

Information System Analysis

Jasmin Opitz, Bijan Parsia, Ulrike Sattler

The University of Manchester
{opitzj|bparsia|sattler}@cs.manchester.ac.uk

Abstract. Ontology-based data access has received a lot of attention recently, yet there is no clear methodology to evaluate a “semantically enriched” information system in general or an ontology based data access system in particular. The quality of such an information system clearly depends on how well your data fits your class-level ontology, and how well these two components fit your queries. This paper presents a generic, flexible framework for this kind of analysis: it can be used, e.g., to compare two class-level ontologies w.r.t. their fitness for a given kind of data and query set. We apply the framework to an example case and show how it helps to answer relevant modelling and representation questions.

1 Introduction

In this paper we present a framework for evaluating the quality of “semantically enriched” information systems (IS). By that we mean IS that distinguish between schema and data and are geared towards answering queries. The idea behind that is to encapsulate domain experts’ background knowledge into the query answering mechanism in order to improve recall and precision. A typical example of such an IS is an ontology-based data access (OBDA) system that uses a class-level ontology (or Tbox) as a schema and stores the data in a database. Queries retrieve tuples of individuals from the database that answer the query w.r.t. the schema.

The proposed evaluation framework measures the well-suitedness of the various components of an IS. It can be applied to any IS that involves a schema, a collection of data, a collection of information requests and a query language (QL). We call this a *modelling approach (MA)* for an IS. Thus, the framework is generic and can be applied to a variety of scenarios, e.g. for comparing different OBDA systems or for comparing different IS using database schemas or for comparing heterogeneous systems. OBDA has received a lot of attention recently and can come in many different fashions, e.g. regarding the expressive power of the ontology or the supported query language [2, 3, 1, 10, 6, 9].

When applying the framework we look at *information requests* as abstractions of queries (they are independent of schema and QL). For each *MA* the framework measures *if* an information request can be answered by a query in a given QL over the given schema and data and *how good* the query is. More precisely, the metrics produced by the evaluation framework are the *fitness* of an *MA*, i.e. the ability of formulating “good” queries, and the *flexibility*, i.e. the number of different “good” ways of expressing a query.

These measurements can help IS designers in taking important design decisions, e.g. whether to use an off-the-shelf schema or one that is specifically tailored to the application or which OBDA technique or tool to use. The measurements also point out which queries can and cannot be answered w.r.t. a given *MA* and how complicated it is to formulate these queries. That allows IS designers to identify, compare and discuss weak and strong points of their *MA* and manage trade-offs between modelling effort, maintainability and scalability.

In the following we will explain the technical details of the evaluation framework and its measurements. Furthermore, we will outline a case study in which we applied the framework to compare different ontology-based *MA* for medical image annotations.

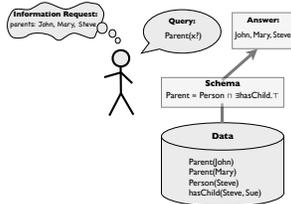


Fig. 1. A modelling approach plus a query and its answers.

2 Information System Evaluation Framework

We start by formalising the relevant components of a (semantically enriched) information system for which we are then going to evaluate and compare different modelling approaches. We will use the term “modelling approach” to describe the whole system consisting of data, schema, (an abstraction of) queries, and a query language as depicted in Figure 1.

2.1 Modelling Approach

A *modelling approach* $MA = (S, D, R, QL)$ consists of

- a schema S : a finite description of the semantics of the data, e.g. a database schema, a logic program, or the TBox of an ontology, which can be empty.
- the data D : e.g. tables and rows in a relational database, ground facts, or ontology ABox assertions.

- a set of information requests R : each $r \in R$ represents the answer to a query of D , and is given as a set (of tuples) over D . Ideally, R should be representative for the queries to be answered by the information system to be built.
- a query language QL : e.g. SQL, (union of) conjunctive queries, OWL class expressions.

An information request asks for tuples of the given data that are relevant for the user. The request needs to be distinguished from the actual query, which is a specific manifestation of the information request formulated in QL , see Figure 1. An information request r can correspond to 0, 1 or more queries in a given query language. The former is the case if there are no queries in QL whose answers would be exactly the tuples in r when asked over S and D , i.e., if QL is unable to express the information request over the given schema and data. In the case that there are one or more queries, some of them might be more easily expressible than others. In Figure 1, we sketch a case where the user wants to retrieve three individuals, John, Mary, Steve, from the database that are known to be parents—but not all of which are explicitly stored as parents. Still, in the presence of the given schema, the query $\text{Parents}(?x)$ can be formulated to retrieve exactly those three individuals.

The only assumptions we make is that the query language QL comes with a semantics that identifies, for a given query q of arity n in QL , data D , and schema S , the set of *certain answers* [4]. More precisely, we assume the existence of an entailment relation \models , and use $\text{Ind}(D)$ for the set of individuals or constants in D to define $\text{cert}(\cdot)$ as follows:

$$\text{cert}(q, S, D) = \{\mathbf{w} \in \text{Ind}(D)^n \mid S \cup D \models q(\mathbf{w})\}.$$

2.2 Applying the Framework

The basic characteristics we want to evaluate is the *fitness* of an MA , i.e. how well the schema and the data are suited to enable the formulation of “fit” queries for answering the given information requests, and the *flexibility* of an MA , i.e. the number of “fit” queries that can be formulated for answering the given information requests. The fitness and flexibility of an MA can be determined by analysing the syntactic, semantic and/or cognitive complexity of the queries that correspond to the information requests and depends on the *fitness function*.

The Fitness Function Different queries that correspond to an information request can vary in length and be more or less complex, e.g. in terms of using relations and constructors such as conjunctions, disjunctions, etc. They can also be more or less difficult to understand from a cognitive perspective. For example, a human user might find a query that uses terms that are actual words (in the sense that they exist in a domain expert’s dictionary) easier to understand than

one that uses anonymous identifiers. The purpose of the fitness function is to capture this complexity.

The framework is parametrized with a *fitness function* f that associates each query q in QL with some value $f(q)$ that is intended to capture its fitness. We only require that f maps QL into a totally ordered set $(M, <)$, e.g. \mathbb{R} or \mathbb{N}^4 , which we call the query’s *fitness value*. Obvious examples of fitness functions are (i) a query’s length, (ii) a query’s length combined with the number of constructors involved, either via some (weighted) summation or into a vector, or (iii) a query’s length combined with the number of terms not to be found in Wikipedia or a suitable lexicon, or any combinations or extensions of these.

The smaller the fitness value, the “better” the query. We read $f(q) < f(q')$ as q being “better” or “fitter” than q' . The framework evaluates the “best queries” for an information request, e.g., the shortest queries. The fitness function induces a partial order on the queries.

The Query Space Each information request $r \in R$ has an associated query space: first, we define *correct queries* $cQ(r, S, D)$ as those that answer exactly an information request r over S and D :¹

$$cQ(r, S, D) = \{q \mid q \text{ is a } QL \text{ query and } cert(q, S, D) = r(D)\}.$$

Next, we define *best queries* $bQ(r, S, D, f)$ as those correct queries whose fitness is maximal. Clearly, best queries depend on how we measure fitness, and thus on the fitness function f :

$$bQ(r, S, D, f) = \{q \in cQ(r, S, D) \mid \text{there is no } q' \in cQ(r, S, D): \\ f(q') < f(q)\}.$$

Since the $bQ(\cdot)$ are the “fittest” queries among the correct queries, any two queries in $bQ(\cdot)$ are equally fit, and we can abbreviate their fitness as follows: for $f(q_i) = f(q_j)$, we set $f(\{q_1, \dots, q_k\})$ to be $f(q_1)$. For an empty set, e.g., if an information request cannot be expressed in QL over S and D , we set $f(\emptyset) = \max_{<}(M)$ if such a maximum exists, i.e., maximally unfit, or to some other *very* unfit value.

If we want to consider the flexibility of an *MA*, we simply need to consider the *number* of best queries, i.e., the cardinality of $bQ(\cdot)$. Depending on the application domain, we can adapt the framework to consider only non-redundant queries for measuring the flexibility. For example, if S is a class-level OWL

¹ Currently, the framework does not consider approximations of *correct* queries for measuring the fitness of the modelling approach, see the discussion in Section 4.

ontology and QL are OWL class expressions for instance retrieval, we can count all elements in $bQ(\cdot)$ that are not structurally equivalent [7].

2.3 Applying the Framework to OWL

We will now specify an instantiation of the framework to evaluate OWL ontology-based data access approaches. This specification is still quite flexible: e.g., we cover both the case where the data resides in a database and the case where it is part of an ontology. An $MA = (\mathbf{T}, \mathbf{A}, R, CL)$ consists of

- a TBox \mathbf{T} , i.e., a set of OWL class-level axioms,² that describes the conceptual model and the terminology of the domain, plus possibly a set of mappings in the sense of [3],
- an ABox \mathbf{A} , i.e., a set of OWL assertions about named individuals, or, in the presence of the above mentioned mappings, tables from a relational database from which these mappings are defined,
- a set R of information requests r , i.e., sets of tuples of OWL individuals, and
- CL is the set of OWL class expressions as a query language.

OWL class expressions are an obvious choice for a query language, but there are more expressive ones such as conjunctive queries, unions of conjunctive queries [4], SPARQL, SPARQL-DL,³ or nRQL [11].

For an ontology-based modelling approach, we suggest a fitness function as follows: $f(q)$ is a *fitness vector* (a, b, c, d) that contains (i) a as the length $|q|$ of q , (ii) b as the number of distinct OWL constructors in q , (iii) c as the role nesting depth of q , and (iv) d as a flag that is set to 1 if q contains unintelligible codes, and 0 if all terms in q are human readable otherwise. We compare the fitness of queries via the lexicographic ordering from the left of their fitness vectors.

A Simple Example We use a simple example to that captures data about parents and their children. We will use this example to illustrate the components of the modelling approach as well as the query space for an information request.

Consider the modelling approach $MA = (\mathbf{T}, \mathbf{A}, R, CL)$ consisting of the following TBox and ABox

$$\begin{array}{ll} \mathbf{T} = \{ & \mathbf{A} = \{ \\ \quad \textit{Father} \equiv \textit{Man} \sqcap \exists \textit{hasChild}.\top, & \quad \textit{Father}(\textit{John}), \\ \quad \textit{Mother} \equiv \textit{Woman} \sqcap \exists \textit{hasChild}.\top, & \quad \textit{Mother}(\textit{Mary}), \\ \quad \textit{Father} \sqsubseteq \textit{Parent}, & \quad \textit{hasChild}(\textit{Mary}, \textit{Tom}), \\ \quad \textit{Mother} \sqsubseteq \textit{Parent} \} & \quad \textit{hasChild}(\textit{John}, \textit{Tom}) \} \end{array}$$

² More precisely, OWL 2 class expression axioms, property axioms, datatype definitions, and keys, see <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>.

³ See http://www.w3.org/2009/sparql/wiki/Main_Page and *entailment regimes*.

Now consider the information request $r(\mathbf{A}) = \{Mary, John\}$, i.e., r retrieve “all parents”. Using OWL class expressions as a query language, the following queries could be considered:

$$\begin{array}{ll}
 q_1 = Parent & q_2 = Father \sqcup Mother \\
 q_3 = \exists hasChild.\top & q_4 = Woman \sqcup Man \\
 q_5 = Father & q_6 = Mother
 \end{array}$$

The correct queries for r are $cQ(r, \mathbf{T}, \mathbf{A}) = \{q_1, q_2, q_3, q_4\}$, and not all of them are equivalent. The queries q_5 and q_6 are not correct because they return only incomplete answers. W.r.t. the above mentioned fitness function, we have only one best query, q_1 , because it is the shortest correct query.

Please note that, w.r.t. the data given here, q_4 is correct for r , and it would be interesting to see what would happen if we extended \mathbf{A} with, say, $Man(Tom)$: either r will change as well to include Tom , or q_4 ceases to be a correct query.

2.4 Using the Evaluation Framework to Compare Modelling Approaches

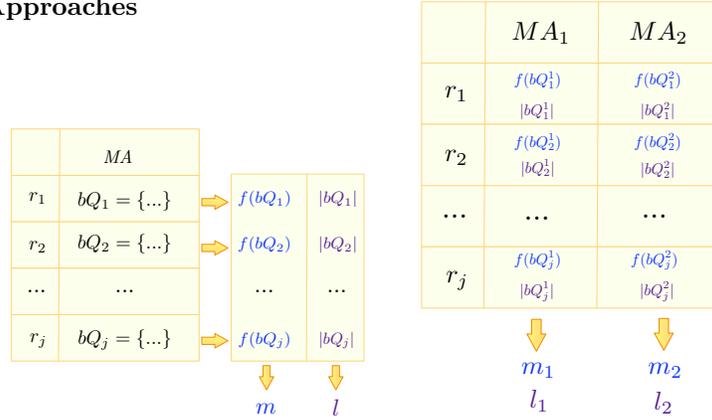


Fig. 2. General and comparative measurements for modelling approaches.

On the left hand side of Figure 2, we have sketched an evaluation of a modelling approach MA where, for each information request $r_i \in R$, we have computed the best queries for r_i , and then their fitness and cardinality. Clearly, if we want to compare two modelling approaches MA_1 and MA_2 , we can do the same and compare, for each information request $r_i \in R$ and each of the two modelling approaches, the fitness and cardinality of the best queries. This can unveil the strengths and weaknesses of the information system to the system designer. For example, if there are information requests for which the set of correct queries is empty, then $f(bQ(r, S, D, f))$ is prohibitively bad. To overcome this, we can then decide whether to select a different, more powerful query language or to change the schema or the way the data is modelled—or whether perhaps that

particular information request is of too little importance for such a change. The measurements can also help to point out where the trade-offs between modelling effort and benefits in terms of easier query answering are. For example, considering an ontology-based modelling approach, whether more modelling effort for a more expressive TBox would be justified for the sake of simpler queries.

In addition, we can aggregate the fitness and flexibility of a modelling approach: this can be interesting if we want to compare two such modelling approaches en gros. In what follows, we use *AGG* to stand for an aggregation function such as *min*, *max*, *avg*, or *count*. This function can be fixed in the particular application of the framework.

We can aggregate both the fitness and the flexibility of a modelling approach: the overall fitness of a modelling approach $f(MA)$ is aggregated over the fitness vectors of all best queries for all information requests, i.e.,

$$m = AGG \bigcup_{r \in R} f(bQ(r, S, D, f)).$$

The overall flexibility of a modelling approach aggregates over the cardinality of all best queries for all information requests, i.e.,

$$\ell = AGG \bigcup_{r \in R} |bQ(r, S, D, f)|.$$

As illustrated in Figure 2, applying the framework to *one* modelling approach $MA = (S, D, R, QL)$ reveals

- for each r , the fitness value of the best queries: $f(bQ_j)$. In particular, it will identify information requests for which it is hard to specify a query in QL and those for which this is impossible.
- for each r , the number of best queries: $|bQ_j|$
- the aggregated fitness value m for the entire MA
- the aggregated flexibility ℓ of the entire modelling approach MA

When comparing different modelling approaches (as shown in Figure 4) we can compare the

- point-to-point fitness for each information request
- overall (aggregated) fitness m of the modelling approaches
- point-to-point flexibility for each information request
- overall (aggregated) flexibility ℓ of the modelling approaches

3 A Case Study: Ontology-Based Annotations

We will now present the application of the evaluation framework in a case study about ontology-based annotations of medical images and their descriptions. The study is described in more detail in [8].

The modelling process involved a number of design decisions. First, we chose to use a module of the established medical ontology SNOMED CT⁴ as the TBox of our annotation ontology⁵ and translated natural language radiology reports of 50 medical images to ABox assertions of that ontology. The textual descriptions contain medical information such as image type, image modalities, clinical findings, body structures and diagnoses. Next, we had to decide whether the ABox assertions should be simple class assertions of the relevant medical terms occurring in the text or whether the ABox should contain class and object property assertions, trying to closely reflect the meaning of the text. Furthermore, the SNOMED CT TBox has a very complex structure containing *role groups* [5] that are used e.g. to model diseases that relate findings to body structures. We had to find a way to translate the textual descriptions in accordance with this complex structure.

3.1 The Modelling Approaches

In the following, we present three different modelling approaches. MA_1 models the data with a simple ABox that contains almost only class assertions: individuals are only linked by a single object property *shows* in order to relate an image to the individuals shown in it. MA_2 uses class and object property assertions that capture the relational structure of the image descriptions. MA_3 uses a slightly different TBox than MA_1 and MA_2 in the sense that we created an additional set of roles and a role hierarchy in order to bypass the SNOMED CT specific role groups. An example of a disease in SNOMED CT that is defined using role groups is *NeoplasmOfLung*. The concept is defined as follows:⁶

$$\text{NeoplasmOfLung} \equiv \text{DisorderOfLung} \sqcap \\ \exists \text{roleGroup} (\exists \text{AssociatedMorphology} . \text{Neoplasm} \sqcap \\ \exists \text{FindingSite} . \text{LungStructure})$$

For MA_3 , we introduced three additional object properties: *shows*, *hasFinding* and *hasLocation* and defined the following role hierarchy:

$$\begin{aligned} \text{roleGroup } o \text{ AssociatedMorphology} &\sqsubseteq \text{hasFinding} \\ \text{roleGroup } o \text{ FindingSite} &\sqsubseteq \text{hasLocation} \\ \text{shows } o \text{ hasFinding} &\sqsubseteq \text{shows} \\ \text{shows } o \text{ hasLocation} &\sqsubseteq \text{shows} \end{aligned}$$

If we want to find all images that show neoplasms in MA_3 , we can formulate a simple OWL class expression query like $\text{Image} \sqcap \exists \text{shows} . \text{Neoplasm}$ and would retrieve images labelled with

⁴ <http://www.ihtsdo.org/snomed-ct/>

⁵ <http://www.cs.man.ac.uk/~opitzj/snomed/snomedLungModuleImageAnnotations.owl>

⁶ To improve readability, we use slightly abbreviated class names and DL syntax.

Image $\sqcap \exists$ roleGroup. \exists AssociatedMorphology.NeoplasmOfLung without having to use the complicated role group construct in the query.

We compare the following modelling approaches:

$$MA_1 = (\mathbf{T}_1, \mathbf{A}_1, R, CL) \quad MA_2 = (\mathbf{T}_1, \mathbf{A}_2, R, CL) \quad MA_3 = (\mathbf{T}_2, \mathbf{A}_3, R, CL)$$

where \mathbf{T}_1 is the original SNOMED CT TBox and \mathbf{T}_2 the TBox with the additional role hierarchy. \mathbf{A}_1 is an ABox with the data formulated in terms of simple class assertions whereas \mathbf{A}_2 and \mathbf{A}_3 use class assertions as well as object property assertions.

3.2 The Information Requests

The set of information requests R is derived from the content of the original, natural language image descriptions: clinical findings, findings located in body parts, complex findings (involving role groups), image types and modalities and combinations of the former. We will now list some representative information requests.

- r_1 : An information request that involves one clinical finding: “All images that show neoplasms.”
- r_2 : An information request that involves two concepts, an image type and an image projection: “All X-ray images with PosteroAnterior (PA) projection.”
- r_3 : An information request that involves a clinical finding combined with a qualifier value: “All images that show left-sided pleural effusions.”
- r_4 : An information request that involves a clinical finding combined with a body structure: “All images that show soft tissue masses in the pleural membrane.”

We expect that MA_1 is good for formulating queries for simple requests (such as those that ask for just one concept, e.g. r_1 and r_2) whereas MA_2 is more appropriate for formulating queries for complex requests that involve relations between concepts, such as r_3 and r_4 . However, we also expect that it is difficult to formulate queries for the more complex requests r_3 and r_4 in MA_1 because simple class assertions cannot capture the semantics of findings that are related to qualifier values or body structures. Furthermore, the measurements should highlight that MA_3 allows the formulation of simpler queries as opposed to MA_2 because the TBox contains the additional role hierarchy that allows use to bypass role groups.

3.3 Results

Tables 1 – 3 illustrate the findings for the three proposed modelling approaches MA_i w.r.t. the information requests r_1 to r_4 . For each of the information requests,

the best queries as well as the fitness values for length, number of distinct constructors and role nesting depth as well as the flexibility (ℓ) are listed for the three modelling approaches.

r_j	bestQueries	Length	Consts	Nesting Depth	Flex
r_1	Image \sqcap \exists shows.Neoplasm	3	2	1	1
r_2	Image \sqcap \exists shows.PlainChestXray \sqcap \exists shows.PAProjection	5	2	1	1
r_3	Image \sqcap \exists shows.PleuralEffusion \sqcap \exists shows.LeftSided	5	2	1	1
r_4	none	100	100	100	0

Table 1. Findings for MA_1 .

r_j	bestQueries	Length	Consts	Nesting Depth	Flex
r_1	Image \sqcap \exists shows.(Disease \sqcap \exists roleGroup.(\exists AssociatedMorphology.Neoplasm))	6	2	2	1
r_2	Image \sqcap \exists hasImageType.PlainChestXray \sqcap \exists hasImageProjection.PAProjection	5	2	1	1
r_3	Image \sqcap \exists shows.(PleuralEffusion \sqcap \exists hasQualifierValue.LeftSided)	5	2	2	1
r_4	Image \sqcap \exists shows.(Disease \sqcap \exists roleGroup.(\exists AssociatedMorphology.SoftTissueMass \sqcap \exists FindingSite.PleuralMembraneStructure))	8	2	3	1

Table 2. Findings for MA_2 .

r_j	bestQueries	Length	Consts	Nesting Depth	Flex
r_1	Image \sqcap \exists shows.Neoplasm	3	2	1	1
r_2	Image \sqcap \exists hasImageType.PlainChestXray \sqcap \exists hasImageProjection.PAProjection	5	2	1	1
r_3	Image \sqcap \exists shows.(PleuralEffusion \sqcap \exists hasQualifierValue.LeftSided)	5	2	2	1
r_4	Image \sqcap \exists shows.(\exists hasFinding.SoftTissueMass \sqcap \exists hasLocation.PleuralMembraneStructure) Image \sqcap \exists shows.(\exists AssociatedMorphology.SoftTissueMass \sqcap \exists FindingSite.PleuralMembraneStructure)	6	2	2	2

Table 3. Findings for MA_3 .

3.4 Evaluation

The results shows that MA_1 allows the formulation of relatively simple queries. However, it is not always possible to formulate a query that returns exactly those tuples that are the certain answers to the information request. As soon as the information request involves nesting of entities, e.g. a finding with a location or a finding with a qualifier, MA_1 does not allow the formulation of a query that is precise enough to return only the correct answers. In this case the fitness values were assigned an exemplary value $\max = 100$, see Table 1 for r_4 . In this information request we want to find images that show soft tissue masses located in the pleural membrane. In our data set there is one image annotation that describes a neoplasm in the pleural membrane and a soft tissue mass in some

other body structure. This image would have been returned with a query like $\text{Image} \sqcap \exists \text{shows} . \text{SoftTissueMass} \sqcap \exists \text{shows} . \text{PleuralMembraneStructure}$, although it is not an answer to the information request. The problem lies in the nature of the data modelling paradigm. The lack of relational structure in the ABox makes it impossible to capture the semantics of the image descriptions appropriately. MA_2 models the data in the ABox using the relational structures defined in the TBox, in particular the properties *shows*, *roleGroup*, *associatedMorphology*, etc. This allows us to formulate queries for all information requests. However, the queries are rather long and nested due to the fact that the complicated role group construct [5] has to be used. The modelling approach MA_3 can capture the semantics of the image descriptions as well as MA_2 and allows us to formulate queries for all information requests. Furthermore, the queries are significantly simpler than those in MA_2 because MA_3 uses a slightly more expressive TBox than MA_2 with which we can bypass role groups.

The three modelling approaches and their measurements expose the evolving design of the retrieval system built in the case study. We started off with a relatively simple ABox that involves little effort compared to the later versions but is not expressive enough to allow the formulation of queries for all information requests. Using a more expressive ABox with object property assertions to relate the class assertions to each other makes it possible to formulate queries for all information requests, however, the queries become significantly more complex. Furthermore, with a little more modelling effort of introducing a small role hierarchy in the TBox we can formulate queries that are as expressive but much simpler than those that came with the original TBox.

The evaluation framework has highlighted the weaknesses of each modelling approach, e.g. the inability or difficulty of formulating queries for information requests. It can also highlight the strengths, e.g. conciseness and flexibility of a modelling approach. The measurements can guide the system engineer and support design decisions. For example, the framework can identify the benefits of changes in the modelling approach and therefore point out whether more modelling effort would be justified.

4 Conclusion and Future Work

We have presented a generic information system evaluation framework that can be used to analyse the fitness and flexibility of modelling approaches. It involves evaluating representative information requests and the complexity of the queries that correspond to these requests as well as the well-suitedness of the components of the modelling approach, i.e. the schema, the data and the queries.

The measurements generated by the framework can be used to highlight strengths and weaknesses of a modelling approach and to compare the fitness

of similar modelling approaches. It also supports engineers in making important design decisions, such as using an off-the-shelf schema or creating one that is tailored to the data or, in general, investing more modelling effort if this leads to significant benefits in the fitness of the modelling approach. The measurements can be used as a basis for discussion when building data access applications.

A next step of our work will be to apply the framework in a case study where we compare more heterogeneous modelling approaches with each other, e.g. an ontology-based modelling approach with one based on databases. Furthermore, we want to extend the framework so that it measures the fitness of queries taking into account not only exact matches to the answers of the respective information requests but also partial results. Finally, we will extend this approach so that it not only evaluates the complexity of *formulating* queries, but also the overall performance and scalability of query answering.

References

1. A. Acciarri, D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati. Quonto: Querying ontologies. In *AAAI*, pages 1670–1671. AAAI Press / The MIT Press, 2005.
2. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable Description Logics for Ontologies. In *AAAI*, pages 602–607. AAAI Press / The MIT Press, 2005.
3. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
4. D. Calvanese, G. D. Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *PODS*, pages 149–158. ACM Press, 1998.
5. R. Cornet and S. Schulz. Relationship Groups in SNOMED CT. In *Medical Informatics in a United and Healthy Europe*, pages 223–227. IOS Press, 2009, 2009.
6. C. Lutz, D. Toman, and F. Wolter. Conjunctive Query Answering in \mathcal{EL} using a Database System. In *OWLED*, 2008.
7. B. Motik, P. F. Patel-Schneider, and B. Parsia. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. Technical report, W3C Recommendation, 2009.
8. J. Opitz, B. Parsia, and U. Sattler. Using Ontologies for Medical Image Retrieval - An Experiment. In *OWLED*, 2009.
9. H. Pérez-Urbina, B. Motik, and I. Horrocks. Rewriting Conjunctive Queries over Description Logic Knowledge Bases. In *SDKB*, pages 199–214, 2008.
10. I. Seylan, E. Franconi, and J. de Bruijn. Effective query rewriting with ontologies over dboxes. In *IJCAI*, pages 923–925, 2009.
11. M. Wessel and R. Möller. A high performance semantic web query answering engine. In *International Workshop on Description Logics*. CEUR, 2005.