Emrooz: A Scalable Database for SSN Observations

Markus Stocker[1], Narasinha Shurpali[2], Kerry Taylor[3], George Burba[4], Mauno Rönkkö[1], and Mikko Kolehmainen[1]

[1] Environmental Informatics Research Group, Department of Environmental Science, University of Eastern Finland, P.O. Box 1627, 70211 Kuopio, Finland
{markus.stocker,mauno.ronkko,mikko.kolehmainen}@uef.fi
[2] Biogeochemistry Research Group, Department of Environmental Science, University of Eastern Finland, P.O. Box 1627, 70211 Kuopio, Finland
narasinha.shurpali@uef.fi
[3] College of Engineering and Computer Science, Australian National University, Acton ACT 2601 Australia
kerry.taylor@acm.org
[4] LI-COR, 4647 Superior Street, Lincoln, Nebraska USA
george.burba@licor.com

Abstract. The design of ontologies for sensor data and metadata has received considerable attention. The most prominent is arguably the Semantic Sensor Network (SSN) ontology. For persistence and retrieval of sensor observations, systems that adopt the SSN ontology most obviously build on an RDF database (triple store). However, large volumes of collected sensor data can be challenging for RDF databases, as the evaluation of SPARQL queries for SSN observations quickly becomes prohibitively expensive. This is arguably due to the fact that triple stores are optimized to efficiently evaluate graph pattern queries, not time series interval queries. As our main contribution, we present *Emrooz*, a scalable database capable of consuming SSN observations represented in RDF and evaluating queries for SSN observations formulated in SPARQL. We present the Emrooz implementation on Apache Cassandra and Sesame and its performance compared to two state-of-the-art RDF databases. The results show that Emrooz query performance outperforms the two RDF databases by orders of magnitude with increasingly large datasets. We motivate the need for scalable databases for SSN observations on a case study in micrometeorology.

Keywords: Sensor Data, Data Management, Query Performance, Ontology, RDF, SSN ontology, Semantic Web, Linked Data, Emrooz

1 Introduction

Emrooz means 'today' in Farsi and is the name of an open source database for SSN [1] observations represented in RDF [2] capable of evaluating queries for SSN observations formulated in SPARQL [3]. Emrooz builds on Apache Cassandra and Sesame [4], which serve in the implementation of Emrooz data and knowledge stores, respectively. The Emrooz code repository is available on GitHub.[5]

Over the past decade, several authors have proposed ontologies with formalized vocabulary for describing sensors—their metadata such as observed properties, operating ranges, and location—and sensor observations—the data collected from sensors such as the observation value and the time at which the observation was made. Compton et al. [5] reviewed some of the efforts in the 'semantic specification of sensors'. Today, the most prevalent ontology for the domain of sensing is arguably the SSN ontology. It has been widely adopted in the literature [6–14].

Ontologies can facilitate the querying, integration and reuse of sensor data as well as ease the management of large networks with heterogeneous sensors. Whereas the volume of metadata about sensors is generally comparatively small—and can thus be easily managed by RDF databases, specifically triple stores—the volume of data collected from sensors is often large.

As we demonstrate in this paper, the performance of state-of-the-art RDF databases in evaluating SSN observation queries quickly degrades with an increasing number of observations. This constitutes an obvious and practical problem for the adoption of the SSN ontology. If unable to answer queries quickly, a technology that promises semantic interoperability, reasoning, and data linked to metadata in graph data structures arguably remains a mere theoretical curiosity. Database systems for SSN observations with good load and excellent query performance are needed for the technology to become viable in practice.

Our focus is on environmental sensor networks [15] and their use in earth and environmental science research. Primarily for this community, we aim at developing a database capable of consuming SSN observations and fast evaluation of corresponding queries formulated in SPARQL. The proposed case study is in micrometeorology, specifically in monitoring of surface-atmosphere energy and trace gas fluxes using a typical LI-COR Eddy Covariance System. As we discuss in more details in Section 3, such systems generate large volumes of data, currently stored as files. Researchers are thus unable to readily retrieve a time series of arbitrary time interval using a declarative query language. Emrooz attempts to address this particular concern for scientists who measure surface-atmosphere fluxes by proposing an approach that merely commits to the SSN ontology and is thus generic with regard to specific sensors, their data and metadata. In addition to advantages such as declarative querying and semantic interoperability of data and metadata, the use of the SSN ontology in Emrooz frees individual researchers from having to design models and database schemata for their sensor data and metadata.

2 Implementation

Emrooz defines two main abstractions: the data store and the knowledge store. The data store supports the persistence of sensor observations. Sensor observations are represented following the SSN ontology as sets of RDF statements

[5] https://github.com/markusstocker/emrooz

(i.e. triples). Accordingly, a sensor observation relates to the measured value, the time at which the observation became available, the sensor that made the observation, and the observed property and feature. The retrieval of sensor observations is enabled by data store query handlers. A data store query handler evaluates a set, $\mathcal{Q}_{so}$, of sensor observation queries, q_{so}.

The knowledge store manages sensor specifications (metadata). A specification for a sensor defines the observed property and the sampling frequency. The observed property relates to a feature. Sensors are specified by creating and relating relevant individuals of SSN classes, typically using an editor such as Protégé.[6] The resulting file can be loaded by the knowledge store. A knowledge store can create query handlers. A knowledge store query handler owns a data store query handler and evaluates a SPARQL SELECT query for SSN observations, q_{ssn}.

Queries q_{ssn} are formulated by some agent, e.g. a user, and must define a time interval $[t_1, t_2[$. The evaluation of such queries occurs in three stages. First, q_{ssn} is translated into a sensor observation query, q_{so}. A sensor observation query $q_{so} \leftarrow (\dot{s}, \dot{p}, \dot{f}, \dot{t_1}, \dot{t_2})$ consists of parameters for the sensor, $\dot{s}$, property, $\dot{p}$, feature, $\dot{f}$, and time interval, $[\dot{t_1}, \dot{t_2}[$. Values for these parameters are extracted from q_{ssn} during translation. The parameters may be bound or unbound. Second, q_{so} is rewritten into a set of sensor observation queries, $\mathcal{Q}_{so}$, that may be a singleton set. This is the case when q_{so} has *defined* sensor, property, and feature. If any of these parameters is undefined, then sensor specifications managed by the knowledge store are utilized to rewrite q_{so} into queries $q^i_{so} \in \mathcal{Q}_{so}$ so that (1) each q^i_{so} matches the defined parameters in q_{so} and (2) the defined tuple $(\dot{s}, \dot{p}, \dot{f})$ matches a sensor specification. Third, a knowledge store query handler for q_{ssn} and a data store query handler for $\mathcal{Q}_{so}$ are composed. The knowledge store query handler evaluates q_{ssn} on the results returned by the data store query handler in evaluating $\mathcal{Q}_{so}$.

Emrooz builds its knowledge store implementation on the Sesame framework for RDF.[7] Of particular interest to Emrooz are Sesame repositories and SPARQL query parsing and evaluation. Sensor specifications are managed by a Sesame repository. Depending on the application, the repository may be volatile or persistent, resident on the local machine or on a remote server. To evaluate q_{ssn}, the knowledge store query handler implementation for Sesame utilizes a volatile repository initialized with RDF statements returned by the composed data store query handler.

The data store implementation builds on Apache Cassandra.[8] Sensor observations are persisted to rows of a `data` table with schema consisting of partition key (row key) of type `ascii`; clustering key (column name) of type `timeuuid`; and column value of type `blob`. The partition key and the clustering key form a compound primary key.

[6] http://protege.stanford.edu
[7] http://rdf4j.org/
[8] http://cassandra.apache.org

The partition key consists of two dash-concatenated parts: a SHA-256 hex string and a date-time string. The hex string is a digest of a message (string) consisting of dash-concatenated identifiers (URIs) for a sensor, a property, and a feature. The date-time string follows the pattern 'yyyyMMddHHmm'. Given a sensor observation and the specification for the related sensor, the date-time string is computed from the observation result time, truncated to the year, month, day, hour, or minute depending on the specified sampling frequency. For instance, for sensors with sampling frequency $]1, 100]$ Hz the computed date-time string is truncated to the hour. The date-time string thus limits the number of sensor observations per partition key for any given $(\dot{s}, \dot{p}, \dot{f})$ tuple. For a sensing device with sampling frequency 10 Hz each row holds 36000 sensor observations.

Sensor observation result times determine clustering keys. Specifically, given a sensor observation, the corresponding result time is translated into a time UUID, which acts as column name. Columns are ordered in time and support fast time interval scans.

Column values are `byte` arrays for the binary-encoded sets of RDF statements corresponding to sensor observations. The representation of sensor observations as sets of RDF statements is handled by an RDF entity representer, implemented in Emrooz. The representer translates entities (Java objects) into corresponding sets of RDF statements. Sets of RDF statements are then converted to `byte` arrays using the Sesame binary RDF writer.

In addition to the SSN ontology, Emrooz also adopts OWL-Time [16] for the representation of temporal entities, GeoSPARQL [17] for the representation of spatial entities, and the Quantities, Units, Dimensions and Data Types Ontologies (QUDT) [18] for the representation of quantities and units, such as the sampling frequency in sensor specifications.

3 Case Study

We evaluate Emrooz comparative performance with data of a typical LI-COR Eddy Covariance System for the direct measurement of CO_2, CH_4, and H_2O fluxes.

Eddy covariance is a method to directly measure surface-atmosphere fluxes of energy and trace gases. It has been employed to monitor fluxes over various ecosystems and for diverse applications, also in climate change research where CO_2 and CH_4 flux measurements by eddy covariance method support determining whether the observed ecosystem is a carbon sink or source. Large data volumes for surface-atmosphere fluxes of energy and trace gases are managed by platforms such as ICOS Carbon Portal. [9]

The installation consists of a LI-7500A Open Path CO_2/H_2O Gas Analyzer, a LI-7700 Open Path CH_4 Analyzer, and a sonic anemometer. The devices operate at 10 Hz sampling frequency. Collected data is typically stored on a USB drive of a LI-7550 Analyzer Interface Unit. The components of a LI-COR Eddy

[9] https://www.icos-cp.eu/

Covariance System are often installed on a tripod, which thus acts as a platform for the devices. For this study, we consider two gas analyzers, the property of mole fraction, and three features for the monitored gases.

The data are available in ZIP archive files. Each archive contains text files with metadata about the site, instruments, and the data files as well as the data for 30 min of measurement. The time period considered in our experiments begins on January 7, 2015 and ends on May 26, 2015. The total number of archive files is 6045. Effectively there should be 6720 archive files for the period but the dataset is incomplete between March 3 and April 12, during which it misses 675 archive files.

For each archive file, the data file of interest is the one containing observation values for CO_2, H_2O, and CH_4. Except for a header spanning the first few lines, this data file consists of a $18\,000 \times 40$ matrix. The number of rows is equivalent to the number of 10 Hz samples in 30 min ($10 \times 60 \times 30 = 18\,000$). Of this matrix, we concentrate on the three columns for measured CO_2 [µmol mol^{-1}], H_2O [mmol mol^{-1}], and CH_4 [µmol mol^{-1}] plus the two columns for date and time. Thus, we expect 54 000 sensor observations per matrix, i.e. per 30 min of measurement. For the January-May period the expected number of sensor observations is 326 430 000. Considering that each sensor observation maps to a set of 15 RDF statements (triples) the expected total number of processed triples is approximately 4.9 billion.

We evaluate the load and query performance of Emrooz on 10 subsets, for 30 minutes, 1 hour, 3 hours, 6 hours, 12 hours, 1 day, 7 days, 1 month, 3 months, and the complete dataset (J-M). All subsets begin on January 7, except those for 1 month (February) and 3 months (February-April). Note that the 3 months subset is incomplete. Query performance is evaluated using a defined query $q_{so} \leftarrow (\dot{s}, \dot{p}, \dot{f}, \dot{t}_1, \dot{t}_2)$, whereby $\dot{f}$ is CO_2 and the time interval $[\dot{t}_1, \dot{t}_2[$ is 10 min. The expected result set size of each query is 6000. We evaluate the query performance as the mean value of three runs per subset. Emrooz performance is compared with two RDF databases: Stardog[10] 2.2 and Blazegraph[11] 1.5.1. For all three systems, we use the integrated Sesame API to load and query sensor observations. For both Stardog and Blazegraph we use persistent disk databases (local triple stores). The disk databases are created first and data is loaded in transactions of maximally approximately 2 million triples. We use Apache Cassandra 2.1.3, Sesame 2.8.1, and Emrooz 0.2.0. The evaluation is performed on a Fujitsu CELSIUS W420 with an i7-3770 3.40 GHz CPU, 4×8 GB DDR3 1600 MHz DIMM memory modules, and 2×1 TB 7200 RPM SATA hard drives.

4 Results and Discussion

We first provide an overview of subset sizes in terms of number of sensor observations, corresponding triples, and distinct triples. Table 1 summarizes the numbers. Sensor observations are represented as sets of triples, including triples

[10] http://stardog.com
[11] http://www.blazegraph.com/bigdata

Table 1. The number of sensor observations and corresponding (distinct) triples per subset. J-M stands for the complete dataset spanning the period January-May. Stardog and Blazegraph evaluations did not terminate on time for this paper; hence the missing count (*) of distinct triples for the 3 M and J-M subsets.

Subset	Observations	Triples	Distinct
30 m	54 000	810 000	648 007
1 h	108 000	1 620 000	1 296 007
3 h	324 000	4 860 000	3 888 007
6 h	647 997	9 719 955	7 775 971
12 h	1 295 997	19 439 955	15 551 971
1 d	2 591 994	38 879 910	31 103 935
7 d	18 140 271	272 104 065	217 683 259
1 M	72 526 464	1 087 896 960	870 317 575
3 M	194 188 107	2 912 821 605	*
J-M	328 715 445	4 930 731 675	*

asserting class membership of sensors, properties, and features. The number of triples is always 15 times the number of observations. The set of distinct triples is smaller because it excludes duplicate triples. We also observe that with the 6 h subset the expected and actual number of sensor observations (and thus triples) differ. We investigated the reason and found that the data file for January 7 at 4 a.m. misses data for 04:00:53.100. Hence the three missing sensor observations in the 6 h subset. We suspect that this also explains differences between expected and actual number of sensor observations in other subsets.

Figure 1 summarizes the *load* performance for the 10 subsets and Emrooz compared to Stardog and Blazegraph. The figure shows that Emrooz is outperformed on small datasets. However, with larger datasets Emrooz outperforms both Stardog and Blazegraph. We attribute this behavior to the apparent gradually increasing cost of committing transactions in Stardog and Blazegraph.

Figure 2 summarizes the *query* performance for the 10 subsets and Emrooz compared to Stardog and Blazegraph. With constant time at roughly 2.3 s, Emrooz outperforms both triple stores—by several orders of magnitude for large datasets. The query performance difference between Emrooz and the two triple stores is, however, not surprising. Given a defined query $q_{so} \leftarrow (\dot{s}, \dot{p}, \dot{f}, \dot{t}_1, \dot{t}_2)$, Apache Cassandra can efficiently retrieve the relevant set of triples by directly addressing the row key and perform a range scan on column names. The resulting set of triples is subsequently processed by Sesame using a (volatile) memory store. In contrast, both Stardog and Blazegraph evaluate the defined tuple $(\dot{s}, \dot{p}, \dot{f})$ as a SPARQL basic graph pattern with expensive joins and resulting in an intermediate result set corresponding to the complete time series eventually filtered to the desired interval $[\dot{t}_1, \dot{t}_2[$.

The results suggest that Emrooz query performance is independent of data store size. However, query performance is dependent on query time interval duration. The effect of varying intervals ranging from 1 s to 60 min is shown in Figure 3. The query with time interval duration 10 min executes in 2.21 s, which

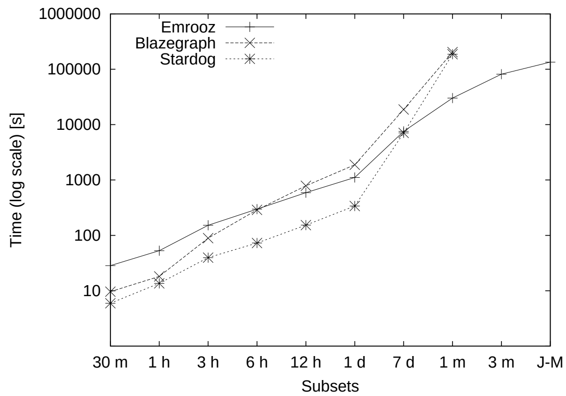

Fig. 1. Load performance for the subsets and Emrooz compared to Stardog and Blazegraph. Stardog and Blazegraph evaluations did not terminate on time for this paper.

is comparable to Figure 2 for variable subsets. Queries with shorter time interval duration evaluate faster while queries with longer time interval duration evaluate slower. Several factors are at play, in particular the time required to evaluate q_{ssn} on larger memory stores in Sesame post-processing and the time required to iterate over result sets of increasing size.

Compared to Stardog and Blazegraph, and comparable triple stores, Emrooz has some constraints. Most obviously, Emrooz cannot manage arbitrary RDF data. Furthermore, Apache Cassandra has no means to perform standard reasoning tasks on SSN observations. Off-the-shelf reasoning can only be performed by Sesame, on the knowledge store and in post-processing query result sets returned by Apache Cassandra. Some level of reasoning pushed down to the Apache Cassandra data store can be implemented by Emrooz using the Sesame knowledge store and query rewriting [19].

Emrooz is currently capable of evaluating SPARQL queries with a basic graph pattern for SSN observations, whereby the observation result time must be constrained by a time interval $[\dot{t}_1, \dot{t}_2[$ specified as `FILTER`. The related sensor, property, and feature may be bound or unbound. SPARQL features such as aggregates and solution modifiers such as `ORDER BY` can be specified over $[\dot{t}_1, \dot{t}_2[$.

Sesame post-processing adds overhead which can be avoided if applications do not require SPARQL. SPARQL adds flexibility, e.g. it enables selecting variables, ordering or filtering results. However, in some applications this flexibility may not be required and does thus not justify the overhead. For instance, a data portal may simply want to return the set of RDF statements matching the user query $q_{so} \leftarrow (\dot{s}, \dot{p}, \dot{f}, \dot{t}_1, \dot{t}_2)$ and leave further processing to the user.

The data in our case study arguably fall into the category of "particularly hard cases" for triple stores. Assuming equally sized datasets, SSN observation query evaluation on data collected from sensor networks with more sensors, properties, and features but sampling at lower frequency are less expensive for triple stores. This is because the tuple $(\dot{s}, \dot{p}, \dot{f})$, as well as its elements, are more

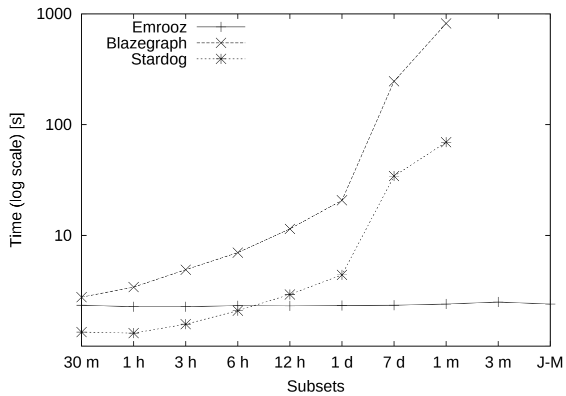

Fig. 2. Query performance for the subsets and Emrooz compared to Stardog and Blazegraph. Stardog and Blazegraph evaluations did not terminate on time for this paper.

selective. The intermediate result sets are smaller and basic graph pattern joins less expensive. Furthermore, more diverse selectivity estimates for triple patterns could give query optimizers more room to find better query plans.

5 Related and Future Work

A number of authors have developed RDF data management systems that build on NoSQL stores such as Apache Cassandra. Cudré-Mauroux et al. [20] provide a comparative evaluation for the load and query performance of several systems that implement an RDF data management layer on top of a NoSQL system. Of particular interest here is CumulusRDF [21], as it also builds on Apache Cassandra and Sesame. However, these systems aim at being RDF databases and thus implement indexes specialized for answering arbitrary SPARQL queries on RDF. In contrast, Emrooz is designed for the management of SSN observations represented in RDF and for the evaluation of SSN observation queries formulated in SPARQL. Emrooz is thus designed to be a scalable time series database for sensor observations represented in RDF according to the vocabulary defined by the SSN ontology.

Authors who developed systems for (historical or streamed) sensor data management have recognized that persisting large volumes of sensor data in an RDF database is hardly viable. Presenting a platform designed to connect (semantic) sensor data with data in the 'Linked Data Cloud', Le-Phuoc et al. [7] resort to a relational database management system for historical sensor data management. Describing a data warehouse for water resource management, Abecker et al. [22] also propose a hybrid approach in which time series sensor data is managed by a relational database system (PostGIS) whereas information objects with more complex relationships are managed by an RDF database. The authors note that "a complete 'semantification' [...] of all data [...] seemed not feasible and promising to us, especially regarding the measurement data."

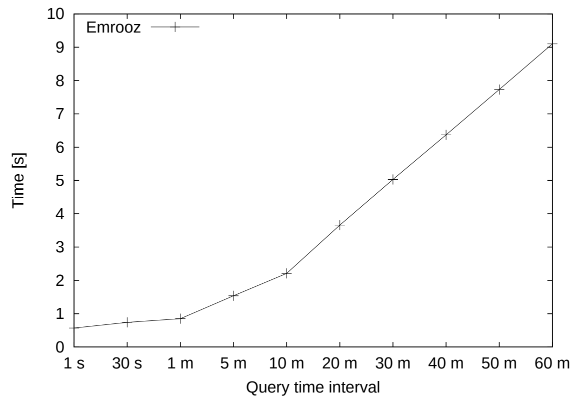

Fig. 3. Emrooz query performance with increasing query time interval duration, from 1 s to 60 min on the 6 h subset. Result set sizes for 1 s and 60 min are 10 and 36 000, respectively.

NoSQL systems have been utilized to manage SSN observations, specifically. Wang et al. [23] present a Hadoop-based system designed to manage SSN observations. The authors describe how their system stores SSN observations to HBase which, however, features an index structure that is typical for RDF statements, akin to the systems surveyed by Cudré-Mauroux et al. [20]. Wang et al. evaluate the performance of various queries. However, to our understanding the queries are not for SSN observations but rather for single triple patterns, e.g. a pattern with bound subject and unbound predicate and object. As such, both the indexing approach and the query performance evaluation are different from those presented in this paper for Emrooz.

There are several potentially interesting directions for future work. First, we plan to extend the implementation so that it supports the management of *dataset* observations represented following the RDF Data Cube (QB) Vocabulary [24]. With this extension, Emrooz could thus manage not only raw sensor data but also processed sensor data. For instance, CO_2 flux sensor data are used to compute Net Ecosystem Exchange (NEE). NEE data are a result of sensor data processing and form a dataset; hence the different vocabulary. Combining the SSN ontology and the QB vocabulary in systems has been demonstrated in the literature, e.g. [6, 25].

Second, Emrooz can be equipped with further features. Command line tools could simplify user interaction with Emrooz. A RESTful service could expose (load) and query functionality for client-server interaction over HTTP. A browser-based client could support the visualization of time series. R[12] and Matlab[13] libraries could enable users to query and persist data from statistical computing environments. Data managed by Emrooz could then be loaded into R data

[12] http://www.r-project.org/
[13] www.mathworks.com/products/matlab/

frames. Results of computations in R could be persisted in Emrooz. Finally, Emrooz could be enhanced with more flexibility in SPARQL query formulation, e.g. filter for a set of properties, as well as reasoning at query time. These enhancements could be supported by extending the existing query rewriting mechanism. A detailed analysis of the SPARQL expressivity covered in Emrooz may also be of interest.

Third, it is interesting to compare Emrooz performance with triple stores that build on SQL and NoSQL databases, such as SDB[14] and CumulusRDF [21], respectively, as well as with non-triple stores, including SQL or OGC standards compliant databases, such as PostgreSQL[15] and 52North,[16] respectively.

6 Conclusion

We have presented Emrooz, a scalable database for SSN observations represented in RDF capable of evaluating queries for SSN observations formulated in SPARQL. We briefly discussed how Emrooz builds on Apache Cassandra and Sesame for its implementations of a data store and a knowledge store, respectively, and how the two stores interact. Emrooz is motivated by the following two contrasting aspects. On one hand, the attractiveness of the RDF data model and the SSN ontology for representing metadata about sensors and what is sensed, as well as for representing data resulting in sensor measurement, is an argument for adopting these technologies in systems. On the other hand, the most obvious approach to SSN observations management using triple stores seems to fail on a fundamental and important requirement, i.e. scalable fast evaluation of SSN observation queries. As we demonstrated for two state-of-the-art triple stores, SSN observation query evaluation becomes quickly prohibitive as store size grows to tens of millions sensor observations. To serve client applications with SSN observations in RDF is attractive for several reasons, including data linked to metadata and formal descriptions of vocabulary semantics. However, for practical viability the underlying SSN observations management system needs to be designed for time series query evaluation.

Acknowledgements

This research is funded by the Academy of Finland project "FResCo: High-quality Measurement Infrastructure for Future Resilient Control Systems" (Grant number 264060).

References

1. Compton, M., Barnaghi, P., Bermudez, L., Garca-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K.,

[14] https://jena.apache.org/documentation/sdb/
[15] http://www.postgresql.org/
[16] http://52north.org/

Kelsey, W.D., Phuoc, D.L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN ontology of the W3C semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web **17**(0) (2012) 25–32
2. Cyganiak, R., Wood, D., Lanthaler, M.: RDF 1.1 Concepts and Abstract Syntax. Recommendation, W3C (February 2014)
3. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language. Recommendation, W3C (March 2013)
4. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In Horrocks, I., Hendler, J., eds.: The Semantic Web – ISWC 2002. Volume 2342 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2002) 54–68
5. Compton, M., Henson, C., Lefort, L., Neuhaus, H., Sheth, A.: A Survey of the Semantic Specification of Sensors. In Taylor, K., Ayyagari, A., Roure, D.D., eds.: Proceedings of the 2nd International Workshop on Semantic Sensor Networks. Volume 522., Washington DC, USA, CEUR-WS (October 2009) 17–32
6. Lefort, L., Bobruk, J., Haller, A., Taylor, K., Woolf, A.: A Linked Sensor Data Cube for a 100 Year Homogenised Daily Temperature Dataset. In Henson, C., Taylor, K., Corcho, O., eds.: Proceedings of the 5th International Workshop on Semantic Sensor Networks. Volume 904., Boston, Massachusetts, CEUR-WS (November 2012) 1–16
7. Le-Phuoc, D., Quoc, H.N.M., Parreira, J.X., Hauswirth, M.: The Linked Sensor Middleware—Connecting the real world and the Semantic Web. In: Proceedings of the Semantic Web Challenge at the 10th International Semantic Web Conference, Bonn, Germany (October 2011)
8. Müller, H., Cabral, L., Morshed, A., Shu, Y.: From RESTful to SPARQL: A Case Study on Generating Semantic Sensor Data. In Corcho, O., Henson, C., Barnaghi, P., eds.: Proceedings of the 6th International Workshop on Semantic Sensor Networks. Volume 1063., Sydney, Australia, CEUR-WS (October 2013) 51–66
9. Calbimonte, J.P., Yan, Z., Jeung, H., Corcho, O., Aberer, K.: Deriving Semantic Sensor Metadata from Raw Measurements. In Henson, C., Taylor, K., Corcho, O., eds.: Proceedings of the 5th International Workshop on Semantic Sensor Networks. Volume 904., Boston, Massachusetts, USA, CEUR-WS (November 2012) 33–48
10. Yu, J., Davis, P., Gould, S., Taylor, K.: Linked Data Approach for Automated Failure Detection in Sewere Rising Mains Using Real-Time Sensor Data. In: Proceedings of the 11th International Conference on Hydroinformatics, New York City, USA (2014)
11. Wu, L.: Representing and Inferring Events from Deforestation Observations. In Gensel, J., Josselin, D., Vandenbroucke, D., eds.: Proceedings of the AGILE'2012 International Conference on Geographic Information Science, Avignon, France (April 2012) 80/392–85/392
12. Llaves, A., Kuhn, W.: An event abstraction layer for the integration of geosensor data. International Journal of Geographical Information Science **28**(5) (2014) 1085–1106
13. Taylor, K., Leidinger, L.: Ontology-Driven Complex Event Processing in Heterogeneous Sensor Networks. In Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., Leenheer, P., Pan, J., eds.: The Semanic Web: Research and Applications. Volume 6644 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 285–299

14. Rinne, M., amd Robin Keskisärkkä, E.B., Nuutila, E.: Event Processing in RDF. In Gangemi, A., Gruninger, M., Hammar, K., Lefort, L., Presutti, V., Scherp, A., eds.: Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns. Volume 1188., Sydney, Australia, CEUR-WS (October 2013)
15. Martinez, K., Hart, J.K., Ong, R.: Environmental Sensor Networks. Computer **37**(8) (2004) 50–56
16. Hobbs, J.R., Pan, F.: Time Ontology in OWL. Working draft, W3C (September 2006)
17. Perry, M., Herring, J.: OGC GeoSPARQL - A Geographic Query Language for RDF Data. Technical Report OGC 11-052r4, Open Geospatial Consortium Inc. (September 2012)
18. Hodgson, R., Keller, P.J., Hodges, J., Spivak, J.: Qudt – quantities, units, dimensions and data types ontologies. Technical report, TopQuadrant, Inc (March 2014)
19. Kontchakov, R., Zakharyaschev, M.: An Introduction to Description Logics and Query Rewriting. In Koubarakis, M., Stamou, G., Stoilos, G., Horrocks, I., Kolaitis, P., Lausen, G., Weikum, G., eds.: Reasoning Web. Reasoning on the Web in the Big Data Era. Volume 8714 of Lecture Notes in Computer Science. Springer International Publishing (2014) 195–244
20. Cudré-Mauroux, P., Enchev, I., Fundatureanu, S., Groth, P., Haque, A., Harth, A., Keppmann, F.L., Miranker, D., Sequeda, J.F., Wylot, M.: NoSQL Databases for RDF: An Empirical Evaluation. In Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K., eds.: The Semantic Web – ISWC 2013. Volume 8219 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 310–325
21. Ladwig, G., Harth, A.: CumulusRDF: Linked Data Management on Nested Key-Value Stores. In Fokoue, A., Liebig, T., Guo, Y., eds.: The 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2011), Bonn, Germany (October 2011) 30–42
22. Abecker, A., Brauer, T., Magoutas, B., Mentzas, G., Papageorgiou, N., Quenzer, M.: A Sensor and Semantic Data Warehouse for Integrated Water Resource Management. In Gómez, J.M., Sonnenschein, M., Vogel, U., Winter, A., Rapp, B., Giesen, N., eds.: Proceedings of the 28th Conference on Environmental Informatics - Informatics for Environmental Protection, Sustainable Development and Risk Management (EnviroInfo 2014), Oldenburg, Germany, BIS-Verlag, Oldenburg (September 2014) 517–524
23. Wang, D., Zhang, X., Gao, H.: HDSW: Semantic Sensor Network System Based on Hadoop. International Journal of Multimedia and Ubiquitous Engineering **9**(12) (2014) 61–72
24. Cyganiak, R., Reynolds, D., Tennison, J.: The RDF Data Cube Vocabulary. Recommendation, W3C (January 2014)
25. Stocker, M., Baranizadeh, E., Portin, H., Komppula, M., Rönkkö, M., Hamed, A., Virtanen, A., Lehtinen, K., Laaksonen, A., Kolehmainen, M.: Representing situational knowledge acquired from sensor data for atmospheric phenomena. Environmental Modelling & Software **58** (2014) 27–47