer instances of a class or related concepts or none at all, the class or concept should be less relevant in the menu bar. On the contrary, those that do have many members should be shown prominently in the menu bar. To achieve this, we obtain the hierarchical structure of the classes or concepts instantiated or used in the dataset. For each class, it is also computed the number of instances and for concepts how many times they are referred to as the topic subject. All this information is retrieved using SPARQL (SPARQL Protocol and RDF Query Language).

Then, the hierarchy is flattened to the number of levels required because this component can generate both global and local menus, i.e., a menu for the whole dataset or just for a subset of it. The site administrator can also configure some parameters: the number of levels in the menu, the number of items in each level, the order of items (alphabetically or by the number of instances) and a list of classes or concepts to omit.

According to these parameters, this component generates the menu applying a recursive algorithm that mainly performs two operations:

- Split the concepts or classes with a large number of members in their narrower related concepts or subclasses.
- Group those with few members into a broader concept or superclass.

This approach allows showing the navigation bar that best fits the data in the dataset at each particular moment. For instance, if the dataset changes from containing mainly data about projects to mainly about publications, the menu would change accordingly to show the part of the dataset structure about publications more prominently.

### **1.3.1.2 Treemaps**

Navigation menus are quite useful because lay users are comfortable with them, most websites feature them and they are used to interacting with them. However, they only provide an overview of the most common classes, those more instantiated.



In order to gain a more detailed overview, websites usually apply the Directory Navigation pattern through different sorts of sitemaps.

However, such detailed overviews are difficult to generate with large heterogeneous semantic datasets, which is the typical case with Linked Data. A conventional approach to obtain an overview and support the exploration of large datasets is to structure them hierarchically [33]. Hierarchies allow users to visualize different abstractions of the underlying data at different levels of detail. Visual representations of hierarchies allow creating simplified versions of the data while still maintaining the general overview.

There are several techniques for visualizing hierarchical structures. One approach to provide high-level overviews is Treemaps [32]. Treemaps use a rectangle to show the tree root and its children. Each child has a size proportional to the cumulative size of its descendants. They are an excellent method to display the size of each node in a hierarchy. However, since Treemaps are not so useful if the user wants to perform an exhaustive exploration or if the item the user is looking for is a known one, we complement Treemaps with more standard overview components like sitemaps or site indexes [34].

We have implemented a Treemap component that shows an interactive overview of the class hierarchy in the dataset. Users can zoom in and zoom out to go deep into the class hierarchy. This way, the Treemap visualization supports the overview and zoom tasks proposed by Shneiderman.

The starting point to generate the Treemap for a dataset is also, like for navigation menus, a data structure capturing how the classes instantiated by the dataset are hierarchically structured, plus how many direct instances does each class have. In this case, however, this structure is not flattened but reproduced as a Treemap with sizes proportional to the number of instances. Classes too small to be represented in the Treemap are grouped and labeled "Others". The Treemap provides an overview of the dataset an information about the size of each of the classes structuring this dataset (it is interactive so users can gain access to classes deeper in the hierarchy than the two first levels shown by default).

### **1.3.1.3 Site Maps and Site Indexes**

The navigation menus presented in Section 1.3.1.1 are quite effective because lay-users are comfortable with them, most websites feature them and users are used to interacting with them. However, they only provide an overview of the most frequent classes, those more instantiated. Treemaps provide a more detailed overview of the dataset but they require much interaction with the user and, because they are less frequent, they are harder to use.

To provide a more detailed overview, many websites usually apply the Directory Navigation pattern through different sorts of sitemaps. Sitemaps act as a navigation aid by providing an overview of the site's content at a single glance. Moreover, they are widely used, so they are more comfortable for end-users. However, they do not provide the visual feedback about the significance of the different parts of the dataset that is provided by Treemaps.

HTML sitemaps are designed to help users find content on the website. A site map is a web page that lists all the pages of a website, customarily organized hierarchically. In the case of large sites, instead of containing links to all the pages, they can list the main pages (e.g., categories) of the site. When the site contains many levels in the structure and many elements on each level, the sitemap functions as a navigation alternative to navigation menus.

In addition to sitemaps and Treemaps, evaluations with users show that there are tasks, like "known item" searches, where they are more comfortable with site indexes [35]. A site index or A-Z index is a navigational and informational tool that lists all the pages or categories alphabetically. While a sitemap provides a general view of the overall site contents, an A-Z index provides access to particular content. An alphabetical list can better suit users' mental model when they are searching for a specific item.

However, while sitemaps can give users context, site indexes provide none. Non-related categories appear in the site index without giving users any additional information. Therefore, we have implemented the site index so that it also provides context information of each class. When the user moves the pointer over a site index entry, an overlay appears showing its more generic parent and its more specific subclasses.

### ***1.3.2 Facets***

Users do not always know exactly what they are looking for and, sometimes, they do not even know what its name is. Sometimes they are unfamiliar with the domain or they want to learn about a particular topic. This is particularly true when exploring Semantic Web datasets. In these cases, exploratory search is a strategy that allows users to refine their search by successive iterations. An exploratory interface such as faceted browsing allows users to find information without a priori knowledge of its schema or underlying ontologies.

With navigation menus or Treemaps, we can make the user aware of the hierarchical structure of a dataset but, once they choose the class of things they are interested in, they face the barrier of not knowing how they are described. In other words, what are the main properties that describe them, which ones are the more

relevant for that particular kind of things, the range of values they have in that particular case, etc.

Faceted navigation is an exploratory technique for navigating a collection of elements in multiple ways, rather than a single and pre-determined order. Faceted interfaces provide a user-friendly way to navigate through a wide range of data collections. Traditional faceted browsers relied on manual identification of the facets and apriori knowledge of the target domain.

When dealing with semantic data, it is possible to automate this process so a faceted semantic browser will be able to handle any dataset without any configuration requirements. Since semantic data facilitates integrating data from different sources, we cannot assume a single fixed schema for all data. Consequently, a semantic data faceted browser should be scalable and generic, not depending on a particular dataset structure.

To compute the facets, Rhizomer performs SPARQL queries that retrieve all the properties for each class, which will become the facets when exploring it, the frequency of each property for the given class and the number of different values that each property can have for the given class. The frequency and the number of different values are used to help decide if a property is worth including it as a facet of the corresponding class faceted view. For instance, a property that is used just in the 10 percent of the class instances descriptions, which has just one possible value or that has a different value for each instance is not very useful in the faceted view.

For datasets with many classes and properties, these queries become too slow to be computed in real-time during user interaction. Consequently, they are calculated the first time Rhizomer is deployed on a dataset and stored in a data structure. They are then updated incrementally whenever the dataset changes.

However, there are also queries that are generated while the user interacts through the facets. For instance, when the user requests the most common values of a class facet. These values are shown as part of the facet when the user clicks on the link "Common values". The result is not just the list of the most common values for that particular class and property. Other previous restrictions set by the user through other facets should also be taken into account. Consequently, what is retrieved is the list of common values for that facet given the set of restrictions on the current faceted view. For instance, the most common names for people change depending on the restrictions made on the "gender" facet.

Consequently, in this case, it is not possible to pre-compute the list of common values and this is done during user interaction. In any case, to reduce the common values to be updated after each user interaction, all facets are initially shown with the list of common values collapsed. They are only expanded as a result of user

interaction and just those that have been expanded are updated when the user sets constraints by checking facet values.

In addition to expandable common values lists, facets feature a search box. This box allows searching among the facet values for a specific one, something especially useful when it is not among the most common ones. However, this requires that the user knows the desired value. To mitigate this problem, the search box features an autocomplete function that allows the user to explore facet values based on what the user has typed so far.

The filters applied so far by the users are converted into filters in the resulting SPARQL query. These filters are used both to compute the common values and to compute the instances of the class for which the faceted view is shown that satisfy all the filters applied so far. For instance, the next code listing shows the SPARQL generated after selecting the class "Encounterid" and then, in the corresponding faceted view, filtering the facets "glimepiride" to the value "Down" and the facet "change" to "No".

The type constraint in line 6 of the listing is generated as a result of switching to the "Encounterid" faceted view, for instance by selecting the class in the navigation menu. The user then expands the available values of the "glimepiride" facet and selects the value "Down", which produces the constraints in lines 7 and 8. Similarly, for the facet "change" the user selects the value "No", which corresponds to the lines 9 and 10.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX professional: <http://...ontologies/2018/4/Professional#>
```

```
SELECT DISTINCT ?r1  
WHERE  
{ ?r1 rdf:type professional:Encounterid .  
  ?r1 professional:glimepiride ?r1var1 .  
  FILTER ( str(?r1var1) = "Down" )  
  ?r1 professional:change ?r1var0 .  
  FILTER ( str(?r1var0) = "No" )  
}
```

### ***1.3.3 Approach Validation***

The validity of this proposed approach as a way to improve the User Experience (UX) when interacting with semantic data has been validated using the Quality in Use Model for Semantic Web Exploration Tools (SWET-QUM) [36].

Through this Quality in Use Model, the quality factors Effectiveness, Efficiency, Satisfaction and Context Coverage were evaluated using a set of standard and custom metrics. The standard ones are based on ISO/IEC 25010:2011 while the added metrics are proposed by SWET-QUM to better measure the standard quality factors in the context of Semantic Web exploration tools.

The results are very promising as the proposed approach, as implemented in Rhizomer, outperforms the other main semantic data faceted navigators featuring pivoting, i.e. Virtuoso Facets and SParallax. This is especially true for effectiveness and efficiency for which the collected metrics allow to affirm, with a 95% confidence, that users needed less time to complete the proposed end-user tasks using Rhizomer.

Full details of the user study are available from [36] and [37]. The only observed drawbacks are regarding Context Coverage. In this case, Virtuoso provides better Task Flexibility, i.e. Virtuoso offers more ways to complete the task, though this usually makes the user interface more complex and negatively impacts users' performance.

More recently, it has been also possible to evaluate the proposed approach using the Benchmark for End-user Structured Data User Interfaces (BESDUI) [38]. In this case, compared to existing tools that also included the relational databases exploration tool Sieuferd, Rhizomer also obtained the best overall results considering both the range of supported tasks and the time efficiency to solve them.

The other evaluated tools are Sieuferd, Virtuoso Facets and PepeSearch. In this case, despite Sieuferd is the tool capable of supporting the greatest range of end-user tasks, it is also the most complex tool and thus the less efficient. On the other hand, while PepeSearch is the tool that allows users to complete the supported tasks quicker, it supports just the 25% of the proposed tasks.

## 1.4 Case study

This research aims to visualize, under different user roles, over 70000 diabetic patient records based on an ontology developed ad-hoc for this open data dataset [39] (other similar datasets of the Pima Indians Diabetes are available in Kaggle <https://www.kaggle.com/uciml/pima-indians-diabetes-database> [40])

The decision to create and ad-hoc ontology is because it is lighter to be used in the visualization tool and the overhead of maintaining the ontology is minimal. Moreover, as the ontology uses international standards such as ATC and ICD codes and terms of other ontologies, it makes easier to reuse it and to map it with other existing ontologies.

The ontology has been created following a bottom-up methodology [41]. We started with more specific classes obtained from the dataset and afterwards they were grouped in more general concepts. This methodology was chosen among top-down and mixed top-down and bottom-up methodologies because we wanted to represent with a minimum overhead the concepts of the dataset. Furthermore, the ontology has been automatically verified using the tool Oops! [42], which enforces best practices during ontology development. The only minor pitfall is some pending annotations.

The following reuse of classes of the ontology have been considered (48 concepts). In brackets there is the name of the concept of the correspondent existing ontology:

- Drugs of ATC classification, 22 concepts: Acetohexamide, Alphablucosidaseinhibitors, Acarbose, Biguanides, Glimepiride, Glinides, Glipizide, Glitazones, Insulin, Metformin, Metforminpioglitazone, Metforminosiglitazone, Miglitol, Nateglinide, Glimepiridepioglitazone, Pioglitazone, Repaglinide, Rosiglitazone, Sulfonylureas, Troglitazone, Tolazamide and Tolbutamide.
- Diabetes mellitus treatment ontology, 8 concepts: Age, Demographicqualifiers, Disease, Drugs (chemical entity), Gender, Hospital, Race (ethnicity) and Weight.
- SNOMED CT, 18 concepts: A1Cresult (Glycosylated Hemoglobin), Admissionsourceid (Admission statuses), Admissiontypeid (Hospital admission), change (Recommendation to change medication), Encounterid (Encounter for check up), Evaluationfinding, Medicalprocedures (Medical procedure), Medicalspecialty, Numberemergency (Emergency), Numberinpatient (Inpatient), Numberoutpatient (Outpatient), Numlabprocedures (Laboratory procedure), Numprocedures (Administrative procedure), Patient discharge (Dischargedispositionid), Patientnbr (Computer record of patient), Person, Payer code (Healthcare payer), Timeinhospital (Time of symptom onset).

On the other hand, the following new classes have been created (14 concepts):

- Drugs not available in ATC classification, 6 concepts: Glipizidemetformin, Glyburidemetformin, Citoglipton, Glyburide, Chlorpropadime, Examide,
- New concepts, 8 concepts: Diagnosis1, Diagnosis2, Diagnosis3, Numberdiagnoses, Diabetesmedications, Nummedications, Maxgluserum and readmitted.

Therefore, they add up 62 classes, which is the number of classes of the ontology.

Three different views will be provided depending on the profile of the user who is looking at the data:

- Patients: in this case, medical procedures information will not be provided, because patients are not interested in administrative information.
- Professional: for this profile, all the information will be shown because the health professional should have the full picture of the patient. Besides, health professionals know medical technical vocabularies such as standards and classifications.
- Policymaker: in this case, we preserve privacy and avoid discrimination, so demographic information of age, gender or race will not be provided.

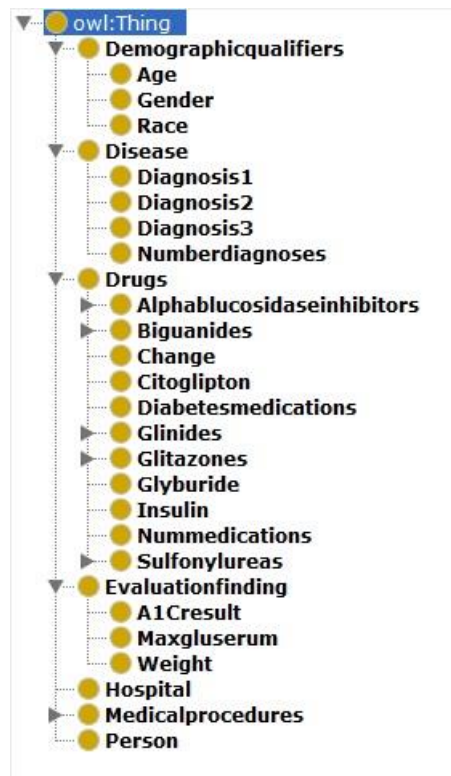


Fig. 1.2. Class diagram of the professional view of the ontology from Protégé.

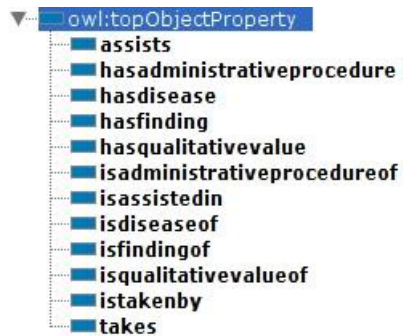


Fig. 1.3. Object properties diagram of the professional view of the ontology from Protégé.

The class diagram of the professional is shown in Fig.1.2. There are 7 classes which are children of Thing: demographic qualifiers, disease, drugs, evaluation finding, hospital, medical procedures and person. Some other taxonomies have been taken into account in the creation of the ontology. For example, the Drug class follows the ATC taxonomy and generic classes of drugs such as Biguanides have been included. All three ontologies for each of the profiles are available under request. The professional ontology has the following metrics without considering the instances:

- Axiom: 171
- Logical axiom count: 96
- Declaration axioms count: 75
- Class count: 62
- Object property count: 13
- Annotation Property count: 1
- DL expressivity: ALCHIF
- SubClassOf: 59
- DisjointClasses: 1
- SubObjectPropertyOf: 1
- InverseObjectProperties: 6
- FunctionalObjectProperty: 5
- ObjectPropertyDomain: 12
- ObjectPropertyRange: 12

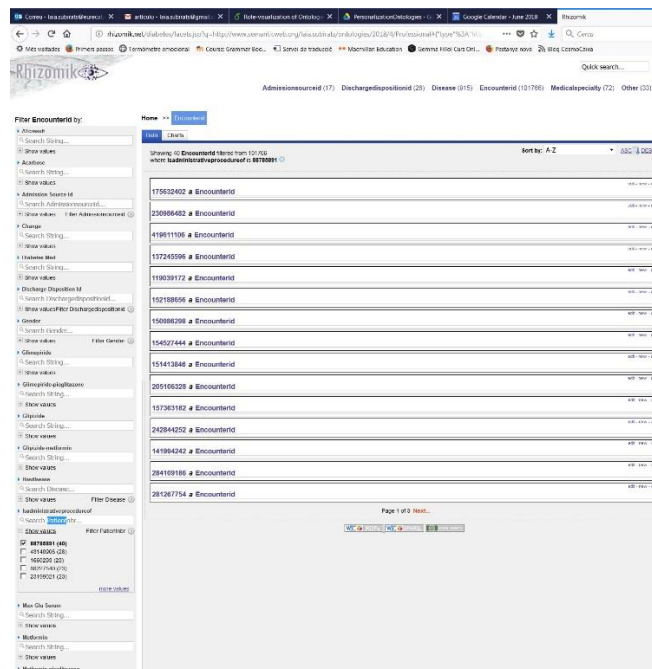
All 71518 patients data has been translated to RDF format with LODRefine freely available from its repository <https://github.com/sparkica/LODRefine>, which is a version of OpenRefine including template-based RDF generation. The user can explore the resulting semantic data, following the approach presented in the previous section, at <http://rhizomik.net/diabetes>.



The object properties diagram of the professional ontology is shown in Fig. 1.3. All the properties have its inverse: assists / is assisted in, has administrative procedure / is administrative procedure of, has disease / is disease of, has finding / is finding of, has qualitative value / is qualitative value of and takes / is taken by.

When exploring the dataset at <http://rhizomik.net/diabetes>, several tabs corresponding to the main classes in the dataset appear in the upper part of the tool as navigation menus to fulfill the data analysis "Overview" task. The menu options include the class label and the number of instances: "Admission Source (25)", "Discharge Disposition (30)", "Disease (915)", "Encounter (101766)", "Medical Specialty (72)", "Patient (71518)" and "Other (33)". The "Other" menu option has the following submenus, corresponding for the classes with fewer instances beyond the available slots in the main menu: "Admission Type (8)", "Gender (3)", "Payercode (17)" and "Race (5)".

When clicking any of this options in the navigation menus, the user switches from the "Overview" task to the "Zoom and Filter" one, as described in the previous section. Consequently, as this task has been implemented using facets following the proposed approach, the faceted view for the corresponding class is shown. In Fig. 1.4 the "Encounter" menu option has been selected and its facets are displayed so the user can filter the corresponding instances using their attributes and properties values.



**Fig. 1.4** Selection of Encounters of a patient in the navigation of Rhizomik in the diabetes case study from <http://rhizomik.net/diabetes>.

Showing 40 **Encounterid** filtered from 101766  
 where **Isadministrativeprocedureof** is **88785891**

**152188656 a Encounterid**

a1Cresult	None
acarbose	No
acetoexamide	No
admission source id	Emergency Room
admission type id	Urgent
age	[20-30)
change	Ch
citoglipton	No
diabetesMed	Yes
discharge disposition id	Left AMA
examide	No
gender	Female
glimepiride	No
glimepiride-pioglitazone	No
glipizide	No
glipizide-metformin	No
glyburide	No
glyburide-metformin	No
hasdisease	V15, 250.11 and 244
insulin	Up
isadministrativeprocedureof	88785891
max glu serum	None
medical specialty	Surgery-General
metformin	No
metformin-pioglitazone	No
metformin-rosiglitazone	No
miglitol	No
nateglinide	No
num lab procedures	33
num medications	10
num procedures	0
number diagnoses	6
number emergency	1
number inpatient	10
number outpatient	4
payer code	BC
pioglitazone	No
race	Caucasian
readmitted	<30
repaglinide	No
rosiglitazone	No
time in hospital	1
tolazamide	No
tolbutamide	No
troglitazone	No

**Fig. 1.5** Description of an encounter of the person with most number of encounters (40) from <http://rhizomik.net/diabetes>.

As part of the "Zoom and Filter" task, the user can use the facets corresponding to the RDF properties describing the selected class instances to filter them. For instance, for the "Encounter" class there is the "is administrative procedure of" facet that corresponds to the same name property that links the encounter to its patient. If the values for this facet are shown, the identifiers for the 5 patients with most encounters are displayed. Then, if for instance patient 88785891 is clicked, the one with most encounters, the 40 encounters for this patient are filtered from the initial selection of 101766 encounters, i.e. all the encounters. Although it is not shown in Fig. 1.4, this kind of filtering can be performed for all facets.

Continuing with the case of the encounters for patient 88785891, it is now possible to switch to the last data analysis task, "Details". For instance, when clicking to the 6th encounter of the page, that with identifier "152188656", the details for the procedure are displayed, e.g. it is for a Caucasian woman in her twenties. As also shown in Fig. 1.5, for this procedure the patient admission to the hospital is through the emergency room with an urgency. Moreover, she changes her medication and takes diabetes medication. If looking at ICD-9, we can see that her diagnosis include hazards to health (V15), Diabetes with ketoacidosis, type I [juvenile type], not stated as uncontrolled (250.11) and Acquired hypothyroidism (244) (see <http://www.icd9data.com>). The dose of insulin increases and she takes 10 medications. Finally, another important attribute is that she has been readmitted less than 30 days.

Apart from individual visualization of patients, we can also have an aggregated visualization of patients. For instance in Fig. 1.6 we can see the average number of medications in purple and the average number of diagnosis in brown of encounters with a stable value of metformin and with an increment in their levels of glimepiride (there are 84 encounters after filtering them based on these facet constraints). The maximum value of diagnosis is about 10 and the maximum value of medications is 66. In the right side of Fig.1.6 the related values appear. These values are admission source ID, discharge disposition ID, gender, and the object properties "has disease" and "is administrative procedure of".

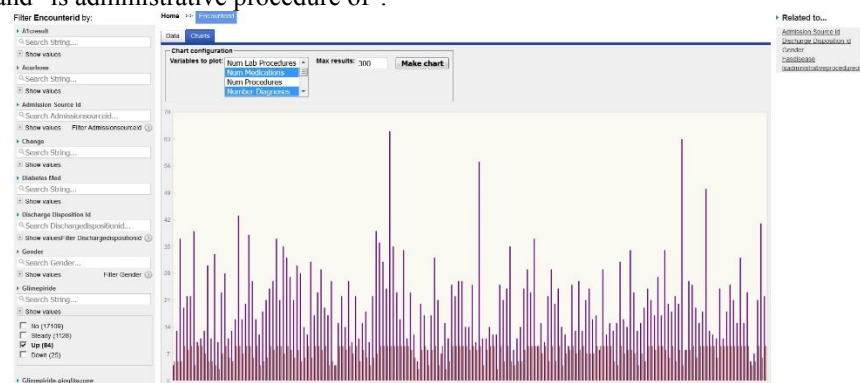


Fig. 1.6 Class diagram of the professional view of the ontology from <http://rhizomik.net/diabetes>.

## 1.5 Conclusions and future work

As shown in the analysis of state of the art, personalization of ontologies visualization is a topic where considerable effort is being put by ontology researchers and especially in the medical domain, where personalized medicine is one of the cornerstones of 4P medicine.

The proposed approach implements the widely-accepted visual data analysis tasks proposed by Shneiderman to the Semantic Web visualization domain. These tasks include, mainly, "Overview", "Zoom and Filter" and "Details". In the proposed approach "Overview" corresponds to providing an overview of the dataset to obtain the full picture of the ontologies used to describe the data. This is implemented with user interface components users are comfortable with as they are used to them: Navigation Menus, TreeMaps, Site Indexes and Site Maps.

The "Zoom and Filter" visual analytics task is also implemented for semantic data, in this case using facets and a faceted view. Finally, "Details" is implemented merely as a listing of the triples describing an RDF subject, thus its properties and values.

To illustrate the feasibility of this approach to the personalization of ontology visualization in the medical domain, it has been applied as part of a use-case to a diabetes dataset and its corresponding ontology. The results are promising and show that the application of this approach has the potential to provide an interactive visualization that can be customized to different user profiles, including lay-users without any technical knowledge beyond Web browsing.

It remains as future work to define the specific data to include or hide, based on real user profiles. The idea is to conduct a card-sorting study in different countries about the personalization of the currently available visualization of both the patient, the professional and policymaker. The implementation of the personalization can be done also using the proposed tool, Rhizomer. It semantically annotates all the generated HTML so it is possible to use CSS (Cascading Style Sheets) to control the visualization of all the user interface elements. Thus, given that all HTML elements for values are annotated with the RDF property they correspond to, a CSS can hide them selectively and CSS for each user profile can be defined.

Finally, other profiles such as service provider could be added in order to have a more complete perspective of the domain. Another challenge would be to integrate the proposed ontology with other existing ontologies such as the Diabetes Mellitus Treatment Ontology and to make it openly available in a biomedical repository such as Bioportal.

#### Abbreviations

ATC: Anatomical Therapeutic Chemical Classification System;  
BFO: Basic Formal Ontology;  
CBR: Case-based reasoning;  
CSS: Cascading Style Sheets;  
DDO: Diabetes Mellitus Diagnosis Ontology;  
DMTO: Diabetes Mellitus treatment ontology;  
EHR: Electronic Health Records;  
HbA1c: Haemoglobin A1c or glicated haemoglobin;  
HTML: Hypertext Markup Language;  
ICD: International Classification of Diseases;  
ICF: International Classification of Functioning, Disability and Health;  
OGMS: Ontology for General Medical Science;  
RDF: Resource Description Framework;  
SNOMED CT: Systematized Nomenclature of Medicine – Clinical Terms;  
SPARQL: SPARQL Protocol and RDF Query Language;  
VOWL: Visual Notation for OWL ontologies;  
WHO: World Health Organization.

#### Ethics approval and consent to participate

Open data is used and it is extracted from UCI machine learning repository. The citation requested has been done.

#### Competing interests

The authors declare that they have no competing interests.

#### Availability of data and materials

Data sharing and code is available under request and the visualization is openly available at <http://rhizomik.net/diabetes>.

#### Author's contributions

R. García., R. Gil and L. Subirats conceived the use case; R. García and R. Gil developed Rhizomik, all authors contributed to the analysis and wrote the paper.

#### Acknowledgments

This research has been partially funded by the Catalonia Competitiveness Agency (ACCÍÓ).

#### References

1. “Stratified, personalised or p4 medicine: a new direction for placing the patient at the centre of healthcare and health education,” tech. rep., Academy of Medical Sciences, 2015.
2. Mayo Clinic, “Patient experience.” <https://www.mayoclinic.org/about-mayoclinic/quality/quality-measures/patient-satisfaction>.
3. A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou, “Ontology visualization methods—a survey,” ACM Comput Surv, vol. 39, no. 10, 2007.

4. N. Noy, N. Shah, P. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. Rubin, M. Storey, C. Chute, and et al., "Biportal: ontologies and integrated data resources at the click of a mouse," *Nucleic Acids Res.*, vol. 37, pp. 170–173, 2009.
5. K. Donnelly, "SNOMED-CT: The advanced terminology and coding system for eHealth," *Stud Health Technol Inform*, vol. 121, pp. 279–290, 2006.
6. P. Whetzel, N. Noy, N. Shah, P. Alexander, C. Nyulas, T. Tudorache, and M. Musen, "Biportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications," *Nucleic Acids Res.*, vol. 39, pp. W541–W545, 2011.
7. L. Subirats, R. Lopez-Blazquez, L. Ceccaroni, M. Gifre, F. Miralles, A. García-Rudolph, and J. Tormos, "Monitoring and prognosis system based on the icf for people with traumatic brain injury," *Int J Environ Res Publ Health*, vol. 12, pp. 9832–9847, 2015.
8. L. Subirats, L. Ceccaroni, R. Lopez-Blazquez, F. Miralles, A. García-Rudolph, and J. Tormos, "Circles of health: towards an advanced social network about disabilities of neurological origin," *J Biomed Inform*, vol. 46, pp. 1006–1029, 2013.
9. M. Calvo, L. Subirats, L. Ceccaroni, M. J. M., de Pablo C., and F. Miralles, "Automatic assessment of socioeconomic impact in cardiac rehabilitation," *Int J Environ Res Publ Health*, vol. 10, pp. 5266–5283, 2013.
10. R. Treede, W. Rief, A. Barke, Q. Aziz, M. Bennett, R. Benoliel, M. Cohen, S. Evers, N. Finnerup, M. First, M. Giamberardino, S. Kaasa, E. Kosek, P. Lavand'homme, M. Nicholas, S. Perrot, J. Scholz, S. Schug, B. Smith, P. Svensson, J. Vlaeyen, and S. Wang, "A classification of chronic pain for icd-11," 2015.
11. World Health Organization (WHO), "Guidelines for ATC classification and DDD assignment. WHO collaborating Centre for Drug Statistics Methodology." <https://www.whooc.no/filearchive/publications/guidelines.pdf>, 2018.
12. S. Lohmann, S. Negru, F. Haag, and T. Ertl, "Visualizing Ontologies with VOWL," *Semantic Web Journal*, 2015.
13. Y. Fukazawa, T. Naganuma, K. Fujii, and S. Kurakake, "Construction and use of role-ontology for task-based service navigation system," in Cruz I. et al. (eds) *International Semantic Web Conference 2006 (ISWC 2006)*. Lecture Notes in Computer Science. vol. 4273, 2006.
14. M. C. Schraefel and D. Karger, "The Pathetic Fallacy of RDF," in *International Workshop on the Semantic Web and User Interaction (SWUI)*, 2006.
15. B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Proceedings of the IEEE Symposium on Visual Languages*, (Boulder, CO, USA), pp. 336–343, IEEE, 1996.
16. "WHO, Facts and Figures about diabetes." <http://www.who.int/diabetes/facts/en>, 2017.
17. R. Kocher and E. Adashi, "Hospital readmissions and the affordable care act: paying for coordinated quality care," *JAMA*, vol. 306, 2011.
18. D. Rubin, "Hospital readmission of patients with diabetes," *Curr Diab Rep.*, vol. 15, no. 17, 2015.
19. "Gencat, Pautes per a l'harmonització del tractament farmacològic de la diabetis mellitus tipus 2. Barcelona: Servei Català de la Salut. Departament de Salut. Generalitat de Catalunya. (Programa d'harmonització farmacoteracèutica de medicaments en l'àmbit de l'atenció primària i comunitària del Servei Català de la Salut 01/2017)," 2017.
20. S. El-Sappagh and M. Elmogy, "A fuzzy ontology modeling for case base knowledge in diabetes mellitus domain," *Engineering Science and Technology, an International Journal*, vol. 20, pp. 1025–1040, 2017.
21. A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Communications*, vol. 7, pp. 39–59, 1994.
22. C. Lee, M. Wang, and H. Hagra, "A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation," *IEEE Trans Fuzzy Syst*, vol. 18, pp. 374–395, 2010.
23. Y. Alfaifi, F. Grasso, and V. Tamma, "Towards an ontology to identify barriers to physical activity for type 2 diabetes," in *In Proceedings of the 2017 International Conference on Digital Health (DH '17)*. ACM, New York, NY, USA, pp. 16–20, 2017.

24. A. Rahimi, S.-T. Liaw, J. Taggart, P. Ray, and H. Yu, "Validating an ontology-based algorithm to identify patients with type 2 diabetes mellitus in electronic health records," *International Journal of Medical Informatics*, vol. 83, no. 10, pp. 768 – 778, 2014.
25. A. Rahimi, N. Parameswaran, P. Ray, J. Taggart, H. Yu, and S. Liaw, "Development of a methodological approach for data quality ontology in diabetes management," *Int. J. E-Health Med. Commun. (IJEHMC)*, vol. 5, p. 5877, 2014.
26. D. Vasant, F. Neff, P. Gormanns, N. Conte, A. Fritsche, H. Staiger, et al., "DIAB: An Ontology of Type 2 Diabetes Stages and Associated Phenotypes," in *Phenotype Day at ISMB 2015*, pp. 24–27, 2015.
27. S. El-Sappagh and F. Ali, "Ddo: a diabetes mellitus diagnosis ontology," *Applied Informatics*, vol. 3, p. 5, Aug 2016.
28. S. El-Sappagh, D. Kwak, F. Ali, and K.-S. Kwak, "Dmto: a realistic ontology for standard diabetes mellitus treatment," *Journal of Biomedical Semantics*, vol. 9, p. 8, Feb 2018.
29. R. Scheuermann, W. Ceusters, and B. Smith, "Toward an ontological treatment of disease and diagnosis," in *Summit on Translational Bioinformatics*, pp. 116–20, 2009.
30. R. Arp, B. Smith, and A. Spear, *Building Ontologies with Basic Formal Ontology*. MIT Press, 2015.
31. P. Morville and L. Rosenfeld, *Information Architecture for the World Wide Web*. Cambridge, MA: O'Reilly Media, 3rd ed. ed., 2006.
32. B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Trans. Graph.*, vol. 11, pp. 92–99, Jan. 1992.
33. N. Elmqvist, P. Dragicevic, and J. Fekete, "Colorlens: Adaptive color scale optimization for visual exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 99, 2010.
34. J. Goldberg and J. Helfman, "Enterprise network monitoring using treemaps," in *Proceedings of the Human Factors and Ergonomics Society*, vol. 49, pp. 671–675, 2005.
35. J. Brunetti, R. García, and S. Auer, "From overview to facets and pivoting for interactive exploration of semantic web data," *Int. J. Semantic Web Inf. Syst.*, vol. 9, p. 120, 2013.
36. J. L. González-Sánchez, R. García, J. M. Brunetti, R. Gil, and J. M. Gimeno, "Using SWETQUM to Compare the Quality in Use of Semantic Web Exploration Tools.," *Journal of Universal Computer Science*, vol. 19, no. 8, pp. 1025–1045, 2013.
37. J. M. Brunetti and R. García, "User-centered design and evaluation of overview components for semantic data exploration," *Aslib Journal of Information Management*, vol. 66, no. 5, pp. 519–536, 2014.
38. R. García, R. Gil, J. M. Gimeno, E. Bakke, and D. R. Karger, "BESDUI: A Benchmark for End-User Structured Data User Interfaces," in *The Semantic Web ISWC 2016, Lecture Notes in Computer Science*, pp. 65–79, Springer, Cham, Oct. 2016.
39. B. Strack, J. DeShazo, C. Gennings, J. Olmo, S. Ventura, K. Cios, and J. Clore, "Impact of hba1c measurement on hospital readmission rates: Analysis of 70000 clinical database patient records," *BioMed Research International*, vol. 11, 2014.
40. J. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Symposium on Computer Applications and Medical Care*, pp. 261–265, IEEE Computer Society Press, 1988.
41. G. M. Uschold, M., "Ontologies: Principles, methods and applications," *Knowledge Engineering Review*, vol. 2, no. 11, 1996.
42. M. Poveda-Villalón, M. Suárez-Figueroa, and A. Gómez-Pérez, "Validating ontologies with oops!," *Knowledge engineering and knowledge management*, pp. 267–281, 2012.