Table 1: Query evaluation with respect to 5 setups (number of answers / running time in seconds)

		ONTOP	LSA	GSA	ONTOPROX	CLIPPER
UOBM	Q_1^u	14,129 / 0.08	14,197 / 0.11	14,197 / 0.43	14,197 / 0.42	14,197 / 21.4
	Q_2^u	1,105 / 0.09	2,170 / 0.15	2,170 / 0.42	2,170 / 0.44	2,170 / 21.3
	Q_3^u	235 / 0.20	235 / 0.24	235 / 0.88	247 / 0.83	247 / 19.6
	Q_4^u	19 / 0.13	19 / 0.15	19 / 0.43	38 / 0.52	38 / 21.4
Telecom	Q_1^t	0 / 2.91	0 / 0.72	0 / 1.91	82,455 / 5.21	N/A
	Q_2^t	0 / 0.72	0 / 0.21	0 / 0.67	16,487 / 198	N/A
	Q_3^t	5,201,363 / 128	5,201,363 / 105	5,201,363 / 538	5,260,346 / 437	N/A

Table 2: ONTOPROX pre-computation time and output size

	UOBM	Telecom
Time (s)	8.47	8.72
Number of mapping assertions	441	907
Number of TBox axioms	294	620
Number of new concepts	26	60
Number of new roles	30	7

- Query Q_1^t asks, for each cable in the telecommunications network, the single segments of which the cable is composed, and the network line (between two devices) that the cable covers. For each cable, it also returns its bandwidth and its status (functioning, non-functioning, etc.).
- Query Q_2^t asks for each path in the network that runs on fiber-optic cable, to return the specific device from which the path originates, and also requires to provide the number of different channels available in the path.
- Query Q_3^t asks, for each cable in the telecommunications network, the port to which the cable is attached, the slot on the device in which the port is installed, and, for each such slot, its status and its type. For each cable, it also returns its status.

For each OBDA instance $\langle\langle\mathcal{T},\mathcal{M},\mathcal{S}\rangle,\mathcal{D}\rangle$, we have evaluated the number of query answers and the query answering time with respect to five different setups:

(1) The default behavior of ONTOP v1.15, which simply ignores all non-*DL-Lite*$_\mathcal{R}$ axioms in $\mathcal{T}$, i.e., using $\langle\mathcal{T}^1,\mathcal{M},\mathcal{S}\rangle$ where $\mathcal{T}^1$ are all the *DL-Lite*$_\mathcal{R}$ axioms in $\mathcal{T}$.
(2) The local semantic approximation (LSA) of $\mathcal{T}$ in *DL-Lite*$_\mathcal{R}$, i.e., using $\langle\mathcal{T}^2,\mathcal{M},\mathcal{S}\rangle$ where $\mathcal{T}^2$ is obtained as the union, for each axiom $\alpha \in \mathcal{T}$, of the set of *DL-Lite*$_\mathcal{R}$ axioms $\Gamma(\alpha)$ entailed by α (Console et al. 2014).
(3) The global semantic approximation (GSA) of $\mathcal{T}$ in *DL-Lite*$_\mathcal{R}$, i.e., using $\langle\mathcal{T}^3,\mathcal{M},\mathcal{S}\rangle$ where $\mathcal{T}^3$ is the *DL-Lite*$_\mathcal{R}$ closure of $\mathcal{T}$ (Pan and Thomas 2007).
(4) Result of ONTOPROX, $\langle \mathsf{rew}(\mathcal{T}), \mathsf{cut}_5(\mathsf{comp}(\mathcal{T},\mathcal{M})), \mathcal{S}\rangle$.
(5) CLIPPER over the materialization of the virtual ABox.

In Table 1, we present details of the evaluation for some of the queries for which we obtained significant results. In Table 2, we provide statistics about the ONTOPROX pre-computations. The performed evaluation led to the following findings:

- For the considered set of queries LSA and GSA produce the same answers.
- Compared to the default ONTOP behavior, LSA/GSA produces more answers for 2 queries out of 4 for UOBM.
- ONTOPROX produces more answers than LSA/GSA for 2 queries out of 4 for UOBM, and for all Telecom queries. In particular, note that for Q_1^t and Q_2^t, LSA and GSA returned no answers at all.
- For UOBM, ONTOPROX answers are complete, as confirmed by the comparison with the results provided by CLIPPER. We cannot determine completeness for the Telecom queries, because the Telecom database was too large and its materialization in an ABox was not feasible.
- Query answering of ONTOPROX is ~3–5 times slower than ONTOP, when the result sets are of comparable size (note that for Q_2^t the result set is significantly larger).
- The size of the new *DL-Lite*$_\mathcal{R}$ OBDA specifications is comparable with that of the original specifications.

7 Conclusions

We proposed a novel framework for rewriting and approximation of OBDA specifications in an expressive ontology language to specifications in a weaker language, in which the core idea is to exploit the mapping layer to encode part of the semantics of the original OBDA specification, and we developed techniques for *DL-Lite*$_\mathcal{R}$ as the target language.

We plan to continue our work along the following directions: *(i)* extend our technique to Horn-$\mathcal{SHIQ}$, and, more generally, to Datalog rewritable TBoxes (Cuenca Grau et al. 2013); *(ii)* deepen our understanding of the computational complexity of deciding CQ-rewritability of OBDA specifications into *DL-Lite*$_\mathcal{R}$; *(iii)* extend our technique to SPARQL queries under different OWL entailment regimes (Kontchakov et al. 2014); *(iv)* carry out more extensive experiments, considering queries that contain existentially quantified variables. This will allow us to verify the effectiveness of RewObda, which was designed specifically to deal with existentially implied objects.

Acknowledgement. This paper is supported by the EU under the large-scale integrating project (IP) Optique (*Scalable End-user Access to Big Data*), grant agreement n. FP7-318338. We thank Martin Rezk for insightful discussions, and Benjamin Cogrel and Elem Güzel for help with the experimentation.