RODI: A Benchmark for Automatic Mapping Generation in Relational-to-Ontology Data Integration

Christoph Pinkel[1], Carsten Binnig[2,3], Ernesto Jiménez-Ruiz[4], Wolfgang May[5], Dominique Ritze[6], Martin G. Skjæveland[7], Alessandro Solimando[8], and Evgeny Kharlamov[4]

[1] fluid Operations AG, Walldorf, Germany
[2] Brown University, USA
[3] Baden-Wuerttemberg Cooperative State University, Mannheim, Germany
[4] University of Oxford, United Kingdom
[5] Göttingen University, Germany
[6] University of Mannheim, Germany
[7] University of Oslo, Norway
[8] Università di Genova, Genoa, Italy

Abstract. A major challenge in information management today is the integration of huge amounts of data distributed across multiple data sources. A suggested approach to this problem is ontology-based data integration where legacy data systems are integrated via a common ontology that represents a unified global view over all data sources. However, data is often not natively born using these ontologies. Instead, much data resides in legacy relational databases. Therefore, mappings that relate the legacy relational data sources to the ontology need to be constructed. Recent techniques and systems that automatically construct such mappings have been developed. The quality metrics of these systems are, however, often only based on self-designed benchmarks. This paper introduces a new publicly available benchmarking suite called *RODI*, which is designed to cover a wide range of mapping challenges in *R*elational-to-*O*ntology *D*ata *I*ntegration scenarios. *RODI* provides a set of different relational data sources and ontologies (representing a wide range of mapping challenges) as well as a scoring function with which the performance of relational-to-ontology mapping construction systems may be evaluated.

1 Introduction

Data integration is a big challenge in industry, life sciences, and the web, where data has not only reached large volumes, but also comes in a variety of formats. Integration increases the utility of data, it provides a unified access point to several databases and allows to analyse them, e.g., by correlating their data and identifying important patterns [3,5].

One of the major challenges in the integration task is to address the heterogeneity of data. A promising recent approach to address this challenge is to use ontologies, semantically rich conceptual models [12], to provide a conceptual integration and access layer on top of databases [27]. The ontology is 'connected' to databases with the help

of *mappings* that are declarative specifications describing the relationship between the ontological vocabulary and the elements of the database schema.

Ontologies are already available in many domains, and many of them can naturally be employed to support integration scenarios. For example, in biology there is the Gene Ontology and in medicine [7] there is the International Classification of Diseases (ICD) ontology. Another recent example is *schema.org*, an ontology to mark up data on the web with schema information. Industrial examples include NPD FactPages ontology [30,17] created for petroleum domain and Siemens ontology [15] created for the energy sector.

Mappings, however, cannot easily be reused since they are typically specific for each source database. Thus, they usually need to be developed from scratch. Creating and curating relational-to-ontology mappings manually is a process that often involves an immense amount of human effort [25]. In order to address this challenge, a number of techniques and systems [18,24,10,32,22,13,28] have been recently developed to assist in the relational-to-ontology data integration problem, either in a semi-automatic fashion or by bootstrapping initial mappings. However, claims about the quality of the created mappings are only based on self-designed benchmarks, which make comparisons difficult. While there already exist some standardized benchmarks or testbeds for data integration scenarios in data warehousing [26] or for ontology alignment [21], these benchmarks do not include the mapping challenges that arise from relational-to-ontology mappings.

In this paper we present a systematic overview of different types of mapping challenges that arise in relational-to-ontology data integration scenarios. Based on these types of mapping challenges, we selected existing ontologies and created corresponding relational databases for our benchmark to have a good coverage of all types. Moreover, the benchmark queries have been designed such that each query targets different mapping challenges. That way, the results of the scoring function for the individual queries can be used to draw inferences on how good different types of structural heterogeneity are supported by a certain integration system.

As the main contribution this paper introduces a new publicly available benchmarking suite[9] called *RODI* which is designed for *R*elational-to-*O*ntology *D*ata *I*ntegration Scenarios. *RODI* provides researchers with a set of different relational data sources (schema and data) and ontologies (only schema) that model data of research conferences (e.g., sessions, talks, authors, etc.). The challenge of the benchmark is to map the schema elements of the relational database to the schema elements of the ontology in order to instantiate the ontology. In addition, the benchmark provides a set of query pairs (i.e., a query over parts of the database and an equivalent query over the ontology). The idea is that each of the query pairs targets schema elements that represent different types of mapping challenges. Moreover, the benchmark also provides a scoring function to evaluate the quality of the mappings created by a certain tool. For covering other forms of heterogeneity, our benchmark provides extension points that allow users to integrate other relational databases, ontologies and test queries.

Thus, *RODI* is an end-to-end integration benchmark to test different mapping challenges. We decided to design an end-to-end integration benchmark instead of evaluating individual artifacts of the data integration process (i.e., correspondences, mappings, ...) since existing systems implement a wide range of different integration approaches that

[9] Download at: `http://www.fluidops.com/downloads/collateral/rodi1.0-2.zip`

do not allow a good way of comparison. For example, a major difference is that some integration systems directly map relational databases to ontologies (e.g., IncMap [24]) while other tools first translate the relational database into an ontology and then apply an ontology alignment technique (e.g., BootOX [10]) resulting in different artifacts during the integration process.

The outline of our paper is the following. Section 2 provides a classification of the different types of mapping challenges. Section 3 gives an overview of our benchmark and describes the details about the ontologies and relational databases as well as the benchmarking queries and the evaluation procedure. Section 4 illustrates the initial use of our benchmark suite by evaluating four mapping generation systems. Finally, Section 5 summarizes related work and Section 6 concludes the paper.

2 Mapping Challenges

In the following we present our classification of different types of mapping challenges in relational-to-ontology mapping. As top level of the classification, we use the standard classification for data integration described by Batini et al. [2]: naming conflicts, structural heterogeneity, and semantic heterogeneity.

2.1 Naming Conflicts

Typically, relational database schemata and ontologies use different conventions to name their artifacts even when they model the same domain and thus should use a similar terminology. While database schemata tend to use short identifiers for tables and attributes that often include technical artifacts (e.g. for tagging primary keys and for foreign keys), ontologies typically use long "speaking" names. Moreover, names in ontologies include IRIs with prefixes (that refer to a namespace). Thus, the main challenge is to be able to find similar names despite the different naming patterns.

Other model differences include the use of plural vs. singular form for entities, common tokenization schemes, use of synonyms etc. that are not present in other data integration scenarios (e.g., relational-to-relational or ontology alignment).

2.2 Structural Heterogeneity

The most important differences in relational-to-ontology integration scenarios compared to other integration scenarios are structural heterogeneities. We discuss the different types of structural heterogeneity covered by *RODI*.

Type Conflicts: Relational schemata and ontologies represent the same artifacts by using different modeling constructs. While relational schemata use tables, attributes, as well as constraints, ontologies use modeling elements such as classes and subclasses (to model class hierarchies), data and object properties, restrictions, etc. Clearly there exist direct (i.e., naive) mappings from relational schemata to ontologies for some of the artifacts (e.g., classes map to tables). However, most real-world relational schemata and corresponding ontologies do not follow any naive mapping. Instead, the mapping

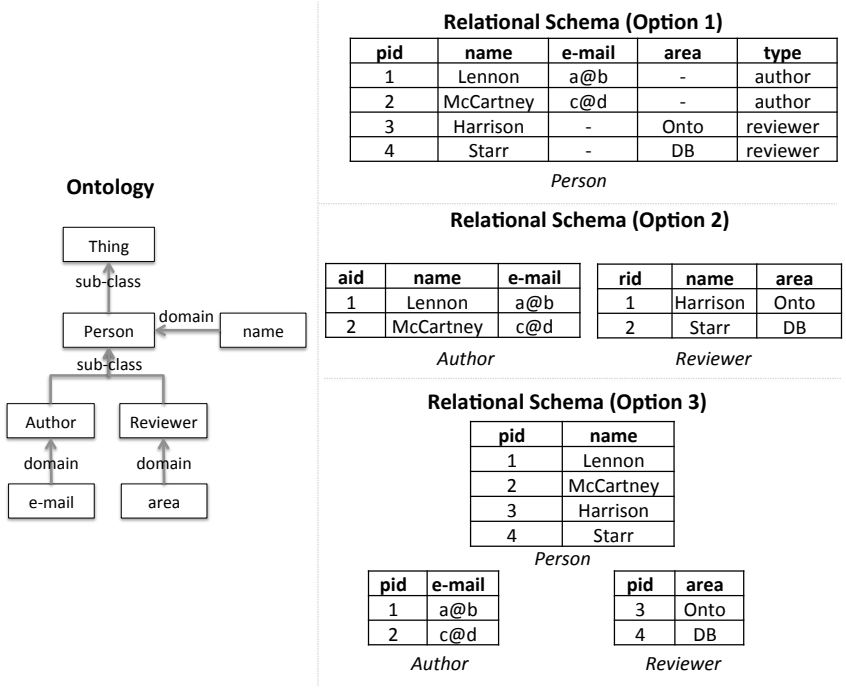

Fig. 1: Class hierarchies – ontology vs. relational schema

rules are much more complex and there exist big differences (i.e., type conflicts) in the way how the same concepts are modeled. One reason is that relational schemata are often optimized towards a given workload (e.g., they are normalized for update-intensive workloads or denormalized for read-intensive workloads) while ontologies model a domain on the conceptual level. Another reason is that some modeling elements have no direct translation (e.g., class hierarchies in ontologies can be mapped to relational schemata in different ways). In the following, we list the different type conflicts covered by *RODI*:

1. *Normalization artifacts:* Often properties that belong to a class in an ontology are spread over different tables in the relational schema as a consequence of normalization.
2. *Denormalization artifacts:* For read-intensive workloads, tables are often denormalized. Thus, properties of different classes in the ontology might map to attributes in the same table.
3. *Class hierarchies:* Ontologies typically make use of explicit class hierarchies. Relational models implement class hierarchies implicitly, typically using one of three different common modeling patterns (c.f., [8, Chap. 3]). In the following we describe those patterns (see Figure 1): (1) In one common variant the relational schema materializes several subclasses in the same table and uses additional attributes to indicate

the subclass of each individual. Those additional attributes can take the shape of a numeric type column for disjoint subclasses and/or a combination of several type or role flags for non-disjoint subclasses. In this case, several classes need to be mapped to the same table and can be told apart only by secondary features in the data, such as the value in a type column. (2) Another common way is to use one table per most specific class in the class hierarchy and to materialize the inherited attributes in each table separately. Thus, the same property of the ontology must be mapped to several tables. (3) A third variant uses one table for each class in the hierarchy, including for possibly abstract superclasses. Tables then use the same primary key to indicate the subclass relationship. This variant has a closer resemblance to ontology design patterns. However, it is also rarely used in practice, as it is more difficult to design, harder to query, impractical to update and usually considered unnecessary.

Thus, the main challenge is that integration tools should be capable to resolve different levels of (de-)normalization and different patterns implementing class hierarchies in a relational schema when mapping a schema to an ontology.

Key Conflicts: In ontologies and relational schemata, keys and references (to keys) are represented in different ways. In the following, we list the different key conflicts covered by *RODI*:

1. *Keys:* Keys in databases are often (but not always) explicitly implemented using constraints (i.e., primary keys and unique constraints). Keys may be composite and in some cases partial keys of a table identify different related entities (e.g., denormalized tables on the relational side). Moreover, ontologies use IRIs as identifiers for individuals. Thus, the challenge is that integration tools should be able to generate mapping rules for creating IRIs for individuals from the correct choice of keys.
2. *References:* A similar observation holds for references. While references are typically modeled as foreign keys in relational schemata, ontologies use object properties. Moreover, sometimes relational databases do not model foreign key constraints at all. In that case an integration tool must be able to derive references from relational schema (e.g., based on the naming scheme or individuals).

Dependency Conflicts: These conflicts arise when a group of concepts are related among themselves with different dependencies (i.e., $1:1$, $1:n$, $n:m$) in the relational schema and the ontology. While relational schemata use foreign keys over attributes as constraints to model 1-1 and 1-N relationships explicitly, they can only model N-M relationships in an implicit way using an additional connection table. Ontologies, on the other hand, model functionalities (i.e., functional properties or inverse functional properties) or they define cardinalities explicitly using min- and max-cardinality restrictions. However, many ontologies do not make use of these constraints and thus are often underspecified.

2.3 Semantic Heterogeneity

Besides the usual semantic differences between any two conceptual models of the same domain, two additional factors apply in relational-to-ontology data integration: (1) the

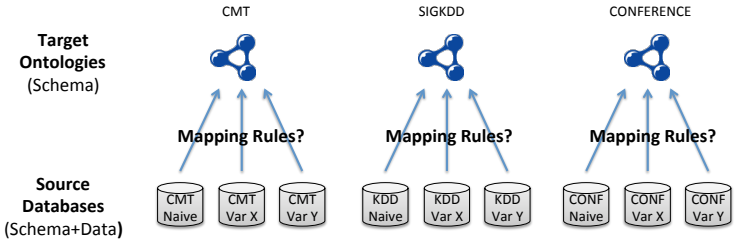

Fig. 2: Overview of the *RODI* benchmark

impedance mismatch between the closed-world assumption (CWA) in databases and the open-world assumption (OWA) in ontologies;[10] and (2) the difference in semantic expressiveness, i.e., databases may model some concepts or data explicitly where they are derived logically in ontologies. The challenge is thus to bridge the model gap. In general, this challenge is inherent to all relational-to-ontology mapping problems.

3 *RODI* Benchmark Suite

In the following, we present the details of our *RODI* benchmark: we first give an overview, then we discuss the details of the data sets (relational schema and ontologies) as well as the queries, and finally we present our scoring function to evaluate the benchmark results.

3.1 Overview

Figure 2 gives an overview of our benchmark. In its basic version, the benchmark provides three target ontologies (T-Box only) and different relational source databases for each ontology (schema and data) varying in the types of mapping challenges that are covered.

As the primary domain for testing, we chose the conference domain: it is well understood, comprehensible even for non-domain experts but still complex enough for realistic testing and it has been successfully used as the domain of choice in other benchmarks before (e.g., by the OAEI [21]). For each ontology, we provide different variants of corresponding databases, each focusing on different types of mapping challenges.

The benchmark asks systems to create mapping rules from the different source databases to their corresponding target ontologies. We call each such combination of a database and an ontology a *benchmark scenario*. For evaluation, we provide query pairs for each scenario to test a range of mapping challenges. Query pairs are evaluated against the instantiated ontology and the provided databases, respectively. Results are compared for each query pair and aggregated in the light of different mapping challenges using our scoring function.

[10] Other notions of impedance mismatch exist (e.g., modeling of values vs. objects). The OWA/CWA notion is most relevant w.r.t. specific mapping challenges.

In order to be open for other data sets and different domains, our benchmark can be easily extended to include scenarios with real-world ontologies and databases. In our initial version, we already provide one such extension from a real-world application of the oil and gas domain.

3.2 Data Sources

In the following, we discuss the data sources (i.e., ontologies and relational schemata) as well as the combinations used as mapping scenarios for the benchmark in more details.

Conference Ontologies. The conference ontologies in this benchmark are provided by the Ontology Alignment Evaluation Initiative (OAEI) [21] and were originally developed by the OntoFarm project.[11] We selected three particular ontologies (CMT, SIGKDD, CONFERENCE), based on a number of criteria: variation in size, the presence of functional coherences, the coverage of the domain, variations in modeling style, and the expressive power of the ontology language used. In SIGKDD, we have fixed a total of seven inconsistencies that we discovered in this ontology.

Relational Schemata. We synthetically derived different relational schemata for each of the ontologies, focusing on different mapping challenges. First, for each ontology we derived a relational schema that can be mapped to the ontology using a naive mapping as described in [11]. The algorithm works by extracting an entity-relationship (ER) model from an OWL DL ontology. It then translates this ER model into a relational schema according to text book rules (e.g., [8]). We extended this algorithm to consider ontology instance data to derive more proper functionalities (rather than just looking at the T-Box as the existing algorithm did). Otherwise, the generated naive relational schemata would have contained an unrealistically high number of $n : m$-relationship tables. The naively translated schemata of the algorithm are guaranteed to be in fourth normal form (4NF), fulfilling normalization requirements of standard design practices. Thus, the naive schemata already include various normalization artifacts as mapping challenges. Also, all scenarios reflect the kind of semantic heterogeneity that is inherent to relational-to-ontology mappings.

From each naively translated schema, we systematically created different variants by introducing different aspects on how a real-world schema may differ from a naive translation and thus to test different mapping challenges:

1. *Adjusted Naming:* As described in Section 2.1, ontology designers typically consider other naming schemes than database architects do, even when implementing the same (verbal) specification. Those differences include longer vs. shorter names, “speaking” prefixes, human-readable property IRIs vs. technical abbreviations (e.g., “hasRole” vs. ”RID”), camel case vs. underscore tokenization, preferred use of singular vs. plural, and others. For each naively translated schema we automatically generate a variant with identifier names changed accordingly.

[11] `http://nb.vse.cz/~svatek/ontofarm.html`

	CMT	CONFERENCE	SIGKDD
Naive	(✓)	(✓)	(✓)
Adjusted Naming	✓	✓	✓
Cleaned Hierarchies	✓	✓	✓
Combined Case	(✓)	(✓)	✓
Missing FKs	-	✓	-
Denormalized	✓	-	-

Table 1: Scenario combinations

2. *Varying Hierarchies:* The most critical structural challenge comes with different relational design patterns to model class hierarchies more or less implicitly, as we have discussed in Section 2.2. We automatically derive variants of all naively translated schemata where different hierarchy design patterns are presented.
3. *Combined Case:* In the real world, both of the previous cases (i.e., adjusted naming and hierarchies) would usually apply at the same time. To find out how tools cope with such a situation, we also built scenarios where both are combined.
4. *Removing Foreign Keys:* Although it is considered as bad style, databases without foreign keys are not uncommon in real-world applications. The mapping challenge is that mapping tools must guess the join paths to connect tables of different entities. Therefore, we have created one dedicated scenario to test this challenge with the CONFERENCE ontology and based it on the schema variant with cleaned hierarchies.
5. *Partial Denormalization:* In many cases, schemata get partially denormalized to optimize for a certain read-mostly workload. Denormalization essentially means that correlated (yet separated) information is jointly stored in the same table and partially redundant. We provide one such scenario for the CMT ontology.

Mapping Scenarios. For each of our three main ontologies, CMT, CONFERENCE, and SIGKDD, the benchmark includes five scenarios, each with a different variant of the database schema (discussed before). Table 1 lists the different versions. All scenarios cover the main semantic challenges and to some degree also the structural challenges. Renamed scenarios cover the naming conflicts challenge. Scenarios with cleaned hierarchies and advanced cases mostly address structural heterogeneity but also stress the challenge of semantic differences more than other scenarios. To keep the number of scenarios small for the default setup, we differentiate between default scenarios and non-default scenarios. While the default scenarios are mandatory to cover all mapping challenges, the non-default scenarios are optional (i.e., users could decide to run them in order to gain additional insights). Non-default scenarios are put in parentheses in Table 1. Similarly, we include scenarios that require mappings of schemata to one of the other ontologies (e.g., mapping a CMT database schema variant to the SIGKDD ontology), but do not consider them as default scenarios either. They represent more advanced scenarios.

Data. In *RODI*, we provide data to fill both the databases and ontologies, as all ontologies are provided as empty T-Boxes, only. All data are first generated as A-Box facts for the different ontologies, and then translated into the corresponding relational data. Actually, for the evaluation it would not be necessary to generate data for the ontologies. However, this design simplifies the evaluation since all databases can be automatically derived from the given ontologies as described before. Our conference data generator deterministically produces a scalable amount of synthetic facts around key concepts in the ontologies, such as conferences, papers, authors, reviewers, and others. In total, we generate data for 23 classes, 66 object properties (including inverse properties) and 11 datatype properties (some of which apply to several classes).

3.3 Queries

We test each mapping scenario with a series of *query pairs*, consisting of semantically equivalent queries against the instantiated ontology and the provided databases, respectively.

Each query pair is based on one SPARQL query, which we then translated into equivalent SQL for each corresponding schema using the same translation mechanism as used for schema translation. To double-check that queries in each pair are in fact equivalent, we manually checked result sets on both ends. Queries are manually curated and designed to test different mapping challenges.

To this end, all query pairs are tagged with categories, relating them to different mapping challenges. All scenarios draw on the same pool of 56 query pairs, accordingly translated for each ontology and schema. However, the same query may face different challenges in different scenarios, e.g., a simple $1:1$ mapping between a class and table in a naive scenario can turn into a complicated $n:1$ mapping problem in a scenario with cleaned hierarchies. Also, not all query pairs are applicable on all ontologies (and thus, on their derived schemata).

3.4 Evaluation Criteria

It is our aim to measure the practical usefulness of mappings. We are therefore interested in the correctness (precision) and completeness (recall) of query results, rather than comparing mappings directly to a reference mapping set. This is important because a number of different mappings might effectively produce the same data w.r.t. a specific input database. Also, the mere number of facts is no indicator of their semantic importance for answering queries (e.g., the overall number of conferences is much smaller than the number of paper submission dates, yet are at least as important in a query about the same papers).

We therefore define precision and recall locally for each individual test (i.e., for each query pair) and use a simple scoring function to calculate averages for different subsets of tests, i.e., for tests relating to a specific mapping challenge.

Unfortunately, precision and recall cannot be measured immediately by naively checking results of query pairs tuple by tuple for equality, as different mappings typically generate different IRIs to denote the same entities. Instead, we define an equivalence measure that is agnostic of entity IRIs.

John
Jane

(a) Reference

Jane
John

(b) Result 1

John

(c) Result 2

Jane
John
James

(d) Result 3

Table 2: Example results from a query pair asking for author names (simplified)

In the following, we define tuple set equivalence based on a more general equivalence of query results (i.e., tuple sets):

Definition 1 (Structural Tuple Set Equivalence). *Let $V = IRI \cup Lit \cup Blank$ be the set of all IRIs, literals and blank nodes, $T = V \times ... \times V$ the set of all n-tuples of V. Then two tuple sets $t_1, t_2 \in \mathcal{P}(T)$ are* structurally equivalent *if there is an isomorphism $\phi : (IRI \cap t_1) \rightarrow (IRI \cap t_2)$.*

For instance, $\{$(urn:p-1, 'John Doe')$\}$ and $\{$(http://my#john, 'John Doe')$\}$ are structurally equivalent. On this basis, we can easily define the equivalence of query results w.r.t. a mapping target ontology:

Definition 2 (Tuple Set Equivalence w.r.t. Ontology ($\sim_O$)). *Let O be a target ontology of a mapping, $I \subset IRI$ the set of IRIs used in O and $t_1, t_2 \in \mathcal{P}(T)$ result sets of queries q_1 and q_2 evaluated on a superset of O (i.e., over O plus A-Box facts added by a mapping).*

Then, $t_1 \sim_O t_2$ (are structurally equivalent w.r.t. O) iff t_1 and t_2 are structurally equivalent and $\forall i \in I : \phi(i) = i$

For instance, $\{$(urn:p-1, 'John Doe')$\}$ and $\{$(http://my#john, 'John Doe')$\}$ are structurally equivalent, *iff* http://my#john is *not* already defined in the target ontology. Finally, we can define precision and recall:

Definition 3 (Precision and Recall under Tuple Set Equivalence). *Let $t_r \in \mathcal{P}(T)$ be a reference tuple set, $t_t \in \mathcal{P}(T)$ a test tuple set and $t_{rsub}, t_{tsub} \in \mathcal{P}(T)$ be maximal subsets of t_r and t_t, s.t., $t_{rsub} \sim_O t_{tsub}$.*

Then the precision of the test set t_t is $P = \frac{|t_{tsub}|}{|t_t|}$ and recall is $R = \frac{|t_{rsub}|}{|t_r|}$.

We observe precision and recall locally on each query test, i.e., based on how many of the result tuples of each query are structurally equivalent to a reference query result set. Table 2 shows an example with a query test that asks for the names of all authors. The corresponding query pair here would be:

```
SQL:     SELECT name FROM persons WHERE person_type = 2.
SPARQL:  SELECT ?name WHERE {?p a :Author; foaf:name ?name}).
```

Result set 1 is structurally equivalent to the reference result set, i.e., it has found all authors and did not return anything else, so both precision and recall are 1.0. Result set 2 is equivalent with only a subset of the reference result (e.g., it did not include those authors who are also reviewers). Here, precision is still 1.0, but recall is only 0.5. In case of result set 3, all expected authors are included, but also another person, James. Here, precision is only 0.66, but recall is 1.0.

To aggregate results of individual query pairs, a scoring function calculates the averages of per query numbers for each scenario and for each challenge category. For instance, we calculate averages of all queries testing $1 : n$ mappings.

3.5 Extension Scenarios

Our benchmark suite is designed to be extensible, i.e., additional scenarios can be easily added. The primary aim of supporting such extensions is to allow actual real-world mapping challenges to be tested on a realistic query workload alongside our more controlled default scenarios.

To demonstrate the feasibility of extension scenarios we added and evaluated one example of an extension scenario in our benchmark suite, based on the data, ontology and queries from *The Norwegian Petroleum Directorate (NPD) FactPages* [30]. The test set contains a small relational database (≈40 MB), but with a relatively complex structure (70 tables, ≈1000 columns and ≈100 foreign keys), an ontology covering the domain of the database (with ≈300 classes and ≈350 properties), and 17 query pairs. The database and ontology are constructed from a publicly available dataset containing reference data about past and ongoing activities in the Norwegian petroleum industry, and the queries in the test set are built from real information needs collected from end-users of the NPD FactPages.

4 Benchmark Results

Setup: In order to show the usability of our benchmark and the usefulness and significance of its results, we have performed an initial evaluation with four systems: *BootOX* [9,16], *IncMap* [24,23], *morph/MIRROR*[12] and *ontop* [29].

(1) *BootOX* (*Boot*strapper of *Ox*ford) is based on the approach called 'direct mapping' by the W3C:[13], i.e., every table in the database (except for those representing $n : m$ relationships) is mapped to one class in the ontology; every data attribute is mapped to one data property; and every foreign key to one object property. Explicit and implicit database constraints from the schema are also used to enrich the bootstrapped ontology with axioms about the classes and properties from these direct mappings. Afterwards, *BootOX* performs an alignment with the target ontology using the LogMap system [31]. (2) *IncMap* maps an available ontology directly to the relational schema. *IncMap* represents both the ontology and schema uniformly, using a structure-preserving meta-graph for both. (3) *morph/MIRROR* (*M*app*I*ngs for *R*db to *R*df generat*OR*) is a tool for generating an ontology and R2RML direct mappings automatically from an RDB schema. *morph/MIRROR* has been implemented as a module of the RDB2RDF engine *morph-RDB* [28]. (4) *ontop* is an ontology-based data access system that also includes a module to automatically compute direct mappings and a simple ontology with the vocabulary used in the mappings. For the last step of aligning to the target ontology we have coupled both *morph/MIRROR* and *ontop* with LogMap in a similar setup to the one used in *BootOX*.

Results: For each of those systems we were running the default scenarios of our benchmark (as discussed in Section 3). We mainly report overall aggregates but also highlight some of the most interesting findings in more detail.

Table 3 shows precision, recall and f-measure averaged over all tests for each scenario. What becomes immediately apparent is that measured quality is relatively modest. Another surprising observation is that for each system, precision, recall and f-measure are always the same per scenario. A manual analysis of results has shown, that the reason for this behavior is linked to the relatively low overall quality: systems did only solve some of the simpler query tests and those tend to result in atomic answers, which may be either correct or incorrect, but nothing in-between. For instance, if a query asks for the number of author instances, the result in either correct ($p = r = f = 1.0$) or incorrect ($p = r = f = 0.0$). Systems did surprisingly well on some tests of medium difficulty, e.g., author names (where, e.g., some other persons could be

[12] `https://github.com/oeg-upm/MIRROR`

[13] `http://www.w3.org/TR/rdb-direct-mapping/`

Scenario	BootOX			IncMap			MIRROR			ontop		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Adjusted naming												
CMT	0.33	0.33	0.33	**0.5**	**0.5**	**0.5**	0.28	0.28	0.28	0.39	0.39	0.39
CONFERENCE	0.33	0.33	0.33	0.26	0.26	0.26	0.27	0.27	0.27	**0.37**	**0.37**	**0.37**
SIGKDD	**0.45**	**0.45**	**0.45**	0.21	0.21	0.21	0.3	0.3	0.3	**0.45**	**0.45**	**0.45**
Cleaned hierarchies												
CMT	0.28	0.28	0.28	**0.44**	**0.44**	**0.44**	0.17	0.17	0.17	0.28	0.28	0.28
CONFERENCE	0.23	0.23	0.23	0.16	0.16	0.16	0.23	0.23	0.23	**0.3**	**0.3**	**0.3**
SIGKDD	**0.16**	**0.16**	**0.16**	0.11	0.11	0.11	0.11	0.11	0.11	**0.16**	**0.16**	**0.16**
Combined case												
SIGKDD	**0.16**	**0.16**	**0.16**	0.05	0.05	0.05	0.11	0.11	0.11	**0.16**	**0.16**	**0.16**
Missing FKs												
CONFERENCE	**0.17**	**0.17**	**0.17**	0.03	0.03	0.03	**0.17**	**0.17**	**0.17**	-	-	-
Denormalized												
CMT	**0.28**	**0.28**	**0.28**	0.22	0.22	0.22	0.22	0.22	0.22	**0.28**	**0.28**	**0.28**

Table 3: Average results of all tests per scenarios. **P**recision, **R**ecall and **F**-measure are all equal as systems fail the more complex tasks while simpler ones are atomic.

mistaken for authors) and scored $p = r = f = 1.0$ in all cases where they submitted any results at all. For the most complex queries, where results could be likely in $]0; 1[$, systems failed the query tests completely. We expect this behavior to change as systems improve in general and overall scores go up.

Best numbers are generally reached for "adjusted naming" scenarios, which are close to the naive ontology translation and thus schemata resemble their corresponding ontologies most closely. Besides the generic model gap and those, these scenarios only test the challenges of naming conflicts and normalization artifacts. Quality drops rapidly for almost all other types of scenarios, i.e., whenever we introduce additional challenges that are specific to the relational-to-ontology modeling gap. With a few exceptions, *BootOX* and *ontop* perform better than the others. Where differences appear between the two of them, *ontop* surprisingly outperforms *BootOX*. Note that those two similar setups differ mostly in that *ontop* only produces a very simple ontology while *BootOX* tries to additionally include some knowledge encoded in the database structure. Results hint that this additional knowledge may be noisy. For CMT, *IncMap* outperforms other systems both adjusted names and cleaned hierarchies. This is interesting, as *IncMap* has been designed to work on typical databases and CMT differs from the other ontologies insofar as it contains relatively flat class hierarchies and results in a somewhat more realistic relational database even when translated naively. The generally low numbers of *morph/MIRROR* come as a surprise. We had expected it to perform similarly to or somewhat better than *BootOX* as it follows the same idea of leveraging knowledge from the database schema to build a better ontology, but does so more systematically. The effect of noise seems to be insufficient as an explanation in this case. As *morph/MIRROR* is still under development, we assume that some of the effects may be related to technical issues that we could not isolate and identify as such.

The drop in accuracy between "adjusted names" and "cleaned hierarchies" is mostly due to the $n : 1$ mapping challenge, introduced by one of the relational patterns to represent class hierarchies which groups data for several subclasses in a single table. Neither of the systems managed to solve even a single test on this challenge.

Adjusted naming	BootOX			IncMap			MIRROR			ontop		
	C	D	O	C	D	O	C	D	O	C	D	O
CMT	0.67	0.0	0.0	0.67	**0.50**	0.0	0.56	0.0	0.0	**0.78**	0.0	0.0
CONFERENCE	0.67	0.0	0.0	0.42	**0.24**	0.0	0.53	0.0	0.0	**0.73**	0.0	0.0
SIGKDD	**0.69**	0.0	0.0	0.34	0.0	0.0	0.46	0.0	0.0	**0.69**	0.0	0.0

Table 4: Average F-measure results for the adjusted naming scenarios. 'C' stands for queries about classes, 'D' stands for queries involving data properties and 'O' stands for queries involving object properties

In the most advanced cases, all systems lose on the additional challenges, although to different degrees. For instance, all systems failed to solve any of the tests specifically targeted to the challenge of denormalization artifacts. (For *BootOX* and *ontop*, there is no difference to the "cleaned hierarchies" scenario as the systems failed the relevant queries already on that simpler scenario.) While *BootOX* stands up relatively well in those most advanced scenarios, *IncMap* records significant further drops. *ontop* failed to produce mappings for the advanced scenario involving missing foreign keys.

All systems struggle with identifying properties, as Table 4 shows. A close look shows that this is in part due to the challenge of normalization artifacts, with no system succeeding in detecting any properties that map to multi-hop join paths in the tables. Here, *IncMap* shows its stronger suit, mapping datatype properties with an average f-measure of up to 0.5. It has however to be noted that we test properties only in the context of their domains and ranges, i.e., to succeed in a property test, a correct mapping at least for its domain class is a precondition, making those tests generally harder.

On NPD FactPages, our extension scenario with real-world data and queries, all four tested systems fail to answer any of the 17 query tests correctly. Given the previous results from the default scenarios, this was to be expected. The query tests in NPD FactPages consist of real-world queries, only. Just as systems failed the most complex queries in the (generally still simpler) default scenarios, they also failed all queries in the extension scenario.

5 Related Work

Mappings between ontologies are usually evaluated only on the basis of their underlying correspondences (usually referred to as ontology *alignments*). The Ontology Alignment Evaluation Initiative [21] provides tests and benchmarks of those alignments that can be considered as de-facto standard. Mappings between relational databases are typically not evaluated by a common benchmark. Instead, authors compare their tools to an industry standard system (e.g., [6,1]) in a scenario of their choice. A novel TPC benchmark [26] was created only recently.

Similarly, evaluations of relational-to-ontology mapping generating systems were based on one or several data sets deemed appropriate by the authors and are therefore not comparable. In one of the most comprehensive evaluations so far, QODI [32] was evaluated on several real-world data sets, though some of the reference mappings were rather simple. IncMap [24] was evaluated on real-world mapping problems based on data from two different domains. Such domain-specific mapping problems could be easily integrated in our benchmark through our extension mechanism.

A number of papers discuss different quality aspects of such mappings in general. Console and Lenzerini have devised a series of theoretical quality checks w.r.t. consistency [4]. In another benchmark, Impraliou et al. generate synthetic queries to measure the correctness and completeness of OBDA query rewriting [14]. The presence of complete and correct mappings is a prerequisite to their approach. Mora and Corcho discuss issues and possible solutions to benchmark the query

rewriting step in OBDA systems [20]. Mappings are supposed to be given as immutable input. The NPD benchmark [19] measures performance of OBDA query evaluation. Neither of these papers, however, address the issue of systematically measuring mapping quality.

6 Conclusion

We have presented *RODI*, a benchmark suite for testing the quality of generated relational-to-ontology mappings. *RODI* tests a wide range of relational-to-ontology mapping challenges, which we discussed of the paper.

Initial results on four systems demonstrate that existing tools can cope with simpler mapping challenges to varying degrees. However, all tested tools fail on more advanced challenges and are still a long way from solving actual real-world problems. In particular, results show that mapping accuracy degrades massively when relational schemata use design patterns that differ greatly from the corresponding ontologies (e.g., in scenarios with "cleaned hierarchies"). We also gave detailed feedback about specific shortcomings to the authors of several of the tested systems, which has already lead to adjustments in one case and will lead to improvements in others.

As the main avenue of future work, we plan to conduct a both broader and deeper evaluation, also involving a greater number of systems. Another interesting aspect would be the addition of further extension scenarios to cover data from a number of application domains out of the box.

Acknowledgements This research is funded by the Seventh Framework Program (FP7) of the European Commission under Grant Agreement 318338, "Optique". Ernesto Jiménez-Ruiz and Evgeny Kharlamov were also supported by the EPSRC projects MaSI^3, Score! and DBOnto.

References

1. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and Ontology Matching with COMA++. In: SIGMOD (2005)
2. Batini, C., Lenzerini, M., Navathe, S.B.: A Comparative Analysis of Methodologies for Database Schema Integration. ACM Comput. Surv. 18(4), 323–364 (1986)
3. Bhardwaj, A.P. et al.: DataHub: Collaborative Data Science & Dataset Version Management at Scale. In: CIDR (2015)
4. Console, M., Lenzerini, M.: Data Quality in Ontology-Based Data Access: The Case of Consistency. In: AAAI (2014)
5. Dong, X.L., Srivastava, D.: Big Data Integration. PVLDB 6(11), 1188–1189 (2013)
6. Fagin R. et al.: Clio: Schema Mapping Creation and Data Exchange. In: Conceptual Modeling: Foundations and Applications (2009)
7. Freitas, F., Schulz, S.: Survey of current terminologies and ontologies in biology and medicine. RECIIS Elect. J. Commun. Inf. Innov. Health 3, 7–18 (2009)
8. Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems – The Complete Book. Prentice Hall, 2nd edn. (2008)
9. Giese, M. et al.: Optique Zooming In on Big Data Access. IEEE Computer (in press) (2015)
10. Haase, P., Horrocks, I., Hovland, D., Hubauer, T., Jiménez-Ruiz, E., Kharlamov, E., Klüwer, J.W., Pinkel, C., Rosati, R., Santarelli, V., Soylu, A., Zheleznyakov, D.: Optique system: towards ontology and mapping management in OBDA solutions. In: WoDOOM (2013)
11. Hornung, T., May, W.: Experiences from a TBox Reasoning Application: Deriving a Relational Model by OWL Schema Analysis. In: OWLED Workshop (2013)

12. Horrocks, I.: What Are Ontologies Good For? In: Evolution of Semantic Systems, pp. 175–188. Springer (2013)
13. Hu, W., Qu, Y.: Discovering Simple Mappings Between Relational Database Schemas and Ontologies. In: ISWC/ASWC (2007)
14. Impraliou, M., Stoilos, G., Cuenca Grau, B.: Benchmarking Ontology-based Query Rewriting Systems. In: AAAI (2013)
15. Kharlamov, E., Solomakhina, N., Özçep, Ö.L., Zheleznyakov, D., Hubauer, T., Lamparter, S., Roshchin, M., Soylu, A., Watson, S.: How Semantic Technologies Can Enhance Data Access at Siemens Energy. In: ISWC (2014)
16. Kharlamov, E. et al.: Optique 1.0: Semantic Access to Big Data – The Case of Norwegian Petroleum Directorates FactPages. In: ISWC (Posters & Demos) (2013)
17. Kharlamov, E. et al.: Optique: Towards OBDA systems for industry. In: The Semantic Web: ESWC 2013 Satellite Events - ESWC 2013 Satellite Events, Montpellier, France, May 26-30, 2013, Revised Selected Papers. pp. 125–140 (2013)
18. Knoblock, C.A. et al.: Semi-Automatically Mapping Structured Sources into the Semantic Web. In: ESWC (2012)
19. Lanti, D., Rezk, M., Slusnys, M., Xiao, G., Calvanese, D.: The NPD Benchmark for OBDA Systems. In: SSWS (2014)
20. Mora, J., Corcho, O.: Towards a Systematic Benchmarking of Ontology-Based Query Rewriting Systems. In: ISWC (2014)
21. Ontology Alignment Evaluation Initiative: `http://oaei.ontologymatching.org`
22. Papapanagiotou, P. et al.: Ronto: Relational to Ontology Schema Matching. AIS SIGSEMIS BULLETIN (2006)
23. Pinkel, C., Binnig, C., Kharlamov, E., Haase, P.: Pay as you go Matching of Relational Schemata to OWL Ontologies with IncMap. In: ISWC (Posters & Demos) (2013)
24. Pinkel, C., Binnig, C., Kharlamov, E., Haase, P.: IncMap: Pay-as-you-go Matching of Relational Schemata to OWL Ontologies. In: OM (2013)
25. Pinkel, C. et al.: How to Best Find a Partner? An Evaluation of Editing Approaches to Construct R2RML Mappings. In: ESWC (2014)
26. Poess, M., Rabl, T., Caufield, B.: TPC-DI: the first industry benchmark for data integration. PVLDB 7(13), 1367–1378 (2014)
27. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. Data Semantics 10, 133–173 (2008)
28. Priyatna, F., Corcho, O., Sequeda, J.: Formalisation and Experiences of R2RML-based SPARQL to SQL Query Translation Using Morph. In: WWW (2014)
29. Rodriguez-Muro, M. et al.: Efficient SPARQL-to-SQL with R2RML mappings. Journal of Web Semantics (in press) (2015)
30. Skjæveland, M.G., Lian, E.H., Horrocks, I.: Publishing the Norwegian Petroleum Directorate's FactPages as Semantic Web Data. In: ISWC. pp. 162–177 (2013)
31. Solimando, A., Jiménez-Ruiz, E., Guerrini, G.: Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In: ISWC (2014)
32. Tian, A., Sequeda, J.F., Miranker, D.P.: QODI: Query as Context in Automatic Data Integration. In: ISWC (2013)