

Sparklis over PEGASE Knowledge Graph: A New Tool for Pharmacovigilance*

Carlos Bobed¹, Laura Douze², Sébastien Ferré¹, and Romaric Marcilly²

¹ Univ Rennes, CNRS, IRISA

Campus de Beaulieu, 35042 Rennes, France

{carlos.bobed-lisbona,sebastien.ferre}@irisa.fr

² Univ. Lille, INSERM, CHU Lille, CIC-IT / Evalab 1403

Centre d'Investigation clinique, EA 2694, F-59000 Lille, France

{laura.douze,romaric.marcilly}@univ-lille.fr

Abstract. Pharmacovigilance is in charge of studying the adverse effects of pharmaceutical products. In this field, pharmacovigilance specialists experience several difficulties when searching and exploring their patient data despite the existence of standardized terminologies (MedDRA).

In this paper, we present our approach to enhance the way pharmacovigilance specialists perform search and exploration on their data. First, we have developed a knowledge graph that relies on the OntoADR ontology to semantically enrich the MedDRA terminology with SNOMED CT concepts, and that includes anonymized patient data from FAERS³. Second, we have chosen and extended a semantic search tool, Sparklis, according to the user requirements that we have identified in pharmacovigilance. We report the results of a usability evaluation that has been performed by human factors specialists to check the benefits of our proposal.

Keywords: Knowledge Graph·Semantic Search·Query Building·Pharmacovigilance·MedDRA·Sparklis·Usability.

1 Introduction

Pharmacology has provided humanity with a huge improvement in life quality. However, while new drugs are thoroughly tested before being released for general consumption, all their possible side effects cannot be completely foreseen. Thus, we need methods to discover and track those adverse effects in order to improve the safety and efficacy of drugs. Pharmacovigilance is defined by the World Health Organization (WHO) as “the science and activities relating to the detection, assessment, understanding and prevention of adverse effects or any other drug-related problem”.

Pharmacovigilance specialists manage and report adverse drug reactions (ADRs) noticed by healthcare professionals and patients to different healthcare authorities. For this, they must codify their reports using one or several terms that closely capture the original verbatim description of the ADRs. In this context, the usefulness of standardized vocabularies to unify the codification of the reports is evident. To this purpose, MedDRA

* This research is supported by ANR project PEGASE (ANR-16-CE23-0011-08), project TIN2016-78011-C4-3-R (AEI/ FEDER, UE), and DGA/FEDER.

³ FDA's Adverse Event Reporting System

(Medical Dictionary for Drug Regulatory Activities)⁴ is recommended by the ICH⁵ for the electronic transmission of individual case safety reports [1], and is used in most countries. However, as pointed out by Bousquet et al. [4], “its main limitation comes from its standard terminological format, which restricts the possibility of accessing terms based on their semantics”. Therefore, depending on their experience, expertise, and interpretation of the meaning of the MedDRA terms, two pharmacovigilance specialists may use different MedDRA terms to code the same report. To solve this problem, Bousquet et al. proposed OntoADR [4], an ontology that describes MedDRA terms by their actual semantics, expressed with a combination of SNOMED CT classes and properties. However, until now, pharmacovigilance specialists could not use those semantic descriptions due to the lack of appropriate tools to help them build the semantic queries required to retrieve sets of MedDRA terms.

In this paper, we present our solution to support pharmacovigilance specialists in the exploration and search of their database of patient cases, the first step in the process of detecting new adverse effects of drugs. First, we studied the existing tools, detecting a lack of proper tools to ease the documentation stage of pharmacovigilance specialists. Then, we built a knowledge graph integrating different knowledge sources, adopting the Semantic Web standards, which makes it possible to have all the relevant data easily accessible, providing the flexibility required to be extended under demand. Finally, in order to improve the way pharmacovigilance specialists search for cases, we adopted and extended Sparklis [5], a query builder that eases the exploration and querying of any SPARQL endpoint, without requiring to master SPARQL itself. This work takes place in the PEGASE project, whose aim is to improve pharmacovigilance and signal detection.

The rest of the paper is as follows. Section 2 presents a literature analysis about usability evaluations of controlled vocabulary searching tools along with a benchmark of existing tools in pharmacovigilance. Section 3 describes the PEGASE Knowledge Graph. Section 4 describes Sparklis and the extensions we have implemented. Section 5 presents the knowledge graph validation we have performed, and the qualitative usability study we have carried out on Sparklis. Finally, Section 6 draws some conclusions, and presents the next stages of the project.

2 Analysis of Existing Tools

Our main objective was to identify the desirable usability qualities and the defects of the tools used by pharmacovigilance specialists. To do so, we performed: (i) a literature review of usability evaluations of tools supporting searches using controlled vocabularies (not limited to pharmacovigilance), and (ii) an evaluation of the tools currently used by French pharmacovigilance specialists. A total of 908 papers were identified in PubMed, Web of Science, and Scopus databases; six of which met all eligibility criteria⁶ and were analyzed in-depth: 1) Walji et al. [14] supports coding odontology diagnostics, 2) Bakshi-Raiez et al. [3] supports coding reason for entrance in the intensive care unit (ICU), 3) Peute et al. [9] supports searching ICU patient data, 4) Shiri et al. [11] supports searching keywords over a multilingual thesaurus, 5) Vega-Gorgojo et al. [13]

⁴ MedDRA® is a trademark of the IFPMA.

⁵ Int. Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use

⁶ Omitted due to space restrictions. The most discriminative one was to include an objective evaluation of the interfaces. See <http://bit.ly/2qA9Jk1> for details on the selection process.

evaluates the search of collaborative tools, and 6) Sutcliffe et al. [12] evaluates visual user interfaces for information search. Besides, we visited four French pharmacovigilance centers to identify the software used, where four main tools were identified:

- BNPV (National Bank of PharmacoVigilance): the main tool used by the pharmacovigilance specialists to report and search cases of ADRs.
- MedDRA browser: it allows searching MedDRA terms along with Standardized MedDRA Queries (SMQ).
- PA.PV: a tool provided by the French Agency for the Safety of Health Products to help pharmacovigilance specialists choose MedDRA terms and SMQs.
- Vigilyze: a world-wide pharmacovigilance database provided by the WHO.

Two human factors specialists went through the different functionalities of each software to identify their significant usability qualities and defects. A total of 62 usability desirable features or defects were learned from the literature review, and 86 from the benchmark of the most used software. Overall, the main lessons learned were:

1. Searches should include synonyms and related terms.
2. One should be able to search terms either by entering keywords (without hierarchy level constraints) or by browsing the term hierarchy.
3. Lists and hierarchies of terms should be ordered in a meaningful way.
4. Display the query, the number, and list of results on the same page.
5. Results should automatically be updated with the query.
6. Provide users with hierarchy information to distinguish similar terms.
7. Inform the user about the number of ADR reports each term triggers.
8. Help users build their request by providing an intuitive interface.
9. Allow saving a request to be reused.

Those recommendations led to the choice of Sparklis, and to its adaptation to the pharmacovigilance’s context. Indeed, Sparklis [5] (see Section 4 for a description) already meets to a large extent criteria 4, 5, 8, and 9 (as well as 7 in an approximate way) by guiding its users in the incremental building of their request while receiving results and lists of suggested terms at every step. The main missing feature was the handling of hierarchies, which concerns criteria 2, 3 and 6, which we have added as a result of the project.

3 PEGASE Knowledge Graph

In this section, we detail the knowledge sources we have used to build the PEGASE Knowledge Graph and its structure. First, we present MedDRA and OntoADR [4] the core ontology we use, and then we move onto the additional sources we have included, Standardized MedDRA Queries (SMQs) and anonymized patient data (obtained from FAERS). The core structure of the PEGASE Knowledge Graph can be seen in Figure 1. It currently contains 3,257,389 triples without taking into account FAERS data (with the patient data of three months, it grows to 28,125,629 triples).

OntoADR This ontology [4] is the result of progressive efforts to semantically structure the Medical Dictionary for Regulatory Activities (MedDRA) terminology. In the context of pharmacovigilance, MedDRA is a reference terminology to code and give a precise description of ADRs and related issues [6]. In particular, OntoADR is an ontology which connects MedDRA terms to SNOMED CT concepts, making it possible to search for MedDRA terms semantically, e.g. to search for all terms about ADRs taking place in the skin. To establish the mapping between MedDRA terms and

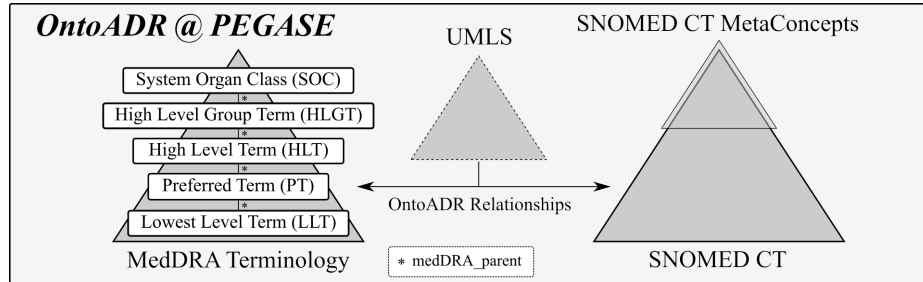


Fig. 1. Main modules of OntoADR within the PEGASE Knowledge Graph.

SNOMED CT, OntoADR exploits the UMLS metathesaurus [2], which provides an initial alignment. Apart from that, several curating steps and algorithms were applied to have a further refined set of relationships.

In order to include OntoADR in our Knowledge Graph, we had to do some adaptations. To model MedDRA, we introduced the concept *MedDRATerm*, which has five different subconcepts corresponding to the five levels of their hierarchy (see Figure 1). However, to model the hierarchy relationship between terms, instead of using the *subclass* relationship (i.e., formal subsumption), we introduced the property *medDRA_parent*. In this way, we can navigate the hierarchy without unexpected potential inferences. For example, “Aortic aneurysm” is a *Preferred Term (PT)*, whose *medDRA_parent* is “Aortic aneurysm and dissections”, a *High Level Term (HLT)*.

To include the SNOMED CT definitions, we had to adapt their representation level. On the one hand, we had MedDRA terms, all of which were instances; on the other hand, we had SNOMED CT terms, all of which were concepts. To solve this mismatch, we materialized SNOMED CT concept hierarchy, and treated the concepts as instances⁷. This allowed us to introduce also different hierarchies to provide different navigation dimensions. In particular, we introduced a top-level hierarchy of SNOMED CT meta-concepts based on the semantic tags that SNOMED CT uses to further refine the concepts’ meaning (see Figure 2). Note that this grouping cohabits with the subclass hierarchy of SNOMED CT concepts. This does not lead to inconsistencies as our knowledge graph is in RDFS, not in OWL. OntoADR relationships between MedDRA terms and SNOMED CT concepts (see [4] for the complete list) were included as they are.

Following with the previous example, “Aortic aneurysm” (MedDRA term) is related to “Aneurysm” (SNOMED concept) via *associatedMorphology* (OntoADR property), and to “Abdominal Aorta Structure” (SNOMED concept) via *findingSite* (OntoADR property). Furthermore, “Abdominal Aorta Structure” is a *BodyStructure* (SNOMED meta concept) and a *rdfs:subclassOf* of “Descending aorta structure” (SNOMED concept). Note how we use the metaconcept hierarchy to group the SNOMED concepts according to their semantic tags.

SMQs SMQs are “groupings of MedDRA terms, ordinarily at the Preferred Term (PT) level that relate to a defined medical condition or area of interest” [7]. In general, SMQs can be seen as disjunctions of terms which are used together in order to perform

⁷ Abusing a little the language, we have flattened them in the RDF graph and allowed for meta-modeling, i.e., classes of SNOMED CT concepts.



Fig. 2. SNOMED meta-concept layer integrated to include the semantic tags hierarchy.

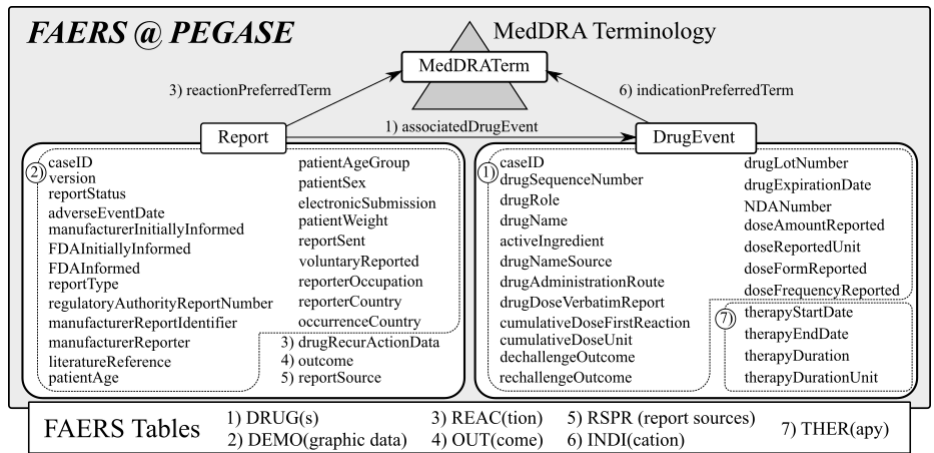


Fig. 3. FAERS integration in the knowledge graph. The companion numbers indicate the source of the data in the original FAERS tables.

searches in a standardized way, although they can be grouped in more complex ways. We added each SMQ as a new node, related to the terms that it includes. The inclusion of SMQs in the graph has two main benefits: on the one hand, pharmacovigilants are used to working with SMQs and they had them directly accessible; and on the other hand, they extend the querying capabilities as now we have the full capabilities of SPARQL on top of them.

FAERS Data Finally, we integrated a source of patient data to show how the integration capabilities of our knowledge graph can help pharmacovigilance specialists to ease their jobs. FAERS (FDA Adverse Event Reporting System) is a dataset containing the anonymized reports of drug adverse events which is gathered and made public quarterly by the U.S. Food and Drug Administration (FDA). The patient data provided by FAERS is split in seven different big tables, which we integrated as shown in the resulting model in Figure 3. Such model was obtained after an evaluation round with the human factors specialists in the project’s team, where we brought the FAERS model closer to the pharmacovigilants’ cognitive process.

4 Sparklis on the PEGASE Knowledge Graph

Sparklis⁸ is a tool for retrieving information from RDF knowledge graphs, with the objective to reconcile the expressivity of SPARQL 1.1 and the usability of point-and-click user interfaces. We here present its principles as well as the main extensions that we have performed to address the needs of the PEGASE project.

4.1 Sparklis: a SPARQL Query Builder in Natural Language

Sparklis is a query builder in natural language that allows people to explore and query SPARQL endpoints without any knowledge of SPARQL [5]. Sparklis is implemented as a Web client running entirely in the browser, which directly connects to SPARQL endpoints to retrieve query results and suggested query elements. It covers a large subset of SPARQL 1.1 SELECT queries: basic graph patterns including cycles, UNION, OPTIONAL, NOT EXISTS, FILTER, BIND, complex expressions, aggregations, GROUP BY, ORDER BY. All those features can be combined in a flexible way. Results are presented as tables, and also on maps and as slideshows. A configuration panel offers a few configuration options to adapt to different endpoints (e.g., GET/POST, labelling properties and language tags).

4.2 Applying Sparklis to the PEGASE Knowledge Graph

Sparklis requires very little configuration to be applied to the PEGASE Knowledge Graph. It is enough to provide the URL of the SPARQL endpoint⁹, and to choose property `rdfs:label` for the labelling of entities, classes, and properties. As the end users are French pharmacovigilance specialists, we also configure the user interface and the labels to the French language.

Figure 4 shows a screenshot of Sparklis on PEGASE data, taken during the process of building a query¹⁰. The current query (at the top) select preferred terms (PT) in MedDRA whose finding site is (a subconcept of) “Skin and subcutaneous tissue structure”, and whose associated morphology is (a subconcept of) various morphologic abnormalities. A first abnormality, “Blister”, has already been selected, and the user is in the process of selecting (at the center) a disjunction of three more abnormalities (“Vesicle”, “Vesiculobullous rash”, “Vesicular rash”). The keyword “vesic” was input at the top of the list of suggested terms in order to ease their retrieval among a long list of suggestions. Sparklis suggests modifications to be applied to the current focus (here, the focus is on the associated morphology of the selected preferred terms): at the middle left, Sparklis suggests classes and properties to refine the query; at the middle center, individuals denoting associated morphologies relevant to the query; and at the middle right, query modifiers and operators (e.g., “and”, “or”, “number of”). The table of results of the current query is shown at each step. Here, it shows the selected preferred terms along with their finding sites and associated morphologies.

⁸ <http://www.irisa.fr/LIS/ferre/sparklis/> (includes the presented extensions)

⁹ The URL is not provided here because of restrictive licences on MedDRA and SNOMED.

¹⁰ A screencast of the whole query building is available at <http://www.irisa.fr/LIS/common/documents/pegase2018/#ExtraCase>.

The screenshot shows the Sparklis interface with a SPARQL endpoint at `http://localhost:3030/meddra/sparql`. The query being constructed is:

```

give me every PT
  whose finding site is (hierarchy) in Skin AND subcutaneous tissue structure (body structure)
  and whose associated morphology is (hierarchy) in
    Blister (morphologic abnormality)
    or something
  
```

Sparklis suggestions to refine the query are shown in the middle section. The current focus is on the `associated morphology (hierarchy)` property. Two suggestion panels are visible:

- Left panel:** Shows a hierarchy of SNOMED concepts. The selected item is `vesic`. It lists properties like `is a body structure`, `has associated morphology of`, `has in onto ADR`, `has a label`, `is the subclass of`, and `has a type`. There are 12 concepts listed.
- Right panel:** Shows a list of entities. The selected item is `vesic`. It lists entities like `anything`, `Vesicle (morphologic abnormality)`, `Vesicular inflammation (morphologic abnormality)`, `Vesiculobullous rash (morphologic abnormality)`, and `Vesicular rash (morphologic abnormality)`. There are 8 entities listed.

The bottom section shows the results of the query in a table format:

PT	finding site	associated morphology
1 Pemphigus b�nign familial	Skin structure (body structure)	Vesiculobullous rash (morphologic abnormality)
2 Imp�tigo bulleux	Skin structure (body structure)	Vesicle (morphologic abnormality)

Fig. 4. Sparklis’ screenshot showing a query under construction (top) on the PEGASE Knowledge Graph, suggestions to refine the query (middle), and query results (bottom).

4.3 Extending Sparklis for PEGASE

Handling hierarchies As the example in Figure 4 shows, it is important to take into account hierarchies of MedDRA terms or SNOMED concepts in the evaluation of queries. For instance, when asking for “preferred terms whose finding site is Skin...”, what is really meant is: “preferred terms whose finding site is any *subconcept* of Skin...”, i.e., any *part of* the skin (e.g., “epidermis”, “subepidermal region”). The correct translation to SPARQL requires the use of property paths like

```
?x ontoadr:findingSite/rdfs:subClassOf* snomed:Skin .
```

A number of problems made it impossible to express in Sparklis, and thus required an extension to handle hierarchies in the building and evaluation of queries. First, it was possible to build a sequence of properties but not to apply the `*`-operator (transitive closure). Therefore, it was possible to reach the parent or grand-parent concepts but not all ancestors at once. Second, assuming the `*`-operator was available, it was tedious to explicitly cross the property defining the hierarchy (`rdfs:subClassOf` in the example), and to apply transitive closure. Third, once an element of the hierarchy was selected (“Skin...” in the example) and the query focus set on it, only that element was visible whereas the user would expect to see the subset of the hierarchy below and above that element.

We solved those problems by modifying Sparklis with several extensions and adaptations. What made it particularly tricky is that the new features had to combine smoothly with the numerous existing features: e.g., crossing properties forward and backward, Boolean coordinations of verb phrases and noun phrases. It was also important to make it in a generic way so as to handle various kinds of hierarchies. The PEGASE knowledge

graph has two hierarchies based on two different properties (`medDRA_parent` for MedDRA terms, `rdfs:subClassOf` for SNOMED concepts), and other applications could have other hierarchies, e.g., hierarchies of geographical places or historical periods. First, we added a `(hierarchy) in` query construct that can turn any property into a hierarchy. It does not only apply transitive closure on the property but translates the current focus in a special way so that all terms below and above the selected terms are retrieved and can be displayed as a tree to render the hierarchical relationships between the terms. For example, the SPARQL translation of the above example, when the focus is on “Skin...” is

```
?x ontoadr:findingSite ?y .
?y rdfs:subClassOf* snomed:Skin ; rdfs:subClassOf* ?focus .
```

Second, we added schema-level declarations to automate the use of the hierarchy feature. For example, a declaration states that the range of property `findingSite` is hierarchically organized by property `rdfs:subClassOf`. Therefore, as soon as property `findingSite` is crossed, a hierarchy construct based on `rdfs:subClassOf` is inserted in the query so that the user immediately sees a hierarchy of concepts in which terms can be selected. We stated a similar declaration for each OntoADR property whose range is made of SNOMED concepts.

Other extensions Finally, a few other less original, yet important, extensions were implemented. We have added support for full-text search (compatible so far with Jena Fuseki and Virtuoso RDF stores), and for multi-selection. The former was required to speed up keyword searches and improve the robustness (e.g., handling accentuated letters). The latter, while it does not increase the expressivity of Sparklis, allows to build coordinations of values in much fewer interactions.

5 Evaluation

We have performed the expert evaluation in two subsequent steps: 1) the human factors specialists in the project team checked Sparklis against the list of recommendations derived from the literature analysis, comparing it to the main tools used by pharmacovigilance specialists in France (see Section 2); then, 2) they analyzed Sparklis over PEGASE Knowledge Graph, after the extensions were implemented (see Section 4.3) performing a *cognitive walkthrough* [10] in order to identify remaining usability issues.

5.1 Qualitative Evaluation

Two human factors specialists checked to which extent the extended version of Sparklis respects the requirements learned from the literature review and the benchmark of existing tools. Table 1 compares the BNPV tool and the extended Sparklis w.r.t. the requirements identified in Section 2. The evaluation showed that Sparklis has a better adherence to the requirements than the BNPV, the tool currently in use. Therefore, Sparklis may provide a better support to pharmacovigilance specialists’ searches. Nonetheless, there is still room for some improvements regarding: criteria 1, synonyms and related terms are not considered; criteria 7, Sparklis provides the number of ADR reports linked to a given term based only on a sample of reports; and criteria 8, although it guides users during the request building, the current interface is not sufficiently intuitive for untrained pharmacovigilance specialists.

Table 1. Adherence of BNPV and Sparklis to the identified requirements.

Requirements	BNPV	Sparklis
1. Include synonyms and related terms	No	No
2. Search by entering terms or by navigating	Yes	Yes
3. Meaningful order	No	Yes
4. Display query, number and list of results together	No	Yes
5. Update automatically results	No	Yes
6. Provide the terms' places within the hierarchy	Incomplete	Yes
7. Inform about the number of ADR reports	No	Incomplete
8. Help users build their request	No	Incomplete
9. Allow saving the request	Yes	Yes

5.2 Cognitive Walkthrough

A cognitive walkthrough [10] was performed, following an ISO standard [8], to identify remaining usability issues in Sparklis. This inspection method focuses on how easy and intuitive for new users is their interaction with an interactive device. Representative pharmacovigilance specialists were asked for four prototypical use cases. Following those use cases, two human factors specialists went through the sequences of tasks supported by Sparklis stepping into pharmacovigilance specialists' shoes. For each subtask, they asked themselves four standardized questions:

- Will pharmacovigilance specialists try to achieve the effect the subtask has?
- Will they notice that the correct action is available?
- Will they understand that the wanted subtask can be achieved by the action?
- Do they get appropriate feedback?

For each question, at each subtask, usability problems leading to answering “no” were noticed. While the evaluation was performed over Sparklis directly, we prepared some illustrative videos to show how the user can deal with the use cases using Sparklis, which are available at <http://www.irisa.fr/LIS/common/documents/pegase2018/>.

Despite the potential support provided by Sparklis to pharmacovigilance specialists, its current graphical user interface might not be intuitive enough from a pharmacovigilance specialists' perspective. While Sparklis has been proved to successfully reduce the gap between users and SPARQL queries, it still requires training in order to go through the first stage of the learning curve. This is specially relevant in this context, where the users (pharmacovigilance specialists) might have neither expertise nor training in computer and information sciences, and their related logic, concepts and wording. For instance, Sparklis provides all the flexibility of SPARQL in order to build the queries; however, it can be argued that this flexibility might confuse pharmacovigilance specialists who use daily a very limited range of requests.

As a conclusion of this evaluation, we can conclude that we might need two different versions of Sparklis interface depending on the user profile (i.e., advanced and beginner users): 1) for advanced users, an interface presenting all the relevant information to help pharmacovigilance specialists build a request, and 2) for beginners, a simplified graphical user interface presenting only the logical operators and concepts which are most likely to be used, and the others on-demand. In this way, we will allow the whole range of pharmacovigilance specialists to take advantage of its power from the start.

6 Conclusions and Future Work

To overcome the limitation of current pharmacovigilance tools, we have built a knowledge graph that integrates: 1) MedDRA, the main pharmacovigilance vocabulary, 2) OntoADR, its semantic enhancement using SNOMED CT, 3) SMQs, and 4) patient data. In order to search and explore that graph, we have adopted Sparklis, a SPARQL query builder in natural language. We conducted a literature review along with a benchmark of existing tools, which led to the detection of several requirements that pharmacovigilance tools have to meet, and which were taken into account in a Sparklis extension. Finally, we carried out an evaluation of the interface following an ISO ergonomics standard [8], showing the potential of our proposal. Usability evaluation with end-users is in progress.

References

1. ICH guideline E2B (R2), Electronic transmission of individual case safety reports, Final Version 2.3, Document Revision February, 2001.
2. UMLS Website. <http://www.nlm.nih.gov/research/umls/>, accessed: September 2018.
3. Bakhshi-Raiez, F., de Keizer, N., Cornet, R., Dorrepaal, M., Dongelmans, D., Jaspers, M.: A usability evaluation of a SNOMED CT based compositional interface terminology for intensive care. *Int. J. of Medical Informatics* **81**(5), 351 – 362 (2012)
4. Bousquet, C., Sadou, É., Souvignet, J., Jaulent, M.C., Declerck, G.: Formalizing MedDRA to support semantic reasoning on adverse drug reaction terms. *Journal of Biomedical Informatics* **49**, 282–291 (2014)
5. Ferré, S.: Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language. *Semantic Web: Interoperability, Usability, Applicability* **8**(3), 405–418 (2017)
6. Harrison, J., Mozzicato, P.: MedDRA®: The tale of a terminology: Side effects of drugs essay. *Side Effects of Drugs Annual*, vol. 31, pp. xxxiii – xli. Elsevier (2009)
7. ICH: Introductory Guide for Standardised MedDRA Queries (SMQs) Version 21.0, Document Revision March, 2018
8. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Standard, International Organization for Standardization, Geneva, CH (Mar 2010)
9. Peute, L.W., de Keizer, N.F., Jaspers, M.W.: The value of retrospective and concurrent think aloud in formative usability testing of a physician data query tool. *Journal of Biomedical Informatics* **55**, 1 – 10 (2015)
10. Polson, P.G., Lewis, C., Rieman, J., Wharton, C.: Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *Int. J. of Man-Machine Studies* **36**(5), 741 – 773 (1992)
11. Shiri, A., Ruecker, S., Bouchard, M., Doll, L., Fiorentino, C.: User evaluation of Searchling and T-saurus: Multilingual thesaurus-enhanced visual interfaces for digital libraries. *Canadian Journal of Information and Library Science* **37**(2), 137–160 (2013)
12. Sutcliffe, A., Ennis, M., Hu, J.: Evaluating the effectiveness of visual user interfaces for information retrieval. *Int. J. Human-Computer Studies* **53**(5), 741 – 763 (2000)
13. Vega-Gorgojo, G., Bote-Lorenzo, M.L., Asensio-Pérez, J.I., Gómez-Sánchez, E., Dimitriadis, Y.A., Jorrín-Abellán, I.M.: Semantic search of tools for collaborative learning with the ontolsearch system. *Computers and Education* **54**(4), 835–848 (May 2010)
14. Walji, M.F., Kalenderian, E., Tran, D., Kookal, K.K., Nguyen, V., Tokede, O., White, J.M., Vaderhobli, R., Ramoni, R., Stark, P.C., Kimmes, N.S., Schoonheim-Klein, M.E., Patel, V.L.: Detection and characterization of usability problems in structured data entry interfaces in dentistry. *Int. J. Medical Informatics* **82**(2), 128 – 138 (2013)