Drug Encyclopedia – Linked Data Application for Physicians

Jakub Kozák(✉), Martin Nečaský, and Jaroslav Pokorný

Faculty of Mathematics and Physics, Charles University in Prague,
Prague, Czech Republic
{kozak,necasky,pokorny}@ksi.mff.cuni.cz

Abstract. The information about drugs is scattered among various resources and accessing it is hard for end users. In this paper we present an application called Drug Encyclopedia which is built on the top of the data mart represented as Linked Data and enables physicians to search and browse clinically relevant information about medicinal products and drugs. The application has been running for more than a year and has attracted many users. We describe the development driven by requirements, data mart creation, application evaluation and discuss the lessons learned.

Keywords: Linked Data application · Drugs · Health care · RDF · SPARQL

1 Introduction

There is a lot of information about medicinal products and drugs which a physician should know. Although they can be found at many places, it is hard to navigate through the piles of data which might be in different languages, formats, quality, etc. And moreover, the information change quickly in time as new products are launched. Therefore we decided to go to the source and asked physicians about their needs in the area of drug related information and also conducted a little survey among them. The results were not surprising. The physicians would like to have a single source of information where clinically relevant information about drugs and medicinal products would be brought together and made available in a user friendly way.

In this paper, we present a web application for physicians, called *Drug Encyclopedia*[1], which is based on semantic web technologies and enables the end users to search and explore the clinically relevant information about medicinal products and drugs in general. The data architecture and application development were mainly driven by the user requirements. The application was designed to match the criteria of easy usage and also flexible development. This was achieved with the help of semantic web technologies as can be seen further in the paper.

[1] http://www.lekovaencyklopedie.cz

M. Arenas et al. (Eds.): ISWC 2015, Part II, LNCS 9367, pp. 41–56, 2015.
DOI: 10.1007/978-3-319-25010-6_3

The application has been running since the end of 2013, attracted thousands of users and found regular users among physicians.

The paper is structured as follows. In Section 2, we describe the user requirements. Then we show the data architecture in Section 3. The application and its evaluation are presented in Section 4 and Section 5 respectively. Related work is described in Section 6 and finally, lessons learned are given in Section 7.

2 Use Case

Before we started to build the application, we had conducted a survey among physicians in the Czech Republic to gather their information needs in the drug domain. We collected 43 responses from physicians (diabetologists, obstetricians, ophthalmologists, etc.) using a web questionnaire complemented with screen shots demonstrating possible functionality. For more details please refer to [7]. The following functionality was considered by the physicians as highly desirable:

1. For a medicinal product and/or active ingredient show its indication, classification group, contraindication and interactions with other medicinal products and/or active ingredients. Moreover, information about risks of prescribing a medicinal product to a pregnant women is required by obstetricians.
2. For a list of medicinal products show whether they have the same active ingredients or belong to the same classification group or whether there are some interactions among them.
3. For a medicinal product show selected parts of textual documentation of the product, specifically extracted from Summary of Product Characteristics.
4. For an active ingredient show advanced clinical information i.e. pharmacological action, pharmacokinetics etc.
5. Find interactions in a set of medicinal products and/or active ingredients.

The use case is then quite simple – end users (i.e. physicians) want to be able to explore and search available information about drugs and medicinal products (coming from highly heterogeneous data sources) in a single application. The application should be easily extensible with further data sources in the future.

3 Data Architecture

Driven by the functional requirements described in the Section 2, we have selected the following data sources with relevant pieces of information for the application:

- **(CZ-DRUGS) Medicinal products registered in the Czech Republic** – data provided by the State Institute for Drug Control (SIDC) about medicinal products marketed in the Czech Republic – including prices.
- **SPC (Summary of Product Characteristics)** – documents attached to each marketed medicinal product and intended for professionals.

- **MeSH (Medical Subject Heading)** – a reference dictionary for linking other sources. It is partially translated to Czech and many other languages.
- **NDF-RT (National Drug File - Reference Terminology)** – data about indications, contraindications and data about pharmacological effects.
- **DrugBank** – data about interactions and descriptions of active ingredients.
- **ATC Hierarchy (Anatomical Therapeutic Chemical Classification System)** – classification of drugs maintained by WHO.
- **NCI Thesaurus** – direct links to FDA SPL.
- **FDA SPL (FDA Structured Product Labeling)** – pregnancy category.
- **MedDRA (Medical Dictionary for Regulatory Activities)** – adverse event classification dictionary.

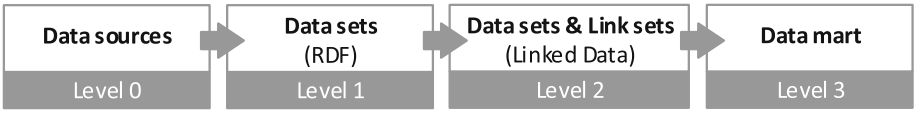

Fig. 1. Data architecture - from data sources to data marts

Because of the heterogeneity of the data sources, we chose to work with Resource Description Format (RDF) [8] and Linked Data (LD) principles [1]. RDF is a format for representing data as triples where each real world entity is represented as a resource (URI) and LD principles give recommendation for publishing RDF. RDF data respecting LD principles are simply called LD.

The data architecture we designed to represent and integrate the selected data sources is depicted in Figure 1. It is logically structured into 4 levels described in the rest of this section.

Level 0 (L0) – Level 2 (L2). We have collected the data sources from L0 described above, transformed them into RDF and loaded them into a triple store. Each data set occupies one graph in the triple store at L1. We have used Virtuoso[2] as a triple store. Then we have applied several strategies for linking the data sets and created a data warehouse represented by data sets and link sets, i.e. L2. The data architecture is shown in the Figure 1 and more details can be found in our previous work [7] and [6]. According to the notation in the Figure 1, our previous work stopped at L2 which represents a LD warehouse of the integrated data sources. The LD warehouse contains about 120 M triples where about 80 M belong to the representation of SPCs.

Level 3 (L3) - Data mart. Because the interlinked data sets are usually large, contain also data irrelevant to a specific application, and it is hard to navigate in them, we introduce Level 3 called *data mart* which should enable developers to create applications more easily. We have borrowed this name from the data

[2] Virtuoso – http://virtuoso.openlinksw.com

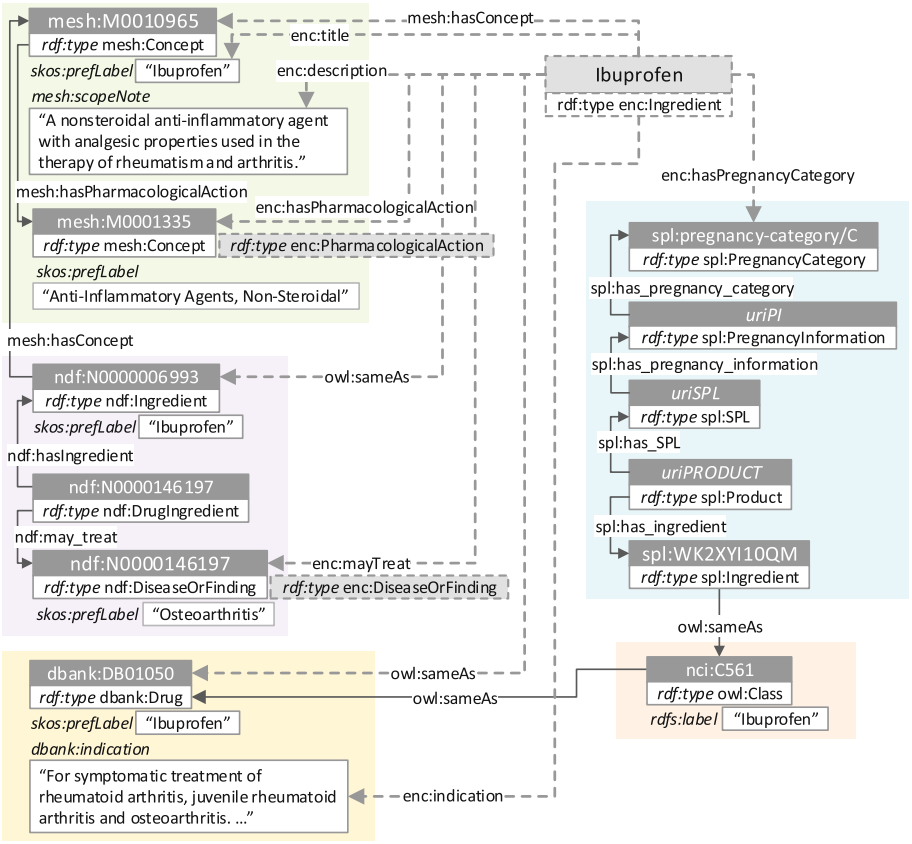

Fig. 2. Part of the triple store. The dashed lines represent the information in the data mart.

warehousing where it means a set of data optimized for specific business or application needs. Here, we have the same need – to make data available in such form that it would be easy and fast to access them. Therefore we make two kinds of optimization – static and dynamic. The *static optimization* simplifies the schema and makes the queries easier to be written. The *dynamic optimization* shortens the paths through the original data sets significantly by adding new triples connecting the resources directly.

To achieve the optimization, we construct the data mart as follows. We add new classes and properties, i.e. we create an application ontology. The new classes are then assigned to the existing resources from L2 in order to distinguish them for the application. The new properties then usually shorten the important paths in the data. Besides that, we also construct new resources which unify underlying resources from different data sets into one. This results into a simpler schema.

All the transformations are represented as SPARQL queries. The schema of the data mart is not derived automatically but rather manually by a domain expert.

We describe the procedure of creating the data mart on the example shown in the Figure 2 where a subset of the data is visualized. Each of the rectangles in the background represents a data set from L1. The dashed lines and rectangles with dashed borders represent new data (triples) from L3 – data mart. The example shows a part of the data for active ingredient *Ibuprofen*.

A new resource was created for the active ingredient *Ibuprofen* which is also linked to the original data sets, e.g. by `owl:sameAs` property. New types are assigned to the important resources from L2, i.e. pharmacological action and disease or finding. These entities are linked to the new active ingredient with properties from the application ontology, e.g., `enc:hasPharmacologicalAction`. This significantly simplifies the schema. When we take a look at the pregnancy category, it is originally available only by traversing several data sets with a long path. We optimize the schema (i.e. also the queries) by connecting the active ingredient directly to the pregnancy category by `enc:hasPregnancyCategory`.

The data mart simplifies the queries and the whole logic. On the other hand, it takes some storage space. Our data mart added 8 M triples to the triple store. The benefits of the data mart are also shown in the Section 4 and the Section 7.

Automation. It is necessary to update the data from the data sources regularly. Most of the data transformation is now done automatically with only minimal manual work needed. The data transformation has been done in *UnifiedViews*[3], an extract-transform-load (ETL) tool for RDF [5]. If the data changes in the original data sources, no manual inference is necessary. If the schema changes, the ETL pipelines have to be adjusted.

4 Drug Encyclopedia – The Application

Drug Encyclopedia is a publicly available web application which enables users to search and explore data about medicinal products and drugs from different data sources. The user can search for a required medical resource (medicinal product, active ingredient, indications, pharmacological actions, etc.) and view its detail. The detail shows information about the resource available in the application data mart. The application is not bound to specific data structures – it is able to show any data associated with the resource in the data mart. (Technically, it is able to show any surrounding of the resource in the RDF representation). From the detail page, the user can browse to other related resources via the links stored in the data mart. We describe how those three features (search, showing detail and browsing) benefit from semantic technologies in the rest of this chapter. We also present two examples of advanced features built in the application.

[3] http://www.unifiedviews.eu

4.1 Full Text Search of Resources in the Application Data Mart

The starting page of the application offers a full-text search field which allows a user to search resources in the data mart. When the user types at least 4 letters, the full-text search engine searches the application data mart for resources whose literal properties values contain the searched string and provides them to the user. The search is implemented as a SPARQL query which can be configured by the administrator to search only for resources of a given type and restrict the search only to given properties of those resources. All available language specific values of the configured properties are searched independently of the current language set by the user as the user interface language. For example, the user may specify the search string in Latin (physicians often use Latin) even if he has set Czech as his current language. The result is always shown in the language selected by the user, i.e. Czech in our case. In the data mart, we currently have textual properties of resources translated to Czech, English and Latin. Therefore, the user can type his search string in any of these three languages. The user can select Czech and English as his language for the application user interface. Latin is not supported for the application user interface.

Semantic technologies enable us to easily implement the above mentioned configuration of the search. We can implement advanced search strategies because of the flexibility of the RDF data model and SPARQL query language. A single SPARQL search query can contain different search strategies for RDF resources of different classes. E.g., we can say that all resources of sub-classes of `skos:Concept` (e.g., ATC group) are searched using the property `skos:prefLabel` while instances of a specific application class `enc:MedicinalProduct` are searched using specific application property `enc:title` which unifies various equivalent properties used for expressing a displayable title of a resource used by different data sets integrated in the application data mart. Moreover, if the schema of the data mart will be extended in the future (e.g., a new class and its instances will be added) it will be very easy to modify the search engine – it will be only necessary to extend the SPARQL search query.

Figure 3 shows the application user interface for searching. Every time the user writes at least 4 letters into the search field, a SPARQL query is executed and the search result is shown in the auto-complete panel (see the Figure 3 (A)). The SPARQL search query is also executed when the user clicks on the search button (see the Figure 3 (B)). Because of the simplicity, the auto-complete panel shows only a subset of the search result of all search strategies.

Formally, each class whose resources should be searched by the application is associated with a *search strategy*. A search strategy is a SPARQL graph pattern. All search strategies are concatenated using the SPARQL `UNION` construct. A search strategy must contain a variable `?resource` which matches the found resource, `?type` which matches the type(s) of `?resource` and `?text` which matches the found textual value of a property of `?resource`. The resulting SPARQL search query then constructs the search results from the union of the search strategies in the following form:

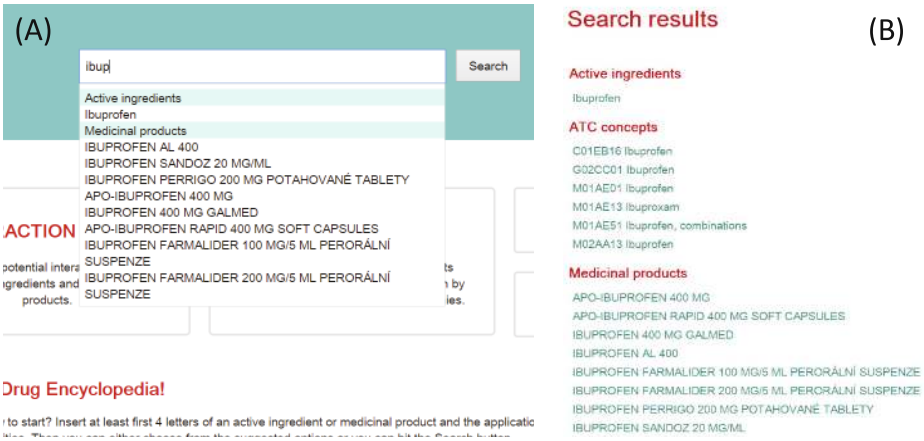

Fig. 3. Searching resources in the application data mart with Drug Encyclopedia

```
?resource a ?type ;
  enc:title ?text .
```

(The application understands the value of `enc:title` as a text to be displayed to the user.) Let us demonstrate the full-text search on a concrete example. When the user types *"ibup"* to the search field, the search engine uses all defined search strategies to explore the data mart. The following strategies are successful (i.e. return a non-empty result):

- The search strategy for class `enc:Ingredient` matches each active ingredient I with a value t of property `enc:title` such that t starts with *"ibup"*. For I, t is assigned to `?text`.
- The search strategy for class `enc:ATCConcept` matches each ATC group A with a value l of property `skos:prefLabel` such that l starts with *"ibup"*. For a found ATC group, it constructs the value of `?text` as a concatenation of the code of the ATC group (`skos:notation`) followed by the title of the ATC group (`skos:prefLabel`).
- The search strategy for class `enc:MedicinalProduct` matches all medicinal products in the same way as the search strategy for `enc:Ingredient`.

The search strategies for other classes, e.g., `enc:DiseaseOrFinding`, do not find any resource for the specified search string. If there is a requirement to change the searching logic (e.g., a new kind of resources is added), it is only necessary to add a new search strategy or to modify the existing one.

4.2 Viewing a Detail of a Resource in the Application Data Mart

When the user clicks on one of the resources found by the search engine, the application shows the detail of that resource. The detail page renders the displayable title of the resource (value of `enc:title`), its displayable description

(value of `enc:description`) and then other properties of the resource stored in the application data mart. Values of each property are displayed in a paragraph with the name (value of `rdfs:label`) of the property in its title.

Figure 4 contains 2 screen shots – a detail of the active ingredient *Ibuprofen* (on the left) and a detail of the medicinal product *Brufen 400*. Both details are displayed using the same page template – the title is shown at the top together with the class of the resource (value of `rdfs:label` of that class). Then there is a description of the resource below the title if it is available. The paragraphs for the values of properties of the resource are displayed below. For example, for *Ibuprofen* (on the left), there are paragraphs *Indication*, *Prevention*, *Contraindication*, *Pharmacological actions*, *Mechanisms of actions*, etc.

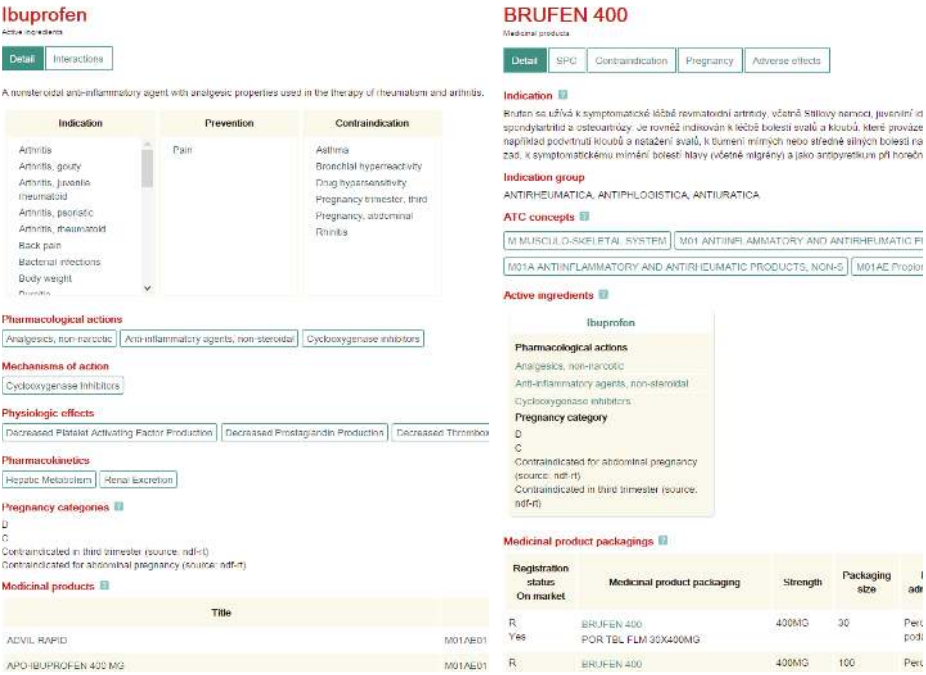

Fig. 4. Showing a detail of a resource (left: active ingredient *Ibuprofen*, right: medicinal product *Brufen 400*) in Drug Encyclopedia

The different layouts for both detail pages in Figure 4 are not given by different page templates. There is only one page template for details of all types of medicinal resources in the application data mart. The different layouts are the result of a mechanism which assigns a specific visual configuration to each property. The configuration is applied anywhere the property appears. However, there is not a specific configuration for each specific property. Instead, the configuration is specified in a more semantic way.

There is a basic configuration saying how literal properties should be displayed and another basic configuration for object properties. If no more specific visual configuration is available, the basic one is applied for displaying the visual paragraph with the values of the property. For example, the basic configuration for object properties is used to display values of properties *Pharmacological actions* and *Mechanisms of action* in the detail of *Ibuprofen* on the left of Figure 4.

Moreover, there can be more specific visual configurations for specific kinds of properties with specific object values. For example, we have a specific visual configuration for the following kinds of object values:

- An object value which is an instance of `skos:Concept` and is related to the displayed resource with `skos:broader`. For this object value we show also the `skos:broaderTransitive`s. The configuration is applied to render the *ATC Concepts* paragraph in the detail of *Brufen 400*. Here, we can see not only the ATC concept *M01AE01* but also all its broader transitive concepts – *M01AE*, *M01A*, *M01*, and *M*.
- Object values which are instances of `enc:Ingredient` - whenever an active ingredient is displayed, we do not display it as a generic object but we show a richer visualization which presents also other selected properties of the active ingredient (we show its pharmacological actions and pregnancy categories in the visualization). This configuration is applied to render the paragraph *Active ingredients* in the detail of *Brufen 400*.
- Object values which are related to an instance of class `enc:Interaction` with `enc:interactionMember` property such that the displayed resource is also related to this instance of `enc:Interaction` with the same property. The meaning of such relationship is that the displayed resource (e.g, a medicinal product, active ingredient or ATC group, etc.) interacts with the object value (again a medicinal product, active ingredient or ATC group, etc.). Moreover, the instance of `enc:Interaction` may be linked to an SPC with `frbr:textChunk` property. If the interaction is linked to an SPC it means that the interaction has been extracted from the SPC using NLP techniques. (We described the extraction mechanism in our previous paper [6]). The visualization shows these interactions as links to the corresponding place in the SPC where the interaction is described. The label of the link is the title of the other interacting object. The user can see this configuration in action when he chooses a detail of a medicinal product packaging (accessible from the detail of a medicinal product). It is applied for rendering the paragraph *Interactions (extracted from SPC)*.

For a detail page, a SPARQL query which extracts data about the displayed resource from the database is executed. There is a basic SPARQL query which extracts all triples with the displayed resource as a subject. The basic query can be extended by the administrator for selected application classes – any SPARQL query which returns the displayed resource and its surroundings can be provided instead of the basic query. The custom query can, e.g., ignore some properties or go beyond the distance 1 for some other resources. As a result, each application class is associated either with the basic query or a custom one.

Then, the application automatically analyzes the obtained data and assigns a visual configuration to each found property related to the displayed resource. This is possible because of the basic visual configurations which assign a visual configuration to each property and are described above. The analysis means that each template is checked whether it can visualize a given property which appears in the data. The template implements a method which is able to traverse the property and its values (and their properties and so on) and check whether the data contain everything what is necessary for the template. This can be an expensive process. However, because the assignment is done only on a very small portion of data, it is computed quickly.

This dynamic execution model is very flexible. We are able to display the detail of any kind of resource and use predefined visual configurations to render the detail page without the need of programming a specific code for each kind of displayed resources. This is useful mainly when new information to existing kinds of resources is added, when the representation of existing information is changed or when completely new kinds of objects are added. The application code has to be updated or extended only when a specific visualization of a property is required. When the data model is changed but there is no requirement to change the visual style, no intervention in the application code is required.

The mechanism built on top of semantic technologies described above also facilitates the maintenance of the consistency of the application logic and visual style within the whole application. A group of properties with the same or similar semantics is displayed and behaves in the same way across the whole application because their visual configuration is chosen on the base of their semantics instead of a programmatic selection directly in the application code. For example, anywhere in the application where a property with `skos:Concept` as its range is displayed, the visual style and behavior is the same if this rule is not explicitly overridden by another visual configuration.

4.3 Browsing Resources

When a detail page is displayed, values of object properties are displayed as links to detail pages of those object values. The basic visual configuration for object properties ensures this behavior. When a more specific configuration is defined it is required to display the objects as links as well. The link is an application URL which contains the URI of the object as a parameter. The mechanism for generating the detail page described above ensures that the detail of the linked object is compiled automatically. Therefore, when the user clicks on the link, the detail page of the object is displayed. By following the links, the user can browse the whole data mart because the data mart forms a connected graph.

The physician can use this browsing feature to explore the application data mart. He can, for example, discover medicinal products which are more suitable for his patient than the one he currently uses. For example, the patient uses the medicinal product *Brufen 400*. However, the patient is pregnant and the active ingredient of this medicinal product is contraindicated in pregnancy (which the physician can see on the detail of *Ibuprofen* in *Pregnancy categories*

paragraph). Therefore, he can browse the data mart to explore active ingredients and medicinal products containing those active ingredients using the links to pharmacological actions, mechanisms of action, etc. from the detail of *Ibuprofen*.

4.4 Advanced Features

The application contains several advanced features. Let us pick up two of them. First, there is an interaction finder. In the application data mart, there are integrated interactions between pairs of active ingredients from different data sources. The application provides a feature which allows a user to specify a set of active ingredients and/or medicinal products and then check the potential interactions among them. The interactions are discovered using a relatively complicated SPARQL query (its effectiveness is discussed in Section 5).

Second, there is a possibility to display SPC documents we have already discussed in the Section 4.2. An SPC document is a textual document with a prescribed structure of sections and subsections which provides a clinical information about a medicinal product to a physician. We have processed all SPC documents of medicinal products available on the Czech market – we converted them to RDF representation (we used *SALT* and *FRBR* ontologies[4]) and annotated medical resources in the text (medicinal products, active ingredients, etc.). The details of this process are described in our previous paper [6]. The application enables a user to view the SPC document from the detail page of each medicinal product. When the SPC document is displayed, the recognized medical resources are displayed as links to their details. Therefore, SPC documents become a part of the browsing feature provided by the application. A user can go to the SPC document from a detail page of a medicinal product and from here he can browse to the details of related medical resources.

5 Evaluation

The application was deployed at the end of 2013 and has been running since then. More than 3,300 unique users used our application at least once and about 57 % of them returned to the application during the time. In this section, we present the evaluation of performance of the application and statistics about usage of the application and user satisfaction.

5.1 Performance

The application is running on a virtual server with dedicated 10 GB of RAM and can use up to 16 cores of CPU (Intel Xeon CPU E5620 @ 2.40GHz). As mentioned before, we use Viruoso as the underlying triple store. We have designed a test to evaluate the performance of the application queries. The test contains 11 different application queries which are run for 10 times sequentially. The run times of the

[4] Can be found at Linked Open Vocabularies: http://lov.okfn.org/

queries and the number of resulting triples are shown in the Table 5.1. All queries run less than 1 second except the detail of an ATC concept and interaction finder. The reason for long run time of interaction finder is the complexity of the task and the query. It checks interaction among 4 entities – 2 medicinal products and 2 active ingredients – and searches for every pair that could possibly interact.

Table 1. Execution time of the application queries in milliseconds and triples in result.

Action	MIN	MAX	AVG	# triples
Searching for *ibup*	363	496	391	53
ATC Concept detail (M01AE01 Ibuprofen)	1,373	1,509	1,440	226
Disease or Finding detail (Ostheoartritis)	571	742	619	960
Ingredient detail (Ibuprofen)	130	195	148	466
Medicinal Product detail (BRUFEN 400)	571	742	619	147
Medicinal Product Packaging detail (BRUFEN 400, POR TBL FLM 10X400MG)	648	891	756	763
Pharmacological Action detail (Anti-inflammatory agents, non-steroidal)	450	690	493	2,540
Mechanism of Action detail (Cyclooxygenase Inhibitors)	110	212	132	542
Physiological Effect detail (Decreased Prostaglandin Production)	184	264	218	907
Pharmacokinetics detail (Renal Excretion)	664	964	218	2,071
Interactions Finder	6,416	10,337	7,065	207

5.2 Application Usage

Here, we present the statistics collected via Google Analytics[5] (GA) during the period between 2014/01/01 and 2015/02/28. The collected data does not contain any sessions triggered by developers of the application because they use a special attribute for accessing the application even for the purpose of presentations. Besides the statistics from GA, we also briefly present results from a questionnaire which was filled by 13 physicians who are users of the application.

GA Statistics. The application has attracted 3,324 unique users during the presented period which means 7,635 sessions and 35,133 pageviews in total. The monthly statistics can be seen in the Figure 5 (A). It results in almost 17 sessions per day. These numbers of sessions and pageviews (and corresponding statistics below) include 2,198 sessions when only homepage was visited and then the application was left immediately. We are not able to filter them out from all of the statistics because GA does not allow such fine grained export. We call these sessions "dead sessions" further on.

The users were coming to the application continuously except 2 peaks in 2014/01 and 2015/01 when little promotion was done. There were almost no users coming during weekends. Besides that, users are returning to the application. 57 % of all users used our application more than once. More detailed analysis shows that 37 % users came to the application more than 9 times. The

[5] Google Analytics – tracks website traffic – http://www.google.com/analytics

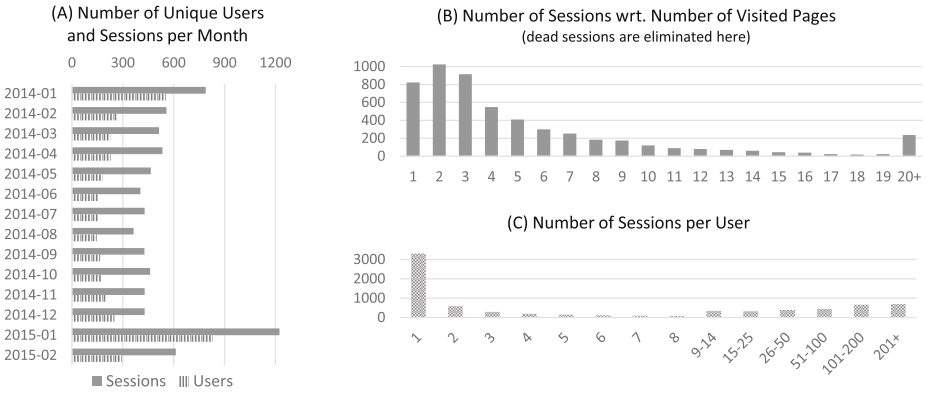

Fig. 5. Statistics about sessions in the application - numbers were exported from GA.

distribution of sessions per user can be found in Figure 5 (C). About 10 % of users were coming from mobile devices.

Interesting facts arise from the analysis of pages visited per session. There is 40 % of sessions with 5 or more visited pages. Thanks to the flow analysis available in GA we may say this represent the exploratory way of work rather than quick information lookup. The distribution of numbers of pages visited in one session is shown in the Figure 5 (B).

We have also set up GA to track types of pages visited. We exclude the static pages (home, about application etc.) from the following statistics. The most of the pageviews come from types medicinal product or its packaging - 11,542 pageviews. Then there were 3,557 pageviews of SPCs, 2,427 pageviews of ingredients, 2,270 pageview of ATC concept and 1,516 pageviews of interactions finder. The interaction finder logs only the first access to the mini application. The number of submitted interaction checks was not logged in the GA.

Survey. We conducted a small survey among the users of *Drug Encyclopedia* and collected 13 responses from the users, mainly physicians. The results of the 3 most important questions can be found in Figure 6. Questions were the following.

Q1 Do you usually find information you are looking for?

Q2 Does Drug Encyclopedia offer more information than comparable information sources in Czech?

Q3 Is Drug Encyclopedia more user friendly than other comparable information sources or products?

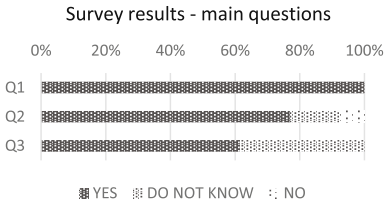

Fig. 6. Brief survey results

The results show the usefulness of the application from the users' perspective.

6 Related Work

LD was used in many domains of interest and many projects were also done in the area of biomedicine. One of the most important projects for sharing biomedical ontologies and terminologies is definitely BioPortal [11], [12]. Other projects try to give users interconnected databases of general biomedical data. These include, e.g., Bio2RDF [2], Linked Life Data[6]. There exist projects dealing specifically with drug related data. Linked Open Drug Data (LODD) was one of the first which published drug data as LD [13]. The major project for drug discovery is currently OpenPHACTS which integrates many databases about drugs [15].

All the projects above provide platforms for data discovery in the biomedical/pharmaceutical domains. They are not intended for end users, i.e. physicians or patients. When a person needs to get information relevant to clinical practice he needs to go somewhere else. Specialized applications have to be developed for this purpose. There is not so many applications build on top of LD.

LODMedics [14] does that for information about drugs. But it is a very simple application. There also exists an application called Pharmer [4] which tries to target semantic prescription of drugs. It uses data from LODD and is rather in the early stage of development. Neither of the mentioned applications allow browse the LD space e.g., show all drugs for a specific pharmacological action.

The complexity of underlying data (often integrated from many data sources) is a tough hurdle for creating such application and is necessary to make it easier e.g., by providing a reasonable API [3] with a simplified data model. The model can be represented by so called application ontology which is a subset of the full reference ontology [10]. This approach was used e.g., in [9].

7 Lessons Learned and Future Work

During our work, we learned several lessons about developing a web application using semantic web technologies. We learned that semantic technologies make the process of integrating various publicly available data sources much more flexible and easier. Moreover, the application code can be more flexible. The proposed mechanism of visual templates helps to reduce the amount of code. This significantly saves time of the database administrators and developers. We further list some important lessons learned.

Iteration of RDF Representation. Using RDF as a data model has several advantages. It is easy to convert the data sources to an RDF representation. At the beginning, it is sufficient to create some RDF representation which reflects the structure of the original data sources one-to-one. In the next iterations, the RDF representation of each data source can be improved by aligning the RDF structure with existing ontologies and vocabularies. Each particular data source and each link set between two data sources should be logically organized in a separate RDF graph. This allows the database administrator to see how each

[6] Linked Life Data – http://linkedlifedata.com

data source and link set was changed and improved during the alignment and linking iterations. This makes the discovery of errors in the alignment and linking procedures easier.

Application Data Mart. The LD warehouse should be further optimized with respect to the typical application SPARQL queries. As we describe in Section 3, optimization means simplification (unification of properties and path shortening). The optimization is useful because the resulting SPARQL queries are more easily maintainable and can be evaluated more effectively by a triple store. Let us note that these simplification is materialized in the database as a set of new triples which should be stored in separate RDF graphs.

Visual Templates for Application Data. One of the most important lessons learned during our work was that the application code should be as generic as possible. This does not mean to create an application which serves as a generic LD browser. Rather it means that there should be a single template which shows all resources in the application data mart in a uniform way and which unifies the visual style of common properties across different kinds of resources. We implemented this practice by introducing generic visual configurations which are applicable to any RDF property and specific visual configurations which are applied to selected properties (details in Section 4). Therefore, it is not necessary to code a specific template for each type of resources. Instead, there is only one template and then visual configurations for particular kinds of properties which are dynamically applied when a resource is displayed. Such approach reduces the amount of code and makes the application code maintenance much easier.

Future Work. Because the LD warehouse and the application are ready for integration of other data sources, we are planning to extend the application in the near future. We would like to add links to current literature about drugs and also spread the application to other countries where local medicinal products are on the market. Moreover, we are also planning to build a mobile application with specific functions for patients.

Acknowledgments. This work was partially supported by the project GACR P103/13/08195S and the project SVV-2015-260222.

References

1. Berners-Lee, T.: Linked Data (2006). http://www.w3.org/DesignIssues/LinkedData.html (accessed: April 24, 2015)
2. Callahan, A., Cruz-Toledo, J., Ansell, P., Dumontier, M.: Bio2rdf release 2: Improved coverage, interoperability and provenance of life science linked data. In: The Semantic Web: Semantics and Big Data, pp. 200–212. Springer (2013)
3. Chichester, C., Harald, L., Harder, T.: Mobile applications driven by open phacts semantic web technology. EMBnet. Journal **19**(B), 21–23 (2013)
4. Khalili, A., Sedaghati, B.: Semantic medical prescriptions-towards intelligent and interoperable medical prescriptions. In: 2013 IEEE Seventh International Conference on Semantic Computing (ICSC), pp. 347–354. IEEE (2013)

5. Knap, T., Skoda, P., Klimek, J., Necasky, M.: Unifiedviews: towards ETL tool for simple yet powerfull RDF data management. In: Proceedings of the Dateso 2015 Workshop, pp. 111–12 (2015)
6. Kozák, J., Necaský, M., Dedek, J., Klímek, J., Pokorný, J.: Using linked data for better navigation in summaries of product characteristics. In: Proceedings of the 6th International Workshop on Semantic Web Applications and Tools for Life Sciences, Edinburgh, UK, December 10, 2013
7. Kozák, J., Nečaský, M., Dědek, J., Klímek, J., Pokorný, J.: Linked open data for healthcare professionals. In: Proceedings of International Conference on Information Integration and Web-Based Applications & Services, pp. 400–409. IIWAS 2013. ACM, New York (2013)
8. Manola, F., Miller, E.: RDF Primer. W3C Recommendation (February 10, 2004)
9. Mejino, J.L.V., Rubin, D.L., Brinkley, J.F.: FMA-Radlex: an application ontology of radiological anatomy derived from the foundational model of anatomy reference ontology. In: AMIA Annual Symposium Proceedings, pp. 465–469 (2008)
10. Menzel, C.: Reference ontologies - application ontologies: either/or or both/and? In: Proceedings of the KI2003 Workshop on Reference Ontologies and Application Ontologies, Hamburg, Germany, September 16, 2003
11. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., Musen, M.A.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research (2009)
12. Salvadores, M., Horridge, M., Alexander, P.R., Fergerson, R.W., Musen, M.A., Noy, N.F.: Using SPARQL to query bioportal ontologies and metadata. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part II. LNCS, vol. 7650, pp. 180–195. Springer, Heidelberg (2012)
13. Samwald, M., Jentzsch, A., Bouton, C., Kallesœ, C., Willighagen, E., Hajagos, J., Marshall, M., Prud'hommeaux, E., Hassenzadeh, O., Pichler, E., Stephens, S.: Linked open drug data for pharmaceutical research and development. Journal of Cheminformatics **3**(1), 1–6 (2011)
14. Shruthi Chari, S.R., Mahesh, K.: LODMedics: Bringing Semantic Data to the Common Man (2014). http://challenge.semanticweb.org/2014/submissions/swc2014_submission_7.pdf
15. Williams, A.J., Harland, L., Groth, P., Pettifer, S., Chichester, C., Willighagen, E.L., Evelo, C.T., Blomberg, N., Ecker, G., Goble, C., et al.: Open PHACTS: semantic interoperability for drug discovery. Drug discovery today **17**(21), 1188–1198 (2012)