Enhancing web portals with Ontology-Based Data Access: the case study of South Africa's Accessibility Portal for people with disabilities

C. Maria Keet[1], Ronell Alberts[2], Aurona Gerber[3], and Gibson Chimamiwa[3]

[1] Faculty of Computer Science, Free University of Bozen-Bolzano, Italy
[2] Intelligent Environments for Independent Living, Meraka Institute, CSIR, Pretoria
[3] Knowledge Systems Group, Meraka Institute, CSIR, Pretoria, South Africa
[1]`keet@inf.unibz.it`, [2,3]`{RAlberts,GChimamiwa,AGerber}@csir.co.za`

Abstract. Web portal software is relatively easy to set up and populate from the perspective of the end-user, but it leaves the back-end database devoid of subject domain semantics due to the requirement for a generic implementation. This approach seriously hampers effective search capabilities to retrieve relevant information. Ontology-based data access (OBDA) could, in theory, solve this problem through adding a semantic 'layer' over such web portal implementations. To this end, we provide and demonstrate the proof-of-concept methodology by enhancing the operational National Accessibility Portal of South Africa. We developed the ADOLENA ontology, which is based on both the semantics in the database and augmented with notions from foundational and related domain ontologies. ADOLENA was then made compliant with the OWL2 profile $DL\text{-}Lite_{\mathcal{A}}$ and mapped to the relational database using the OBDA Plugin for Protégé. Experimentation with OBDA queries unequivocally demonstrates its advantages compared to the portal's keyword-based search.

1 Introduction

The vast majority of extant operational databases are not Semantic Web enabled databases, and fall into two major categories: the well-designed tailor-made databases with a rich database schema and the convenience databases that are generated with a default schema irrespective of the subject domain semantics, such as a database that runs at the back-end of a Joomla-powered website. The main advantage of the second type lies with the user-cum-content-creator, for s/he is not dependent on system administrators to keep the implementation up-to-date. This was also a requirement deemed crucial by the stakeholders of the South African National Accessibility Portal (NAP) that provides information about disabilities, assistive devices, such as wheelchairs and talking thermometers, and so forth, which has been aligned with the goal of the African Decade of Persons with Disabilities (1999 to 2009) to empower persons with disabilities. While this freedom may be liberating for the content-creator, it seriously hampers effective search and retrieval of the data, which for a small website might still be browseable but this does not work anymore for web portals. Hence, the need for enrichment of the implementation arises so as to enable advanced query capabilities (without disrupting the content additions by users), which, invariably, can only be done by adding subject domain information that then has to be connected to the database content.

Over recent years, theory of linking ontologies to data and an overall framework of Ontology-Based Data Access (OBDA) has been maturing [1, 2], with the first (prototype) tools having been made available, such as the OBDA Plugin [3–5] and DataMaster [6]. Given that in the web portal setting one cannot do away with the database to incorporate it in the ABox of a knowledge base, due to the large volume of data that is updated regularly and a web portal has to return query answers quickly, with the current state of technologies, we have to resort to the OBDA-approach of [1–5]. It implements the DL language $DL\text{-}Lite_{\mathcal{A}}$ (which is one of the OWL 2 profiles), is equipped with the lean OBDA-enabled reasoner DIG-QuOnto and plugin for Protégé both to create the mappings between ontology and database and to query the data through the ontology. The functionality of the Protégé OBDA Plugin with associated $DL\text{-}Lite_{\mathcal{A}}$-tailored reasoners have been demonstrated with well-designed tailor-made databases by their developers [3–5] and as such the core of the science and Semantic Web technology reported in this paper is not novel. However, (i) a real use case on feasibility of realising mappings with extant operational databases is lacking, (ii) a comparison between presence and absence of OBDA technology for a given database has not been assessed with live data, and (iii) any how-to methodology for implementing OBDA is informally known by the developers and only sparsely described in the two demo papers ([3, 4]). Our contribution aims to fill this experimental gap. In addition, by not just taking a well-designed database that has already rich semantics in the database schema, but one with comparatively hidden semantics, we push the envelope with regards to the OBDA mappings, which is generally perceived to be the constraining factor for deploying the technology. Further, by taking a typical web portal application (Content Management System, CMS) instead of a boutique database, the results presented here may be generalisable to semantically enhancing other web portals, too.

The remainder of the paper is structured as follows. We outline the materials and methods in section 2. The newly developed domain ontology, OBDA mappings, and experimental data with the test results of the OBDA-enhanced portal compared to the plain NAP is presented and discussed in section 3. We close with conclusions and current work in section 4.

2 Methodology

To realise the proof-of-concept examination of semantically enriching an existing web portal with OBDA, we devised the following methodology:

1. Develop experimental domain ontology, which comprises:
   (a) consulting top-level categories from foundational ontologies as top-down development approach;

   (b) reverse engineering the contents of the 'groups' and 'subgroups' tables from the portal database (alternatively called, e.g., 'Section' and 'Category' in Joomla-like CMSs) as bottom-up approach;
   (c) sourcing related domain ontologies, taxonomies, and the like for potential of reusing them in whole or in part; and
   (d) verifying the resultant ontology with the domain experts;
2. Ensure that the ontology is 'simple' enough to be expressible in $DL\text{-}Lite_{\mathcal{A}}$, including conversion or removing of violating axioms, if necessary, and verify that this new ontology still conforms to the understanding of the subject domain and represents reality faithfully;
3. Enrich the database with knowledge from the ontology by either adding tables or instances, if necessary;
4. Define OBDA mappings between the ontology and SQL queries over the portal database through the Protégé OBDA plugin;
5. Test SPARQL queries with the Protégé OBDA plugin and compare the query answers with those of the near-analogue in the web portal;
6. Provide web interface to the OBDA Plugin to allow users to use the enhanced search capabilities.

The materials used for this procedure are as follows. The extant ontologies consulted with the aim of potential reuse are: DOLCE [7] and BFO[4] foundational ontologies, and SNOMED, ICD10, ICIDH-2[5] [8] and the ISO devices classification as ontology-like artifacts; to the best of our knowledge, no ontology about disabilities exists. The NAP database (d.d. 1-8-2008) contains 20 tables, of which 6 store terms for so-called "services", "groups", and "topics", which are candidates for concepts in the ontology, and 12 tables that store the "contents" data in about 20k cells, such as device vendors and documents with information about disabilities. The groups and data are added by content-creators, such as domain experts and stakeholders (e.g., the QuadPara Association and the SA National Council for the Blind). The novel Abilities and Disabilities OntoLogy for ENhancing Accessibility (ADOLENA) was developed with Protégé 4.0 alpha build 64, and the mappings defined in Protégé v3.3.1 using the OBDA Plugin version 20080721[6]. The queries were performed through both the OBDA plugin over the "content" tables in the NAP database and through the web portal interface at http://www.napsa.org.za/.

3 Results and Discussion

In this section we report on the outcomes of ontology development, conversion from the comprehensive version to one that is better suited for use in OBDA, the ontology-to-database mappings, and the striking differences in query answers be-

[4] http://www.ifomis.org/bfo
[5] WHO's disabilities classification: http://www.who.int/classifications/icf/site/index.cfm
[6] Downloadable from http://www.inf.unibz.it/~rodriguez/OBDA/files/obda-plugin/current/. The stable release will be available on the TONES website.

tween the standard web portal and OBDA-enhanced querying. All data files are available online at http://ksg.meraka.org.za/wiki/SNAP_OBDA_experiment.

3.1 ADOLENA

The experimental domain ontology ADOLENA—the Abilities and Disabilities OntoLogy for ENhancing Accessibility—was developed through a combination of top-down and bottom-up activities. The basic structuring principle in the current, openly available, experimental ADOLENA—which contains 141 classes, 16 object properties, and has a DL expressivity $\mathcal{SHIQ}$—deals with 4 principal concepts: Ability, Disability, Device, and Functionality. A device assistsWith some ability, ameliorates some disability, and hasFunction some functionality, a disability affects abilities (with inverse isAffectedBy). In addition, there is initial additional knowledge about ServiceProviders, such as a device Vendor, that certain devices requiresAbility an ability, and Person with BodyParts (e.g., to say that a prothese replaces some body part).

This ontology was build in Protégé with the built-in FaCT++ reasoner by, first, creating a preliminary seed ontology based on the current structure of NAP by manually examining the "services", "groups", and "topics" tables of the database. Clearly, an automated extraction from table content to taxonomy is preferable, which could be integrated with an overall approach of logic-based reverse engineering (e.g., [9]). Second, DOLCE's Endurant, Perdurant, and several of their subtypes were added to help categorising the seed terms from the database. DOLCE, however, does not deal with so-called 'realizables' [10], such as ability and function. For instance, the ability to Reach does not mean the actual process of reaching out to take up the cup of coffee but the capability to do so, likewise for functionality, where a device might never realize the functionality (e.g., it is not used after all); hence, we have added BFO's notion of Realizable to ADOLENA. Third, the informal ICIDH-2 classification of the WHO is being analysed so as to incorporate an 'ontological rendering' of it in ADOLENA, likewise for the ISO assistive devices standardization. Further extensions are planned, including reusing the FMA to relate abilities and disabilities to body parts. It must be noted, however, that even between the preliminary seed ontology with 40 entities and the current ADOLENA with 141 classes, satisfiability checking is human-noticeably slower with ADOLENA; thus, some care will have to be taken how and with how much knowledge adolena should, and practically can, be extended. Feedback from domain experts occurred during each of the three steps. Thus, ADOLENA is, at present, an experimental ontology to show the proof-of-concept to the stakeholders and to develop a comprehensive general methodology for OBDA and web portals; the next version aims to cover the subject domain more comprehensively.

Transformation to *DL-Lite*$_{\mathcal{A}}$. As mentioned, the full ADOLENA is represented in a language with DL expressivity $\mathcal{SHIQ}$. We could either make the OBDA mappings with ADOLENA, or simplify the ontology—be it manually or automatically—to the OWL 2 profile[7] for $DL\text{-}Lite_{\mathcal{A}}$ and make the OBDA mappings with that one. For experimental purposes, we have carried out two manual simplifications from ADOLENA.owl to the AdolenaSlim.owl and to AdolenaSmall.owl[8]. The former was created by using a previous created comparison between ontology languages [11] and communication with two of the $DL\text{-}Lite_{\mathcal{A}}$ developers. This was compared with the outcome of the automated syntactic transformation (AdolenaDLlite), noting that:

1. AdolenaDLLite is guaranteed to be within $DL\text{-}Lite_{\mathcal{A}}$ expressiveness, whereas this can only be tested for AdolensSlim/AdolenaSmall through running the same algorithm on the DIG-QuOnto;
2. $DL\text{-}Lite_{\mathcal{A}}$ is not a pure fragment of OWL 2 (or $\mathcal{SROIQ}$ [12, 11]), because it can handle identification constraints and role values [13, 1, 2], which, by using the OWL2full-tailored Protégé tool, were not available for use in ontology development. Thus, with the used software versions, any conversion from an expressive OWL ontology results in a version that is only in a fragment of $DL\text{-}Lite_{\mathcal{A}}$;
3. To go from ADOLENA to AdolenaSlim, only *deletions* were made, based on a semantic analysis and domain expert judgement:
   (i) removal of MultipleDisability due to the qualified number restriction "affects $\geq$ 2 Ability", which is ontologically ambiguous anyway and the $DL\text{-}Lite_{\mathcal{A}}$ rendering deemed too cumbersome given the already non-trivial OBDA mappings (see below),
   (ii) removal of the remaining qualified number restriction (3x),
   (iii) the remaining defined classes (3x) were changed into primitive classes with only necessary conditions, and
   (iv) the relational properties (ir)reflexivity and transitivity of partof and properpartof were removed.
   (v) the axioms with the universal quantifications should have been reassessed or removed (removed in AdolenaSmall; where also less important classes and properties were removed but a few data type properties added so as to generate nicer query results, see below) and the full existential quantifications should have been manually remodelled, but this was not carried out for this experiment in full.

   The automated transformation includes also *rewritings* and *approximations*, e.g., it has rewritten several axioms by introducing new placeholder concepts and roles; hence,
4. the automated transformation into AdolenaDLLite is a closer approximation to ADOLENA than either AdolenaSlim or AdolenaSmall is.

---

[7] http://www.w3.org/TR/owl2-profiles/

[8] We have chosen not to make the mappings with ADOLENA, because an OBDA-mapping for a Protégé object property in a non-$DL\text{-}Lite_{\mathcal{A}}$ ontology might be lost in the automatic transformation to $DL\text{-}Lite_{\mathcal{A}}$-compliance by the DIG-QuOnto reasoner, be it that it was an unsupported axiom in the expressive ontology—hence, discarded—or an approximated one through newly introduced roles and concepts. In such an occasion, executing SPARQL query to retrieve data, which first approximates the ontology on-the-fly, would be empty.

Advanced transformations and remodelling as indicated in item 3v were not done manually because it is non-trivial to do and demotivating whilst knowing that there is a tool that can do it automatically (the DIG-QuOnto) whereas the modeller would have to re-invent all the transformation and approximation algorithms. Eventually, it was decided to use AdolenSlim and AdolenaSmall in this experiment instead of AdolenaDLlite, because, first, the automated rewritings and approximations appeared to hamper domain expert understanding of the represented knowledge because it was unclear how the rewriting and approximations had happened to each concept, role, and axiom (e.g. to trace the function of each new Aux_Class1, $NewRole\_1$ and so forth and what it replaced). Second, as a knock-on effect, it was perceived to complicate declaring the OBDA mappings. From an ontology developer perspective, it thus would be better if the on-the-fly approximation were a separate, explicit, OBDA development step. A straight-forward nice GUI rendering hiding such transformation details could solve it (the successful precedent has been set with the ICOM ontology editor [14] already), and generating explanations may be helpful as well.

AdolenaSlim as well as AdolenaSmall were used for OBDA after observing that the ontology had to be saved in the RDF/XML format instead of the OWL/XML format so as to maintain backward compatibility between Protégé alpha 4.0 and its earlier version 3.3.1 for which the OBDA Plugin has been developed (the OBDA plugin for Protégé alpha 4.0 is currently under development).

3.2 Mapping and OBDA-enhanced Querying

Linking the ontology to the NAP database. Declaring OBDA mappings in the light of a semantics-poor database requires considerable knowledge about the database, SQL, and the theory behind the mappings. As with the previous steps, the linking has been carried out manually, though noting that preliminary results to automate this procedure [15] look promising for rich database schemas. For illustrative purpose, we demonstrate a mapping from a class in the ontology to SQL queries over the NAP database, which concern wheelchairs, because it is directly relevant to several persons with disabilities who are members of one of the participating groups (Intelligent Environments for Independent Living). For instance, one can make a mapping (and successfully retrieve the motorised wheelchairs from the database) for Motorised_Wheelchair as follows:

```
HEAD: NAP:Motorised_Wheelchair(getObj($name,$description))

BODY: select distinct name_contentelement.name,
             description_contentelement.description
                from name_contentelement, description_contentelement,
                  content, i18ncontent, rainbowcontent
                    where rainbowcontent.grouping_grouping_id = 9280
                        ...
                        and content.description_fk =
                          description_contentelement.description_ce_id
```

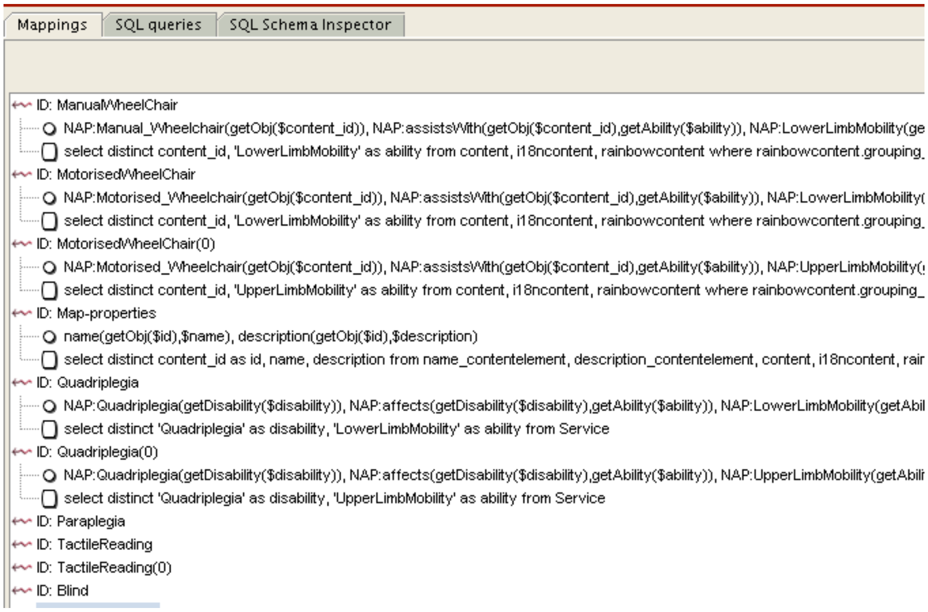

Fig. 1. Screenshot with some of the mappings made in the OBDA Plugin.

where `HEAD` has the class from the ontology and the combination of the `name` and `description` values in the database that will be turned into objects in the ontology, i.e., assuming that devices are identified by their name and description, and the `BODY` contains the actual SQL query to the database. While the above is a sensible query to retrieve human-readable information about motorised wheelchairs in a standard database environment, what we actually have in the database is that the devices, including motorised wheelchairs, are identified (in database terms, primary key) by their `content_id`, hence, the following OBDA mapping is in this case the correct one.

```
HEAD: NAP:Motorised_Wheelchair(getObj($content_id))

BODY: select distinct content_id
          from content, i18ncontent, rainbowcontent
                where rainbowcontent.grouping_grouping_id = 9280
                      and rainbowcontent.rainbowcontent_id =
                           i18ncontent.rainbowcontent_rainbowcontent_id
                      and i18ncontent.i18ncontent_id =
                           content.i18ncontent_i18ncontent_id
                      and content.language = 0"
```

The mapping queries for the other classes and properties in the ontology are similar; see Fig.1 for an example, among many, in the OBDA Plugin's interface. More mappings are available on the experiment's webpage.

Sophisticated queries. While obviously we can query for instances of concepts and roles just like in SQL, we shall focus on the more interesting queries that are difficult, or even impossible to do with standard database technology; put differently, we focus on novel query features.

First, let us combine reasoning over the ontology with database queries. Given the OBDA mappings, we want to retrieve "*all devices that assist with upper limb mobility*", where we have in the ontology the taxonomy of devices under Device, which is related to Ability through the assistsWith role, and UpperLimbMobility $\sqsubseteq$ Ability. The database itself does not contain any specific data about relating devices to abilities and not even about abilities themselves. With OBDA, however, the query answer is not empty; in fact, it returns all motorised wheelchairs, as depicted in Fig. 2. This answer is returned thanks to OBDA's query evaluation: by availing of the knowledge represented in the ontology, it finds that it is MotorisedWheelchair (a type of Device) that has an assistsWith relation to UpperLimbMobility; hence, the final query to the database is only the one to retrieve motorised wheelchairs.

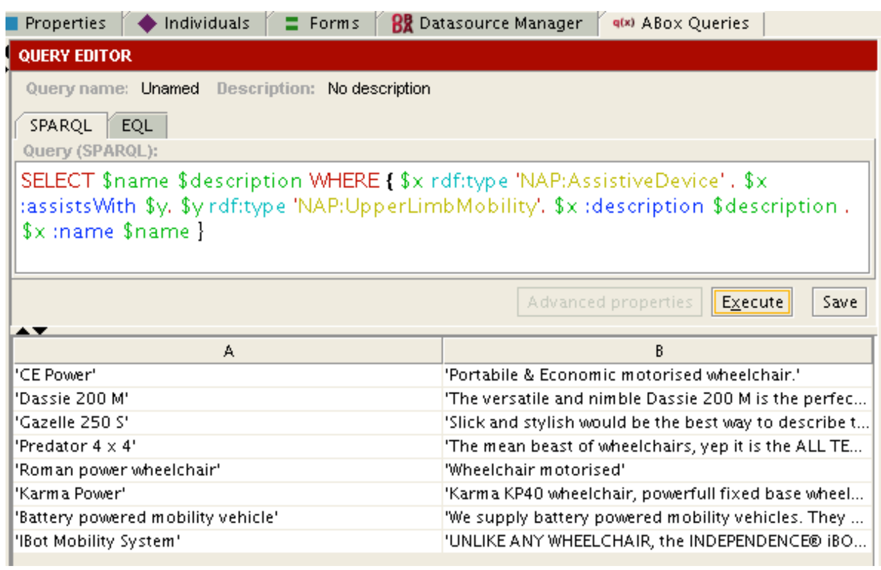

Fig. 2. Screenshot of the OBDA query results for devices that assist with upper limb mobility.

A different type of query is one over the ontology itself with, e.g.,"*show me all abilities that are affected by quadriplegia*", taking into account that we have, in addition to a taxonomy of types of Ability, the affectedBy/affects roles and Quadriplegia as a type of Disability. Given the current state of the ontology, it returns UpperLimbMobility and LowerLimbMobility.

The third example that illustrates a typical query pattern as identified by the domain experts is, e.g., "*retrieve all devices for paraplegia*", where in the ontology we have Paraplegia $\sqsubseteq$ Disability and, in the full ADOLENA, at the top of the two sub-taxonomies the ameliorates role between Device and Disability. This is, in fact, analogous to devices that assist with an ability, and the appropriate query is, in Datalog syntax,

`q(x) :- Device(x), ameliorates(x,y), Paraplegia(y)`

If, on the other hand, we would not have this particular ameliorates in the ontology, then the query to circumvent this gap would have been

`q′(x) :- Device(x),assistsWith(x,y),Ability(y),affects(z,y),Paraplegia(z)`

which causes not only a considerably slower performance in query answering but also complicates the mappings because there as no instances of Ability and Disability in the database. How to identify the best strategy for successful and satisfactory completion of a query scenario—in this case solving it by having (or adding) a mere relation in the ontology versus changing the mappings and queries—is fertile ground to devise guidelines as to where to start looking for a solution and what to do in which occasion.

Comparison of OBDA versus keywords. Given the mappings, and the immediate user requirements, we now would like to retrieve data from the NAP database. To this end, we first want to retrieve data about Wheelchairs. The SPARQL query that returns only and all wheelchairs is

`SELECT ?x WHERE {?x rdf:type 'NAP:Wheelchair'}`

The keyword search in NAP can have two approximations: (i) simple keyword search and (ii) 'advanced' keyword search. Obtaining information about wheelchairs, one retrieves well over 150 results ranging from wheelchair as assistive device, news about wheelchairs, recreation, and classifieds (see Fig.3). In the advanced search, "wheelchair" + Category returns 0 results, and "wheelchair" + Category + all checkboxes in Provinces, Interest Groups and all Disabilities returns 22 hits, all of them within the assistive devices. Clearly, with the OBDA enhancement we obtain immediately the targeted results specifically about wheelchairs, neither being distracted by the many irrelevant hits nor having to do guesswork in the 'advanced keyword' search. We obtain analogous results with similar queries, which are available on the webpage as well.

Let us have a look at the more sophisticated queries from the previous section. Take the phrase "devices that assist with upper limb mobility" in the simple search and in the advanced search where the following check boxes were checked: Category and Specific Information, Physical (in Disabilities), all Provinces, and all Interest Groups. The portal's returned hits (screenshots available online) differ most likely due to the combination of categorisation of the content items by the content providers, the limited set of check boxes one can select, and that only system administrators can change it. Compare this with OBDA and the query answer in Fig. 2. Here we have obtained all MotorisedWheelchairs, but not the Protheses that the NAP answer contains as first query answers in addition to the wheelchairs later on in the keyword search result. The reason for the

Fig. 3. Screenshot of the first-page results of the NAP keyword search on wheelchairs.

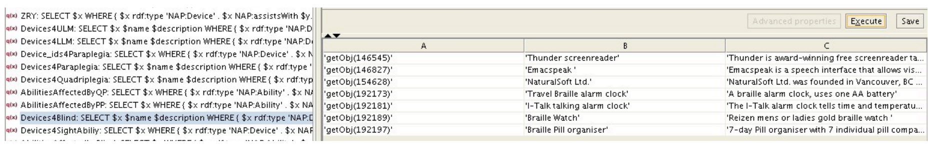

Fig. 4. The query answers to 'devices for the blind' obtained with OBDA.

difference in query answers is that ADOLENA and its slimmed versions do have an assistsWith relation between MotorisedWheelchair and UpperLimbMobility, but not yet between Prothese and UpperLimbMobility. There is no technical limitation for not having this relation in the ontology. It simply had not been added because the ontology is still under development and the focus was to test the proof-of-concept of linking an ontology to an operational database and combine reasoning over the ontology itself with querying the database that does not have any table with instance data about abilities. This has been demonstrated successfully; thus, simply adding "Prothese $\sqsubseteq$ $\exists$assistsWith.UpperLimbMobility" to ADOLENA and the corresponding mappings in the OBDA plugin fully solves the observed difference.

The last example of the comparison is a very nice illustration between sub-optimal meaning-unaware string-based searches and semantics-rich querying. Take, for instance, "devices for the blind" (`q(x):-Device(x), ameliorates(x,y), Blind(y)` or analogous to `q'` in the the previous paragraph). The NAP's results (screenshots available online) has the first device on page 2: a video magnifier, which is, however, not relevant to *blindness* but rather to *low-vision*. The first *relevant* device is a talking alarm clock, on page 3 (i.e., in the range of hit numbers 10-15). On the other hand, when we examine the OBDA results depicted in Figure 4, 7 devices—both Braille and talking devices—were returned. As before, this query answer is more appropriate to the original query than the NAP keyword search results.

The last step in the methodology—providing web interface to the OBDA Plugin to allow users to use the enhanced search capabilities—is currently under investigation with respect to the user requirements and tool availability. The two main paradigms to let end users formulate queries without resorting to a standard query language such SPARQL, are through a version of Query By Diagram or a near-natural language interface. Such tools for OWL do exist, e.g. [16, 17], but they are self-standing applications that do not operate through a web interface, which, for a web portal to function as a web portal, is an essential requirement. In addition, in our case study setting, it is not yet clear if additional user-query paradigms are necessary for people with disabilities. Therefore, this last step has yet to be implemented to stakeholder satisfaction. This HCI topic of user-acceptable query interface is now considered not only to be a familiar engineering issue by the NAP developers but also to be an impetus to make a larger contribution to technologies for independent living thanks to the 'hidden intelligence' at the back-end.

In addition to showing the realistic option to semantically enhance web portals, the demonstration of OBDA-enhancements in a HealthCare & Life Sciences setting may help to scope the planning of activities on KnowledgeBase improvements during the new lease of the W3C HCLS IG[9], noting that the same technology can be used also for ontology-based database integration [18, 19] and to obtain relative good performance with large databases [4].

4 Conclusions

We have successfully demonstrated enhancing an existing web portal, the NAP, with OBDA by developing an experimental domain ontology, ADOLENA, converting it to an OBDA-compliant one, declaring ontology-to-database SQL query mappings, and performing SPARQL queries to retrieve answers that were significantly better than the standard keyword-based searches. In addition, the methodology proposed is generalisable to other web portals that are based on generic software. The procedure both to conduct the steps as well as to link the outputs of each step to the input of the next one, however, has been carried out manually, which will benefit from more automation. Indeed, ongoing research and tool development are moving in the direction of automation of these steps.

Current and future work includes adding more knowledge to the ontology, finalising the last step of web-based OBDA querying, and providing a more detailed methodology and automated workflow environment for OBDA-focussed systems development.

Acknowledgements. We would like to thank Diego Calvanese, Giuseppe De Giacomo, Mariano Rodríguez, and Fabio Savo for their feedback and suggestions on the manual and automated transformations to $DL\text{-}Lite_{\mathcal{A}}$-compliant ontologies and OBDA mappings and software usage support.

[9] http://www.w3.org/2001/sw/hcls/

References

1. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. Journal of Automated Reasoning **39**(3) (2007) 385–429
2. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. on Data Semantics **X** (2008) 133–173
3. Rodriguez-Muro, M., Lubyte, L., Calvanese, D.: Realizing Ontology Based Data Access: A plug-in for Protégé. In: Proc. of the Workshop on Information Integration Methods, Architectures, and Systems (IIMAS'08), IEEE Computer Society (2008)
4. Poggi, A., Rodriguez, M., Ruzzi, M.: Ontology-based database access with DIG-Mastro and the OBDA Plugin for Protégé. In: Proc. of OWLED'08 DC. (2008)
5. Acciarri, A., Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Palmieri, M., Rosati, R.: QuOnto: Querying Ontologies. In: Proc. of AAAI'05. (2005) 1670–1671
6. Nyulas, C., OConnor, M., Tu, S.: DataMaster – a plug-in for importing schemas and data from relational databases into Protege. In: Proceedings of the 10th International Protege Conference. (2007) Stanford Medical Informatics.
7. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Ontology library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003). (2003)
8. Ruggieri, A.P., Elkin, P.L., Solbrig, H., Chute, C.G.: Expression of a domain ontology model in unified modeling language for the World Health Organization International Classification of Impairment, Disability, and Handicap, version 2. In: Proc AMIA Symp. 2001. (2001) 573–577
9. Lubyte, L., Tessaris, S.: Extracting ontologies from relational databases. In: Proc. of DL'07. (2007) 387–395
10. Jansen, L.: Tendencies and other realizables in medical information sciences. The Monist, Special Issue on Biomedical Ontologies **90** (2007) 4
11. Keet, C.M., Rodríguez, M.: Toward using biomedical ontologies: trade-offs between ontology languages. In: AAAI 2007 Workshop Semantic eScience (SeS 2007). Volume WS-07-11 of AAAI Technical Report., AAAI (2007) 65–68
12. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. Proceedings of KR-2006 (2006) 452–457
13. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Can OWL model football leagues? In: Proc of OWLED 2007. CEUR-WS (2007) 6-7 June 2007, Innsbruck, Austria.
14. Fillottrani, P., Franconi, E., Tessaris, S.: The new ICOM ontology editor. In: Proc. of DL'06. (2006) Lake District, UK. May 2006.
15. Lubyte, L., Tessaris, S.: Supporting the design of ontologies for data access. In: Proc. of DL'08. (2008)
16. Dongilli, P., Franconi, E.: An Intelligent Query Interface with Natural Language Support. In: Proc. of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2006), Melbourne Beach, Florida, USA (May 2006)
17. Fadhil, A., Haarslev, V.: OntoVQL: a graphical query language for OWL ontologies. In: Proc, of DL'07. (2007) Bressanone, Italy.
18. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R., Ruzzi, M.: Data integration through DL-liteA ontologies. In: Proc. of the Int. WS. on Semantics in Data and Knowledge Bases (SDKB 2008). LNCS, Springer (2008)
19. Amoroso, A., Esposito, G., Lembo, D., Urbano, P., Vertucci, R.: Ontology-based data integration with MASTRO-I for configuration and data management at SELEX Sistemi Integrati. In: Proc. of SEBD'08. (2008) 81–92