

Aiding the Data Integration in Medicinal Settings by Means of Semantic Technologies^{*}

Vít Nováček¹, Loredana Laera², and Siegfried Handschuh¹

¹Digital Enterprise Research Institute, National University of Ireland, Galway
IDA Business Park, Lower Dangan, Galway, Ireland
E-mail: `first_name.last_name@deri.org`

²Department of Computer Science
University of Liverpool, UK
E-mail: `lori@csc.liv.ac.uk`

Abstract. The paper introduces basic features of a novel ontology integration framework that explicitly takes the dynamics and data-intensiveness of many practical application scenarios into account. We motivate our research partially by the needs of bio-medicine scenarios that have been recently identified within the search for semantics-enabled solutions. In this context, we show a concrete example of the integration process in the life-sciences settings. Moreover, we elaborate a possible bio-medicine industry application domain of the presented framework and explain the benefits of the proposed semantic solution.

1 Introduction

Ontologies have been recently considered as a valuable extension of traditional data-management techniques, since they allow to add a machine comprehensible *meaning* to the traditional repositories (e.g. databases, natural language resources). Thus we can not only query, but also reason with the knowledge contained within the data, inferring implicit facts on a mathematically well-founded basis. Ontologies can also facilitate data integration by means of ontology mapping techniques, which is a very sought-after feature in the practical scenarios the Semantic Web solutions can provide.

1.1 Motivation

The domain of medicine, which is one of the most important application domains we take into account, suffers from lack of mechanisms that would allow to efficiently query, integrate and manage constantly changing and growing data in health-care applications. Ontologies naturally provide a solution to this situation, however, there are still open issues concerning the dynamic and data-intensive character of (not only) the medicinal knowledge.

^{*} This work has been supported by the EU IST 6th framework's Network of Excellence 'Knowledge Web' (FP6-507482), the 'PIPS' project (FP6-IST 2004-507019) and partially by Academy of Sciences of the Czech Republic, 'Information Society' national research program, the grant AV 1ET100300419.

Ontology construction in medicine is usually the result of collaboration (which involves cooperation among ontology engineers and domain experts) through a manual process of the extraction of knowledge. However, it is not always feasible to process all the relevant data and extract the knowledge from them manually, since we might not have a sufficiently large committee of ontology engineers and/or dedicated experts at hand in order to process new data anytime it occurs. This implies a need for (partial) automation of ontology extraction and management processes in dynamic and data-intensive medical environments. This can be achieved by ontology learning [1]. Within the Knowledge Web EU NoE, we have developed a lifecycle [2] of the ontology development process supporting appropriate mechanisms for dealing with the large amounts of knowledge that are *dynamic* in nature. Within the lifecycle’s implementation, the ontology integration is one of the most important problems, forming the focus of this paper. We have followed certain practical requirements in this context:

1. the ability to process new knowledge (resources) automatically whenever it appears and when it is inappropriate for humans to incorporate it;
2. the ability to automatically compare the new knowledge with a “master” ontology that is manually and collaboratively designed and select the new knowledge accordingly;
3. the ability to resolve possible major inconsistencies between the new and current knowledge, possibly favouring the assertions from presumably more complex and precise master ontology against the learned ones;
4. the ability to automatically sort the new knowledge according to user-defined preferences and present it to them in a very simple way, thus further alleviating human efforts in the task of final incorporation of the knowledge.

On one hand, using the automatic methods, we are able to deal with large amounts of changing data. On the other hand, the final incorporation of new knowledge is to be decided by the expert human users, repairing possible errors and inappropriate findings of the automatic techniques. The key to success and applicability is to let machines do most of the tedious and time-consuming work and provide people with concise and simple suggestions on ontology integration.

1.2 Structure of the Paper

The rest of the paper is organized as follows. Section 2 presents the basic features of the integration technique. In Section 3, we give a simple illustrative example of concrete usage of our integration approach. Section 4 discusses a realistic medicine application domain in which our lifecycle framework can help. Section 5 concludes the paper reporting also on the current state of the implementation and identifying the steps needed to be taken to make the framework industry-mature.

2 Basic Features of the Integration Framework

We call our ontology lifecycle framework DINO, which is an abbreviation of its three key elements – *Dynamics*, *INtegration* and *Ontology*. However, the first two can also be *Data* and *INtensive*. All these features express the primary aim of our efforts – *to make the knowledge efficiently and reasonably manageable in data-intensive and dynamic domains*.

As emphasised above, the key novelty of the DINO ontology lifecycle framework is its support for integration of changing knowledge in data-intensive domains. A detailed description of the technical innards is given in [3, 4]. In the following list, we describe only the very basic features of the framework:

- **OWL standard conformance** – the OWL (DL flavour) Semantic Web W3C standard is supported in all phases of the integration by default.
- **Ontology learning** – newly coming data (e.g. patient records, scientific papers, clinical reports) are automatically processed by an ontology learning component.
- **Reference ontology developed by the community** – the master reference domain ontology is maintained by domain experts by means of a simple-to-use ontology development portal interface.
- **Ontology alignment/negotiation** – learned ontology is merged with the master reference by means of automatic negotiation of an agreed ontology alignment.
- **Inconsistency resolution** – possible inconsistencies are resolved using a reasoning engine and simple heuristics, producing an integrated ontology; also, a natural language representation of inconsistencies found can be presented to users in order to let them tackle it manually if needed.
- **Extension triples generation** – triples extending the master ontology are computed by comparison with the integrated one.
- **Mapping triples to natural language suggestions** – from the extending triples, easy-to-comprehend natural language suggestions sorted according to preferences supported by users are generated.

3 Usage Example

In the following we provide a simple illustrative example of the concrete usage of DINO integration mechanism. Imagine a medical institution that has developed an ontology O_M (see the master O_M ontology in Figure 3) covering the basic concepts in clinical practice and research, possibly with help of ontology engineering experts when deploying the DINO framework. The ontology may need to be extended by new information in research (e.g. when new treatments or diagnosis methods are developed and published). Related information can be found in respective documents (research papers, industry white-papers, etc.). Figure 1 presents a sample text fragment with the respective learned OWL ontology O_L (we omit the namespace for simplicity).

The ontologies O_L and O_M are aligned and negotiated (see Figure 2). The preferences have been chosen on the basis of the ontological information of O_L and O_M .

The O_M ontology and the ontology O_A , consisting of axioms produced from the negotiated mappings are shown in Figure 3.

When trying to merge the O_M and O_L ontologies into an integrated ontology (O_I), we find out that there is one inconsistency – “*disease*” is said to be a subclass of “*dysfunction*” and vice versa, which creates a cycle in the taxonomy. Therefore we remove the respective “invalid” assertion that originated from the O_L ontology. On the other hand, we can extend the learned knowledge based on range and domain of the “*DiscoveredUsing*” property. We can infer new assertions on the instantiation of “*cerebellar astrocytoma*” (instance of “*Manifestation*”) and “*CT*” (instance of “*DiagnosisProcedure*”).

Now we can produce the triples (with O_L equivalent labels replaced by those from O_M) from the O_I merge, together with respective suggestions based on the differences

```

... while cerebellar astrocytoma
is usually discovered by means of
CT...using a diagnostic procedure
of scanning...GVHD, an immune
dysfunction...GVHD, a disease being a type
of dysfunction...

...
<owl:ObjectProperty rdf:ID="discovered-by"/>
<owl:Thing rdf:ID="CT"/>
<owl:Thing rdf:ID="cerebellar-astrocytoma">
  <discovered-by rdf:resource="#CT"/>
</owl:Thing>
<owl:Class rdf:ID="diagnostic-procedure"/>
<owl:Class rdf:ID="immune-dysfunction"/>
<owl:Class rdf:ID="dysfunction"/>
<owl:Class rdf:ID="scanning">
  <rdfs:subClassOf rdf:resource="#diagnostic-procedure"/>
</owl:Class>
<immune-dysfunction rdf:ID="GVHD"/>
<owl:Class rdf:ID="disease">
  <rdfs:subClassOf rdf:resource="#dysfunction"/>
</owl:Class>
...

```

Fig. 1. A text sample and the learned O_L ontology

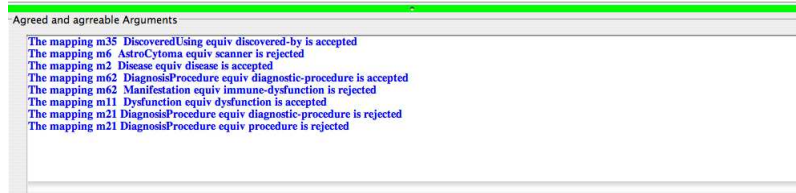


Fig. 2. Negotiated mappings

between O_I and O_M . We present the sorted triples and their transformations into natural language statements¹ in Table 1.

<AstroCytoma rdf:ID="cerebellar-astrocytoma"/>	+0.667: CEREBELLAR ASTROCYTOMA is a <i>new instance</i> of ASTROCYTOMA.
<Manifestation rdf:ID="cerebellar-astrocytoma"/>	+0.667: CEREBELLAR ASTROCYTOMA is a <i>new instance</i> of MANIFESTATION.
<DiagnosisProcedure rdf:ID="CT"/>	+0.389: CT is a <i>new instance</i> of DIAGNOSIS PROCEDURE.
<immune-dysfunction rdf:ID="GVHD"/>	+0.333: GVHD is a <i>new instance</i> of IMMUNE DYSFUNCTION.
<owl:Class rdf:ID="scanning"> <rdfs:subClassOf rdf:resource="#DiagnosisProcedure"/> </owl:Class>	-0.444: A <i>new class</i> SCANNING is a <i>sub-class</i> of DIAGNOSIS PROCEDURE.
<owl:Thing rdf:ID="cerebellar-astrocytoma"> <DiscoveredUsing rdf:resource="#CT"/> </owl:Thing>	-0.667: CEREBELLAR ASTROCYTOMA is DISCOVERED USING CT.
<owl:Class rdf:ID="immune-dysfunction"> <rdfs:subClassOf rdf:resource="#Dysfunction"/> </owl:Class>	-0.833: A <i>new class</i> IMMUNE DYSFUNCTION is a <i>sub-class</i> of DYSFUNCTION.

Table 1. Extension triples and the respective NL suggestions induced by the integrated O_I ontology

Note that the above example may be also used if we just need to align and possibly extend the ontology with another institution’s knowledge base – the only difference is that we do not perform the ontology learning and also omit retractions in the integration process. This can be applied in the critical task of inter-mediation of medicine information, for example.

¹ They are preceded by respective sample relevance values, corresponding to {Scanning, discover, cytoma} and {subclass, disease, dysfunction} sets of preferred and unwanted terms, respectively.

```

...
<owl:ObjectProperty rdf:ID="InstrumentalProperty"/>
<owl:ObjectProperty rdf:ID="DiscoveredUsing">
  <rdfs:subPropertyOf rdf:resource="#InstrumentalProperty"/>
  <rdfs:range rdf:resource="#Manifestation"/>
  <rdfs:domain rdf:resource="#DiagnosisProcedure"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Manifestation"/>
<owl:Class rdf:ID="Procedure"/>
<owl:Class rdf:ID="DiagnosisProcedure">
  <rdfs:subClassOf rdf:resource="#Procedure"/>
</owl:Class>
<owl:Class rdf:ID="SoftTissueCytoma"/>
<owl:Class rdf:ID="AstroCytoma">
  <rdfs:subClassOf rdf:resource="#SoftTissueCytoma"/>
</owl:Class>
<owl:Class rdf:ID="Disease">
<owl:Class rdf:ID="Dysfunction">
  <rdfs:subClassOf rdf:resource="#Disease"/>
</owl:Class>
...
...
<owl:ObjectProperty rdf:ID="DiscoveredUsing">
  <owl:equivalentProperty rdf:resource="#discovered-by"/>
</owl:ObjectProperty>
<AstroCytoma rdf:ID="cerebellar-astrocytoma"/>
<owl:Class rdf:ID="DiagnosisProcedure">
  <owl:equivalentClass rdf:resource="#diagnostic-procedure"/>
</owl:Class>
<owl:Class rdf:ID="immune-dysfunction">
  <owl:subClassOf rdf:resource="#Dysfunction"/>
</owl:Class>
<owl:Class rdf:ID="Dysfunction">
  <owl:equivalentClass rdf:resource="#dysfunction"/>
</owl:Class>
...

```

Fig. 3. A master O_M ontology sample and the respective mapping

4 Selected Application Domain – Longitudinal Electronic Health Record

Several application domains have been discussed according to the use case areas identified in [5]. Although, these areas are rather broad, we can focus here on the needs that our ontology lifecycle/integration framework can (at least partially) cover for one selected domain. Four another related ones are covered by our work [3].

The main topic in *longitudinal electronic health record* activities is development of standards and platforms supporting creation and management of long-term electronic health records of particular patients. These should be able to integrate various sources of data coming from different medical institutions a patient may have been treated in during his whole life.

The integration of different data sources requires automated technologies to facilitate this task. Common abstract conceptual structure of the electronic health record needs to be populated and/or extended by concrete data, present very often in unstructured natural language form. The electronic health record should also be open to efficient and expressive querying.

Ontologies bound to patient data resources in particular institutions can very naturally support integration of respective data into longitudinal electronic health records. Once there is an ontology describing the underlying data, we can directly use the integration mechanism presented here in order to manage the needed integration semi-automatically. Moreover, the DINO framework can serve for easy and laymen-oriented ontology development already at the particular institutions' side. Support for ontology learning directly facilitates the population/extension. Querying of ontology-enabled electronic health records is straightforward in our framework, since it is possible using the state of the art OWL DL reasoning tools.

5 Conclusions and Future Work

The key contribution of our work is the development of an ontology integration framework conforming to the requirements specified in Section 1.1. Moreover, we have described a sample life-sciences industry application domain that can benefit from

our semantic solution. Concerning the state of the framework itself, we have recently completed initial draft implementation of the DINO integration technique in line with the architecture and algorithms described in [3, 4]. Implementation of function returning natural language representation of suggestions and inconsistencies is very naïve and hard-coded now, however, a working connection with general natural language generation tools developed within the SEKT EU project should be ready very soon. Combination of the DINO integration technique and MarcOnt Portal [6] within an extension of Protégé state of the art ontology editing and maintenance tool [7] is currently being implemented in order to provide stand-alone and coherent framework for all the phases of an ontology lifecycle presented in [2].

Now we are in the phase of intensive testing and debugging of the whole DINO integration proof-of-concept implementation. The testing data we take into account are mainly PubMed digital archive² as ontology learning resource pool and (fragments of) Galen ontology³ as a master knowledge base.

In line with preparing an industry-mature implementation of the framework, we plan to continuously evaluate and improve its implementation according to demands of interested partners in the medicine industry (possibly, but not only within the application domains identified in our research) and also in other application fields.

References

1. Maedche, A., Staab, S.: Ontology learning. In Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer-Verlag (2004) 173–190
2. Nováček, V., Handschuh, S., Laera, L., Maynard, D., Völkel, M., Groza, T., Tamma, V., Kruk, S.R.: Report and prototype of dynamics in the ontology lifecycle (D2.3.8v1). Deliverable 238v1, Knowledge Web (2006)
3. Nováček, V., Laera, L., Handschuh, S.: Dynamic integration of medical ontologies in large scale. In: Proceedings of WWW2007/HCLSDI Workshop, ACM Press (2007) In press.
4. Nováček, V., Dabrowski, M., Kruk, S.R., Handschuh, S.: Extending community ontology using automatically generated suggestions. In: Proceedings of FLAIRS 2007, AAAI Press (2007) In press.
5. Eichelberg, M.: Requirements analysis for the ride roadmap. Deliverable D2.1.1, RIDE (2006)
6. Dabrowski, M.: Marcont initiative technical report. Technical report, DERI, Digital Enterprise Research Institute (2006)
7. Gennari, J.H., Musen, M.A., Ferguson, R.W., Grosso, W.E., Crubezy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies* **58** (2003) 89–123

² See <http://www.pubmedcentral.nih.gov/>.

³ Its OWL DL translation, see <http://www.co-ode.org/galen/>.