ABSTRACT

RDF and SPARQL are established standards for data interchange and querying on the Web. While they have been shown to be useful and applicable in many scenarios, they are not sufficiently adequate for dealing with streams of data and their intrinsic continuous nature. In the last years data and query languages have been proposed to extend both RDF and SPARQL for streams and continuous processing, under the name of RDF Stream Processing – RSP. These efforts resulted in several models and implementations that, at a first look, appear to propose alternative syntaxes but equivalent semantics. However, when asked to continuously answer the same queries on the same data streams, they provide different answers at disparate moments due to the heterogeneity of their operational semantics. These discrepancies render the process of understanding and comparing continuous query results complex and misleading. In this work, the authors propose RSP-QL, a comprehensive model that formally defines the semantics of an RSP system. RSP-QL makes explicit the hidden assumptions of currently available RSP systems, allows defining a formal notion of correctness for RSP query results and, thus, explains why available implementations provide different answers at disparate moments.

Keywords: Conceptual Model, Continuous Query Processing, Correctness, RDF Streams, RSP-QL Semantics
INTRODUCTION

One of the key ingredients of the Ubiquitous Web is the management of data streams, which may be obtained from a range of data sources, from social networks to environmental sensors. These data streams may be expressed using RDF, possibly according to well-established vocabularies, such as the W3C Semantic Sensor Network Ontology (Compton et al., 2012). In these cases, we can talk about RDF streams, which are formally defined as potentially unbounded sequences of time-varying RDF statements or graphs. In recent years, several RDF Stream Processing (RSP) systems have emerged, which allow querying RDF streams using extensions of SPARQL that include operators that take into account the streaming nature of these dynamic data sources (Barbieri, Braga, Ceri, Della Valle, & Grossniklaus, 2010; Calbimonte, Jeung, Corcho, & Aberer, 2012; Phuoc, Dao-Tran, Parreira, & Hauswirth, 2011; Anicic, Fodor, Rudolph, & Stojanovic, 2011). These systems are heterogeneous in terms of syntax and capabilities (due to the choice of operators and syntax selected to extend SPARQL). In addition, they implement different evaluation semantics for a set of constructs that may look similar in principle (for example, they may handle time window operators differently). These engines have different assumptions on how the query processing and delivery of results take place, which makes it difficult to describe, compare, understand and evaluate their behaviour.

In this paper, we address the following research question: is it possible to create a formal RDF stream processing model, including its evaluation semantics, which can be used to describe existing RSP systems? For this purpose, we propose RSP-QL, a unifying formal model for representing and processing RDF streams that reflects the different semantics of existing RSP systems. RSP-QL extends the SPARQL model and also takes into account two existing models coming from the streaming data world: CQL (Arasu, Babu, & Widom, 2006) and SECRET (Botan et al., 2010). CQL is a continuous extension of SQL: its semantics define a formal model with three kinds of operators (S2R, R2R and R2S) that process and transform streams and relations. SECRET is a framework to characterise and analyse the operational semantics of window operators. A second contribution of this paper is to show how this formal model can be used to test whether an RSP system is correct or not. RSP-QL extends our previous work (Dell’Aglio, Balduini, & Della Valle, 2013) that was focused on formalising the notion of correctness in RSP query processing and on the development of an oracle – a system that tests whether an RSP implementation works in accordance to the corresponding evaluation semantics of the language that it gives support to. We have shown that this oracle, based on the RSPQL model can effectively model the behaviour of existing engines, and assess the correctness of their results. As a result of our experiments we detected errors in existing implementations, some of which have been now fixed by the corresponding system implementers.

The remainder of the paper is organised as follows. After a brief recap on RDF and SPARQL, we formally define the notion of RDF streams and the evaluation semantics of a generic SPARQL extension that allows handling RDF streams (which we name RSP-QL). Our formal definitions are based on the existing representation and evaluation semantics for RDF and SPARQL. We then show that existing RSP systems can be represented as instances of the RSPQL query model, highlighting the differences among them, e.g. different strategies to evaluate the continuous queries and different ways to manage the sliding windows. Next, we formally define the notion of correctness in RSP systems, and we explain how to use it to check whether system implementations are computing the correct answers. Finally, we present the conclusion and final remarks.
Building Virtual Communities through a De-Marginalized View of Knowledge Networking
www.igi-global.com/chapter/building-virtual-communities-through-marginalized/29300?camid=4v1a

The Human Semantic Web Shifting from Knowledge Push to Knowledge Pull
www.igi-global.com/article/human-semantic-web-shifting-knowledge/2809?camid=4v1a